SIEMENS EDA

Calibre® AutoFix User's Manual

Software Version 2021.2
Document Revision 14

SIEMENS

Revision History

Revision	Changes	Status/ Date
14	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes listed in the *Calibre Release Notes* for this products are reflected in this document. Approved by Michael Buehler.	Released April 2021
13	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes listed in the *Calibre Release Notes* for this products are reflected in this document. Approved by Michael Buehler.	Released January 2021
12	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes listed in the *Calibre Release Notes* for this products are reflected in this document. Approved by Michael Buehler.	Released October 2020
11	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes listed in the *Calibre Release Notes* for this products are reflected in this document. Approved by Michael Buehler.	Released July 2020

Author: In-house procedures and working practices require multiple authors for documents. All associated authors for each topic within this document are tracked within the Siemens EDA documentation source. For specific topic authors, contact the Siemens Digital Industries Software documentation department.

Revision History: Released documents maintain a revision history of up to four revisions. For earlier revision history, refer to earlier releases of documentation which are available on https://support.sw.siemens.com/.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction to Calibre AutoFix

Calibre® AutoFix is an all-in-one solution that addresses the differences in routing rules and sign-off rules by narrowing the gap between physical design and physical verification in place and route environments.

Calibre AutoFix

Calibre AutoFix is a software application that performs automatic layout routing correction for place and route designs. The comprehensive Calibre verification environment validates the corrected routing geometry in the layout to provide sign-off quality DRC-clean designs.

As technology nodes move towards 22 nm and beyond, design rules become greater in number and in complexity. This increase in design rule complexity challenges place and route engines, which operate under numerous constraints. The development of new place and route commands to handle sign-off DRC rules often lags behind the rule development.

Inconsistencies between the internal place and route rules and the sign-off DRC rules mean that a design may pass DRC verification inside the router but show numerous DRC error results at the physical verification stage. This behavior can lead to increased repair iterations and subsequently longer chip development times.

Additionally, place and route engines use abstract design components, typically in the form of LEF databases that contain cell, block, or IP layout information. The abstract blocks may not always capture the required data to correctly define the DRC boundary conditions at the level of the IP and hierarchical blocks. Multiple instantiations of these IP blocks under the constraints of the increasingly complex rules can lead to numerous potential DRC issues at the IP interface.

Calibre AutoFix currently supports the following design tools and databases:

- Synopsys® IC Compiler™ 2011.09-sp5 and newer
- Cadence® Encounter® Digital Implementation System
- 5.6 or 5.7 LEF/DEF design database

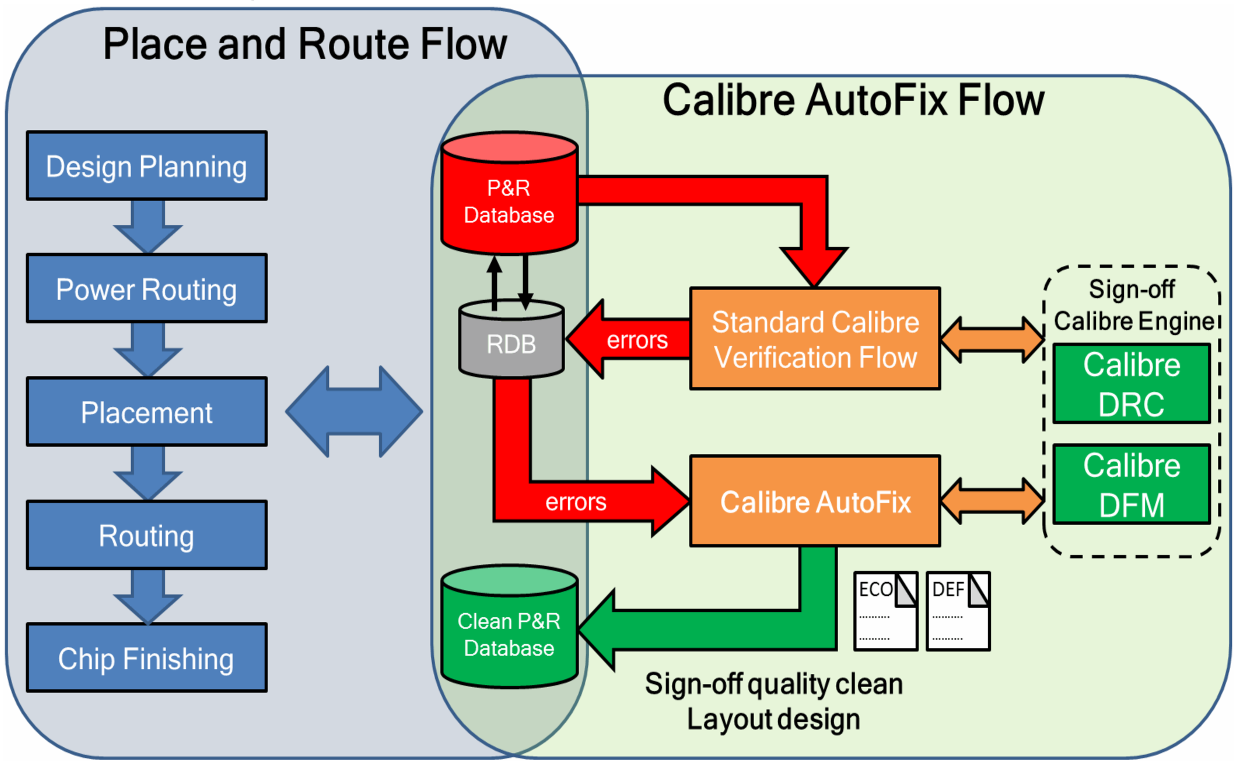
Related Topics

Calibre AutoFix Usage Examples

Workflow

Calibre AutoFix fits into a typical place and route workflow.

Figure 1-1. Overview of the Calibre AutoFix Workflow



When a place and route design is fully routed and DRC clean inside the routing tool environment, it must then be verified with sign-off quality DRC rules. Calibre is typically used to run sign-off verification checks. An error browser is used to view and debug any rule check results.

In traditional place and route flows, verification results are used by physical design engineers to adjust routing geometry in the place and route tool until the errors are no longer present. This verification and rerouting procedure iterates until all errors are corrected, or deemed safe to waive. This can contribute to significant design overhead.

With Calibre AutoFix, this time-consuming process can be circumvented. When Calibre AutoFix is invoked, it takes the generated verification results from a Calibre® nmDRC™ run and automatically adjusts routing geometry in the layout to correct selected errors. The tool exports an adjusted DEF file, or generates a Tcl ECO script with commands that correct the design within Synopsys IC Compiler.

Related Topics

Calibre AutoFix Usage Examples

Input and Output Files

Calibre AutoFix

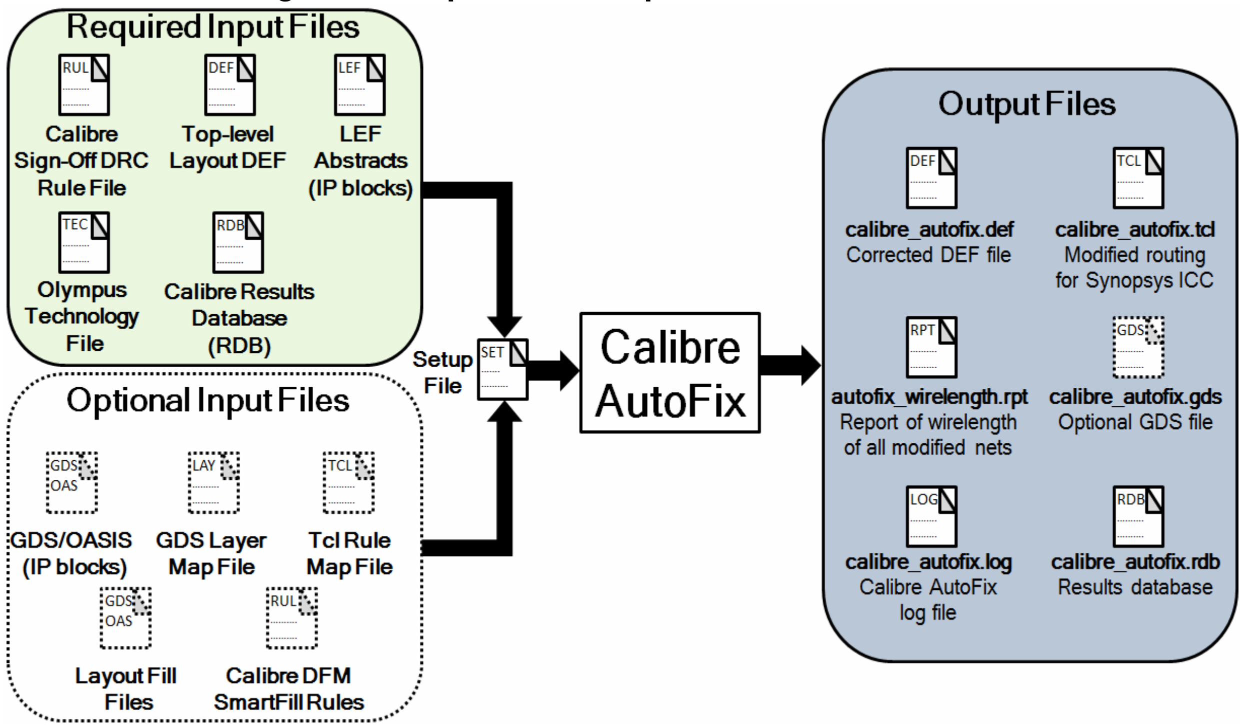
Requirements

Input and Output Files

The flow of input and output files that Calibre AutoFix uses and generates are shown. The setup file defines the inputs, outputs, and settings for the Calibre AutoFix run.

For a description of the required inputs, see "Requirements" on page 14. For details on supported commands and syntax in the setup file, see "Setup File Commands" on page 33.

Figure 1-2. Inputs and Outputs to Calibre AutoFix



Note

The Olympus Technology File is required for 32 nm and below. For any technology node, the Calibre layers and place and route layers must be mapped (either in the GDS abstracts, or with a GDS layer map file).

Related Topics

Requirements

Setup File Example

Requirements

Certain required tools, licenses, and input files are required in order to run Calibre AutoFix.

