



SIEMENS EDA

Calibre® Auto-Waivers™ User's and Reference Manual

Software Version 2021.2
Document Revision 14

Unpublished work. © 2021 Siemens

This material contains trade secrets or otherwise confidential information owned by Siemens Industry Software, Inc., its subsidiaries or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this information is strictly limited as set forth in Customer's applicable agreement with Siemens. This material may not be copied, distributed, or otherwise disclosed outside of Customer's facilities without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This document is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made. Siemens disclaims all warranties with respect to this document including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement of intellectual property.

The terms and conditions governing the sale and licensing of Siemens products are set forth in written agreements between Siemens and its customers. Siemens' **End User License Agreement** may be viewed at: www.plm.automation.siemens.com/global/en/legal/online-terms/index.html.

No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

TRADEMARKS: The trademarks, logos, and service marks ("Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' trademarks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux[®] is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Support Center: support.sw.siemens.com

Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

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 <i>Calibre Release Notes</i> 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 <i>Calibre Release Notes</i> 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 <i>Calibre Release Notes</i> 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 <i>Calibre Release Notes</i> 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

Revision History

Chapter 1

Calibre Auto-Waivers Overview	13
Calibre Auto-Waivers Workflow	13
Calibre Auto-Waivers Versus the .waived File Method	16
Waiver Terminology	17
Usage Models for Calibre Auto-Waivers	20
DRC Automatic Waivers	21
ERC Automatic Waivers	22
DFM Automatic Waivers	22
Calibre PERC Waivers	23
Requirements for Running Calibre Auto-Waivers	23
Calibre Environment Variables	25
SVRF Command Support	27
DFM Operation Support	27
ERC Layer Operation Support	28
Discouraged Layer Operation Use	28
Unsupported Layer Operations in waiver_flow	29
Unsupported Elements	29
Encryption Support	30
Syntax Conventions	30
Known Issues and Workarounds	32

Chapter 2

Getting Started With Calibre Auto-Waivers	33
Third-Party Design Environments	35
Generating Waivers for DRC, ERC, or DFM	36
Using Waivers in the DRC, ERC, or DFM Flows	38
Density Window Waivers	42
Method 1: Generating Density Waivers with DRC RVE	42
Method 2: Generating Density Waivers with waiver_flow	44
Running Density Waiver Verification	45
Using Cell Extent Waivers	46
Frequently Asked Questions	48

Chapter 3

Waiver Shape Interactions	53
Waiver Shape Interaction Types	54
SINGLE Interactions	55
MULTI Interactions	58
Waiver Type Summary	60

Waivers for Edge and Edge Cluster Outputs	61
Block-Level Waivers (Gray Box)	62
Chapter 4	
Waiver Shape Generation with Calibre RVE	71
Requirements for Waiver Export with Calibre RVE	71
Waiver Shape Generation Flow	74
Creating Embedded Waiver Shapes Using Calibre RVE	75
Creating Non-Embedded Waiver Shapes Using Calibre RVE	81
Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE	86
Creating IP_MATCH Signature Checksums Using Calibre RVE	91
Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE	93
Waiver Export of Error Edge Clusters by Calibre RVE for DRC	95
Waiver Options in Calibre RVE	97
Chapter 5	
Waiver Shape Generation with the waiver_flow Tool	99
Writing the Waiver Cells File for waiver_flow	101
Writing the Waiver Criteria File for waiver_flow	101
Writing the Waiver Setup File for Waiver Generation	102
Generating Waivers with the waiver_flow Tool	104
waiver_flow	107
Creating Waiver Cell Rule Checksum Annotations with waiver_flow	109
Waiver Cell Precision and Magnification Considerations	110
Chapter 6	
Waiver File Descriptions	113
Calibre Auto-Waivers File Usage	113
Waiver Setup File	116
Waiver Setup File Format for Waiver Generation	117
Waiver Setup File Format for Waiver Verification	126
Waiver Description Files	143
Waiver Criteria File Description	144
Waiver Criteria File Format	147
Standard Form Criteria	150
EXCLUDE_CELL	154
INCLUDE_CELL	156
WAIVE_EXTENT	158
WAIVE_MARKER	161
WAIVE_PATTERN	164
WAIVE_DENSITY	167
P2P	171
Waiver Cells File Format	173
Waiver Cells File Description	174
Batch Mode File Usage	176

Table of Contents

Chapter 7

Running a Physical Verification with

Waiver Shapes..... 179

Physical Verification with Waivers	180
Writing the Waiver Setup File for Waiver Verification	181
Writing the Waiver Criteria File for Waiver Verification	183
Performing a Waiver Run	184
Performing a DRC Waiver Run Using Multiple Waiver Criteria Files	189
Using Rule Checksums in calibre -waiver	191
Differences in Generated SVRF Rules When Running TVF Rules with Waivers	193

Chapter 8

Calibre Auto-Waivers Utilities 195

drc_waiv_asc2gds	196
checksum_util	203
waiver_util	208

Appendix A

Waiver Cell Description and Text Annotations..... 217

Waiver Cell Description	218
Intermediate Container Cells	219
Text Object Annotations	221
Geometry Checksum Annotations	224
Rule Checksum Annotations	224
Calibre Version Text Annotations	225
IP_MATCH Checksum Annotations	226
Dummy Waiver Cells with Text Annotations	227
Waiver Cells With Instances of Error Cells	228
Support For Waivers in Cloned Cells	230

Appendix B

Waiver Setup File Statement Usage 233

Statement Usage	233
-----------------------	-----

Appendix C

Calibre Auto-Waivers Messages 239

Calibre Auto-Waivers Runtime Messages	240
drc_waiv_asc2gds Runtime Messages	252
checksum_util Runtime Messages	253
waiver_util Runtime Messages	256

Index

Third-Party Information

List of Figures

Figure 1-1. Single Waiver in Specific Cell	14
Figure 1-2. Waiver Interactions Between Cells.	14
Figure 1-3. Workflow for Calibre Auto-Waivers	15
Figure 2-1. Getting Started With Calibre Auto-Waivers Flow	34
Figure 3-1. Waiver Overlap Percentages.	54
Figure 3-2. SINGLE 100 100	55
Figure 3-3. SINGLE 100 1	56
Figure 3-4. SINGLE 1 100	57
Figure 3-5. SINGLE 50 50	57
Figure 3-6. MULTI 100 100	58
Figure 3-7. MULTI 100 1	59
Figure 3-8. MULTI 1 100	59
Figure 3-9. MULTI 50 50	60
Figure 3-10. Calibre RVE Waiver Shape Export From Edge Pairs.	61
Figure 4-1. Calibre RVE Waiver Generation Flow	74
Figure 4-2. Creating Waiver-Corrected IP Using Calibre RVE	75
Figure 4-3. Setup Waiver Options	97
Figure 5-1. Waiver Shape Generation in Batch Mode Using waiver_flow	100
Figure 5-2. waiver_flow I/O	108
Figure 6-1. Without Debug Properties.	133
Figure 6-2. With Debug Properties	133
Figure 6-3. result_overlap Waiver Threshold	169
Figure 7-1. I/O for Waiver Verification in DRC and ERC	180
Figure A-1. Intermediate Container Cell	219
Figure A-2. Flattening of Waiver Shapes	229

List of Tables

Table 1-1. Calibre Auto-Waivers Versus .waived File	16
Table 1-2. Required Licenses for Calibre Auto-Waivers Tools	23
Table 1-3. Calibre Auto-Waivers Environment Variables	25
Table 1-4. Syntax Conventions	31
Table 1-5. List of Issues for Calibre Auto-Waivers	32
Table 2-1. Calibre Auto-Waivers FAQ	48
Table 3-1. Quick Overview of EXCLUDE_CELL Examples	64
Table 4-1. Waiver Setup File Parameters Used in Calibre RVE Waiver Export	94
Table 4-2. Setup Waiver Options Contents	97
Table 6-1. Waiver Generation Setup File Statements	117
Table 6-2. Intermediate Cell Output Conditions with MERGE YES	119
Table 6-3. Waiver Verification Setup File Statements	126
Table 6-4. Waiver Criteria File Statements	144
Table 7-1. Calibre RVE waived.rdb Properties	187
Table A-1. Text Object Annotations	221
Table B-1. Waiver Setup File Statements Usage	233
Table C-1. Calibre Auto-Waivers Error Messages	240
Table C-2. Calibre Auto-Waivers Warning Messages	245
Table C-3. Calibre Auto-Waivers Note Messages	250
Table C-4. drc_waiv_asc2gds Messages	252
Table C-5. checksum_util Messages	253
Table C-6. waiver_util Messages	256

Chapter 1

Calibre Auto-Waivers Overview


The Calibre® Auto-Waivers™ tool enables you to generate waivers for rules that are considered safe to ignore. You can then apply the waivers, which are geometrical shapes that resemble the original error result region, automatically in a verification run wherever an error result for the same waived rule appears.

Calibre Auto-Waivers Workflow	13
Calibre Auto-Waivers Versus the .waived File Method	16
Waiver Terminology	17
Usage Models for Calibre Auto-Waivers	20
Requirements for Running Calibre Auto-Waivers	23
Calibre Environment Variables	25
SVRF Command Support	27
Syntax Conventions	30
Known Issues and Workarounds	32

Calibre Auto-Waivers Workflow

When verifying a design, the foundry can waive certain rule check violations. These waivers need to be managed in the results output so that they can be preserved for analysis and validation. Inspecting such waivers manually consumes much time.

Tip

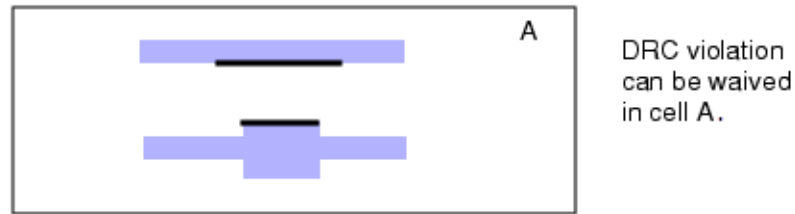
 If you are new to Calibre Auto-Waivers, or are looking for condensed material showing the basic DRC usages, refer to “[Getting Started With Calibre Auto-Waivers](#)” on page 33 before reading this chapter. The procedures in the Getting Started chapter may be all you need to get your waiver flow working.

To reduce the amount of time spent in managing waivers, Calibre provides these methods:

- Calibre Auto-Waivers manages waivers by generating waiver shapes that are merged with your design so that you can apply the waivers automatically during a verification run. These waivers take into account hierarchical polygon interactions, as well as interactions of multiple waivers that may apply to a single result.
- Calibre RVE enables you to mark rule checks as waived. In later runs, you can load the waiver (*.waived*) file and all violations that match the waiver file are marked as waived.

Waivers of rule check violations in specific cells are issues that Calibre RVE addresses through its waiver (*.waived*) file. The following figure shows an example.

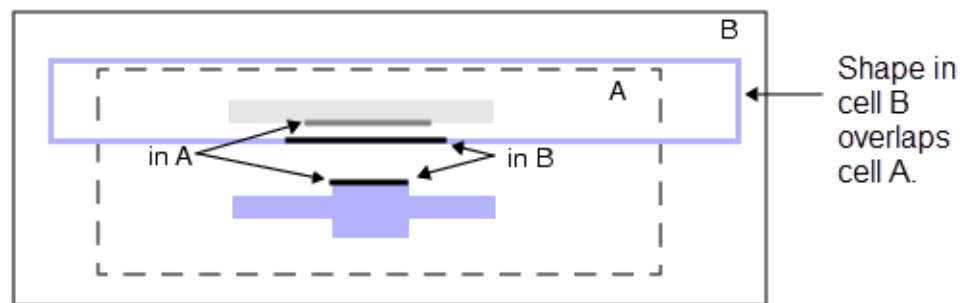
Figure 1-1. Single Waiver in Specific Cell



When using a *.waived* file in Calibre RVE, the waiver applies in all instances of cell A. Afterward, you can load the waiver file to mark all instances of this violation as waived.

However, what happens when there is an interaction between shapes in cell A and overlapping shapes from the same layer in a parent cell? The following figure shows an example.

Figure 1-2. Waiver Interactions Between Cells



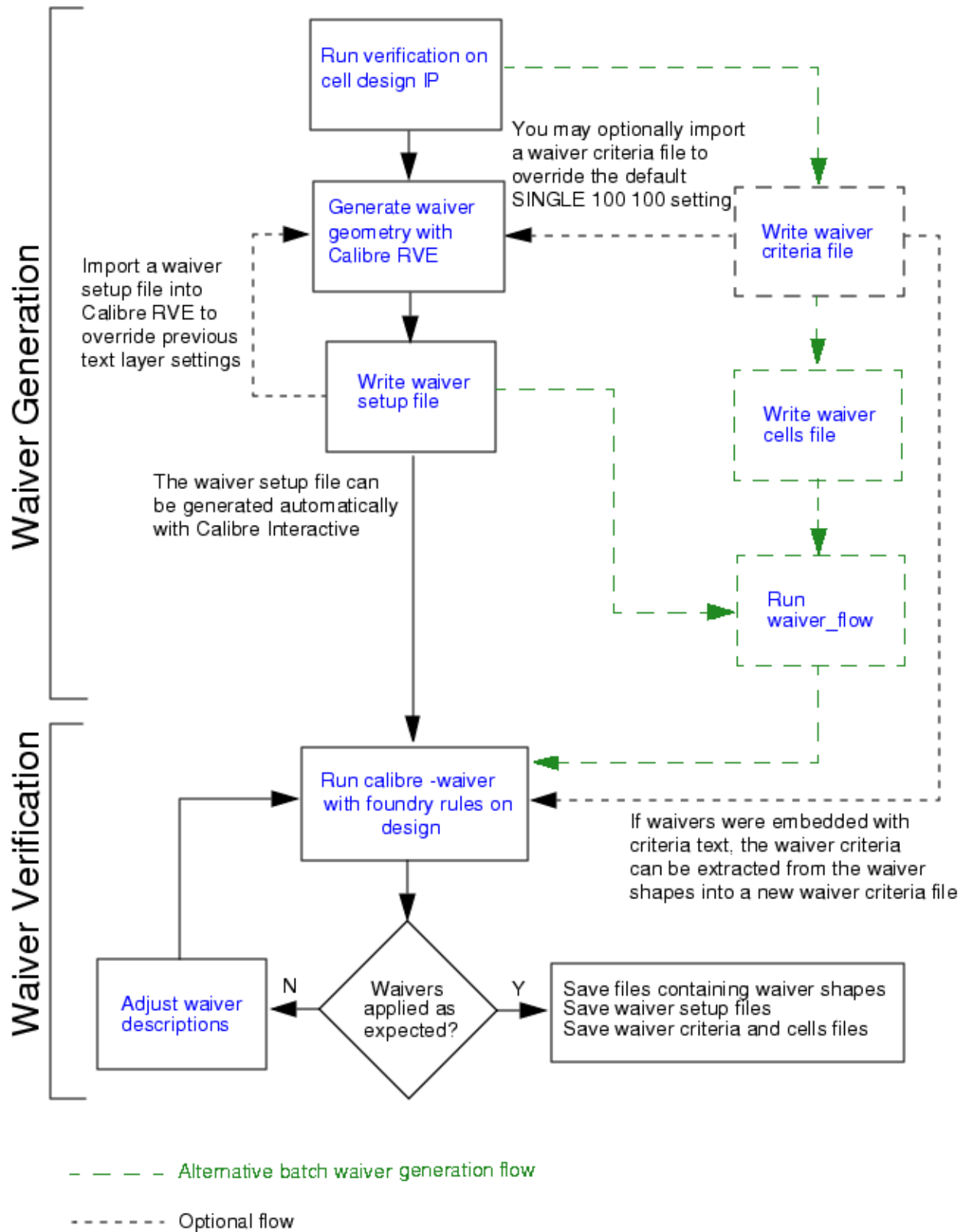
If you connect the endpoints of the rule check violation edges at the top level (cell B in this case) to form a trapezoidal region, the result region in cell B is somewhat wider than a similar region in cell A. The overlap between the two areas is not 100 percent.