Design Tools

You must have Calibre 2021.2, Router AutoFix application, and one or more of these tools installed:

- Synopsys IC Compiler 2011.09-sp5 or newer
- Cadence Encounter Digital Implementation System that supports LEF/DEF 5.6 or newer
- Third-party place and route tool using 5.6 or 5.7 LEF/DEF design database

Licenses

Calibre AutoFix supports DRC fixing mode. You must have the following licenses:

Table 1-1. Required Licenses for Calibre AutoFix

Single CPU Licenses	Simultaneous Multithreading (SMT)
Calibre AutoFix	1 Calibre AutoFix
Router AutoFix	1 Router AutoFix
Router AutoFix MT Advanced	1 Router AutoFix MT Advanced
Required Calibre licenses for the fixing mode	Required Calibre licenses for the fixing mode and for the number of physical cores (CPUs) used on the machine

For example, to correct DRC issues on a 8 CPU SMT machine (8 physical cores + 8 virtual cores = 16 logical cores) where your requested CPU configuration for the DRC run is 5 CPUs (5 physical cores), requires the following:

- One Calibre AutoFix license
- One Router AutoFix license
- One Router AutoFix MT Advanced license
- Five Calibre® nmDRC™/Calibre® nmDRC-H™ license pairs

As a second example, to correct DRC issues on a 4 CPU SMT machine (4 physical cores + 4 virtual cores = 8 logical cores), where your requested CPU configuration for the DRC run is 5 CPUs (4 physical cores + 1 virtual core), requires the following:

- One Calibre AutoFix license
- One Router AutoFix license
- One Router AutoFix MT Advanced license
- Four Calibre® nmDRC™/Calibre® nmDRC-H™ license pairs

Note

When using SMT, only the physical cores require a license. No licenses are consumed by the virtual cores.

For more information on licensing and enabling SMT, refer to the *Calibre Administrator's Guide*.

Environment

Calibre AutoFix requires that you set up the run environment:

- Set the CALIBRE_HOME environment variable to point to the location of your Calibre software tree. See "Setting the CALIBRE_HOME Environment Variable" in the *Calibre Administrator's Guide* for details.
- Have a valid installation of the Calibre and Router AutoFix applications and set your path environment variable to point to the Router AutoFix application. See "Installing Calibre AutoFix" on page 19 for details.

Required Input Files

The following input files are required:

- A sign-off Calibre rule file specific to your technology. The rule file must contain:
  - Sign-off design rules.
  - GDS layer mapping for your design if you do not have a GDS layer map file.
- DEF file containing the logical netlist and physical design information.
- LEF files that define any design abstracts (such as IP blocks) used in the layout.
- Calibre results database (RDB) file that contains errors to be fixed.
- Olympus Tcl technology file that specifies design rules for the router. This is required for designs that are 32 nm and below. This file is provided by the foundry and may

contain layer mapping between LEF and router layers to GDS layers used by Calibre. See "set_autofix_tech_file" on page 38 for complete details.

- Calibre AutoFix setup file to configure environment and tool options. The setup file must define the paths to all required files.

Optional Input Files

The following input files are optional unless noted otherwise:

- Fill file that contains fill shapes in GDS or OASIS®[1] format. This file is necessary if you want Calibre AutoFix to consider fill shapes during the run.
- Calibre DFM SmartFill rule file. The Calibre rule file for fill shapes.
- GDS or OASIS files used to write out a final layout. The GDS/OASIS files can define one or more of the following items:
    - Design abstracts (for example, IP blocks or cells).
    - GDS layer file that contains mapping from LEF and router tool layers to GDS layers used by Calibre. This file is required if the tech file does not have layer mapping information.
- Tcl rule map file. This file is only required if your process technology is not supported. Please contact your AE for more details.

Related Topics

Calibre AutoFix

Workflow

Input and Output Files

Setup File Commands

Syntax Conventions

The command descriptions use font properties and several metacharacters to document the command syntax.

The commands in "Setup File Commands" on page 33 follow the syntax conventions outlined in Table 1-2.

1. OASIS® is a registered trademark of Thomas Grebinski and licensed for use to SEMI®, San Jose. SEMI® is a registered trademark of Semiconductor Equipment and Materials International.

Table 1-2. Syntax Conventions

Convention	Description
Bold	Bold fonts indicate a required item.
Italic	Italic fonts indicate a user-supplied argument.
`Monospace`	Monospace fonts indicate a shell command, line of code, or URL. A bold monospace font identifies text you enter.
Underline	Underlining indicates either the default argument or the default value of an argument.
UPPercase	For certain case-insensitive commands, uppercase indicates the minimum keyword characters. In most cases, you may omit the lowercase letters and abbreviate the keyword.
[ ]	Brackets enclose optional arguments. Do not include the brackets when entering the command unless they are quoted.
{ }	Braces enclose arguments to show grouping. Do not include the braces when entering the command unless they are quoted.
‘ ’	Quotes enclose metacharacters that are to be entered literally. Do not include single quotes when entering braces or brackets in a command.
| or ||	Vertical bars indicate a choice between items. Do not include the bars when entering the command.
…	Three dots (an ellipsis) follows an argument or group of arguments that may appear more than once. Do not include the ellipsis when entering the command.
Example: **DEVice** {*element_name* [‘(’*model_name*‘)’]} ***device_layer*** {**pin_layer** [‘(’*pin_name*‘)’] …} [‘<’*auxiliary_layer*‘>’ …] [‘(’*swap_list*‘)’ …] [BY NET | BY SHAPE]	

Chapter 2
Getting Started with Calibre AutoFix

You can install and invoke Calibre AutoFix, and use the examples to demonstrate the usage flow for third-party tools.

Installing Calibre AutoFix

The installation of Calibre AutoFix requires the installation and configuration of the Router Autofix application.

Prerequisites

In order to run Calibre AutoFix, you must have a valid installation of the Calibre tree and the Router Autofix application. For instructions and requirements on installing the Calibre software tree, see "Installation Information" in the *Calibre Administrator's Guide*.

Procedure

1. Access the Support Center web site:

   https://support.sw.siemens.com

2. Navigate to the Nitro-SoC (Olympus-SoC) product web page and choose the **Downloads** tab.

3. From the "Restrict content to Version" dropdown list, select the most recent Olympus-SoC release, and click the Olympus-SoC download link.

4. Review and accept the Software Terms and Conditions.

5. On the Product Download web page:

   a. Select either the GUI installation pack or the Command line installation pack.

   b. Download and install the application in the usual manner.

   The self-extracting archive *.exe* contains a built-in help document that explains how to install the required software.

6. When you have finished installing Olympus-SoC, add the installation's binaries directory to your path.

C-shell:

```
setenv PATH ${MGC_HOME}/bin:<Olympus_install_dir>/sierra/bin/
:${PATH}
```

Bourne shell:

```
export PATH=$PATH:${MGC_HOME}/bin:<Olympus_install_dir>/sierra/bin
```

7. Test your installation:

   a. Write a simple setup file. See "Setup File Example" on page 22.

   b. Perform one of the example procedures in "Calibre AutoFix Usage Examples" on page 27.

Related Topics

Error and Warning Messages

Calibre AutoFix Invocation and Configuration

Specific commands, switches, and arguments are used to invoke Calibre AutoFix from within a third-party tool or from the command line.

You control the configuration of the Calibre AutoFix run with a setup file containing commands and options.

Ways to Invoke Calibre AutoFix

There are two scenarios for invoking Calibre AutoFix.

- Calibre AutoFix may be invoked from within Synopsys IC Compiler and Cadence Encounter by entering **calibre_autofix *setup_file*** at the tool command prompt.

  Note
  Third-party tool licenses are consumed while Calibre AutoFix is running.

- Calibre AutoFix may be invoked from the command line using the **calibre -autofix *setup_file*** command and switch.

Related Topics

Calibre AutoFix Usage Examples

Setup File Example

calibre -autofix

calibre -autofix

The **-autofix** switch is required to run Calibre AutoFix from the command line.

Usage

calibre -autofix {***setup_file*** [-nowait | -wait *n*] | **-version**}

Arguments

- **-autofix**

  Required switch that enables Calibre AutoFix.

- ***setup_file***

  Required argument that specifies the path to your Calibre AutoFix setup_file. See "Setup File Commands" on page 33 to view the supported commands and syntax.

- -nowait | -wait *n*

  Optional argument that specifies the length of time that Calibre AutoFix should queue for a license. The options are described as follows:

  -nowait — Calibre AutoFix queues for approximately 10 seconds before attempting to acquire substitute licenses. This option is equivalent to specifying -wait 0.

  -wait *n* — Calibre AutoFix queues for a maximum of *n* minutes. If the license is unavailable after queueing for *n* minutes, Calibre attempts to acquire any substitute licenses, or exits if no suitable substitution licenses are defined. The maximum value for *n* is 45000.

- **-version**

  Required switch for running tool version checking only. If specified, version information is output to the transcript window (Calibre version from $MGC_HOME, and router version from $PATH). If a version is unsupported, a warning is issued.

Related Topics

Error and Warning Messages

Calibre AutoFix Usage Examples

Setup File Example

A setup file is used to configure the settings and define the files for a Calibre AutoFix run.

Setup files are Tcl-based and must be written manually prior to invoking Calibre AutoFix. See "Setup File Format" on page 34 for a description of the format and commands used in the setup file.

The following figure shows an example setup file that you can modify to match your environment, design data, and tool settings.

Figure 2-1. Calibre AutoFix Setup File

```
#
###specify environment settings
set_cpus 2
set_output_dir autofix_output

####set to icc if using Synopsys IC Compiler
set_output_eco_file icc

####or

####if using ATopTech tools, uncomment next line
####set_output_eco_file atop -map "layer_map_file"

###specify rules, tech files, and options for your foundry
set_calibre_rule_file  "tsmc_28nm.rul"
set_tech_options -node tsmc28 -top_layer metal7
set_autofix_tech_file "tsmc28_rules.tcl" -metal_stack TECH_5x1y1z1a
set_gds_layer_map_file "fix.map"

###specify rdb file that contains your Calibre nmDRC results
set_input_rdb_file "ldpc_dec_pcd.drc.results"

###set design specific paths
set_lef_file "ram_block.lef io_pads.lef"
set_input_def_file "design_data.def.gz"

###Calibre AutoFix options
set_fill_notch true

###Select checks that produce DRC errors you want to fix (see .rdb file)
set_select_checks {Metal2_space1 Metal5*}

###Specify nets that you want to remain unchanged by Calibre AutoFix
set_dont_touch_nets {clk clka}

###Select which mode to run Calibre AutoFix (the default is DRC)
set_calibre_mode drc
#set_calibre_mode drc_all
#set_calibre_mode check_inputs
#set_calibre_mode lfd
#set_calibre_mode preroute
```

Related Topics

Calibre AutoFix Usage Examples

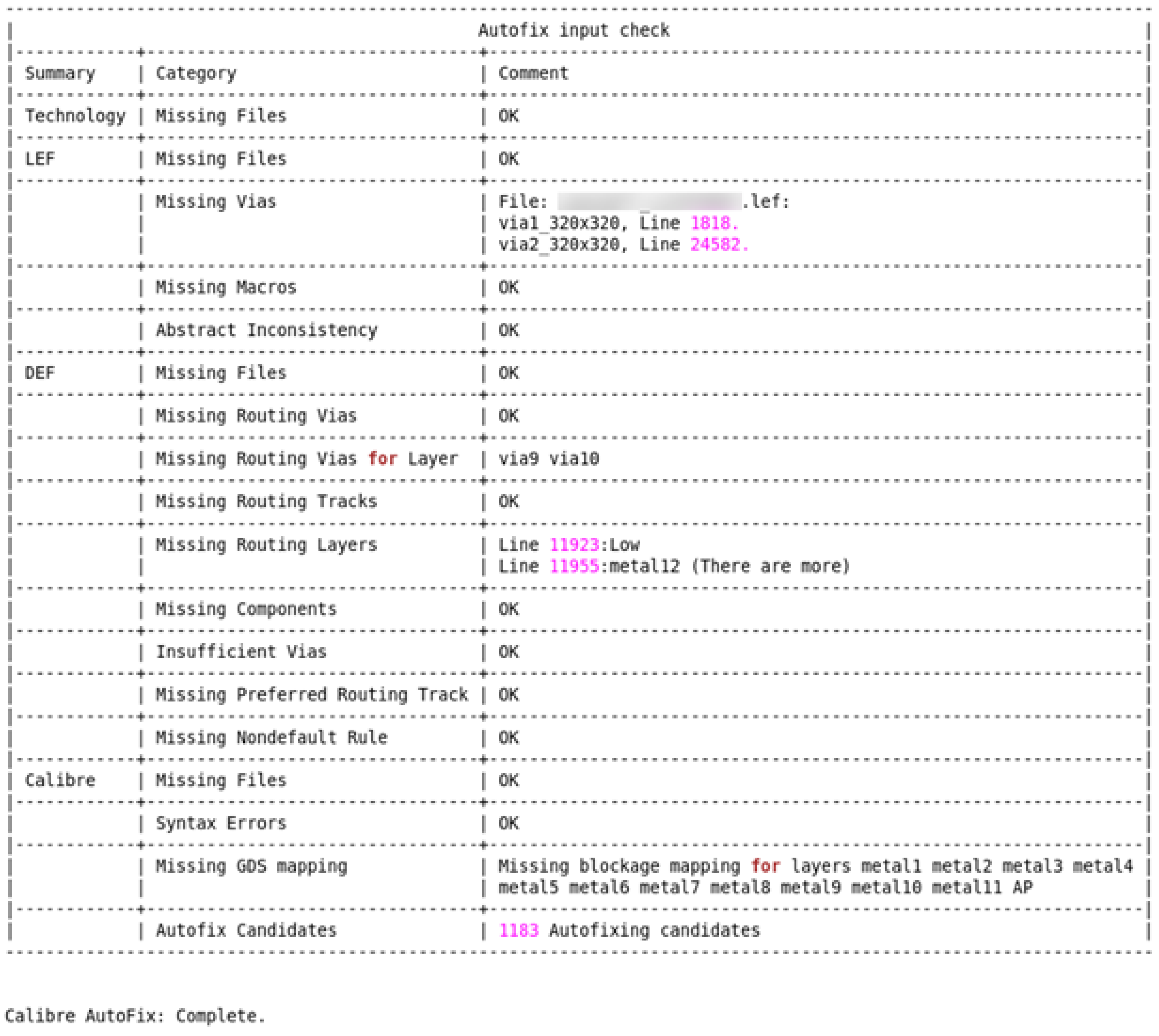
Input Check Table Example

The input check table is used as a sanity check to verify the inputs and report potential problems before a Calibre AutoFix run is performed.

The input check table is reported to the transcript when Calibre AutoFix is run in the check_inputs mode. See the set_calibre_mode command for the setup file specifications for running Calibre AutoFix in the check_inputs run mode.

The following figure shows an example of an input check table that is reported to the transcript.

Figure 2-2. Calibre AutoFix Input Check Table



```
-----------------------------------------------------------------------------------------------------------
|                                             Autofix input check                                         |
|------------+---------------------------------+----------------------------------------------------------|
| Summary    | Category                        | Comment                                                  |
|------------+---------------------------------+----------------------------------------------------------|
| Technology | Missing Files                   | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
| LEF        | Missing Files                   | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Vias                    | File: [redacted].lef:                                    |
|            |                                 | via1_320x320, Line 1818.                                 |
|            |                                 | via2_320x320, Line 24582.                                |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Macros                  | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Abstract Inconsistency          | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
| DEF        | Missing Files                   | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Routing Vias            | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Routing Vias for Layer  | via9 via10                                               |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Routing Tracks          | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Routing Layers          | Line 11923:Low                                           |
|            |                                 | Line 11955:metal12 (There are more)                      |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Components              | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Insufficient Vias               | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Preferred Routing Track | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing Nondefault Rule         | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
| Calibre    | Missing Files                   | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Syntax Errors                   | OK                                                       |
|------------+---------------------------------+----------------------------------------------------------|
|            | Missing GDS mapping             | Missing blockage mapping for layers metal1 metal2 metal3 metal4 |
|            |                                 | metal5 metal6 metal7 metal8 metal9 metal10 metal11 AP    |
|------------+---------------------------------+----------------------------------------------------------|
|            | Autofix Candidates              | 1183 Autofixing candidates                               |
-----------------------------------------------------------------------------------------------------------

Calibre AutoFix: Complete.
```

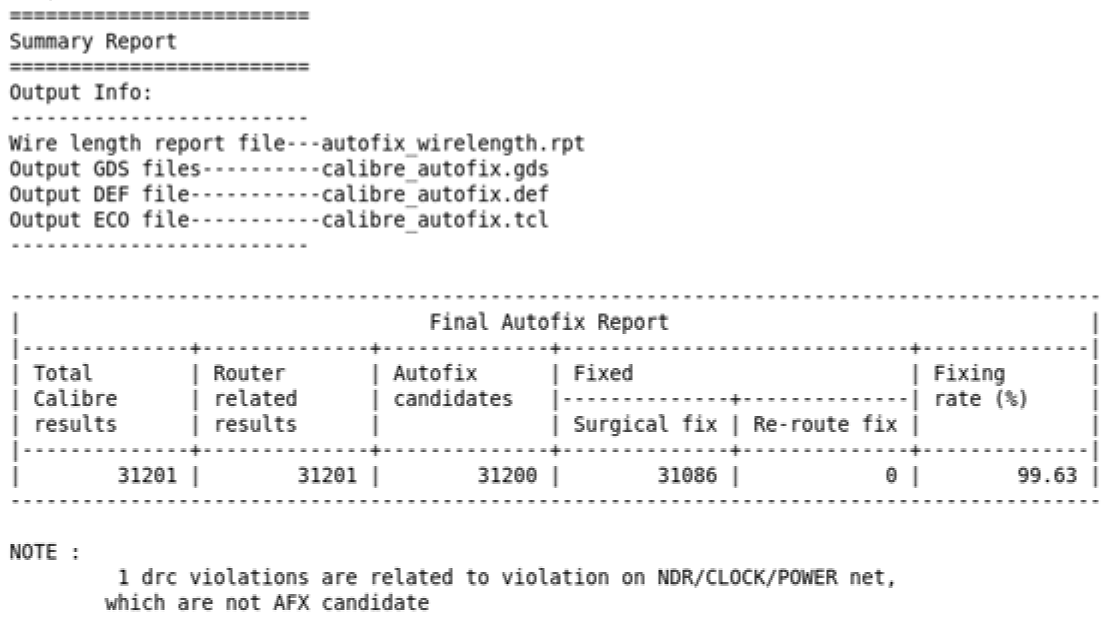
Summary and Final AutoFix Report Example

The Summary and Final AutoFix Report information shows the outputs, results, and fixing information for a Calibre AutoFix run.

The run Summary and Final AutoFix Report information is reported at the end of the transcript. This information includes a list of the generated output files, fixing information, and metrics for assessing a Calibre AutoFix run. Calibre DRC violations (results) and candidate information for Calibre AutoFix fixing are reported.

The following figure shows an example of the Summary and Final AutoFix Report information in the Calibre AutoFix transcript.

Figure 2-3. Calibre AutoFix Summary and Final AutoFix Report Example



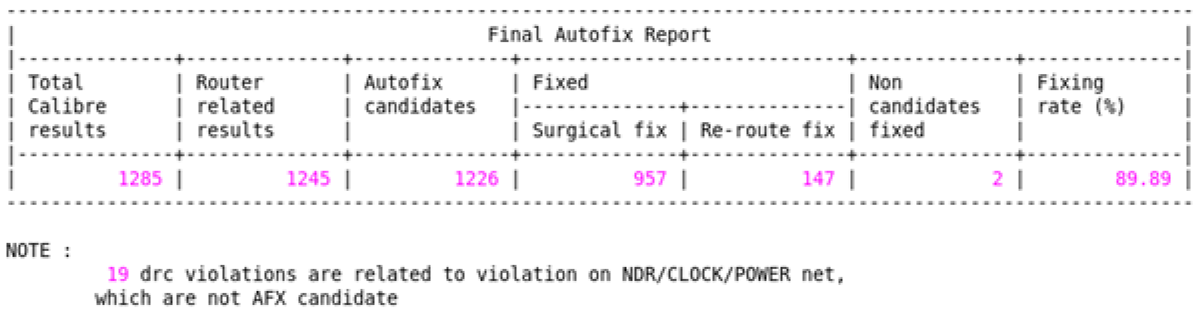
```
=========================
Summary Report
=========================
Output Info:
-------------------------
Wire length report file---autofix_wirelength.rpt
Output GDS files---------calibre_autofix.gds
Output DEF file----------calibre_autofix.def
Output ECO file----------calibre_autofix.tcl
-------------------------

----------------------------------------------------------------------------------------------
|                                   Final Autofix Report                                     |
|--------------+--------------+--------------+-----------------------------+-------------|
| Total        | Router       | Autofix      | Fixed                       | Fixing      |
| Calibre      | related      | candidates   |--------------+--------------| rate (%)    |
| results      | results      |              | Surgical fix | Re-route fix |             |
|--------------+--------------+--------------+--------------+--------------+-------------|
|        31201 |        31201 |        31200 |        31086 |            0 |       99.63 |
----------------------------------------------------------------------------------------------

NOTE :
         1 drc violations are related to violation on NDR/CLOCK/POWER net,
        which are not AFX candidate
```

The following figure shows an example of the Final AutoFix Report in the transcript with information for both candidates and non-candidates fixed. Any non-candidates fixed during the run are for the case when an NDR, clock, or power net violation is indirectly fixed due to the repair of violations between the mapped and selected candidates.

Figure 2-4. Final AutoFix Report with Non-Candidates Example



```
------------------------------------------------------------------------------------------------------
|                                        Final Autofix Report                                        |
|--------------+--------------+--------------+-----------------------------+--------------+--------------|
| Total        | Router       | Autofix      | Fixed                       | Non          | Fixing       |
| Calibre      | related      | candidates   |--------------+--------------| candidates   | rate (%)     |
| results      | results      |              | Surgical fix | Re-route fix | fixed        |              |
|--------------+--------------+--------------+--------------+--------------+--------------+--------------|
|         1285 |         1245 |         1226 |          957 |          147 |            2 |        89.89 |
------------------------------------------------------------------------------------------------------

NOTE :
         19 drc violations are related to violation on NDR/CLOCK/POWER net,
        which are not AFX candidate
```

The following table provides a description of the field contents of the Final AutoFix Report.

Table 2-1. Final AutoFix Report Content

Field	Description
Total Calibre Results	The number of DRC violations (results).

Table 2-1. Final AutoFix Report Content (cont.)