If you had generated a waiver for the violation in [Figure 1-1](#), consider whether it should still apply in [Figure 1-2](#). A waiver from a *.waived* file would not apply in this case, but it could be that the foundry would consider this to be a waive-able error.

Calibre Auto-Waivers addresses such issues. It enables you to determine the percentage of overlap between a waiver shape and a violation result region that permits a waiver to apply. Calibre Auto-Waivers also addresses issues when multiple waiver interactions may intersect a violation result region. It also enables you to apply waivers hierarchically throughout the design.

The following figure shows the typical waiver generation and usage flow for Calibre Auto-Waivers.

Figure 1-3. Workflow for Calibre Auto-Waivers



Related Topics

[Waiver Terminology](#)

[Requirements for Running Calibre Auto-Waivers](#)

Calibre Auto-Waivers Versus the .waived File Method

You can use both the Calibre Auto-Waivers and Calibre RVE tools to waive rule check errors. Historically, Calibre RVE has used an ASCII file (*.waived* extension) to mark rule checks as waived. Calibre RVE also supports Calibre Auto-Waivers. The Calibre Auto-Waivers flow offers significant advantages in flexibility.

The following table shows some of the differences between the two waiver methods.

Table 1-1. Calibre Auto-Waivers Versus .waived File

Calibre Auto-Waivers	Waivers Using <i>.waived</i> File
Applies waivers using automated methods that insert geometry into your design to enforce waivers. Calibre RVE or the <code>waiver_flow</code> executable generates the geometry.	User generates the <i>.waived</i> text file by marking checks in Calibre RVE as waived.
Generates waivers per rule check and per cell.	Waivers are specified per rule check and per cell.
Causes error waivers at the cell level to apply at other levels of the hierarchy also.	Error waivers at the cell level frequently do not apply at other levels in the hierarchy.
Applies waivers based upon the percentage of coverage of a waiver shape with an actual violation.	No similar capability.
Applies a waiver based upon the number of distinct waiver shapes that intersect a result region.	No similar capability.
Produces a layout file of design geometry embedded with waiver geometry. Also, you can use Calibre Auto-Waivers to produce non-embedded waiver geometry in a separate file that gets merged with the main design at runtime for waiver application.	Produces a text file used by Calibre RVE to apply waivers for future DRC, DFM, or ERC runs.
Produces the usual DRC, DFM, or ERC Results Database for violations along with separate RDBs showing waiver usage.	Interacts with DRC, DFM, or ERC Results Database to show waiver results.

Table 1-1. Calibre Auto-Waivers Versus .waived File (cont.)

Calibre Auto-Waivers	Waivers Using .waived File
Annotates output waiver geometries with text objects that assist in classifying the waivers.	No similar capability.

Calibre Auto-Waivers can use Calibre RVE to generate waiver shapes. This capability includes loading a .waived file into Calibre RVE and then exporting the waivers.

The [drc_waiv_asc2gds](#) command enables batch conversion of a .waived file to a GDS or OASIS^{®1} format waiver database.

Related Topics

[Waiver Terminology](#)

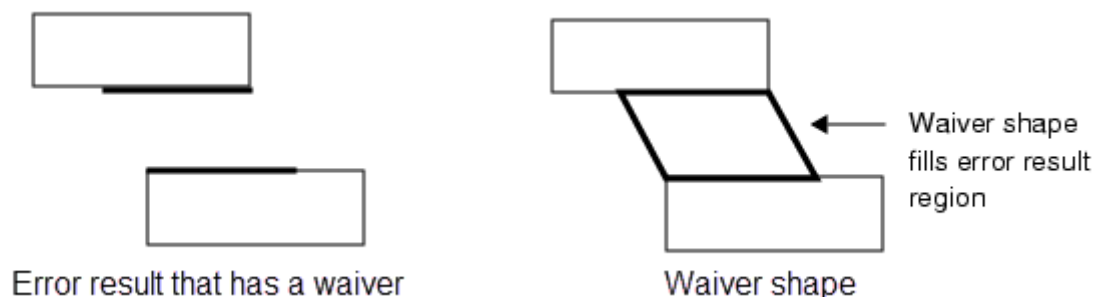
[Requirements for Running Calibre Auto-Waivers](#)

[Calibre Auto-Waivers Workflow](#)

Waiver Terminology

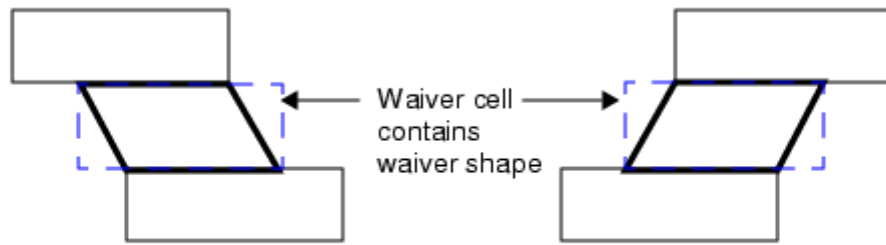
There are several terms specific to Calibre Auto-Waivers that are used throughout this manual. It is highly recommended that you fully understand the terminology and concepts before using the tool.

- **Waiver** — A design rule error that the foundry allows not to be fixed. Also referred to as a *result waiver*.
- **Waiver shape** — A polygon created during a waiver generation run. A waiver shape defines a waiver within a waiver cell.



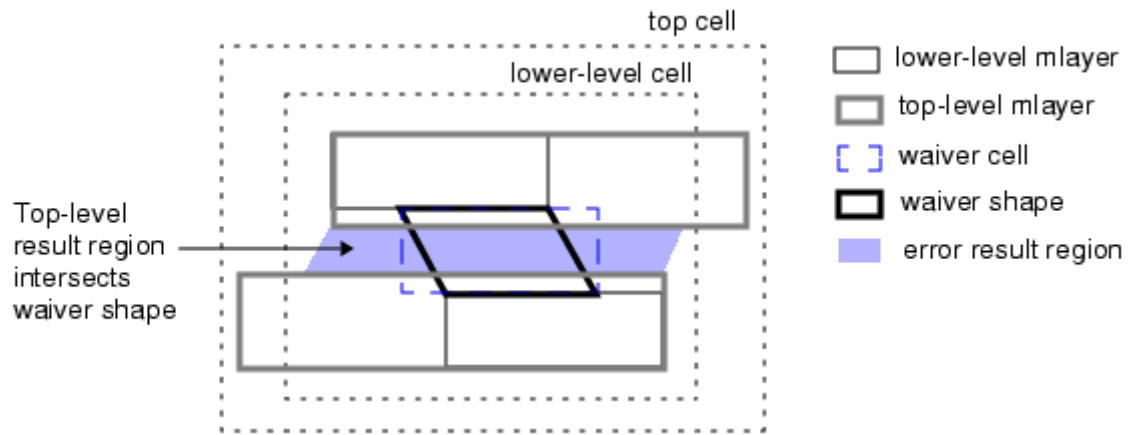
1. OASIS is a registered trademark of Thomas Grebinski and licensed for use to SEMI[®], San Jose, California. SEMI is a registered trademark of Semiconductor Equipment and Materials International

- **Waiver cell** — A cell created during waiver shape generation. The cell contains a waiver shape.

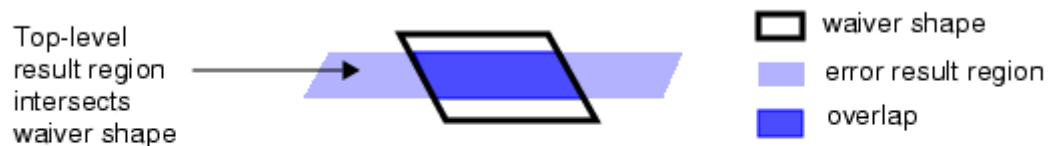


Waiver cell is placed everywhere the waiver applies.

- **Error result region** — A polygonal area defined by an error result, to which a waiver may apply. Such regions are defined in a similar way as the REGION keyword of the “ENClosure”, “EXTernal”, and “INTernal” operations.



- **Result overlap** — The percentage of the error result region that is overlapped by the waiver shape.



$$\text{Result overlap} = (\text{area } \text{overlap} / \text{area } \text{error result region}) * 100$$

$$\text{Waiver overlap} = (\text{area } \text{overlap} / \text{area } \text{waiver shape}) * 100$$

- **Waiver overlap** — The percentage of the waiver shape that is overlapped by the error result region.

- **Waiver type** — The type of interaction between the waiver shape and the error result region. There are two primary types:
 - SINGLE** — [SINGLE Interactions](#) enforce the condition that only one waiver shape can interact with a result region in order for the waiver to apply.
 - MULTI** — [MULTI Interactions](#) allow more than one waiver shape to overlap a result region.

A primary waiver type specified with overlap values, such as SINGLE 100 100, is referred to as a standard form waiver statement.

These terms are discussed in detail under the sections “[Waiver Shape Interaction Types](#)” on page 54 and “[Waiver Criteria File Format](#)” on page 147.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Requirements for Running Calibre Auto-Waivers](#)

Usage Models for Calibre Auto-Waivers

To improve flexibility and management, you can generate waivers within your design database or in separate layout databases. Waivers that are saved along with your layout design are called embedded waiver shapes, and waivers that are saved in external databases are called non-embedded waiver shapes.

Calibre Auto-Waivers generates waiver shapes that are used to waive design rule violations. You can use these waiver shapes in the following ways:

- **Embedded waiver shapes** — This method often applies when IP cell libraries are used. This method is suitable for DRC, ERC, and DFM applications and it involves two stages:
 - a. The waiver shapes are instantiated in waiver cells, which are embedded in original design cells. This results in waiver-corrected IP. The IP is then used in the main design. For details see “[Creating Embedded Waiver Shapes Using Calibre RVE](#)” on page 75.
 - b. Users of the waiver-corrected IP design perform verification on their main design. The waivers in the corrected design cells are then enforced automatically because the waiver shapes are embedded in the design. For details see “[Physical Verification with Waivers](#)” on page 180.
- **Non-embedded waiver shapes** — This method can be used alone or in combination with embedded waiver shapes. In this method, waiver shapes are not embedded in the primary design, they are kept in separate files. This method is suitable for DRC, ERC, and DFM applications and it involves two stages:
 - a. Waiver shapes are generated in a separate file. For details, see “[Creating Non-Embedded Waiver Shapes Using Calibre RVE](#)” on page 81.
 - b. The waiver cell file is read in along with your main design during the verification run, and the waivers are applied automatically.

The embedded and non-embedded waiver shape methods can be combined. The embedded method has these advantages:

- A library cell name can change many times throughout the lifetime of a library. If you embed the waiver shapes into the IP cells, you do not have to worry about cell naming conventions to ensure proper layout merging in later runs.
- If you are extracting cells from an existing design that has been embedded with waiver shapes, you do not have to re-merge the existing layout with a separate file of waiver shapes.
- If you modify a cell to which a waiver applies, the corrected geometry is already in the cell. You can see what the modification does to the corrected design cell because the correction is embedded in the layout file.

- The designer can see the waiver shapes in the design when performing DRC. With non-embedded waiver shapes, this will not be the case.

The non-embedded waiver shape method has these advantages:

- Because the waiver shapes are not kept in the primary design, there is less risk that the waiver geometry and cells will be altered by a designer.
- The waiver shapes and the primary design can be maintained using differing database precisions.

For answers to frequent questions, see “[Frequently Asked Questions](#)” on page 48.

The usage also depends on the type of Calibre run.

DRC Automatic Waivers	21
ERC Automatic Waivers	22
DFM Automatic Waivers	22
Calibre PERC Waivers	23

DRC Automatic Waivers

Calibre Auto-Waivers can use embedded or external waiver shapes in your DRC flow.

Specified DRC results can be waived interactively with Calibre® RVE, or in batch mode with waiver_flow, where Calibre RVE is the recommended method.

Also see “[Getting Started With Calibre Auto-Waivers](#)” on page 33 for basic DRC procedures in Calibre Auto-Waivers.

Calibre nmDRC Reconnaissance

The Calibre Auto-Waivers tool supports Calibre nmDRC Reconnaissance using the “calibre -waiver -recon” invocation arguments.

The Calibre nmDRC Reconnaissance tool performs automated rule check selection that provides fast feedback to help designers identify systemic design issues that may result in many errors. A Calibre nmDRC Recon run primarily excludes context-dependent rules, such as connectivity, delta v, and multi-patterning rule checks, that are likely to result in long run times and are unlikely to provide actionable results for an unfinished layout.

The excluded rule checks require significant context awareness and usually generate hundreds or thousands of violations at different locations across the chip; usually such violations are resolved indirectly as the design matures. By contrast, Calibre nmDRC Recon rules are fast and identify results that are typically geographically close to the source of the design flaw. The faster and more targeted Calibre Recon run enables you to identify and fix significant issues in your layout before performing sign-off verification runs.

A single Calibre RealTime Digital license is required for jobs with -recon, regardless of the number of CPUs.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

ERC Automatic Waivers

The ERC automatic waivers can use embedded or external waiver shapes, just like DRC or DFM automatic waivers.

The main difference is an ERC run occurs in LVS verification. ERC automatic waivers are often used to waive ERC errors in lower-level blocks before the full chip is assembled. This is because the lower-level blocks may not have paths to desired signals when those blocks are not in the context of the full chip. The waivers are needed to avoid false errors when verifying the lower-level blocks outside of the full-chip context. For this use model, embedded waiver shapes may be a better choice for doing ERC waivers.

Encrypted rules are not supported for ERC waivers.

Also see “[Specifying Calibre Auto-Waivers for ERC](#)” in the *Calibre Interactive User’s Manual* for information on running ERC automatic waivers using Calibre Interactive.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

DFM Automatic Waivers

The DFM automatic waivers flow is similar to DRC and ERC automatic waivers, except that it produces a binary DFM Database rather than ASCII results databases.

DFM automatic waivers currently supports Critical Feature Analysis (CFA) through rules that use [DFM Analyze](#) and [DFM Property](#). Also see “[Specifying Calibre Auto-Waivers for DFM CFA](#)” in the *Calibre Interactive User’s Manual* for information on running DFM automatic waivers using Calibre Interactive.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

Calibre PERC Waivers

The Calibre Auto-Waivers license also enables waivers for Calibre® PERC™.

Calibre PERC supports waivers in two contexts: topology waivers, which are based upon netlists, and geometry waivers for Logic-Driven Layout (LDL results). See “[Calibre PERC Waiver Flows](#)” in the *Calibre PERC User’s Manual* for details about both waiver types.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

Requirements for Running Calibre Auto-Waivers


Calibre Auto-Waivers requires its own license and existing Calibre licenses for each verification tool. Depending on the use case, you may also choose to configure the tool by writing a waiver setup file.