Field	Description
Router related results	The number of results that are mapped and selected minus the results inside the IP.
AutoFix candidates	The number of results with violations caused by objects that can be modified by the router. To be considered a candidate, the error type must be mapped and selected. Results that cannot be fixed by Calibre AutoFix are removed from this number.
Fixed — Surgical fix	The number of results fixed by the Calibre AutoFix “pre-route” stage.
Fixed — Re-route fix	The number of results fixed by the “rip-up and re-route” stage.
Non-candidates fixed	The number of “non-candidates” results (NDR, clock, or power net violations), fixed by Calibre AutoFix.
Fixing rate (%)	The percent of fixed Calibre AutoFix candidates minus any fixed non-candidates.

Calibre AutoFix Usage Examples

You can perform the procedures for running Calibre AutoFix for the supported place and route tools. Calibre AutoFix uses the error results contained within the specified Calibre results database file to correct the design. Nets or vias causing DRC errors are moved or regenerated iteratively until the design passes the rules in the sign-off Calibre nmDRC rule file.

Using Calibre AutoFix with Synopsys IC Compiler

You can invoke and run Calibre AutoFix with a design generated in Synopsys IC Compiler.

Prerequisites

- The requirements for running Calibre AutoFix are met. See “Requirements” on page 14.
- A layout design database is available to verify and modify with Calibre AutoFix.
- Full sign-off Calibre nmDRC verification is accessible.

Procedure

1. Open your layout design in Synopsys IC Compiler.
2. Ensure that the design passes any internal place and route DRC checks.
3. Run a full sign-off Calibre nmDRC verification on your design in the usual manner.
4. Carefully examine the DRC verification results in your Calibre results database file (RDB) and choose the checks to be added to the setup file for Calibre AutoFix to correct. The violations must be contained to your routing layers.
5. Open and review the Calibre AutoFix setup file in the text editor of your choice. See “Setup File Example” on page 22.

   In addition to specifying the Calibre sign-off rule file and technology files in your setup file, make sure you also do the following:

   - Define the DEF file to be exported in Step 7:

     ```
     set_input_def_file
     ```

   - Define the RDB file generated in Step 3:

     ```
     set_input_rdb_file
     ```

   - Add any checks chosen in Step 4:

     ```
     set_select_checks
     ```

For Synopsys IC Compiler, you must also instruct Calibre AutoFix to generate an ECO script using the following setup file command:

```
set_output_eco_file icc
```

The ECO script is sourced in Synopsys IC Compiler following a Calibre AutoFix run.

6. Make the necessary edits to the setup file and save it in the directory from which you plan to run Calibre AutoFix.

   Figure 2-1 on page 23 shows a commented setup file that you can modify for your design.

7. Export your design as a DEF file in one of two ways:

   - Use the Synopsys IC Compiler internal write_def command.
   - Choose **File** > **Export** > **Write DEF** and specify any desired options.

   Note

   The DEF file is the primary design that you wish to correct with Calibre AutoFix. This is the DEF file defined in Step 5. You may specify multiple DEF files in the setup file.

8. Use either of the following procedures to run Calibre AutoFix:

   - **Invoke Calibre AutoFix from within Synopsys IC Compiler:**

     i. Invoke Calibre AutoFix by entering the following command at the Synopsys IC Compiler command prompt:

     ```
     calibre_autofix path_to_setup_file
     ```

     Calibre AutoFix writes all corrections to a Synopsys ECO Tcl script named *calibre_autofix.tcl* in your output directory and automatically sources the file in Synopsys IC Compiler. The script instructs Synopsys IC Compiler to perform all the geometrical fixes generated and validated by Calibre AutoFix.

   - **Invoke Calibre AutoFix from the command line:**

     i. Invoke Calibre AutoFix by entering this command at the command line:

     ```
     calibre -autofix path_to_setup_file
     ```

     Calibre AutoFix writes all corrections to a Synopsys ECO Tcl script named *calibre_autofix.tcl* in your output directory. The script must be manually sourced on the original design database. The script instructs Synopsys IC Compiler to perform all the geometrical fixes generated and validated by Calibre AutoFix.

     ii. Load your original design in Synopsys ICC Compiler.

     iii. Manually source the *calibre_autofix.tcl* ECO file to apply the changes to your design database.

9. View the changes in Synopsys IC Compiler and run any appropriate internal design tool checks.

Results

Calibre AutoFix was invoked to correct a design generated in Synopsys IC Compiler. Calibre AutoFix used the DRC results in the RDB file to perform iterative sign-off DRC verification for each geometrical adjustment in the exported DEF file.

If Calibre AutoFix was invoked from within Synopsys IC Compiler, the commands required to reproduce the fixes in Synopsys IC Compiler were exported to the *calibre_autofix.tcl* ECO Tcl file and automatically sourced in Synopsys IC Compiler.

If you ran Calibre AutoFix from the command line, you applied the changes to your design database by loading the original design in Synopsys IC Compiler and manually sourcing the *calibre_autofix.tcl* ECO file.

Related Topics

Error and Warning Messages

Setup File Commands

Using Calibre AutoFix with Cadence Encounter Digital Implementation System

Using Calibre AutoFix with Cadence Encounter Digital Implementation System

You can invoke and run Calibre AutoFix using a Cadence Encounter design database.

Prerequisites

- The requirements for running Calibre AutoFix are met. See "Requirements" on page 14.
- A layout design database is available to verify and modify with Calibre AutoFix.
- Full sign-off Calibre nmDRC verification is accessible.

Procedure

1. Open your layout design in Cadence Encounter.
2. Ensure that the design passes any internal place and route DRC checks.
3. Run a full sign-off Calibre nmDRC verification on your design in the usual manner.
4. Carefully examine the DRC verification results in your Calibre results database file (RDB) and choose the checks to be added to the setup file for Calibre AutoFix to correct. The violations must be contained to your routing layers.
5. Open and review the Calibre AutoFix setup file in the text editor of your choice. See "Setup File Example" on page 22.

In addition to specifying the Calibre sign-off rule file and technology files in your setup file, make sure you also do the following:

- Define the DEF file to be exported in Step 7:

  ```
  set_input_def_file
  ```

- Define the RDB file generated in Step 3:

  ```
  set_input_rdb_file
  ```

- Add any checks chosen in Step 4:

  ```
  set_select_checks
  ```

6. Make the necessary edits to the setup file and save it in the directory from which you plan to run Calibre AutoFix.

   Figure 2-1 on page 23 shows a commented setup file that you can modify for your design.

7. Use either of the following procedures to run Calibre AutoFix.

   - **Invoke Calibre AutoFix from within Cadence Encounter:**

     i. Export your design as a DEF file using the internal **defOut** command with any desired options.

     Note
     The DEF file is the primary design that you wish to correct with Calibre AutoFix. This is the DEF file defined in Step 5. You may specify multiple DEF files in the setup file.

     ii. Invoke Calibre AutoFix by entering the following command at the Cadence Encounter command prompt:

     ```
     calibre_autofix path_to_setup_file
     ```

     Calibre AutoFix corrects the specified rule results on the exported DEF file and automatically imports the changes back into the Cadence Encounter database.

   - **Invoke Calibre AutoFix from the command line:**

     i. Export your design as a DEF file using the internal **defOut** command with any desired options.

     Note
     The DEF file is the primary design that you wish to correct with Calibre AutoFix. This is the DEF file defined in Step 5. You may specify multiple DEF files in the setup file.

     ii. Exit Cadence Encounter.

iii. Invoke Calibre AutoFix by entering this command at the command line:

```
calibre -autofix path_to_setup_file
```

Calibre AutoFix writes all corrections to a new DEF file named *calibre_autofix.def* contained in your Calibre AutoFix output directory.

iv. Invoke Cadence Encounter and load your original design.

v. Load the corrected DEF file by using the **defIn** command with the -nets option as follows:

```
defIn calibre_autofix.def -nets
```

The **defIn** command imports the routing changes for each net contained in the *calibre_autofix.def* file into your existing design database. This ensures that only the NETS section of the DEF file is read.

8. View the changes in Cadence Encounter and run any appropriate internal design tool checks.

Results

Calibre AutoFix was invoked to correct a design generated in Cadence Encounter. Calibre AutoFix used the DRC results in the RDB file to perform iterative sign-off DRC verification for each geometrical adjustment in the exported DEF file.

If Calibre AutoFix was invoked from within Cadence Encounter, the changes were automatically applied to your design database.

If you ran Calibre AutoFix from the command line, the design was exported to a new DEF file, and the fixed output was manually imported into Cadence Encounter using the **defIn** -nets command.

Related Topics

Error and Warning Messages

Using Calibre AutoFix with Synopsys IC Compiler

Setup File Commands

Chapter 3
Setup File Commands

You can use specific commands to configure your Calibre AutoFix setup file.

Setup File Format

Input file for: Calibre AutoFix invocation

Required: Yes

The setup file is a Tcl-based file that you create manually and use to specify the rules, files, commands, and options for configuring a Calibre AutoFix run.

Format

A Calibre AutoFix setup file must conform to the following formatting and syntax rules:

- ASCII text
- One command plus option(s) per line
- Create comments for a full line by beginning the line with a pound sign (#)

See "Setup File Example" on page 22 for a sample setup file that you can modify for your environment, design data, and settings. The setup file is required when invoking Calibre AutoFix. Refer to "Calibre AutoFix Invocation and Configuration" on page 21 for the ways to invoke Calibre AutoFix within a third-party tool or a command line.

Parameters

The setup file commands use specific syntax described in "Syntax Conventions" on page 16. The Required column of the following setup file command table indicates whether the command is required in the setup file.

Table 3-1. Command List for Calibre AutoFix Setup File

Name	**Description**	**Required?**
Process and Library Setup Commands		
set_autofix_tech_file	Defines the path to an Olympus-SoC Tcl rules file.	Yes, for 32 nm and below
set_calibre_mode	Sets the desired run mode for Calibre AutoFix.	No
set_calibre_rule_file	Defines the path to the sign-off Calibre rule file. The rule file must match the rule file used to generate the results database.	Yes
set_gds_layer_map_file	Defines the path to a GDS layer map file.	No
set_lef_file	Defines the path to the LEF files used in your design.	Yes
set_tech_options	Defines the routing technology for your design.	Yes
Design Input Commands		
set_input_def_file	Defines the paths to the DEF files used in your design.	Yes

Table 3-1. Command List for Calibre AutoFix Setup File (cont.)

Name	Description	Required?
set_input_rdb_file	Defines the path to an existing Calibre results database (RDB) file.	Yes
Output Commands		
set_output_def_file	Defines the name of the DEF generated by Calibre AutoFix that replaces the default DEF file name.	No
set_output_def_units	Specifies the database units of the generated DEF file.	No
set_output_dir	Defines the path of the output directory where all files are written that replaces the default output directory path.	No
set_output_eco_file	Instructs Calibre AutoFix to generate an Engineering Change Order (ECO) Tcl file for Synopsys IC Compiler or ATopTech tools.	Yes, if using Synopsys IC Compiler or ATopTech tools
set_output_gds_file	Defines the name of the generated GDS file.	No
set_log_file	Defines the name of the generated log file that replaces the default log file name.	No
set_output_rdb_file	Defines the name of the generated Calibre results database that replaces the default file name.	No
set_report_shorts_on_pg_nets	Specifies whether short violations for power and ground nets are reported.	No
set_wire_length_rpt	Defines the name of the generated wire length report file that replaces the default report name.	No
Fixing Commands		
set_blockage_spacing_type	Sets the spacing type for blockages.	No
set_cpus	Defines the number of CPUs to use during a Calibre AutoFix run.	No
set_dont_care_routing_layers	Defines routing layer name(s) for Calibre AutoFix to ignore.	No
set_dont_touch_nets	Defines nets that are not modified during a Calibre AutoFix run.	No
set_fill_notch	Defines whether step and notch violations are fixed by re-routing or inserting fill.	No

Table 3-1. Command List for Calibre AutoFix Setup File (cont.)

Name	Description	Required?
set_fix_custom_signal	Allows Calibre AutoFix to change pre-route wires to detail wires when fixing errors.	No
set_fix_min_area	Defines whether minimum area violations are fixed during routing.	No
set_fix_ndr_nets	Specifies whether nets with a non-default rule (NDR) property are modified during a Calibre AutoFix run.	No
set_ignore_cell_overlap	Defines whether cell placement overlaps are checked and reported	No
set_m1_routing	Defines whether routing is allowed on the metal 1 layer.	No
set_route_options	Defines custom options for the router during a Calibre AutoFix run.	No
set_select_checks	Selects which checks to fix during a Calibre AutoFix run.	No
set_silent	Specifies whether to print detailed messages to the transcript during a run.	No
set_skip_cell_overlap	Specifies whether to skip the checking of overlapping cells in DRC fixing modes.	No
set_unselect_checks	Unselects checks from a Calibre AutoFix run.	No

Process and Library Setup Commands

Calibre AutoFix uses commands in the setup file for configuring the technology, library, and other setup options.

The setup file commands use specific syntax described in "Syntax Conventions" on page 16. The Required column of Table 3-2 indicates whether the command is required in the setup file.

Table 3-2. Process and Library Setup Commands

Name	Description	Required?
set_autofix_tech_file	Defines the path to an Olympus-SoC Tcl rules file.	Yes, for 32 nm and below
set_calibre_mode	Sets the desired run mode for Calibre AutoFix.	No
set_calibre_rule_file	Defines the path to the sign-off Calibre rule file. The rule file must match the rule file used to generate the results database.	Yes
set_gds_layer_map_file	Defines the path to a GDS layer map file.	No
set_lef_file	Defines the path to the LEF files used in your design.	Yes
set_tech_options	Defines the routing technology for your design.	Yes

set_autofix_tech_file

Process and library setup command for: Calibre AutoFix setup file

Required: Yes, for design technologies 32 nm and below

Defines the path to an Olympus-SoC Tcl rules file.

Usage

set_autofix_tech_file ***techFileName*** **-metal_stack** ***metalStackName***

Arguments

- ***techFileName***

  Required argument that specifies the path to an Olympus DRC technology file that contains internal routing rules for the process technology that you are using in your design. This file must be obtained from your foundry.

- **-metal_stack** ***metalStackName***

  Required argument and parameter that specifies the metal stack configuration of your technology. The metal stack configuration must match the type of process defined in your sign-off Calibre rule file.

Description

Use this command to define the path to your Olympus DRC Tcl technology file. A technology file is a Tcl script that specifies internal place and route technology rules used for routing. This command is required for designs using technology nodes that are 32 nm and below. Calibre AutoFix performs syntax checking of this file. If an error is encountered, the tool outputs the error to the transcript and exits.

You can obtain the Olympus DRC technology file from your foundry.

Examples

Define the Olympus-SoC Tcl rule file used for a specified process technology and metal stack:

```
set_autofix_tech_file ./tech/tsmc28_rules.tcl -metal_stack TECH_5x1y1z1a
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_calibre_mode

Process and library setup command for: Calibre AutoFix setup file

Required: No

Sets the desired run mode for Calibre AutoFix.

Usage

set_calibre_mode {**drc** | **preroute** [-mode {power | signal}]
| **drc_all** [-skip {preroute | preroute_signal | signal}] | **check_inputs**}

Arguments

- **drc**

  Required keyword that specifies Calibre nmDRC mode for Calibre AutoFix. If **drc** is specified, Calibre AutoFix verifies and corrects DRC rule checks only. This is the default behavior if the set_calibre_mode command is not specified.

- **preroute**

  Required keyword that specifies that pre-route wires and vias causing DRC violations are removed or modified to correct the error.

- -mode {power | signal}

  Optional keywords that specify the type of pre-route wires to fix. This option can only be specified with the **preroute** argument. The -mode options are described as follows:

  power — Argument that instructs Calibre AutoFix to correct errors in power/ground wires. This is the default.

  signal — Argument that instructs Calibre AutoFix to correct errors in signal wires.

- **drc_all**

  Required keyword that specifies that Calibre AutoFix runs in Calibre nmDRC mode for pre-route (power), pre-route signal, and signal fixing. The default is to fix all.

- -skip {preroute | preroute_signal | signal}

  Optional keywords that specify the stage of fixing to skip during complete drc flow. This option can only be specified with the **drc_all** keyword. The -skip options are described as follows:

  preroute — Argument that instructs Calibre AutoFix to skip fixing of errors in power/ground wires during the preroute fixing stage.

  preroute_signal — Argument that instructs Calibre AutoFix to skip fixing of errors in signal wires during the preroute fixing stage.

  signal — Argument that instructs Calibre AutoFix to skip fixing of errors in signal wires during the detailed routing stage.

- **check_inputs**

  Required keyword and mode that verifies inputs before a Calibre AutoFix run is performed. When this mode is specified, any issues that may prevent a successful run are reported to the transcript in an input check table.

Description

Use this command in your setup file to define the desired run mode for Calibre AutoFix. You can correct DRC or pre-route errors and specify the types of nets for Calibre AutoFix to fix. The default run mode is **drc.** When you specify **check_inputs** mode, the following items are verified and reported:

- Technology, LEF, DEF, GDS layer map, and Calibre files
- Macros, components, vias, routing vias, tracks, and layers
- Abstract consistency, macro/pad cell opens and shorts to macro blockage
- Syntax

See "Input Check Table Example" on page 23 for an example of an input check table report.

Examples

Example 1

Run Calibre AutoFix in DRC mode:

```
set_calibre_mode drc
```

Example 2

Run Calibre AutoFix to correct violations from signal pre-route wires and vias:

```
set_calibre_mode preroute -mode signal
```

Example 3

Run Calibre AutoFix in **check_inputs** mode to verify the inputs and report potential problems before the Calibre AutoFix run is performed. The **check_inputs** mode outputs an input check table to the transcript and a macro/pad short and open results database file:

```
set_calibre_mode check_inputs
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_calibre_rule_file

Process and library setup command for: Calibre AutoFix setup file

Required: Yes

Defines the path to the sign-off Calibre rule file. The rule file must match the rule file used to generate the results database.

Usage

set_calibre_rule_file ***ruleFileName***

Arguments

- ***ruleFileName***

  Required argument that specifies the path to the sign-off Calibre rule file.

Examples

Define the Calibre rule file used for DRC sign-off checking:

```
set_calibre_rule_file ./RULES/calibreDRC_rul.encrypt
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_gds_layer_map_file

Process and library setup command for: Calibre AutoFix setup file

Required: No

Defines the path to a GDS layer map file.

Usage

set_gds_layer_map_file ***layerMapName***

Arguments

- ***layerMapName***

  Required argument that specifies the path to the file that contains layer map information.

Description

Use this command to define the path to the GDS layer map file. For any technology node, the Calibre layers and place and route layers must be mapped (either in the GDS abstracts, or with a GDS layer map file).

When using LEF-based technology files, all metal and via layers used in your layout database must be mapped to the correct GDS layers before writing out a GDS file or invoking Calibre AutoFix. The layer map is a Tcl-formatted file that usually contains a set of set_gds_layer_map commands used by the routing tool. It may also contain the report_gds_layer_map command. Typically, you may obtain a GDS layer map file from your foundry.

Examples

Specify the file containing the GDS layer map information for your technology node:

```
set_gds_layer_map_file ./layer_map.tcl
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_lef_file

Process and library setup command for: Calibre AutoFix setup file

Required: Yes

Defines the path to the LEF files used in your design.

Usage

set_lef_file *lef_file* [*lef_file*…]

Arguments

- ***lef_file*** [*lef_file* …]

  Required argument that specifies the paths to the LEF files that compose your layout design. At least one ***lef_file*** must be specified.

Description

Use this command to define the pathnames to the LEF files for abstract blocks used in your layout design. A technology LEF file should only be specified when using technology nodes of 45 nm and above.

Examples

Specify the LEF files for the IP blocks in a 45 nm or above technology node design:

```
set_lef_file ./LEF/ram_block.lef ./LEF/mux.lef
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_tech_options

Process and library setup command for: Calibre AutoFix setup file

Required: Yes

Defines the routing technology for your design.

Usage

set_tech_options -node *techNodeName*
[-map *ruleMapFile*] [-top_layer *topLayerName*] [-setup_files *setupFilesName*]

Arguments

- **-node *techNodeName***

  Required argument and keyword that specify the foundry process technology used for the layout design. Contact your Siemens representative for assistance with technology node specification.

- -map *ruleMapFile*

  Optional argument and parameter that specify the path to a Tcl file that contains a list of mappings between Calibre rule checks and the corresponding place and route layer and DRC rule class.

  This file is required for LFD mode. For DRC mode, it is only required for certain process technologies specified with the **-node** argument. Contact your Siemens representative for assistance with technology node specification.

- -top_layer *topLayerName*

  Optional argument and parameter that specify the top routing layer of the layout design.

- -setup_files *setupFilesName*

  Optional argument and parameter that is required for certain process technology nodes. An error message is generated if this file is not specified for these technology nodes. Contact your Siemens representative for assistance with technology node specification. This file includes the values for Calibre environment variables defined in the rule deck and is sourced when Calibre AutoFix is invoked.

Note

The file referred to by the -setup_files argument is different than the Calibre AutoFix setup file.

Description

Use this command to define the routing technology used in your layout design. Specify this command with the other rule and technology commands in the Calibre AutoFix setup file.

Examples

Define the sign-off rule DRC rule followed by the specific routing technology for the design:

```
set_calibre_rule_file ./RULES/calibreDRC_32nm.rul
set_tech_options -node <my_tech32> -top_layer metal8
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

Design Input Commands

The Calibre AutoFix setup file requires two design input commands to read in the top level design and the Calibre results database.

The setup file commands use specific syntax described in "Syntax Conventions" on page 16. The Required column of Table 3-3 indicates whether the command is required in the setup file.

Table 3-3. Design Input Commands

Name	Description	Required?
set_input_def_file	Defines the paths to the DEF files used in your design.	Yes
set_input_rdb_file	Defines the path to an existing Calibre results database (RDB) file.	Yes

set_input_def_file

Design input command for: Calibre AutoFix setup file

Required: Yes

Defines the paths to the DEF files used in your design.