The section “[Licensing: Physical Verification Products](#)” in the *Calibre Administrator’s Guide* contains complete licensing information.

Table 1-2. Required Licenses for Calibre Auto-Waivers Tools

Mode	Required License(s)
Generating DRC or DFM waivers	Calibre Auto-Waivers and Calibre® nmDRC™/Calibre® nmDRC-H™
Generating ERC waivers	Calibre Auto-Waivers, Calibre nmDRC/Calibre nmDRC-H, and Calibre® nmLVS™/Calibre® nmLVS-H™
Calibre® YieldServer waivers	Calibre Auto-Waivers and Calibre® YieldAnalyzer/Calibre® YieldEnhancer
DRC verification with waivers calibre -drc -waiver	Calibre nmDRC/Calibre nmDRC-H and (Calibre Auto-Waivers or an additional Calibre nmDRC/Calibre nmDRC-H license pair)

Table 1-2. Required Licenses for Calibre Auto-Waivers Tools (cont.)

Mode	Required License(s)
DFM verification with waivers calibre -dfm -hier -waiver	Calibre nmDRC/Calibre nmDRC-H, Calibre YieldAnalyzer/YieldEnhancer and (Calibre Auto-Waivers or an additional Calibre nmDRC/Calibre nmDRC-H license pair)
ERC verification with waivers calibre -spice -waiver (hierarchical) or calibre -lvs -waiver (flat)	Calibre nmDRC/Calibre nmDRC-H, Calibre nmLVS/Calibre nmLVS-H, and (Calibre Auto-Waivers or additional Calibre nmDRC/Calibre nmDRC-H or Calibre nmLVS/Calibre nmLVS-H license pairs)
Calibre RVE	Calibre RVE
Multi-threaded runs for the above applications	Suitable number of additional Calibre nmDRC/ Calibre nmDRC-H and Calibre nmLVS/Calibre nmLVS-H license pairs, plus additional Calibre Auto-Waivers licenses for each license pair (or appropriate substitute licenses).  Note: The mt2cpu license is not supported for the waiver_flow executable.

The following items are also required:

- Layout database in GDS, OASIS, LEF/DEF, or OpenAccess format.
- Waiver setup files. These files are used for waiver shape generation tools and calibre -waiver. For details see “[Waiver Setup File](#)” on page 116 and “[Writing the Waiver Criteria File for Waiver Verification](#)” on page 183.
- Waiver criteria and waiver cells files. These files are required for waiver generation in waiver_flow. The waiver criteria file is optional for waiver generation in Calibre RVE and waiver verification with calibre -waiver. For details see “[Waiver Description Files](#)” on page 143.

Be aware of the following limitations for rule files used with a Calibre Auto-Waivers run:

- The rule file may not include [Layout Allow Duplicate Cell](#) NO and [Layout Input Exception Severity](#) DUPLICATE_CELL settings other than 3 or 4 when

WAIVER_DATABASE is specified in the waiver setup files. Layout Allow Duplicate Cell NO is allowed when MERGE YES is specified. Similarly, Layout Input Exception Severity 2 is allowed in this situation.

- Dual-database comparison mode is not supported.

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)


Calibre Environment Variables

Calibre tools require that the CALIBRE_HOME environment variable be set.

Table 1-3. Calibre Auto-Waivers Environment Variables

Environment Variable	Description
CALIBRE_HOME	See “ Setting the CALIBRE_HOME Environment Variable ” in the <i>Calibre Administrator’s Guide</i> for details. For lists of Calibre environment variables, see “ Calibre Environment Variables ” in the <i>Calibre Administrator’s Guide</i> and Environment Variables of the <i>Calibre RVE User’s Manual</i> .

Table 1-3. Calibre Auto-Waivers Environment Variables (cont.)

Environment Variable	Description
CALIBRE_WAIVER_IGNORE_SUMMARY_HIER	<p>Set this environment variable to a non-null value to instruct the tool to ignore the HIER option in the DRC Summary Report specification statement and directly generate the unwaived rules. The “by cells” statistics are not written.</p> <p>As of 2019.3, Calibre Auto-Waivers directly generates results for rules that are not waived in order to improve runtime.</p> <p>Rules that are not waived will be directly generated only if the following requirements are met:</p> <ul style="list-style-type: none"> • The DRC Results Database type is ASCII. • The DRC and DFM Summary Report does <i>not</i> have the HIER option (this is the default behavior). <p> Note: This limitation is disabled when this environment variable is set.</p> <ul style="list-style-type: none"> • Not all unwaived rules can be directly generated. The rules must meet the following criteria: <ul style="list-style-type: none"> • They must not appear in a DRC Check Map statement that is not ASCII. • They must not have Density operations (whether it is a direct output, or assigned to an intermediate layer inside the rule check) with RDB or Print options • They must not have DFM Measure operations (whether it is a direct output, or assigned to an intermediate layer inside the rule check)
CALIBRE_WAIVER_OLD_ENGINE	<p>If you encounter any of the issues described in “Known Issues and Workarounds” on page 32, you have the option to use the Calibre Auto-Waivers engine that was released in 2016.4 and earlier versions.</p> <p>To use the legacy Calibre Auto-Waivers engine, set the CALIBRE_WAIVER_OLD_ENGINE environment variable to 137955. For example, in C-shell, do the following before invoking a waiver run:</p> <pre>setenv CALIBRE_WAIVER_OLD_ENGINE 137955</pre>

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

SVRF Command Support

Most of the commonly used layer operations for DRC rule checks are processed by default by the waiver_flow executable and calibre -waiver.

See “[DRC Automatic Waivers](#)” on page 21 for information on SVRF operations performed for calibre -drc -waiver.

[Calibre Auto-Waivers Workflow](#)

[Usage Models for Calibre Auto-Waivers](#)

DFM Operation Support	27
ERC Layer Operation Support	28
Discouraged Layer Operation Use	28
Unsupported Layer Operations in waiver_flow	29
Unsupported Elements	29
Encryption Support	30

DFM Operation Support

Certain DFM operations are supported by Calibre Auto-Waivers.

The [DFM Copy](#), [DFM Property](#), [DFM RDB](#), and [DFM Space](#) operations are supported for DRC waiver generation and verification runs.

The [DFM Analyze](#) and DFM Property operations are supported for DFM Critical Feature Analysis waiver runs.

The tool supports waiving of the DFM Analyze operation's output when it has only one input layer. If you have multiple DFM Analyze operations with the same input layer in the same rule check, the input layer is waived for each DFM Analyze operation and appears multiple times under the rule check in the waived RDB file. In the case of multiple input layers, the waiver operation will be ignored and the tool outputs a warning message.


The operation results output is processed for automatic waiver generation. The DFM RDB operation is placed inside of a rule check, and this rule check generates a separate results database from the global DRC Results Database. If such a rule check is specified in a waiver_flow run, then the results that are sent to the RDB are also used to generate waiver shapes in the waiver output files. DFM RDB results can also be waived manually using Calibre RVE. Waivers are automatically applied in a calibre -waiver run to the RDB.

The DFM RDB GDS and OASIS statement and options are supported with the following limitations:

- The dynamic results reporting and direct generation features are disabled.


- Any rule check operation that uses this command is incorrectly reported in the summary file as “Not Executed”, which is not accurate. The checks are executed.
- The Calibre run will terminate unexpectedly if the layer used in the DFM RDB GDS or OASIS statement is not saved in the DFMDB. The layer must be used in at least one rule check to be waived.
- A warning is printed when the DFM RDB GDS and OASIS statement will not be waived.

Note

 The tool always adds a DFM Database statement to generate the *waiver.dfmdb* database, and overwrites any existing DFM Database statement. Prior to 2019.4, the tool used the user-specified statement.

If there are no rule checks in the rule file, the tool issues an error message. Prior to 2019.4, if there were no rule checks, but the you included the DFM Database statement with the “DEVICES” option, the devices were extracted and the run completed successfully.

Note

 In Calibre Auto-Waivers, the DFM RDB operation should not be used in a rule check where there are other layer operations that have output to the DRC Results Database. In such a case, the rule check should be rewritten as separate rule checks, one for standard DRC output, and the other for DFM RDB output.

ERC Layer Operation Support

If the `waiver_flow -erc` option is used, then the operation is enabled for use by default, along with the Net Area family of layer operations. ERC waivers are highly context-dependent because of connectivity considerations. Note that the ERC Pathchk specification statement is not supported for waivers.

Discouraged Layer Operation Use

Some layer operations are generally discouraged for automated waivers because the waiver output is highly dependent upon the state of the design. Discouraged operations may be used, but you should be aware of their main limitation: sensitivity to the completeness of the design and the entire connectivity layer set.

Layer operations that are generally discouraged for automated waivers include these:

- Layer operations (DRC flows) that are not enabled with an Calibre nmDRC-H license alone.

- Connectivity-dependent layer operations such as [\[Not\] Net](#), [Net Interact](#), and operations that use the [\[Not\] Connected](#) or [By Net](#) keywords.

Note



Connectivity is required for most ERC waivers, such waivers are highly dependent on design context.

The [Net Area](#), [Net Area Ratio](#), and [Net Area Ratio Accumulate](#) operations do not produce waiver shape output by default in waiver_flow. These are connectivity-related operations, so they are generally discouraged in the DRC waiver context. You can cause them to produce output in waiver_flow by using the NOT_IGNORE statement in the waiver generation setup file, or by waiving results in Calibre RVE. Use the waivers carefully because they are dependent on the state of the design.

- The TVF runtime layer operation is ignored by default. It only produces output in conjunction with the NOT_IGNORE statement in the waiver generation setup file. The TVF runtime operation is not supported for a calibre -waiver run; however, the waiver shapes can be detected using Tcl code in the TVF runtime context.

Unsupported Layer Operations in waiver_flow

Some layer operations are not supported in waiver_flow only. Results from such operations may have waiver shapes produced for them by other tools such as Calibre RVE.

In the DFM family of operations, [DFM Analyze](#), [DFM Copy](#), [DFM Property](#), [DFM RDB](#), or [DFM Space](#) are supported, but all others are not. The layer operations that are not supported by waiver_flow can have waiver shapes generated for them using Calibre RVE or [drc_waiv_asc2gds](#).

Unsupported Elements

Some layer operations are not supported for automated waivers at all.

These operations include the following:

- Result output to auxiliary RDB files generated by Net Area Ratio is not waived. Density RDB output is not waived by default; however, a WAIVE_DENSITY statement in the waiver criteria file enables Density RDB waivers.
- Additional Calibre nmDRC layer operations not supported for waivers are shown here:

Density Convolve	Extent Drawn	TDDRC ¹
------------------	--------------	--------------------

1. These operations generate waiver shapes but are not qualified for waiver flows.

Encryption Support

You must follow certain guidelines to ensure correct waiver application for encrypted checks.

Review the following guidelines for handling rules with TVF encryption:

- For specification statements that control data processing in general, such as DRC Results Database and ERC Results Database, it is recommended *not* to use encryption. This is because the Calibre Auto-Waivers tool adds similar statements in the intermediate files for generating waivers. If these lines are encrypted, they can cause errors in the run.
- If the rule file has a rule check using the following format:

```
rule {  
    <encrypted_section>  
    <open_operation>  
}
```

The rule check is treated as completely encrypted.

- Waivers can only be generated from encrypted rule checks using Export Waivers in Calibre RVE or with the drc_waiv_asc2gds utility. Waivers cannot be generated for encrypted rules using waiver_flow.

Note



Waivers generated from encrypted rules by Calibre RVE or the drc_waiv_asc2gds utility cannot be applied using the old engine.

- SVRF encrypted operations are not supported for ERC waivers.
- SVRF encrypted operations are not supported within rule check blocks.
- The following specification statements are not supported when encrypted:

DFM Select Check	DRC Results Database	Layout Input Exception Severity
DFM Select Check By Layer	DRC Results Database Precision	Layout Path
DRC Check Map	ERC Results Database	Layout Primary
DRC Magnify Results	LVS Summary Report	

Syntax Conventions

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

Table 1-4. Syntax Conventions

Convention	Description
Bold	Bold fonts indicate a required item.
<i>Italic</i>	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.
<u>Underline</u>	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 { <i>element_name</i> ['('model_name')']} <i>device_layer</i> { pin_layer ['('pin_name')'] ...} ['<'auxiliary_layer'>' ...] ['('swap_list')' ...] [<u>BY NET</u> BY SHAPE]	

Related Topics

[Calibre Auto-Waivers Workflow](#)

[Requirements for Running Calibre Auto-Waivers](#)

Known Issues and Workarounds

If you encounter issues using this release of Calibre Auto-Waivers, refer to this topic for possible causes and workarounds.

The issues in this topic are related to a change in the internal engine for Calibre Auto-Waivers. If you encounter any of these issues, set the CALIBRE_WAIVER_OLD_ENGINE environment variable to 137955. For example, to set the environment variable in C-shell, do the following:

```
setenv CALIBRE_WAIVER_OLD_ENGINE 137955
```

Table 1-5. List of Issues for Calibre Auto-Waivers

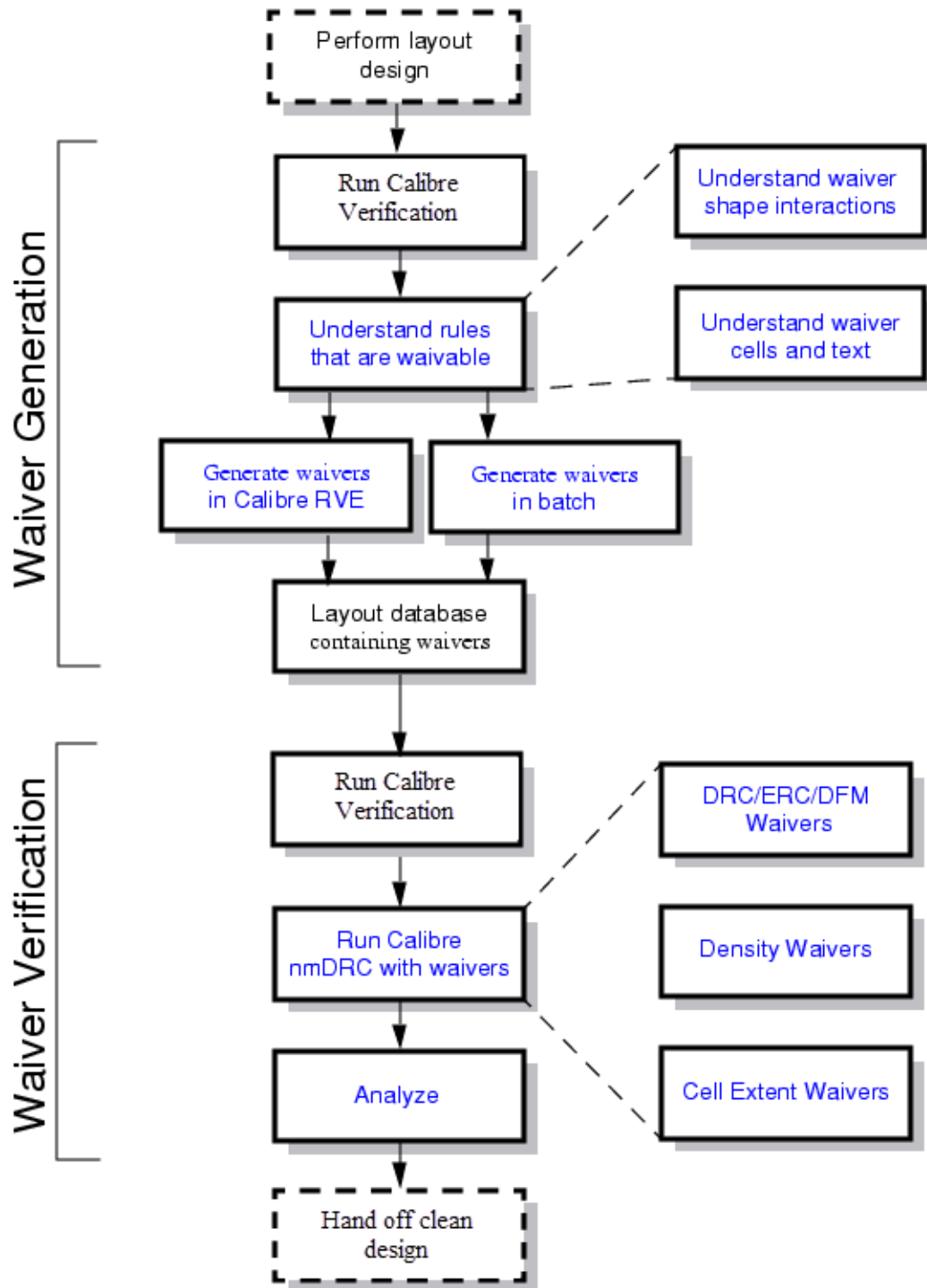
Description	DR #
The following statements are not currently supported by either engine: <ul style="list-style-type: none">• Layout Window Clip• Density Convolve• DFM Analyze• DFM Defaults RDB NODAL• DFM Measure• DFM Transition• DFM Spec Fill	N/A
If DRC Check Map is specific in an encrypted rule check, Calibre Auto-Waivers incorrectly writes the results from this check into the ASCII results database instead of the layout file.	1246262
Warning messages for missing side RDBs are not generated for the following operations: Density, DFM RDB, DFM Analyze, and NAR.	1243087