Usage

set_input_def_file *def_file* [*def_file* …]

Arguments

- ***def_file*** [*def_file* …]

  Required argument that specifies the paths to the DEF files that compose your place and route design. At least one ***def_file*** must be specified. DEF 5.6 and 5.7 are supported.

Description

The DEF files contain the logical and physical information for your design, where the ***def_file*** list defines a routed design with errors that require correction. Routing layers and tracks for preferred routing direction must be defined in the input DEF files. The valid routing layers are metal 2, … up to -top_layer, where -top_layer is the top routing layer in your design. See "set_m1_routing" on page 72 for the case of including metal 1 routing in the valid routing layers.

Examples

Specify the input DEF file for the design. More than one DEF file may be specified:

```
set_input_def_file ./top_level.def
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_input_rdb_file

Design input command for: Calibre AutoFix setup file

Required: Yes

Defines the path to an existing Calibre results database (RDB) file.

Usage

set_input_rdb_file ***rdbFileName*** [-origin {center | lower_left}]

Arguments

- ***rdbFileName***

  Required argument that specifies the path to a Calibre RDB input error file. The RDB file must contain violations to fix.

- -origin {center | lower_left }

  Optional argument that specifies how Calibre AutoFix processes the Calibre RDB input error file for design fixing.

  center — Use if the origin of the DEF file is different than the GDS origin, and the origin of the GDS is defined at the center of the die.

  lower_left — Use if the origin of the DEF file and the GDS are the same. This is the default.

Description

Use this command to define the path to the Calibre RDB input error file. The file must be generated from the same design and sign-off rule file that you plan to use with Calibre AutoFix. Calibre AutoFix uses this file to resolve violations in the layout design. If the -origin option is specified, Calibre AutoFix automatically applies the specified origin to the Calibre RDB input error file at the beginning of the run.

Examples

Example 1

Define the Calibre input RDB file with violations to be fixed:

```
set_input_rdb_file calibre_results.rdb
```

Example 2

Define the Calibre input RDB file with violations to be fixed and offset the RDB file error markers by the specified origin:

```
set_input_rdb_file calibre_results.rdb -origin center
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

Output Commands

Calibre AutoFix uses commands in the setup file to configure the output file options and locations.

The setup file commands use specific syntax described in "Syntax Conventions" on page 16. The Required column of Table 3-4 indicates whether the command is required in the setup file. If a command is not specified, default behavior results unless noted otherwise.

Table 3-4. Output Commands

Name	Description	Required?
set_output_def_file	Defines the name of the DEF generated by Calibre AutoFix that replaces the default DEF file name.	No
set_output_def_units	Specifies the database units of the generated DEF file.	No
set_output_dir	Defines the path of the output directory where all files are written that replaces the default output directory path.	No
set_output_eco_file	Instructs Calibre AutoFix to generate an Engineering Change Order (ECO) Tcl file for Synopsys IC Compiler or ATopTech tools.	Yes, if using Synopsys IC Compiler or ATopTech tools
set_output_gds_file	Defines the name of the generated GDS file.	No
set_log_file	Defines the name of the generated log file that replaces the default log file name.	No
set_output_rdb_file	Defines the name of the generated Calibre results database that replaces the default file name.	No
set_report_shorts_on_pg_nets	Specifies whether short violations for power and ground nets are reported.	No
set_wire_length_rpt	Defines the name of the generated wire length report file that replaces the default report name.	No

set_output_def_file

Output command for: Calibre AutoFix setup file

Required: No

Defines the name of the DEF generated by Calibre AutoFix that replaces the default DEF file name.

Usage

set_output_def_file ***defFileName***

Arguments

- ***defFileName***

  Required argument that specifies the name of your output DEF file.

Description

Use this command to define the name of the output DEF file. The output DEF contains the corrected layout design that is the output of Calibre AutoFix. The output file is generated in the directory specified by the set_output_dir setup file command. If this command is not specified, the default output DEF file name is *calibre_autofix.def*.

Examples

Specify the name of the corrected output DEF file for the design:

```
set_output_def_file top_level_fixed.def
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_output_def_units

Output command for: Calibre AutoFix setup file

Required: No

Specifies the database units of the generated DEF file.

Usage

set_output_def_units {**100** | **200** | **1000** | **2000**}

Arguments

- {**100** | **200** | **1000** | **2000**}

  Required value that specifies the number of database units in one user unit for the output DEF file.

Description

Use this command to specify the database units that are written to the output DEF file specified by the set_output_def_file command. The value controls the
UNITS DISTANCE MICRONS *value* statement in the DEF file.

Examples

Define 2000 database units in one user unit for the output DEF file:

```
set_output_def_units 2000
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_output_dir

Output command for: Calibre AutoFix setup file

Required: No

Defines the path of the output directory where all files are written that replaces the default output directory path.

Usage

set_output_dir *dirName*

Arguments

- ***dirName***

  Required argument that specifies the path to your output directory.

Description

Use this command to define the path to the output directory. If a directory with the same name already exists, Calibre AutoFix adds the output files underneath the existing directory, overwriting existing files in the event of a conflict. The application aborts if a file is specified instead of a directory. If this command is not specified, a directory called *Autofix* is created in your working directory.

Examples

Specify the directory path for the output files:

```
set_output_dir ./AutoFix_Output
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_output_eco_file

Output command for: Calibre AutoFix setup file

Required: Yes, when using Calibre AutoFix with Synopsys IC Compiler or ATopTech tools

Instructs Calibre AutoFix to generate an Engineering Change Order (ECO) Tcl file for Synopsys IC Compiler or ATopTech tools.

Usage

set_output_eco_file {**icc** | **atop -map** ***LayerMapFile***}

Arguments

- **icc**

  Specifies that an ECO Tcl file for Synopsys IC Compiler is created after a Calibre AutoFix run. This parameter is required if you are using Synopsys IC Compiler.

- **atop**

  Specifies that an ECO Tcl file containing commands for use with ATopTech tools is created after a Calibre AutoFix run. This parameter is required if you are using ATopTech tools.

- **-map** ***LayerMapFile***

  Required argument and parameter that specify the path to a file that maps DEF layer names to ATopTech layer names. This argument is only required with the **atop** argument. The map file must be in the following format:

  DEF_layer1_name ATOP_layer1_name

  DEF_layer2_name ATOP_layer2_name

Description

The **set_output_eco_file** command is only required if you are using Synopsys IC Compiler or ATopTech tools; the command is not required for other tools.

Calibre AutoFix with Synopsys IC Compiler

If you are using Calibre AutoFix with Synopsys IC Compiler, you must set this command to **icc**. When set to **icc**, Calibre AutoFix generates an ECO Tcl file in the form of a script that contains Synopsys IC Compiler commands used to fix routed databases.

Calibre AutoFix writes the ECO Tcl file, *calibre_autofix.tcl*, to the default output directory or to a specified output directory using the set_output_dir setup file command. The Calibre AutoFix ECO Tcl file corrections are applied to the routed design database in one of two ways:

- If Calibre AutoFix was invoked outside of Synopsys IC Compiler, you must open your routed design in Synopsys IC Compiler and source *calibre_autofix.tcl* to apply any corrected DRC errors.

- If you launched Calibre AutoFix from within Synopsys IC Compiler, the ECO Tcl file is sourced automatically.

See "Using Calibre AutoFix with Synopsys IC Compiler" on page 27 for further details.

Calibre AutoFix with ATopTech tools

If you are using Calibre AutoFix with ATopTech place and route tools, you must set this command to **atop -map *LayerMapFile***. When set to **atop -map *LayerMapFile***, Calibre AutoFix generates an ECO Tcl file in the form of a script that contains ATopTech tool commands used to fix routed databases.

Calibre AutoFix writes the ECO Tcl file, *calibre_autofix.tcl*, to the default output directory or to a specified output directory using the set_output_dir setup file command. The Calibre AutoFix ECO Tcl file corrections are applied to the routed design database by sourcing the *calibre_autofix.tcl* ECO script from within the ATopTech tool.

Examples

Example 1

Generate an ECO Tcl file with routing corrections for Synopsys IC Compiler:

```
set_output_eco_file icc
```

Example 2

Generate an ECO Tcl file with routing corrections for ATopTech tools and ATopTech layer mapping:

```
set_output_eco_file atop -map layer_map_file
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_output_gds_file

Output command for: Calibre AutoFix setup file

Required: No

Defines the name of the generated GDS file.

Usage

set_output_gds_file ***gdsFileName***

Arguments

- ***gdsFileName***

  Required argument that specifies the name of your output GDS file.

Description

Use this command to define the name of the output GDS file. The output GDS contains the corrected layout design that is the output of Calibre AutoFix. The output file is generated in the directory specified by the set_output_dir setup file command. If this command is not specified, no GDS file is saved.

Examples

Specify the name of the GDS file for the corrected output layout:

```
set_output_gds_file top_level_fixed.gds
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_log_file

Output command for: Calibre AutoFix setup file

Required: No

Defines the name of the generated log file that replaces the default log file name.

Usage

set_log_file *logFileName*

Arguments

- ***logFileName***

  Required argument that specifies the name of the output log file that Calibre AutoFix generates.

Description

Use this command to define the path to the log file for Calibre AutoFix. The default name is *calibre_autofix.log*. The output file is generated in the directory specified by the set_output_dir setup file command.

Examples

Specify the name of the Calibre AutoFix log file in the output directory:

```
set_log_file autofix.log
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_output_rdb_file

Output command for: Calibre AutoFix setup file

Required: No

Defines the name of the generated Calibre results database that replaces the default file name.

Usage

set_output_rdb ***rdbFileName***

Arguments

- ***rdbFileName***

  Required argument that specifies the name of the output results database file that Calibre AutoFix generates.

Description

Use this command to define the name of your output results database (RDB) file. The RDB file is generated from a Calibre AutoFix run. The output file is saved in the directory specified by the set_output_dir setup file command. If this command is not specified, the results are written by default to *calibre_autofix.rdb*.

Examples

Define the Calibre AutoFix generated RDB output file:

```
set_output_rdb_file calibre_autofix.rdb
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_report_shorts_on_pg_nets

Output command for: Calibre AutoFix setup file

Required: No

Specifies whether short violations for power and ground nets are reported.

Usage

set_report_shorts_on_pg_nets {**<u>false</u>** | **true**}

Arguments

- **<u>false</u>**

  Required keyword that disables the reporting of shorts for power and ground nets. This is the default.

- **true**

  Required keyword that enables the reporting of shorts for power and ground nets.

Description

Use this command to specify whether shorts for power and ground nets are reported in the generated *OPENSandSHORTS.rdb* file and the input check table from a Calibre AutoFix run. The default is to exclude the reporting of shorts for power and ground nets.

Examples

Specify that shorts are reported for power and ground nets:

```
set_report_shorts_on_pg_nets true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_wire_length_rpt

Output command for: Calibre AutoFix setup file

Required: No

Defines the name of the generated wire length report file that replaces the default report name.

Usage

set_wire_length_rpt ***rptFileName***

Arguments

- ***rptFileName***

  Required argument that specifies the name of the output wire length report file that Calibre AutoFix generates.

Description

Use this command to define the name of the wire length report file. The file contains information on the length of all modified nets in the layout. The output file is generated in the directory specified by the set_output_dir setup file command. If this command is not specified, the report is written by default to *autofix_wirelength.rpt*.

Examples

Define the name of the output wire length report file:

```
set_wire_length_rpt wire_length.rpt
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

Fixing Commands

Calibre AutoFix uses commands in the setup file for configuring fixing behavior at run time.

The setup file commands use specific syntax described in "Syntax Conventions" on page 16. The Required column of Table 3-5 indicates whether the command is required in the setup file. Commands that are not required, may be added to the setup file for additional control of the Calibre AutoFix run.

Table 3-5. Fixing Commands

Name	Description	Required?
set_blockage_spacing_type	Sets the spacing type for blockages.	No
set_cpus	Defines the number of CPUs to use during a Calibre AutoFix run.	No
set_dont_care_routing_layers	Defines routing layer name(s) for Calibre AutoFix to ignore.	No
set_dont_touch_nets	Defines nets that are not modified during a Calibre AutoFix run.	No
set_fill_notch	Defines whether step and notch violations are fixed by re-routing or inserting fill.	No
set_fix_custom_signal	Allows Calibre AutoFix to change pre-route wires to detail wires when fixing errors.	No
set_fix_min_area	Defines whether minimum area violations are fixed during routing.	No
set_fix_ndr_nets	Specifies whether nets with a non-default rule (NDR) property are modified during a Calibre AutoFix run.	No
set_ignore_cell_overlap	Defines whether cell placement overlaps are checked and reported.	No
set_m1_routing	Defines whether routing is allowed on the metal 1 layer.	No
set_route_options	Defines custom options for the router during a Calibre AutoFix run.	No
set_select_checks	Selects which checks to fix during a Calibre AutoFix run.	No
set_silent	Specifies whether to print detailed messages to the transcript during a run.	No
set_skip_cell_overlap	Specifies whether to skip the checking of overlapping cells in DRC fixing modes.	No
set_unselect_checks	Unselects checks from a Calibre AutoFix run.	No

set_blockage_spacing_type

Fixing command for: Calibre AutoFix setup file

Required: No

Sets the spacing type for blockages.

Usage

set_blockage_spacing_type {**<u>none</u>** | **zero**}

Arguments

- **<u>none</u>**

  Required keyword that does not set any cell blockage spacing type. This is the default.

- **zero**

  Required keyword that sets the cell blockage spacing type to zero.

Description

Use this command to set the spacing type for cell blockages. This is used by Calibre AutoFix to fix or avoid end-of-line errors encountered in routing operations.

Examples

Specify the cell blockage spacing type to zero:

```
set_blockage_spacing_type zero
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_cpus

Fixing command for: Calibre AutoFix setup file

Required: No

Defines the number of CPUs to use during a Calibre AutoFix run.

Usage

set_cpus *number*

Arguments

- ***number***

  Required value that specifies the number of CPUs to use during a run. The ***number*** must be an integer greater than or equal to 1. The default is 1. The maximum number of CPUs specified should not exceed the number of local CPUs available.

Examples

Specify the run to use four CPUs:

```
set_cpus 4
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_dont_care_routing_layers

Fixing command for: Calibre AutoFix setup file

Required: No

Defines routing layer name(s) for Calibre AutoFix to ignore.

Usage

set_dont_care_routing_layers ‘{’***layer_name*** [*layer_name*…]‘}’

Arguments

- ‘{’***layer_name*** *layer_name*…‘}’

  Required argument that specifies a list of routing layer names for Calibre AutoFix to ignore during DRC runs or input check reporting.

Description

Use this command to define routing layer names that you want Calibre AutoFix to ignore and not report as missing routing layers. When Calibre AutoFix encounters a layer name in this list, a warning is issued, and the run continues to completion. At least one layer name must be specified. An empty list results in the consideration of all valid routing layers for a run. The primary use of this command is when the top routing layer name in the Calibre AutoFix technology file does not match the corresponding routing layer name in the DEF file.

Examples

If the top layer routing name in your Calibre AutoFix technology file is “AP”, and you want to ignore the top routing layer name “metal14” in the DEF file, the following statement must appear in your Calibre AutoFix setup file:

```
set_dont_care_layers {metal14}
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_dont_touch_nets

Fixing command for: Calibre AutoFix setup file

Required: No

Defines nets that are not modified during a Calibre AutoFix run.

Usage

set_dont_touch_nets '{'*net_list*'}'

Arguments

- **'{'*net_list*'}'**

  Required argument that specifies a list of nets to leave unchanged. The ***net_list*** should be a list of strings.

Description

Use this command to define nets that you do not want Calibre AutoFix to change. Layout objects associated with these nets are not moved, deleted, or edited in any way by Calibre AutoFix when performing corrections. Nets that you want to remain unaffected by a Calibre AutoFix run must be specified in the ***net_list*** argument.

Examples

Example 1

If you want to keep the existing routing for the net "clk", the following statement must appear in your Calibre AutoFix setup file:

```
set_dont_touch_nets {clk}
```

Example 2

If you want to specify that a number of nets remain unchanged, you must use the following statement:

```
set_dont_touch_nets {clk clk_analog ADC_bus*}
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_fill_notch

Fixing command for: Calibre AutoFix setup file

Required: No

Defines whether step and notch violations are fixed by re-routing or inserting fill.

Usage

set_fill_notch {**false** | **true**}

Arguments

- **false**

  Required keyword that specifies to fix step and notch violations by re-routing the geometry. This is the default.

- **true**

  Required keyword that specifies to fix step and notch violations by using fill operations.

Description

Use this command to define how minimum step and notch violations are fixed by Calibre AutoFix.

This command is ignored when running in LFD mode.

Examples

Use re-routing to fix step and notch violations:

```
set_fill_notch false
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_fix_custom_signal

Fixing command for: Calibre AutoFix setup file

Required: No

Allows Calibre AutoFix to change pre-route wires to detail wires when fixing errors.

Usage

set_fix_custom_signal {**<u>false</u>** | **true**}

Arguments

- **<u>false</u>**

  Required keyword that does not allow Calibre AutoFix to change the properties of wires. This is the default.

- **true**

  Required keyword that allows Calibre AutoFix to change pre-route wires to detail wires.

Description

Use this command to allow Calibre AutoFix to change pre-route wires to detail wires. The router corrects errors in pre-route wires by changing them to detail wires and re-routing them.

Examples

Allow Calibre AutoFix to change pre-route wires to detail wires during error fixing:

```
set_fix_custom_signal true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_fix_min_area

Fixing command for: Calibre AutoFix setup file

Required: No

Defines whether minimum area violations are fixed during routing.

Usage

set_fix_min_area {**false** | **true**}

Arguments

- **false**

  Required keyword that specifies for Calibre AutoFix to not fix nets with minimum area violations. This is the default.

- **true**

  Required keyword that specifies for Calibre AutoFix to fix nets with minimum area violations.

Description

Use this command to enable or disable fixing of nets with minimum area violations during routing.

Examples

Enable Calibre AutoFix to fix minimum area violations:

```
set_fix_min_area true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_fix_ndr_nets

Fixing command for: Calibre AutoFix setup file

Required: No

Specifies whether nets with a non-default rule (NDR) property are modified during a Calibre AutoFix run.

Usage

set_fix_ndr_nets {**true** | **<u>false</u>**}

Arguments

- **true**

  Required keyword that allows Calibre AutoFix to change nets with an NDR property. Calibre AutoFix applies the NDR when making changes to a net with an NDR property.

- **<u>false</u>**

  Required keyword that does not allow Calibre AutoFix to change nets with an NDR property. This is the default.

Description

Use this command to specify if Calibre AutoFix can change nets with an NDR property during the rip-up and re-route stage.

Examples

Allows Calibre AutoFix to change nets with an NDR property:

```
set_fix_ndr_nets true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_ignore_cell_overlap

Fixing command for: Calibre AutoFix setup file

Required: No

Defines whether cell placement overlaps are checked and reported.

Usage

set_ignore_cell_overlap {**false** | **true**}

Arguments

- **false**

  Required keyword that enables cell overlap placement checking. When this keyword is set, cell overlap information is reported to the transcript. This is the default.

- **true**

  Required keyword that disables cell overlap placement checking. When this keyword is set, cell overlap information is not reported to the transcript.

Description

Use this command to specify whether Calibre AutoFix checks and reports overlapping cell placements or ignores them. When cell overlap checking is enabled and cell overlaps exist in a design, an error is generated and the run exits. Calibre AutoFix reports the first ten overlapping cells to the transcript.

Examples

Instruct Calibre AutoFix to check and report cell placement overlaps (this is the default):

```
set_ignore_cell_overlap false
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_m1_routing

Fixing command for: Calibre AutoFix setup file

Required: No

Defines whether routing is allowed on the metal 1 layer.

Usage

set_m1_routing {**<u>false</u>** | **true**}

Arguments

- **<u>false</u>**

  Required keyword that instructs Calibre AutoFix not to use metal 1 as a routing layer. No adjustments to the metal 1 layer are performed. This is the default. The valid routing layers are metal 2, … up to -top_layer, where -top_layer is the top routing layer in your design.

- **true**

  Required keyword that instructs Calibre AutoFix to make corrections using the metal 1 layer, if possible. The valid routing layers are metal 1, … up to -top_layer, where -top_layer is the top routing layer in your design.

Description

Use this command to enable or disable routing on the metal 1 layer. When Calibre AutoFix performs corrections on a design, it does not use the metal 1 layer by default. If you want Calibre AutoFix to use the metal 1 routing layer, then set this command to **true**.

Examples

Make routing corrections using metal 1 if possible:

```
set_m1_routing true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_route_options

Fixing command for: Calibre AutoFix setup file

Required: No

Defines custom options for the router during a Calibre AutoFix run.

Usage

set_route_options [-accept {strictly_improved | number_improved}] [-run_limit *number*]

Arguments

- -accept {strictly_improved | number_improved}

  Optional and default argument that specifies the type of fixing mode for the router to use to reduce the number of DRC error counts. The options for -accept are described as follows:

  strictly_improved — Instructs Calibre AutoFix to reduce the total DRC error count but not introduce new DRC error types. This is the default option.

  number_improved — Instructs Calibre AutoFix to reduce the total DRC error count but may introduce new DRC error types.

  By default, the **set_route_options** behavior is as follows:

  ```
  set_route_options -accept strictly_improved
  ```

- -run_limit *number*

  Optional argument and value that enable Calibre AutoFix to iterate run cycles. The *number* must be a positive integer between 1 and 100. If not specified, Calibre AutoFix automatically determines the termination point.

Description

Use this command to specify options to control the behavior of Calibre AutoFix during the rip-up and re-route stage.

Examples

Enable Calibre AutoFix to iterate for five cycles and reduce the total DRC error count, but not create new DRC error types:

```
set_route_options -accept strictly_improved -run_limit 5
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_select_checks

Fixing command for: Calibre AutoFix setup file

Required: No

Selects which checks to fix during a Calibre AutoFix run.