Chapter 2

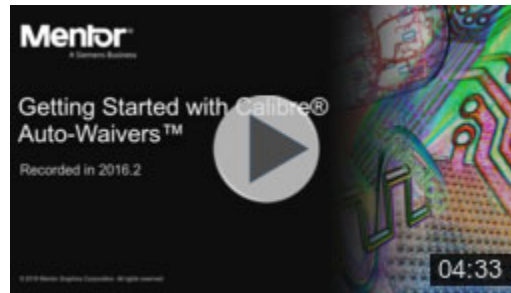
Getting Started With Calibre Auto-Waivers

You can perform waiver generation in batch mode or interactively using Calibre® Interactive™ or Calibre® RVE™. Calibre RVE is the preferred method for generating waivers. Waiver application can also be done in batch mode and Calibre Interactive.

Figure 2-1. Getting Started With Calibre Auto-Waivers Flow



The following video provides an introduction to the Calibre Auto-Waivers flow:



Third-Party Design Environments.....	35
Generating Waivers for DRC, ERC, or DFM.....	36
Using Waivers in the DRC, ERC, or DFM Flows.....	38
Density Window Waivers	42
Using Cell Extent Waivers	46
Frequently Asked Questions.....	48

Third-Party Design Environments

You can export waiver shapes into an existing design database (embedded waivers), or as separate layout files (non-embedded), or both. The waiver shape export method has implications for how waivers are used in third-party design tools.

If you are using a third-party design environment, you can import waiver geometry in one of the following ways:

- If the waiver geometry is embedded in design cells, then importing the design cells brings in the waiver geometry as well, so long as the waiver geometry and text layers are imported.
- If the waiver geometry is non-embedded, then the waiver cells can be imported separately. They can then be referenced from the main design in order to apply waivers at GDS or OASIS streamout.
- If the waiver geometry is non-embedded, then the waiver cells could be kept as GDS or OASIS files outside of the third-party design environment. The non-embedded waiver cell files can be merged with the streamed-out primary design by referencing all the files in the rule file [Layout Path](#) statement. The files containing non-embedded waiver geometry must then be referenced using WAIVER_DATABASE statements in the waiver setup file.
- If you are running an OpenAccess version of Cadence Virtuoso, waiver shapes can be directly imported to a Virtuoso design database from Calibre RVE. See [“Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE”](#) on page 86.

Related Topics

[Generating Waivers for DRC, ERC, or DFM](#)

Generating Waivers for DRC, ERC, or DFM

You can generate waivers for DRC, ERC, or DFM Critical Feature Analysis using Calibre RVE, Calibre Interactive, or the waiver_flow utility.

Prerequisites

- Review the “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Ensure you have written out the final edits to your layout IP.
- Ensure that one of these two statements is used in the rule file for DRC:

```
DRC CELL NAME YES CELL SPACE XFORM
```

```
DRC RESULTS DATABASE filename TOP
```

Caution



DRC RESULTS DATABASE ... TOP is not recommended for most applications. Reporting flat results to the top cell space creates much larger results databases, which can increase run time.

- For ERC waivers, the following statement must be used in the rule file:

```
ERC CELL NAME YES CELL SPACE XFORM
```

- ASCII results databases are available for the cells of interest.
- Review “[Creating Embedded Waiver Shapes Using Calibre RVE](#)” on page 75 and “[Creating Non-Embedded Waiver Shapes Using Calibre RVE](#)” on page 81.

Procedure

1. Identify the rule checks that produce violations that can be waived.
2. Open the design in your layout editor and a global ASCII results database in Calibre RVE.

Note



Do not perform edits on the design while performing the rest of this procedure.

3. Identify the cells in the design affected by waivers.


Typically, you should waive errors at the lowest possible level of the hierarchy.

4. Ensure that the layout editor is set up to retain text objects based upon layer and texttype when writing layout data.

5. Generate your waivers in Calibre RVE as follows:

- a. Right-click the results you want to waive and then choose **Waive > Waive**.

Note

 The waived results could already have been captured in an existing *.waived* file that is imported by DRC RVE.

- b. Choose **Tools > Export Waivers**.

- c. Select one of the following procedures to specify how your waivers are exported:

Waiver Type	Procedure
Embedded	<p>Merge waiver shapes with an existing design database:</p> <ol style="list-style-type: none"> 1. Click the Design tab and specify the path to an existing design database using the “Output file” field. 2. Enable the “Merge from file” and “Output file” checkboxes under the Options tab. <p>This merges the waiver shapes into the design and writes the output to the Output file.</p>
Non-embedded	<p>Create a new design database for the waiver shapes:</p> <ol style="list-style-type: none"> 1. Click the Design tab and specify the path to a new design database using the “Output file” field. 2. Disable the “Merge from file” option under the Options tab if you do not want the waiver shapes embedded in the design. <p>The waiver geometry is exported separately from the main design geometry to the “Output file” design.</p>

- d. Click the **Criteria** tab, and then enable the “Annotate waiver cells with waiver criteria text” checkbox.

- e. Click **Export** to create the file with waiver shapes and click through any additional message dialog boxes, as necessary.

6. Open a new text file and write the waiver setup file.

This file defines the inputs and output for a waiver run. See “[Writing the Waiver Setup File for Waiver Verification](#)” on page 181 for an example. The following is a basic verification setup file:


```

WAIVER_CRITERIA_EXTRACT /home/workarea/waiver_criteria
LAYER_NUMBER            1234
DATATYPE_NUMBER          5678
CALIBRE_WAIVER_NUMBER    AUTO

```

Layer 1234.5678 should not be written to or mapped in the rule file. The design should not contain data on this layer.

Note

 For advanced nodes, if your rule file precision and input layout precision is different, specify PRECISION_CONVERSION YES in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

7. Add one of the following sets of statements to the waiver setup file:

- If *all* of your waiver shapes are embedded in your main design, add this statement to the waiver setup file:

```
MERGE YES
```

This means your [Layout Path](#) design(s) have all waiver shapes embedded in them.

- If at least some of your waiver shapes are *not* embedded in the primary design and exist in separate GDS files, then specify MERGE NO. You should add the following statements to the waiver setup file for each file that contains external waiver shapes:

```
WAIVER_DATABASE /home/design/external_waivers1.gds  
WAIVER_DATABASE /home/design/external_waivers2.gds
```

Results

Step 5 generates either a design cell with embedded waiver geometry, or a waiver cell with waiver geometry. These are viewable in your layout editor.

The waiver criteria text annotations produced in Step 5 are SINGLE 100 100, if no waiver criteria file is specified. These are the default and safest values for waiver criteria. If the results are not as expected, try writing a criteria file that applies SINGLE 98 98, or lower, to specified rules. Import the file using the **Input waiver criteria file(s)** field under the **Criteria** tab.

The file produced in Step 6 is used for a Calibre physical verification run with waivers.

To verify the waivers, see “[Using Waivers in the DRC, ERC, or DFM Flows.](#)”

Related Topics

[Waiver Shape Generation Flow](#)

[Physical Verification with Waivers](#)

[Writing the Waiver Cells File for waiver_flow](#)


Using Waivers in the DRC, ERC, or DFM Flows

You can waive approved error results by applying the waivers in a verification run. This procedure assumes you have generated waiver shapes for your design.

Prerequisites

- The procedure “[Generating Waivers for DRC, ERC, or DFM](#)” on page 36 has been completed.
- If using a waiver-corrected IP library, the IP cells are used in your finished design.
- A waiver setup file and rule file are configured properly for your run.
- If the library vendor did not embed the waiver criteria as text objects in the IP, then a waiver criteria file is also needed. See “[Text Object Annotations](#)” on page 221 for more information about waiver cell text object annotations.

Note

 For advanced nodes, if your rule file precision and input layout precision is different, specify PRECISION_CONVERSION YES in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

Procedure

1. If you have any non-embedded waiver geometry files that are external to the main design, then include statements like this in the waiver setup file:

```
MERGE NO
WAIVER_DATABASE external_file1.gds
WAIVER_DATABASE external_file2.gds
```

2. If *all* of your waiver geometry is embedded in the Layout Path design(s), then include this in the waiver setup file:

```
MERGE YES
```

3. Enter one of the following commands to do a verification run with waivers applied:

Run Type	Invocation Command
DRC	<code>calibre -drc -hier -waiver waiver_setup_file \</code> <code>[drc_options] [-recon] rules</code>
ERC	<code>calibre -spice layout.spice -waiver waiver_setup_file \</code> <code>[lvs_options] rules</code>
DFM CFA	<code>calibre -dfm -hier -waiver waiver_setup_file \</code> <code>[dfm_options] rules</code>

Results

Waivers in the design are applied automatically. The DRC or ERC run produces the following files viewable in Calibre RVE:

File	Description
ASCII Results Database	DRC or ERC errors
<i>waived.rdb</i>	RDB file containing original DRC results that were waived
<i>used_waiver.rdb</i>	RDB file containing the waiver shapes that were applied successfully to the DRC result
<i>unused_waiver.rdb</i>	RDB file containing waivers that were not used

Note



To provide information in the results about the cell in which a waiver was applied, include the [ENABLE_WAIVER_CELL_REPORTING](#) YES statement in your waiver setup file before a verification run with waivers. This option creates a WAIVER_CELL column in the results database.

A DFM CFA run has the waivers applied automatically. You can open the DFM Database to view results. Waived errors do not appear in the database.

The run also produces a *calibre-waiver.summary* file that contains run-time data and messages.

The end of the transcript includes information about the run time. For example:

```
--- DFM CPU TIME = 0 + 3 REAL TIME = 1 LVHEAP = 8/14/14 SHARED = 0/1 MALLOC = 63/63/63
    ELAPSED TIME = 5
--- YIELD SERVER CPU TIME = 7 REAL TIME = 19 LVHEAP = 1153/1170/1170 MALLOC = 432/432/472
    ELAPSED TIME = 24
--- TOTAL RULECHECKS EXECUTED = 60
--- TOTAL RESULTS GENERATED = 0 (0)
--- TOTAL DFM RDB RESULTS GENERATED = 213845 (286887)
--- DRC RESULTS DATABASE FILE = error.db (ASCII)
```

Where the CPU times are calculated as follows:

```
DFM CPU TIME = <DFM_CPU_time for local machine> + <DFM_CPU_time for remotes>
YIELD SERVER CPU TIME = <YS_CPU_time for local machine>
Total cpu time = <DFM_CPU_time for local machine> + <YS_CPU_time for local machine>
```

The YieldServer time for remote CPUs is not included.

Note



You can optionally view waived results in Calibre RVE as soon as they are available. To enable this feature, do one of the following before performing a run with waivers:

- Include ENABLE_DYNAMIC_RESULTS_REPORTING YES in the waiver setup file.
 - Enable the “Enable dynamic results reporting” checkbox on the **DRC Run** tab in Calibre Interactive.
-

Related Topics

[Waiver Shape Generation Flow](#)

[Writing the Waiver Cells File for waiver_flow](#)

[Physical Verification with Waivers](#)

Density Window Waivers


Top-level Density results are merged into single shapes that can be comprised of multiple windows. Typically, you should waive results on a per-window basis rather than on a merged-window (top-level) basis to ensure the intended result.

Method 1 shows how to use Calibre RVE to generate waiver shapes of *merged* (top-level) Density results from the DRC Results Database. The waivers are then applied both on a per-window and per-merged-result basis.

Method 2 shows how to generate Density window result waivers using `waiver_flow`. Waiving of Density results using this method causes waivers to be applied on a per-Density-window basis rather than on a merged-results basis. Method 2 enforces waivers on the unmerged windows.

The final procedure in this section demonstrates how to perform a verification run using Density waivers.

Note

 It is more common to waive Density results using either cell extents or marker layers than what is shown in this section. If you want to waive Density results within the extent of a cell (such as an IP cell that has been verified as Density clean), see [“Using Cell Extent Waivers”](#) on page 46. Waiving both individual Density results and results inside the extents of waiver cells is possible.

Method 1: Generating Density Waivers with DRC RVE.....	42
Method 2: Generating Density Waivers with <code>waiver_flow</code>	44
Running Density Waiver Verification	45

Method 1: Generating Density Waivers with DRC RVE

You can use Calibre RVE to waive merged (DRC results database top-level) results interactively.

Prerequisites

- You have met [“Requirements for Running Calibre Auto-Waivers”](#) on page 23.
- You have a [DRC Results Database](#) with [Density](#) results (for Method 1 only).
- You are familiar with using Calibre RVE to generate waiver shapes. See [“Generating Waivers for DRC, ERC, or DFM”](#) on page 36.

Procedure

1. Open your design in a layout viewer.
2. Open your DRC results database with Density results in Calibre RVE.
3. Highlight the Density DRC result you want to waive using Calibre RVE.

For more information about using DRC RVE with Density results, see [“Analyzing Density Results with Calibre RVE”](#) in the *Calibre Verification User’s Manual*.

4. Right-click the result you want to waive, and then choose **Waive > Waive**.

Note



Waived results can be preselected by importing an existing *.waived* file.

5. Repeat Steps 3 and 4 until you have waived all the results that interest you.
6. Export waivers as follows:
 - a. Choose **Tools > Export Waivers**.
 - b. Click the **Design** tab, then type *waived.gds* in the “Output file” field.
 - c. Click the **Criteria** tab, then enable the “Annotate waiver cells with waiver criteria text” option.
 - d. (Optional) Set the “Input waiver criteria file(s)” field to the path of a waiver criteria file that contains waiver type and overlap values for specified density checks.

For example, if a waived density check is named *density_rule_1*, write a criteria file as follows:

```
density_rule_1 WAIVE_DENSITY 99
```

The 99 is somewhat arbitrary, but is a good starting point to test.

Note



Do not enable “Merge from file” under the **Options** tab.

- e. Click **Export**.

The waivers are exported according to your settings. You should verify that the exported layout has waiver shapes embedded in it.

7. Write a waiver setup file as follows:

```
WAIVER_CRITERIA EXTRACT waiver_criteria  
MERGE NO  
WAIVER_DATABASE waived.gds  
CALIBRE_WAIVER_NUMBER AUTO
```

8. Save the file as *waiver_setup*.

9. Proceed with “[Running Density Waiver Verification](#)” on page 45.

Related Topics

[Method 2: Generating Density Waivers with waiver_flow](#)

Method 2: Generating Density Waivers with waiver_flow

You can generate batch Density waivers for per-window results with the waiver_flow utility.

Prerequisites

- You have met the “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- You have a [DRC Results Database](#) with [Density](#) results (for Method 1 only).

Note



In order to generate waivers for density violations, your results database must have been generated with the following statement in your rule file:

```
DRC MAXIMUM RESULTS DENSITY ALL
```

- You are familiar with using Calibre RVE to generate waiver shapes. See “[Generating Waivers for DRC, ERC, or DFM](#)” on page 36.

Procedure

1. Write a waiver criteria file with this statement:

```
density_rule WAIVE_DENSITY 99
```

The 99 means the area of a Density results window must be overlapped by 99 percent in order for the result to be waived. (The 99 is somewhat arbitrary, but is a good starting point to test.)

Save the file as *waiver_criteria*.

2. Write a waiver cells file with this statement:

```
cell density_rule
```


The cell is the name of a cell for which you want to waive results for the density_rule.

Save the file as *waiver_cells*.

3. Write a waiver setup file as follows:

```
INPUT_LIBRARY design.gds
RULE_FILE rules
WAIVER_CRITERIA waiver_criteria
WAIVER_CELLS waiver_cells
LAYOUT_SYSTEM GDSII
MERGE NO
WAIVER_DATABASE waived.gds
```

Note

 For advanced nodes, if your rule file precision and input layout precision is different, specify `PRECISION_CONVERSION YES` in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

Save the file as *waiver_setup*.

4. Run the following command:

```
waiver_flow waiver_setup -turbo
```

Waivers are generated for the waiver rule in Step 1 in a file called *waived.gds*.

5. Check the *calibre-waiver.summary* file to ensure there were no problems in the *waiver_flow* run.

Related Topics

[Method 1: Generating Density Waivers with DRC RVE](#)

Running Density Waiver Verification

You apply Density waivers in a physical verification run similar to DRC or ERC waivers. You will run density waiver verification whether you used Method 1 or 2 to generate your Density waivers.

Prerequisites

- You have met the “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- You have completed either “[Method 1: Generating Density Waivers with DRC RVE](#)” on page 42 or “[Method 2: Generating Density Waivers with waiver_flow](#)” on page 44.
- Density waivers were generated using the DRC Maximum Results ALL statement.

Procedure

Enter the following command:

```
calibre -drc -hier -waiver waiver_setup -turbo rules
```

Results

Waivers in the design are enforced automatically. The run also produces a *calibre-waiver.summary* file that contains runtime data and messages and the following files viewable in DRC RVE:

File	Description
ASCII Results Database	DRC errors
<i>density_waived.rdb</i>	RDB file showing applied Density window waivers
<i>used_waiver.rdb</i>	RDB file containing the actual waiver shapes that were applied successfully to the DRC result
<i>unused_waiver.rdb</i>	RDB file showing potential Density window waivers that were not used. There are no properties available in this file.

Related Topics

[Using Cell Extent Waivers](#)

[Waiver Description Files](#)

[Waiver Shape Generation with Calibre RVE](#)

Using Cell Extent Waivers

You can waive rule check results within the extent of a cell, which can be used for any rule check for which you want to waive DRC or ERC results that touch the extent of a cell. Cell extent waivers are often useful in cases where an unplaced cell is known to be clean, but the cell has Density errors when placed in the design. This happens because the Density window grid uses a different basepoint in the full design.

If you want to waive individual Density results, see Method 1 under “[Density Window Waivers](#)” on page 42.

Waiving errors using a cell’s extent does not depend upon generating waiver shapes, so this procedure does not require Calibre RVE or waiver_flow to generate waiver shapes, but you can use either of those methods with this procedure.

Waiving DRC or ERC results based upon intersections with a marker layer is similar to performing cell extent waivers. To do this, the WAIVE_MARKER command is used in the [Waiver Criteria File Format](#).

See “[Waiver Description Files](#)” on page 143 for a complete description of syntax and semantics for the waiver criteria and waiver cells files.

Prerequisites

- Review the “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Ensure the waiver setup file is properly configured. See “[Generating Waivers for DRC, ERC, or DFM](#)” on page 36” for an example of a basic waiver setup file.

Procedure

1. Identify the cells and rules in which you want to waive results within the cells’ extents.
2. In your waiver criteria file, enter a line similar to the following for each rule for which you want waivers:

```
rule SINGLE 0 0 WAIVE_EXTENT cell
```

Multiple cells can be specified with the WAIVE_EXTENT statement. The second value is the result overlap percentage value for enforcing waivers. In this case, the cell extent must overlap the DRC-style result by at least 1 database unit for the result to be waived.

3. If you want to waive Density results on a per-window basis, then enter this in the waiver criteria file for each rule you want to waive:

```
density_rule WAIVE_DENSITY 99 WAIVE_EXTENT cell
```

These two statements work together to waive the density_rule results. The 99 value means the extent of the cell must overlap 99 percent of the area of a Density window in order for the window to be waived.

4. Run waiver verification:

```
calibre -drc -hier -waiver waiver_setup_file -turbo rules
```

Results

Waivers in the design are enforced automatically. The run produces the following files viewable in Calibre RVE:

File	Description
ASCII Results Database	DRC errors
<i>density_waived.rdb</i>	RDB file containing original DRC results that were waived
<i>used_waiver.rdb</i>	RDB file containing the actual waiver shapes that were applied successfully to the DRC result
<i>unused_waiver.rdb</i>	RDB file containing waivers that were not used
<i>density_waived.rdb</i>	RDB file showing applied Density waivers, if any

The run also produces a *calibre-waiver.summary* file that contains runtime data and messages.

Related Topics

[Density Window Waivers](#)

[Waiver Description Files](#)

[Waiver Shape Generation with Calibre RVE](#)

Frequently Asked Questions

This topic provides solutions to common issues related to the Calibre Auto-Waivers flow.

Table 2-1. Calibre Auto-Waivers FAQ

Question	Answer	Cross-References
Which waiver criteria should be used?	SINGLE 98 98 is a good place to start. Changing the values to lower numbers waives more DRC errors.	“Waiver Shape Interactions” on page 53
When should you merge the waiver geometry with your design?	Do this when you want to simplify the flow so the waivers and the design are in a single file. Also do this if the waiver subcells are already placed in the appropriate cell.	
How do you merge waiver geometry with your design?	For Calibre RVE, choose Tools > Export Waivers > Options tab > Merge from File option. When using waiver_flow batch generation, use the MERGE YES statement in the waiver setup file.	“Waiver Setup File Format for Waiver Verification” on page 126
When should you not merge waiver geometry with your design?	Do this when you want to restrict who can edit the waivers. Also do this if you want to keep the layout design simpler. Use the MERGE NO statement in the waiver setup file.	
How do you apply waivers when they are both embedded in the primary design and stored as external waiver shapes in separate files?	Specify MERGE NO in the waiver setup file. Also specify WAIVER_DATABASE for each file that contains waiver shapes.	

Table 2-1. Calibre Auto-Waivers FAQ (cont.)

Question	Answer	Cross-References
Should you use a waiver criteria file or embed the waiver criteria as text objects in the waiver geometry cells?	<p>Embedding the criteria as text makes for a simpler flow because there are fewer files to manage. A waiver criteria file can always be extracted from the text objects at a future time. Embedding text objects makes the waiver criteria more difficult to modify; the waiver criteria file is easier to maintain in this respect.</p> <p>If waiver criteria other than SINGLE 100 100 is desired, a waiver criteria file should be written and specified before exporting from Calibre RVE.</p>	Waiver Description Files
How do you embed waiver criteria text objects in the waiver geometry cells?	<p>If using Calibre RVE to export waivers, then use the “Annotate waiver cells with waiver criteria text” option.</p> <p>Alternatively, the waiver_util tool, specified with -mergedesc, can embed waiver criteria text objects in layout files.</p>	Waiver Shape Generation with Calibre RVE waiver_util
How do you extract waiver criteria from text objects in waiver geometry cells?	Use the WAIVER_CRITERIA EXTRACT option in the waiver setup file.	Waiver Setup File Format for Waiver Verification
What if you change the name of your IP cells?	<p>This situation can be handled by using the IP_MATCH command in the waiver cells file.</p> <p>Alternatively, you can change the <i>cellname</i> strings within the corresponding Waive\$wv\$... waiver cell names to match the altered cell names. The <i>rulecheck</i> strings in the waiver cell names should not be changed.</p>	Waiver Cell Description

Table 2-1. Calibre Auto-Waivers FAQ (cont.)


Question	Answer	Cross-References
What if the geometry in your IP cell changes?	So long as this does not affect rules that have error waivers, this should have no effect on waivers. Otherwise, waiver geometry may need to be regenerated or waiver criteria adjusted.	
What if your rule changes for which there are waivers?	<p>Depending on the change, this can cause waivers to no longer apply. If a waiver no longer applies due to a rule change, you should regenerate the waiver shapes.</p> <p>To ensure that this does not occur unintentionally, use rule checksums when you generate and apply your waivers.</p> <p> Note: To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See “Calibre Environment Variables” on page 25.</p>	Rule Checksum Annotations, Using Rule Checksums in calibre -waiver, and Creating Waiver Cell Rule Checksum Annotations with waiver_flow
Can you edit the Waiver\$wv\$... cells?	This is not recommended. It is better to adjust your waiver criteria instead. If a waiver cell is edited and RUN_LAYOUT_CHECKSUM YES is specified in the waiver setup file, this can cause a checksum warning for any mismatches.	Waiver Cell Description and Waiver Setup File Format for Waiver Verification
Which layout database formats are supported?	Any database format supported by the Layout System specification statement. Ensure cell names, layer numbers, and text objects are output properly.	Waiver Setup File Format for Waiver Verification

Table 2-1. Calibre Auto-Waivers FAQ (cont.)

Question	Answer	Cross-References
Can you waive ERC errors?	Yes. The procedure you should use is similar to the Library Flow for DRC. Your ERC waiver shapes will be embedded in lower-level cells and the waivers applied automatically, just as in DRC.	Generating Waivers for DRC, ERC, or DFM. Using Waivers in the DRC, ERC, or DFM Flows.
You already have a DRC Results Database or an Calibre RVE <i>.waived</i> file. Can you generate waiver shapes for these?	Yes. The <code>drc_waiv_asc2gds</code> tool is used for that purpose.	drc_waiv_asc2gds
You have a waiver description file, how do you translate it into the waiver criteria and waiver cells files?	The <code>WAIVER_DESCRIPTION</code> statement is deprecated, but if it is specified in your waiver setup file, Calibre Auto-Waivers translates the file into the correct criteria and cells formats. Alternatively, <code>waiver_util -splitdesc</code> can translate legacy waiver description files.	Waiver Description Files waiver_util
Checksums are not written to the waiver shapes when they are exported from Calibre RVE. Calibre RVE reports that the waiver cell rule check name does not match any rule file check name, so no checksums are written.	Check results in Calibre RVE can be appended with a “::<” string separator and numerical value. This causes a mismatch between the check name in the rule file and the check name in the waiver cell when the waiver is exported. In this case, checksums cannot be generated. To avoid this issue, enable the “Truncate synthesized check names when exporting waivers” option on the Calibre RVE Setup Waiver Options pane. This removes the additional text in the check name when waiver cells are created.	Waiver Options in Calibre RVE

Table 2-1. Calibre Auto-Waivers FAQ (cont.)

Question	Answer	Cross-References
What if I change the parent cell name of a cell that contains waivers?	<p>Assume there is a cell name in the waiver database file of the following form:</p> <pre>Waive\$wv\$_checkname_in_cellname</pre> <p>The waiver cell is contained in a parent cell named <i>cellname</i>.</p> <p>If you change the parent cell name (for example, <i>prefix_cellname</i>) without modifying the name in the waiver cells file, the cell will not be waived because Calibre Auto-Waivers uses the old parent cell name</p>	<p>“Waiver Cell Description” on page 218.</p>

Related Topics

[Generating Waivers for DRC, ERC, or DFM](#)

[Using Waivers in the DRC, ERC, or DFM Flows](#)

[Waiver Shape Interactions](#)

Chapter 3

Waiver Shape Interactions

The waiver criteria file controls how waiver shapes are applied in a physical verification run. It is important to understand the different types of interactions that waiver shapes can have with an error result region and how the criteria can be used to determine whether a result is waived.

Waiver shapes for edge and edge cluster outputs behave differently than result regions, so some consideration is necessary when generating and applying results for these types of outputs.

The following video describes how waiver shapes interact with error results:



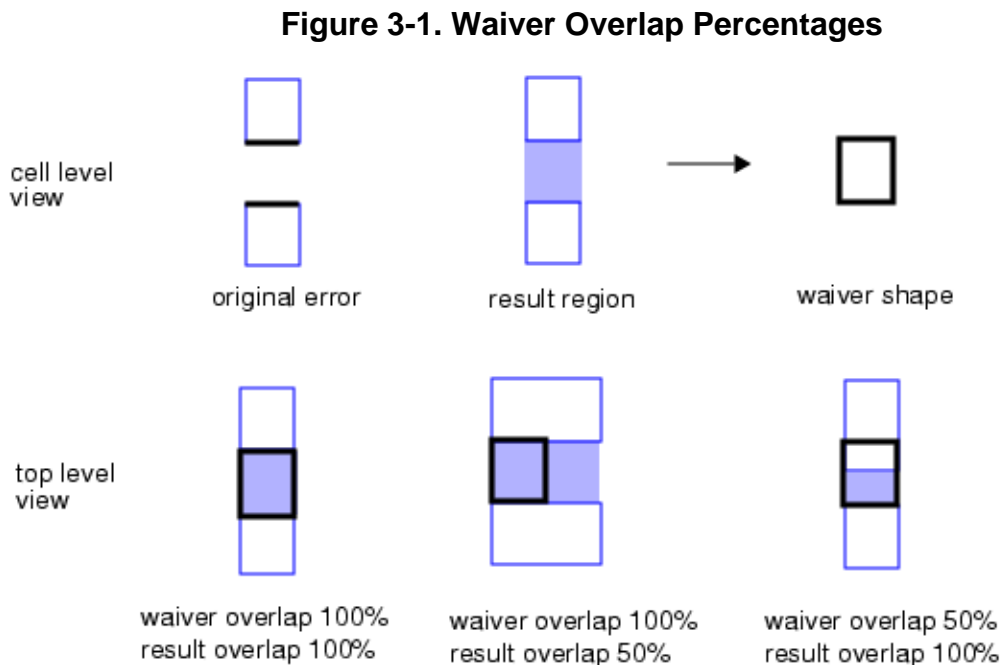
Waiver Shape Interaction Types	54
Waiver Type Summary	60
Waivers for Edge and Edge Cluster Outputs	61
Block-Level Waivers (Gray Box)	62

Waiver Shape Interaction Types

Waiver shapes are generated either using Calibre RVE interactively or running the waiver_flow executable in batch mode.