Usage

set_select_checks ‘{’*rule* [*rule* ...]**‘}’**

Arguments

- **‘{’*rule*** [*rule*...]**‘}’**

  Required string list that specifies the Calibre nmDRC rule check names that you want Calibre AutoFix to correct. The ***rule*** list must match rules contained in the rule file specified with the set_calibre_rule_file setup file command. You may use wildcard (*) expressions to match rule check names.

Description

Use this command to define the Calibre nmDRC rule checks that you want Calibre AutoFix to correct.

Note

This command takes precedence over any select check operations in your Calibre rule file. When this command is issued, Calibre AutoFix unselects all checks in your rule file and then only selects the rules contained in the ***rule*** list.

Examples

Instruct Calibre AutoFix to correct only the specified rule checks:

```
set_select_checks {met1_space met2_space M3.* via3_int met*_int antenna*}
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_silent

Fixing command for: Calibre AutoFix setup file

Required: No

Specifies whether to print detailed messages to the transcript during a run.

Usage

set_silent {**false** | **true**}

Arguments

- **false**

  Required keyword that disables silent mode. When this keyword is set, detailed messages are displayed to the transcript window during a run. This is the default.

- **true**

  Required keyword that enables silent mode. When this keyword is set, detailed messages are not displayed to the transcript window during a run.

Description

Use this command to specify whether detailed messages about the Calibre AutoFix runtime are displayed to the transcript window output.

Examples

Do not display detailed runtime messages to the transcript window:

```
set_silent true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_skip_cell_overlap

Fixing command for: Calibre AutoFix setup file

Required: No

Specifies whether to skip the checking of overlapping cells in DRC fixing modes.

Usage

set_skip_cell_overlap {**<u>false</u>** | **true**}

Arguments

- **<u>false</u>**

  Required keyword that instructs Calibre AutoFix not to skip cell placement overlap checking in DRC fixing modes. When the **<u>false</u>** argument is set, cell overlaps are checked. This is the default.

- **true**

  Required keyword that instructs Calibre AutoFix to skip checking of cell placement overlaps in DRC fixing modes.

Description

Use this command to specify whether Calibre AutoFix skips the checking of cell placement overlaps in DRC fixing modes. This command does not affect **check_inputs** mode, which always checks and reports overlapping cells. Setting this command to **true** can save runtime. However, you should only set this command to **true** if you have already run **check_inputs** mode on the same design and verified that there are no overlapping cells.

Examples

Instruct Calibre AutoFix to skip the checking of cell placement overlaps during DRC fixing modes:

```
set_skip_cell_overlap true
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

set_unselect_checks

Fixing command for: Calibre AutoFix setup file

Required: No

Unselects checks from a Calibre AutoFix run.

Usage

set_unselect_checks ‘{’*rule* [*rule* ...]‘**}**’

Arguments

- ‘**{**’***rule*** [*rule* ...]‘**}**’

  Required string list that specifies the Calibre nmDRC rule check names that you *do not* want Calibre AutoFix to correct. The ***rule*** list must match rules contained in the rule file specified with the set_calibre_rule_file setup file command. You may use wildcard (*) expressions to match rule check names.

Description

Use this command to define the Calibre nmDRC rule checks you do not want Calibre AutoFix to correct.

Examples

Instruct Calibre AutoFix to not correct the specified rule checks:

```
set_unselect_checks { IO* M1.* ant_rule5 M6*}
```

Related Topics

Setup File Format

Process and Library Setup Commands

Design Input Commands

Output Commands

Fixing Commands

Error and Warning Messages

Appendix A
Error and Warning Messages

The Calibre AutoFix tool may create different kinds of informational messages in the form of error messages and warning messages.

Error Messages

When certain conditions are encountered, Calibre AutoFix issues an error message.

Error messages are more serious than warning messages and should be investigated or resolved before the run proceeds. Refer to Table A-1 for descriptions of error messages and possible causes and solutions.

Table A-1. Error Messages and Possible Causes

Description	Possible Causes and Solutions
Calibre AutoFix is only supported on Red Hat Linux®[1], aborting.	Verify your platform environment is correct for running Calibre AutoFix.
Calibre Rules File must be set.	The set_calibre_rule_file statement must be specified in your setup file.
Calibre Rules File <file_name> is not readable!	Check your set_calibre_rule_file statement in the setup file to ensure that the path to your Calibre rule file is correct and readable.
DEF File <file_name> is not readable!	Check your set_input_def_file statement in the setup file to ensure that the path to your DEF file is correct and readable.
DEF File must be set.	The set_input_def_file statement must be specified in your setup file.
File: <def_file_name>, Line <line_number>: No such via <via_name>.	An undefined via is used in the input DEF file(s). The input DEF file name and line number where the error occurred are shown. All vias used in your design must be defined in either DEF or LEF(s).
File: <def_file_name>, Line <line_number>: No such spacing rule <ndr_rule_name>. Creating wire with default.	Check that the <ndr_rule_name> at <line_number> is specified in the NONDEFAULTRULES section of the input DEF file.

Table A-1. Error Messages and Possible Causes (cont.)

Description	Possible Causes and Solutions
GDS Map File <file_name> is not readable!	Check your set_gds_layer_map_file statement in the setup file to ensure that the path to your input GDS layer map file is correct and readable.
Illegal setting <setting> for command "<command>". Please specify …	Check your command statement in the setup file to verify the specified value is correct. Refer to “Setup File Format” on page 34.
LEF File must be set.	The set_lef_file statement must be specified in your setup file.
LEF File <file_name> is not readable!	Check your set_lef_file statement in the setup file to ensure that the paths to your input LEF files are correct and readable.
lib ref = '<library_macro>' is undefined.	Check that the <library_macro> used in the DEF is defined in your input LEF files. This error causes Calibre AutoFix to exit.
Line <line_number>:METALx routing layer is not defined.	Check that the valid routing layers are defined in the input DEF. This error causes Calibre AutoFix to exit in “drc” and “drc_all” run modes. For valid routing layers, refer to “set_input_def_file” on page 48.
Missing Preferred Routing Track(s): Mx …	Check that the valid routing layers in the input DEF have TRACKS defined for the preferred routing direction. This error causes Calibre AutoFix to exit in “drc” and “drc_all” run modes. For valid routing layers, refer to “set_input_def_file” on page 48.
Mode <mode_name> is not legal, must be one of drc lfd preroute drc_all check_inputs pm.	You did not specify a required parameter for set_calibre_mode in your setup file. Specify one of the following: “drc”, “preroute”, “drc_all”, or “check_inputs”.
No Autofix candidates in this design.	There are no candidates for Calibre AutoFix to repair. This error causes Calibre AutoFix to exit in any fixing mode. To be considered a candidate, the error type must be mapped and selected. DRC violations related to NDR, clock, or power net are non-candidates.
Please select at least one check.	At least one valid Calibre nmDRC rule check from your rule file must be specified if the set_select_checks setup file command is used in your setup file.

Table A-1. Error Messages and Possible Causes (cont.)

Description	Possible Causes and Solutions
Please specify at least one critical net.	At least one net must be specified in the argument list if the set_dont_touch_nets command is used in your setup file.
Please specify at least one input def file.	Specify one top-level DEF file with the set_input_def_file setup file command. Hierarchical DEF is not supported.
Please specify at least one lef file.	At least one LEF file must be specified with the set_lef_file setup file command. Do not add the top-level LEF as an input to LEF.
Please specify "-metal_stack" option with "set_autofix_tech_file" command.	See "set_autofix_tech_file" on page 38 for the correct command syntax.
RDB File must be set.	The set_input_rdb_file statement must be specified in your setup file.
RDB File <file_name> is not readable!	Check your set_input_rdb_file statement in the setup file to ensure that the path to your input results database file is correct and readable. The results database file must match your layout design database.
Reporting first 10 of <total_number> overlapping cells.	Remove cell overlaps before proceeding with the run. This error causes Calibre AutoFix to exit. See "set_ignore_cell_overlap" on page 71.
Router is not found, please check your path or see the *Calibre AutoFix User's Manual*.	See "Requirements" on page 14 for a list of the required environment variables and software installations.
Rule Map File <file_name> is not readable!	Verify that the set_tech_options -map option points to a valid rule map file. Contact your Siemens representative for assistance with technology node specification.
Tech Node must be set.	The set_tech_options statement must be specified in your setup file with the technology node of your process. Contact your Siemens representative for assistance with technology node specification.
Technology node name must be specified.	The set_tech_options statement must be specified in your setup file with the technology node of your process. Contact your Siemens representative for assistance with technology node specification.

Table A-1. Error Messages and Possible Causes (cont.)

Description	Possible Causes and Solutions
Technology node <node_name> is not supported …	Contact your Siemens representative for assistance with technology node specification.
TECH File <file_name> is not readable!	Check your set_autofix_tech_file statement in the setup file to ensure that the path to your Tcl tech file is correct and readable.
Tracks not defined for layer 'Mx'.	Check that the valid routing layers in the input DEF have TRACKS defined. This error causes Calibre AutoFix to exit in “drc” and “drc_all” run modes. For valid routing layers, refer to “set_input_def_file” on page 48.
Undeclared component <instance_name> in the COMPONENT section.	Check the COMPONENTS section of the DEF file for the missing <instance_name>. This error reports up to ten missing instance names with line numbers and causes Calibre AutoFix to exit.
Unknown option <option_name> for "set_autofix_tech_file" command.	Check that option_name is specified correctly.
Unknown option <option_name>" for "set_tech_options" command.	Check that option_name is specified correctly.
Value for option <option_name> is not specified.	A value set is missing from the specified option argument in your setup file.

1. Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

Warning Messages

When certain conditions are encountered, Calibre AutoFix issues a warning message.

You should review all warning messages generated during the Calibre AutoFix run and decide whether or not they are potential problems for your design. Refer to Table A-2 for descriptions of warning messages and possible causes and solutions.

Table A-2. Warning Messages and Possible Causes

Description	Possible Causes and Solutions
An empty metal list is specified for command "set_dont_care_routing_layers". All routing layers will be considered.	You must specify at least one routing layer name in the argument list for the set_dont_care_routing_layers setup file command.

Table A-2. Warning Messages and Possible Causes (cont.)

Description	Possible Causes and Solutions
File: <lef_file_name>, No such via '<via_name>'; Line <line_number>.	A via used in the LEF file is not defined in the input LEF and DEF file(s). The LEF file name and line number where the via is specified are shown.
Ignored invalid metal "<layer_name>" specified in command "set_dont_care_routing_layers".	Verify that <layer_name> is specified in the set_dont_care_routing_layers setup command. If so, Calibre AutoFix continues to the end of the run.
RouterAutofix version x.xxxxx.x.xxx is not qualified. Qualified version is x.xxxxx.x.xxx.	Check that the specified router version in your $PATH variable is qualified.

Index

Third-Party Information

Details on open source and third-party software that may be included with this product are available in the *<your_software_installation_location>/legal* directory.