The waiver criteria and waiver cells files (see “[Waiver Description Files](#)” on page 143) determine how waiver shapes are generated in a batch run and how waivers are applied in a error waiver run. (Calibre RVE does not require waiver description files, but a waiver criteria file can be used to load non-default criteria text in the Calibre RVE Export Waived Results dialog box.)

As shown in the following figure, waiver shapes can intersect error result regions when the waiver cells are placed in the original design. The following figure shows the concepts of waiver overlap and result overlap for single waiver shapes.



The percentage of a waiver shape that overlaps a result region is the *waiver overlap*.
The percentage of a result region that overlaps a waiver shape is the *result overlap*.

The selection of the waiver type, along with the waiver overlap and result overlap percentages, define how waivers are generated and applied in DRC, DFM, or ERC.

The following examples show various combinations of waiver types and overlaps, and syntactical elements that are used to define waivers for the tool. You should start by testing the SINGLE 98 98 setting to see how many waivers this enforces.

Waiver classifications come in two types:

SINGLE *waiver_overlap result_overlap*

MULTI *waiver_overlap result_overlap*

The **SINGLE** keyword specifies only single waiver shape interactions are waived. The **MULTI** keyword means multiple waiver shape interactions are waived. The *waiver_overlap* and *result_overlap* parameters are integer values that specify overlap percentages of waiver area and result area, respectively.

These keywords are entered in a waiver criteria file. See “[Waiver Criteria File Format](#)” on page 147 for details.

SINGLE Interactions	55
MULTI Interactions	58

SINGLE Interactions

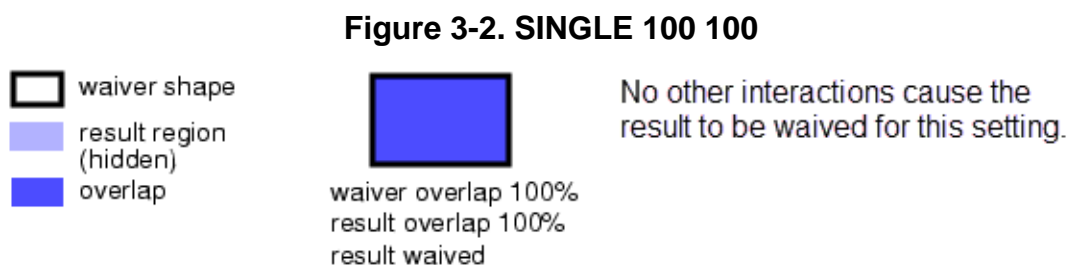
A **SINGLE** interaction enforces the condition that only one waiver shape can interact with a result region in order for the waiver to apply. Single interactions are the most conservative method of applying a waiver.

The following video includes animations that describe **SINGLE** interactions:



SINGLE 100 100

SINGLE 100 100 is the most conservative waiver setting. It means one waiver shape must share 100 percent of its area with the result region, and must completely overlap the result region.



Use this setting if you have any uncertainty about which setting to use because it is the most conservative. It waives only those error results that match the waiver shape exactly. If you need to relax the waiver criteria, you can always do that in subsequent runs.

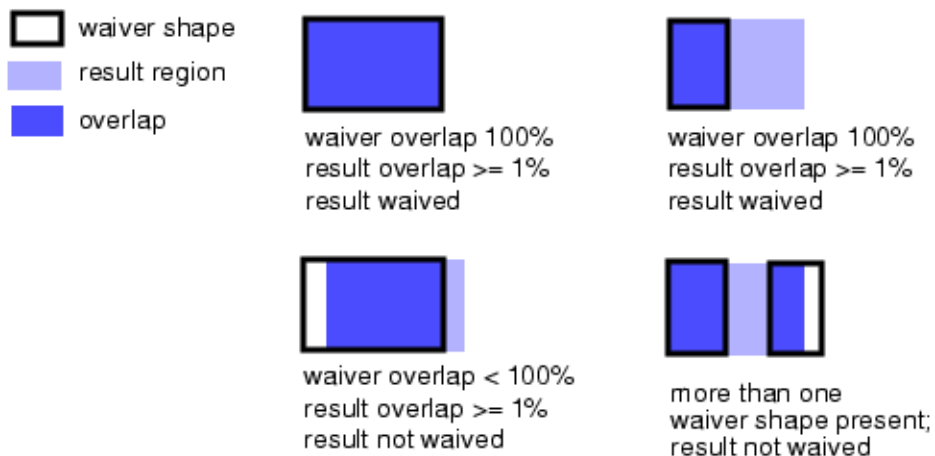
Note

Because of differing calculations for grid snapping of skew edges, SINGLE 100 100 can cause differing results based upon run mode (that is, 32-bit, 64-bit, flat, hierarchical, MT, hyperscaling, and operating system can all have an effect on grid snapping). A slightly more relaxed setting of SINGLE 98 98 may be preferable if there are grid-snapping issues because of skew edges, or you are seeing result count differences based upon run mode.

SINGLE 100 1

SINGLE 100 1 means one waiver shape must share 100 percent of its area with the result region, and the result region must have at least 1 percent overlap with the waiver shape.

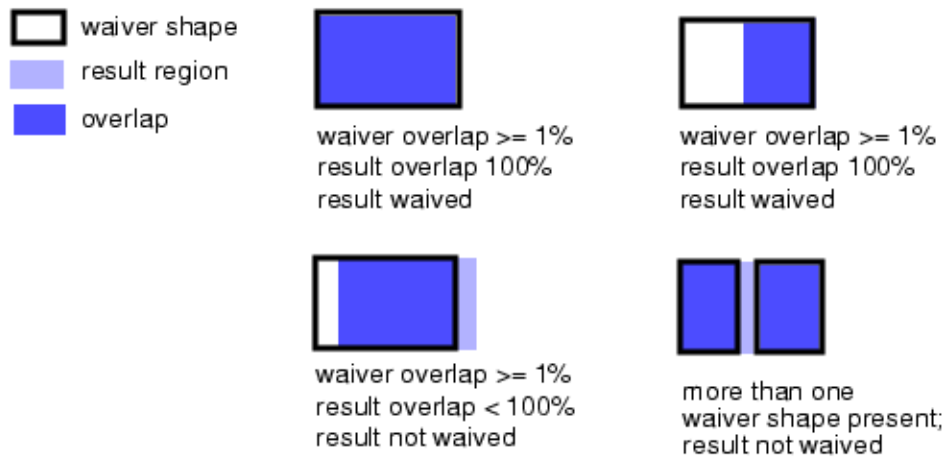
Figure 3-3. SINGLE 100 1



SINGLE 1 100

SINGLE 1 100 means one waiver shape must have at least 1 percent overlap with the result region, and the result region must be completely covered.

Figure 3-4. SINGLE 1 100



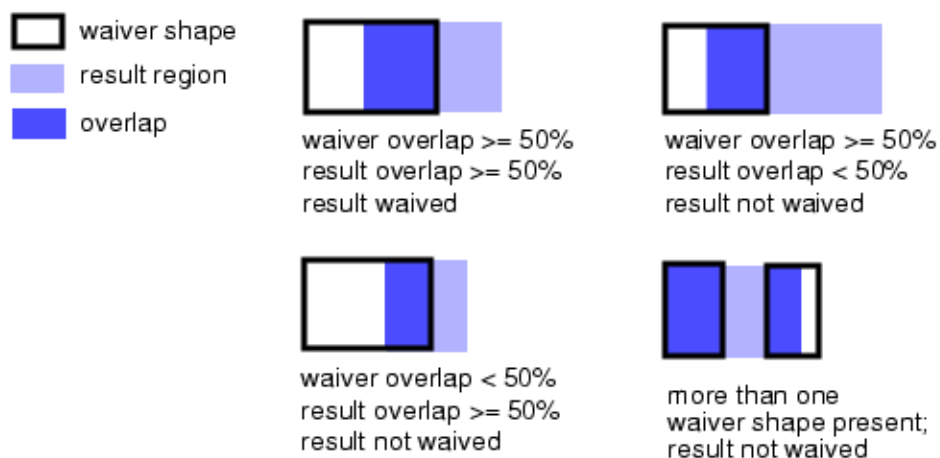
SINGLE 1 1

SINGLE 1 1 means a mutual 1 percent overlap of a single waiver shape with a result region causes a waiver. This setting may be useful for observing all (or nearly all) potential single-interaction waivers.

SINGLE 50 50

SINGLE 50 50 means the waiver overlap and result overlap must both be greater than or equal to 50 percent for the waiver to apply.

Figure 3-5. SINGLE 50 50



Related Topics

[Waiver Shape Interaction Types](#)

[MULTI Interactions](#)

MULTI Interactions

A MULTI interaction enables more than one waiver shape to overlap a result region.

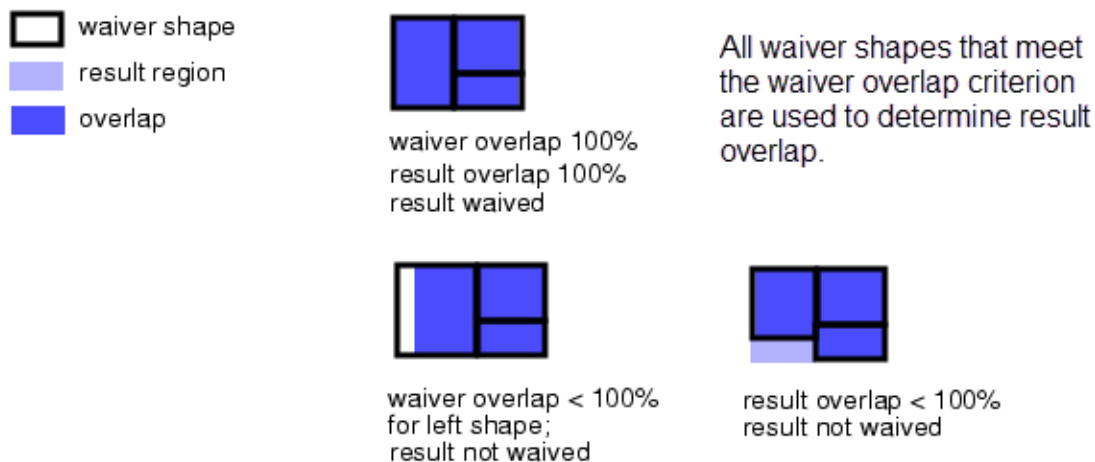
For all waiver shapes that meet the waiver overlap criterion, the sum of the areas (intersections between these areas are counted once) that overlap the result region is computed. If that sum meets the result overlap criterion, then the waiver is applied. In cases where only a single waiver shape overlaps a result region, the MULTI interaction behaves the same as the SINGLE interaction.

The MULTI 0 0 setting can be useful when used in conjunction with rule checks that use Calibre Pattern Matching. This setting does not waive everything for a given design cell and rule. Where there is a non-zero overlap between the waiver shape and results region, then waivers are enforced.

MULTI 100 100

MULTI 100 100 means there must be at least one waiver shape that shares 100 percent of its area with the result region, and all waiver shapes that meet the 100 criterion for waiver overlap must completely cover the result region.

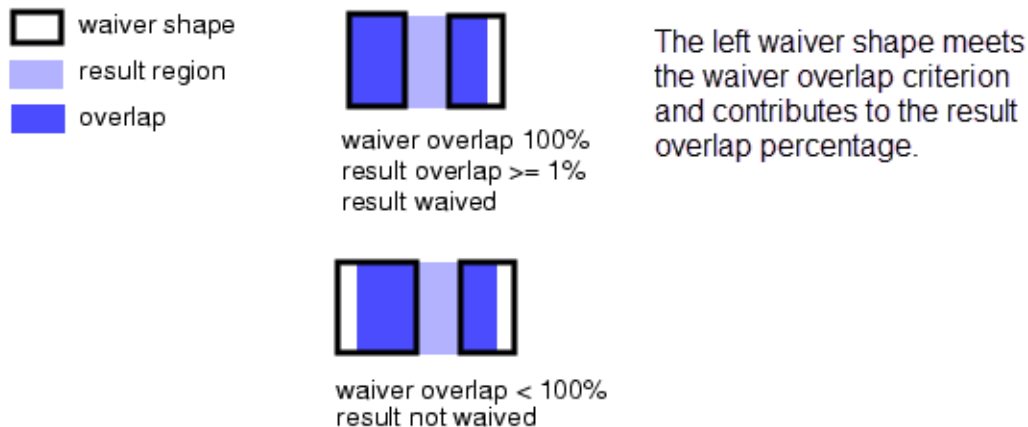
Figure 3-6. MULTI 100 100



MULTI 100 1

MULTI 100 1 means there must be at least one waiver shape that shares 100 percent of its area with the result region, and the result region must have at least a 1 percent overlap with all waiver shapes that meet the 100 criterion.

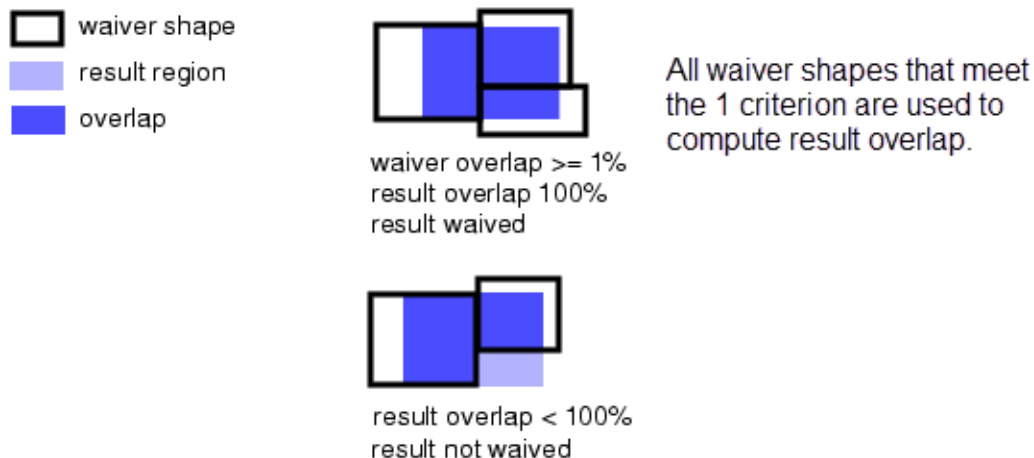
Figure 3-7. MULTI 100 1



MULTI 1 100

MULTI 1 100 means there must be at least one waiver shape that has at least a 1 percent overlap with the result region, and all waiver shapes that meet the first criterion must completely cover the result region.

Figure 3-8. MULTI 1 100

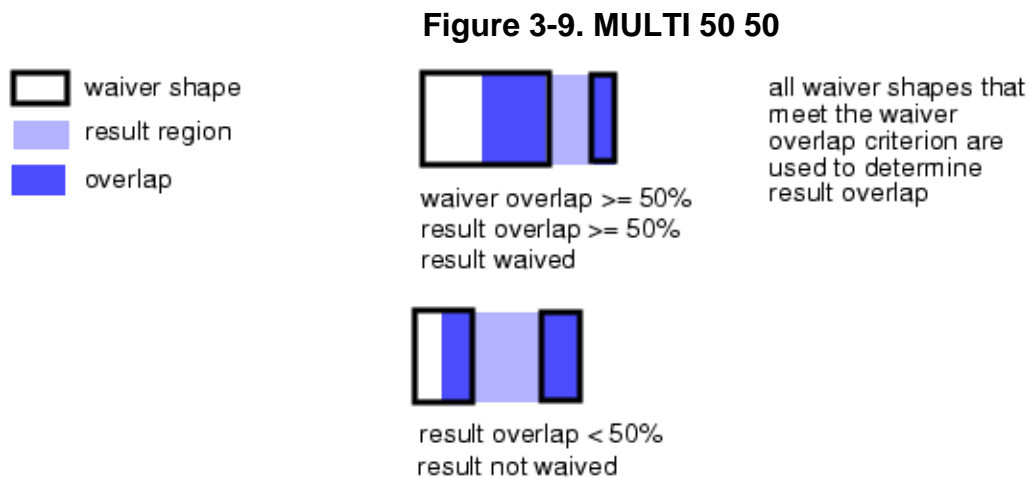


MULTI 1 1

MULTI 1 1 means that a mutual 1 percent overlap of a waiver shape with a result region causes a waiver. More than one waiver shape may be present. This is the least restrictive setting and results in the most waivers of any setting.

MULTI 50 50

MULTI 50 50 means there must be at least one waiver shape that shares 50 percent of its area with the result region, and the result region shares 50 percent of its area with all waiver shapes that meet the waiver overlap criterion.



Related Topics

[Waiver Shape Interaction Types](#)

[SINGLE Interactions](#)

Waiver Type Summary

The waiver and overlap criteria control the relative conservativeness of the various waiver settings. The more conservative a setting is, the fewer waivers are enforced. Also, the more conservative a setting is, the closer the waiver shape (or shapes) must exactly overlay the shape of the error result region in order for the waiver to apply.

The settings behave as follows:

- The waiver overlap and result overlap percentage settings become more conservative as they approach 100.
- For a given set of waiver overlap and result overlap percentages, SINGLE is the more conservative waiver type; MULTI is the less conservative.

You should start with SINGLE 98 98 to see how many waivers this enforces. You can relax the criteria if you find this setting is too restrictive.

Related Topics

[Waiver Shape Interaction Types](#)

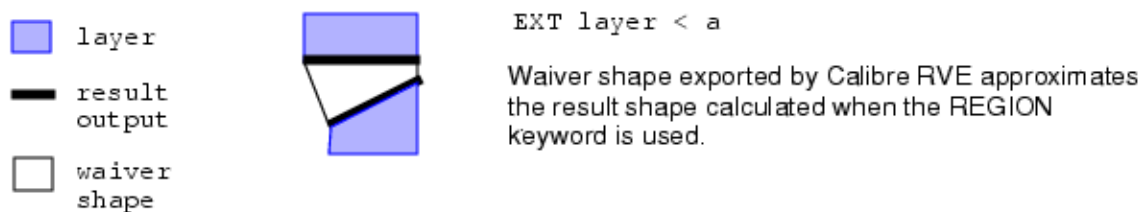
[Waivers for Edge and Edge Cluster Outputs](#)

Waivers for Edge and Edge Cluster Outputs

Rule checks can output single edges or edge clusters. Waivers are based upon area coverage, but edges have no area; hence, Calibre Auto-Waivers converts edges and edge clusters to polygons in order to apply waivers.

When the DRC result is an error edge cluster and the original edge cluster does not form a rectangle, the resulting waiver shape may not match the result shape produced by the Calibre run. This can happen when the edge clusters from [Enclosure](#), [External](#), and [Internal](#) operations form non-Manhattan polygons, as shown in [Figure 3-10](#).

Figure 3-10. Calibre RVE Waiver Shape Export From Edge Pairs



Calibre RVE and the waiver_flow tool perform the same operations on edge clusters. A verification run (calibre -waiver) handles edge clusters with minor differences. See “[Edge Clusters in Calibre RVE and waiver_flow](#)” on page 61 and “[Edge Clusters in calibre -waiver](#)” on page 62.

Tip

i When waivers generated from edge clusters are applied in a verification run, waived edge outputs are expanded into polygons and waived error clusters are converted into polygons in the *waived.rdb* database. If you specify RETAIN_RESULT_TYPE YES in the waiver verification setup file, the original edge clusters are written to the *waived.rdb* file, instead of converted results.

Edge Clusters in Calibre RVE and waiver_flow

Calibre RVE and the waiver_flow tool attempt to create the same polygon a Calibre nmDRC run would with the REGION keyword included in the rule check.

In most cases, the waiver shape exported by Calibre Auto-Waivers is close to the actual result shape. To ensure an exact match, use the REGION keyword in the layer operation to produce polygon result output. Calibre Auto-Waivers uses the polygon result output as the waiver shape. You may also use the waiver_flow tool, as described in “[Writing the Waiver Cells File for waiver_flow](#)” on page 101.

Edge Clusters in calibre -waiver

In order to apply waivers in a verification run with calibre -waiver, edges and edge clusters must be converted to polygons.

To accomplish this, calibre -waiver does the following:

- Output edges are converted to polygons by expanding the edges by 5 dbu.
- Output edge clusters are converted to regions, such as the REGION keyword of the Enclosure, External, and Internal operations.

These adjustments occur when applying waivers.

Related Topics

[Waiver Shape Interaction Types](#)

Block-Level Waivers (Gray Box)

You can use the waiver criteria file to remove incomplete blocks from your verification run or only include certain blocks. Unfinished blocks can impact performance and introduce large numbers of DRC results. Using the Calibre Auto-Waivers tool to selectively ignore unfinished blocks enables you to focus on real design errors that need to be fixed, without editing the foundry rule file.

Note



You can also use Calibre nmDRC to exclude unfinished blocks, but this methodology can potentially result in errors at interacting regions. The following SVRF statement can be used to remove areas from your DRC run:

```
LAYOUT WINDEL CELL cell_to_exclude HALO dist HIER BY_LAYER marker_layer
```

Try It!



Calibre Gray Box Tutorial and Example Kit (eKit)

This eKit uses example data to demonstrate how to perform block exclusion using different waiver criteria options.

Go to [this page](#) on Support Center to download the eKit (Documentation tab, Document Types=Getting Started Guide). The link goes to the latest release.


- [Overview of Excluding Blocks with Calibre Auto-Waivers](#)
- [Exclude Cells Using Extents and Markers](#)
- [Exclude Regions in Specified Cells](#)
- [Waive Entire Regions in Excluded Blocks](#)

- [Waive Blocks and Preserve Certain Layers](#)
- [Include Cells in the Chip Context](#)

Overview of Excluding Blocks with Calibre Auto-Waivers

The **EXCLUDE_CELL** waiver criteria file option enables you to exclude geometries within a design block from the waiver verification run. When you specify a cell to exclude from the run, all geometries in the region that belong to the cell are removed from any loading or processing steps. This is different from **WAIVE_EXTENT**, which loads the geometry for the waived region.

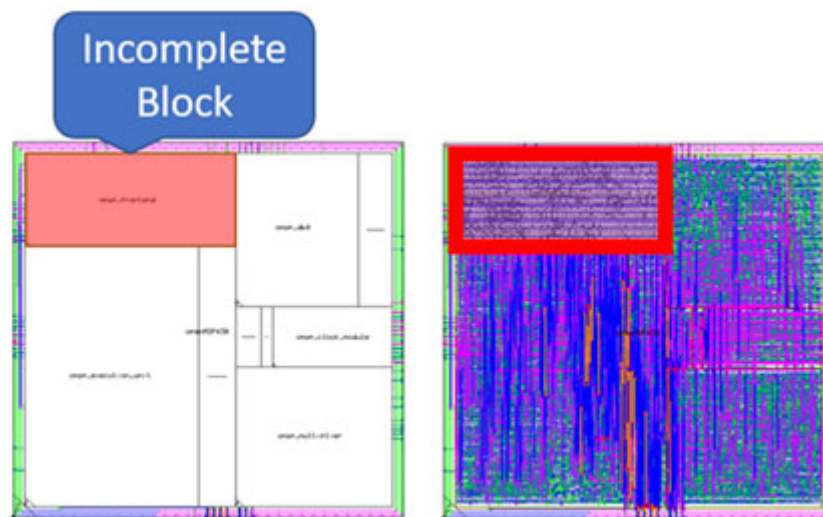
Note

 A single Calibre RealTime Digital license is required for jobs with **EXCLUDE_CELL** in the criteria file, regardless of the number of CPUs.

The **EXCLUDE_CELL** option is specified as follows:

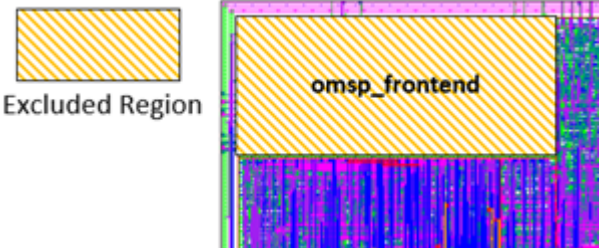
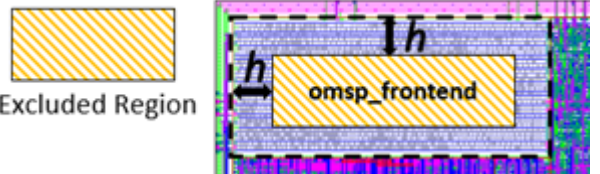
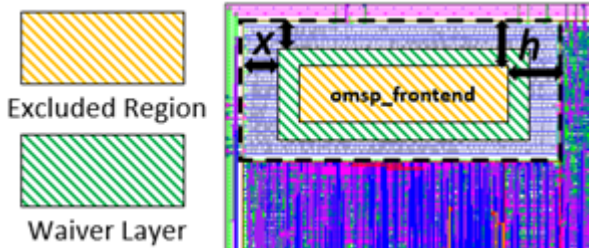
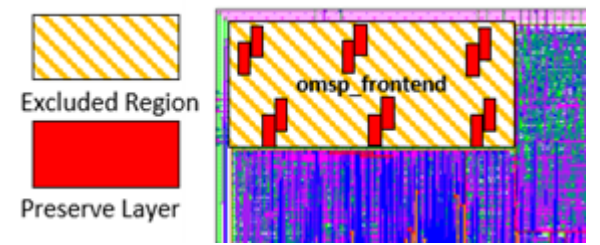
```
EXCLUDE_CELL cell_name [cell_name ...] [HALO value] [BY_LAYER layer_name]
[PRESERVE layer]
```

The **HALO** keyword enables you to specify an outside-facing region of the block to *include* in the run. For example, assume you are performing chip-level verification on the following design:



The top-left block has been placed and connected to the surrounding cells in the context of the top-level, but the contents of the block are still under development. In this scenario, it is useful to verify the connections (interface regions) between the incomplete block and its surrounding cells, but ignore the interior contents of the block that may generate numerous errors. Only geometries from the specified cell are removed.

Table 3-1. Quick Overview of EXCLUDE_CELL Examples

Waiver Criteria Statement	Waived Components
<p>Remove the entire block from processing:</p> <pre>EXCLUDE_CELL omsp_frontend</pre>	
<p>Remove the block from processing, but keep the distance h from the edge of the block.</p> <pre>EXCLUDE_CELL omsp_frontend HALO h</pre>	
<p>Remove the block from processing, but keep the distance h from the edge of the block. Waive all errors for <i>ruleA</i> within the waiver layer, which is x distance from the extents of the cell.</p> <pre>EXCLUDE_CELL omsp_frontend HALO h <i>ruleA</i> WAIVE_EXTENT x where x must be less than h.</pre>	
<p>Remove the entire block from processing, but keep a layer (P1) that is ready to be checked:</p> <pre>EXCLUDE_CELL omsp_frontend PRESERVE P1</pre>	

The BY_LAYER option is useful for non-rectilinear blocks. The tool uses the layer specified in the BY_LAYER argument to calculate the extents of the block.

Exclude Cells Using Extents and Markers

Excluding cell geometries can introduce new errors in the interacting regions near the excluded cells. By combining the EXCLUDE_CELL statement and the WAIVE_EXTENT or WAIVE_MARKER UNDERSIZE keywords, you can remove geometries from a cell and waive specified rules from the interacting region.

The following example excludes the contents of Cell_A, but retains the polygons at the boundary within a width of 0.4. The UNDERSIZE keyword waives any errors within waiver ring of 0.3:

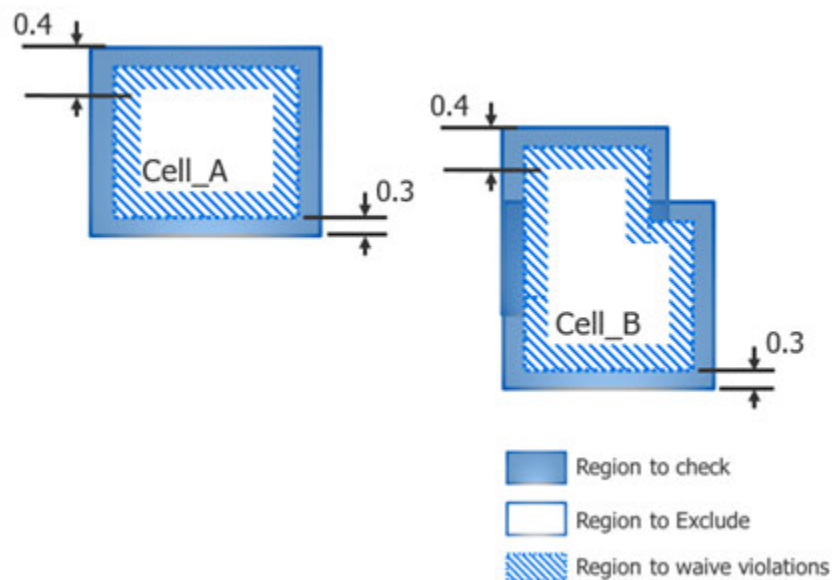
```
EXCLUDE_CELL Cell_A HALO 0.4
* SINGLE 98 98 WAIVE_EXTENT cell_A UNDERSIZE 0.3
```

The following example excludes the contents of non-rectangular Cell_B, and undersizes the marker_layer by 0.3, following the same shape of the cell. As with the previous example, the tool waives results within the undersized region:

```
EXCLUDE_CELL Cell_B halo 0.4 BY_LAYER marker_layer
* SINGLE 98 98 WAIVE_MARKER marker_layer UNDERSIZE 0.3
```

The OVERSIZE keyword for WAIVE_EXTENT operates similarly, but in the reverse direction to UNDERSIZE.

Use EXCLUDE_CELL with WAIVE_EXTENT for rectangular cells and WAIVE_MARKER with rectilinear cells.

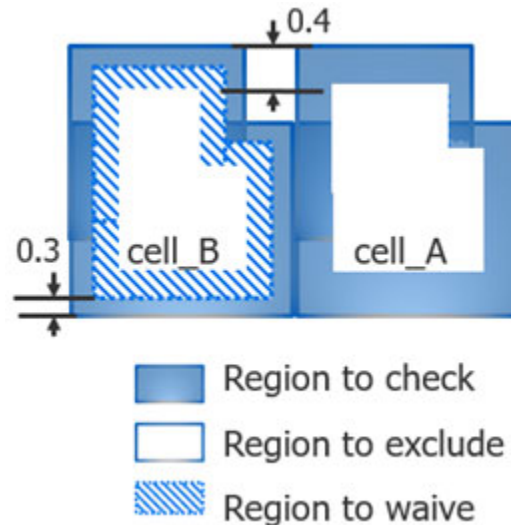


Exclude Regions in Specified Cells

You can further refine the block exclusion by specifying a list of cells to which an exclusion layer applies. The marker layer must exist inside the specified cells. A useful technique is to use the place and route boundary layer as the marker layer, which usually covers the extents of the cell.

In the following example, both cell_A and cell_B contain rectilinear marker layer representing the extents of the cell. The EXCLUDE_CELL statement excludes the contents of cell_B and cell_A, but only uses the rectilinear marker_layer to waive violations from cell_B.

```
EXCLUDE_CELL cell_B cell_A HALO 0.4 BY_LAYER marker_layer  
  
* SINGLE 98 98 WAIVE_MARKER marker_layer UNDERSIZE 0.3 INSIDE cell_B
```



Waive Entire Regions in Excluded Blocks

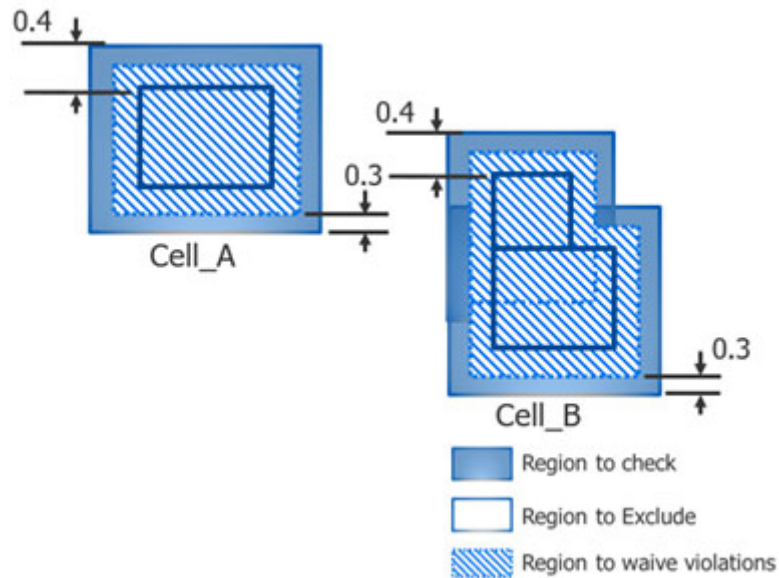
You can use the EXCLUDE_CELL FULL keyword to generate a marker layer over the entire area in which to waive all top-level results. This waives violations due to partial overlapping geometry over the block. If this option is not specified, there can still be connectivity, density, or other violations from omitting all of the geometries in the extent.

Exclude the entire Cell_A (undersized by 0.3) and use the waiver layer to waive the violations around and over the excluded area:

```
EXCLUDE_CELL Cell_A HALO 0.4  
  
* SINGLE 98 98 WAIVE_EXTENT cell_A FULL UNDERSIZE 0.3
```

Exclude the contents of non-rectangular Cell_B, and undersize the marker_layer by 0.3, following the outline of the cell, and waive the violations introduced around the excluded area:

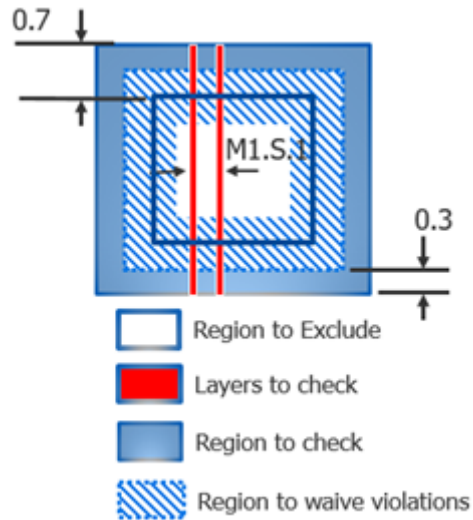
```
EXCLUDE_CELL Cell_B HALO 0.4 BY_LAYER marker_layer
* SINGLE 98 98 waive_marker marker_layer FULL UNDERSIZE 0.3
```



Waive Blocks and Preserve Certain Layers

In some cases, you might want to exclude an unfinished block, but still check layers inside the block that are known to be complete. The PRESERVE keyword includes the specified *layer_name* in the verification run. Multiple layers can be specified.

```
EXCLUDE_CELL cell_A HALO 0.7 PRESERVE M1
* SINGLE 98 98 WAIVE_EXTENT cell_A UNDERSIZE 0.3
```

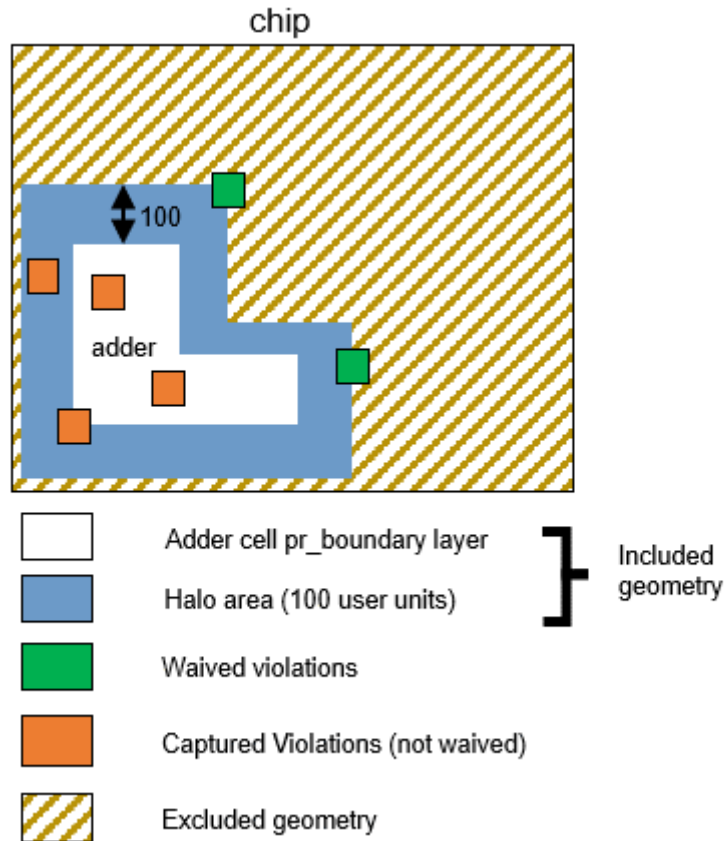


Include Cells in the Chip Context

As an alternative to the EXCLUDE_CELL statement, you can specify to only include certain cells in the layout using the INCLUDE_CELL statement. This enables you to only include specific cells in a DRC run, with an optional halo region around the cell to capture the interface region between other blocks. The rest of the chip is excluded from the verification run. This

capability is useful for block designers that need to validate their blocks in the context of a full chip without expending time checking the rest of the design. For example:

```
INCLUDE_CELL adder HALO 100 BY_LAYER pr_boundary
```



If you do not want to waive violations, you can use the NONE waiver criteria format. See [“INCLUDE_CELL”](#) on page 156 for details.

Chapter 4

Waiver Shape Generation with Calibre RVE

You can generate waivers using Calibre RVE or the waiver_flow tool. Calibre RVE is the preferred method for generating waiver shapes because you can view the results as you waive them. Waiver shapes are the primary mechanism for applying waivers in a verification run.

For error verification waivers topics, view these sections:

- [“Physical Verification with Waivers”](#) on page 180
- [“Writing the Waiver Criteria File for Waiver Verification”](#) on page 183
- [“Performing a DRC Waiver Run Using Multiple Waiver Criteria Files”](#) on page 189
- Also see [“Frequently Asked Questions”](#) on page 48.

Requirements for Waiver Export with Calibre RVE.....	71
Waiver Shape Generation Flow	74
Waiver Export of Error Edge Clusters by Calibre RVE for DRC	95
Waiver Options in Calibre RVE.....	97

Requirements for Waiver Export with Calibre RVE

You must meet several requirements before running Calibre RVE for DRC.

- Calibre Auto-Waivers license and a Calibre RVE license.
- One of the following, depending on the type of waivers:
 - For DRC waivers, the DRC results can be reported in the cell space or at the top level (flat) in each ASCII results database.


To report results by cell (recommended), use the following statement in your DRC rule file:

DRC CELL NAME YES CELL SPACE XFORM

To report results at the top level, use the following statement in your DRC rule file:

DRC RESULTS DATABASE *filename* TOP


Caution

 DRC RESULTS DATABASE ... TOP is not recommended for most applications. Reporting flat results to the top cell space creates much larger results databases, which can increase run time.

- o For ERC waivers, the results must be reported by cell using the following statement in your rule file:

ERC CELL NAME YES CELL SPACE XFORM

Caution

 The DRC RESULTS DATABASE ASCII PSEUDO command should not be specified in your rule file for waiver export with Calibre RVE.

- o Using Calibre Interactive:
 - For DRC runs, click **DRC Options** and enable the “Output cell errors in cell space” checkbox. See [“Reporting DRC Results in Local Cell Space or Top Cell Space”](#) in the *Calibre Interactive User’s Manual*.
 - For ERC runs, click **LVS Options** and enable the “Output ERC errors in cell space” checkbox on the **ERC** tab. See [“Running ERC Checks”](#) in the *Calibre Interactive User’s Manual*.

Results must be reported by cell in order for the waiver to apply to each placement of a child cell in the parent cell. If results are not reported by cell, then the waivers you select only apply to one placement of the waiver in the parent cell. If results are reported flat (DRC RESULTS DATABASE *filename* TOP), then each result is reported using top level coordinates.


- o Your results database open in Calibre RVE, with at least one waived result.

You can waive results by right-clicking a result, or importing a *.waived* file.

If your results are in a binary DFM Database, you can waive results by opening an RDB file using the **Chip Summary** tab’s **View > Browse Hierarchy Errors** menu.

- o Your cell library or chip design is open in an editor that supports Calibre RVE.

Caution


 Ensure that you write out your layout after performing any modifications *prior to* exporting waivers from Calibre RVE. Failure to save layout edits before merging the waiver shapes with the finished design can cause unwaived errors in future Calibre runs.

- o (Optional) A waiver setup file. Calibre RVE can read the layer and datatype settings for waiver shapes and text annotations from the waiver setup file and use them during waiver export. You can specify the waiver setup filename with the environment variable MGC_RVE_WAIVER_SETUP_FILE or specify the filename

in the “Export Waived Results” dialog box. See [“Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE”](#) on page 93.

- (Optional) A waiver criteria file. If waivers are exported with embedded criteria text, the default criteria is SINGLE 100 100. To export waivers with non-default embedded criteria text, you may import a waiver criteria file with different criteria settings.

Tip

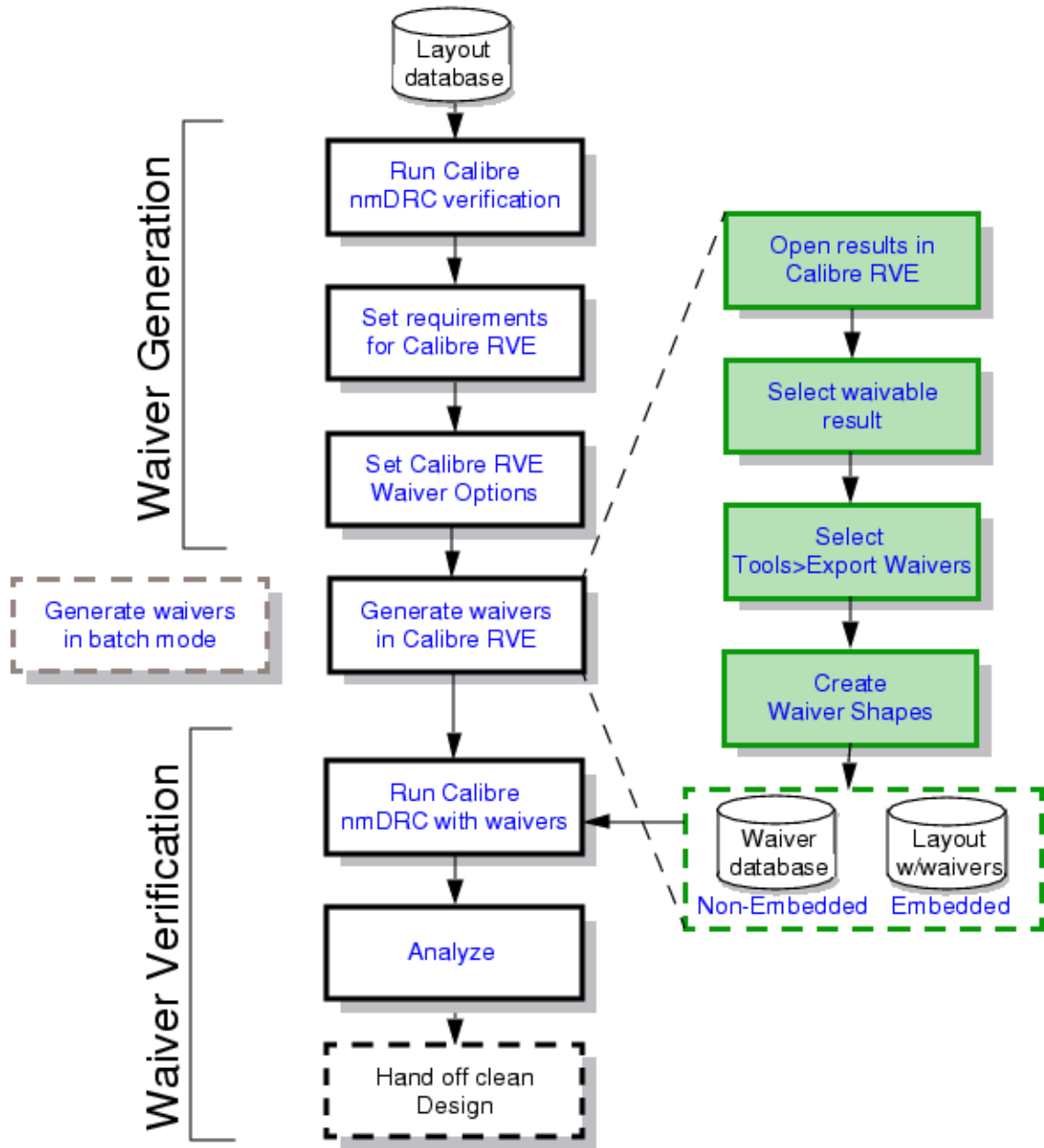
 If your waived results include edge clusters from Enclosure, External, or Internal layer operations, and those edge clusters could form non-Manhattan polygons, then it is recommended to change the corresponding layer operations in your rule file to use the REGION keyword before running DRC with waivers.

Also see [“Waiver Export of Error Edge Clusters by Calibre RVE for DRC”](#) on page 95.

Waiver Shape Generation Flow

There are two primary methods of generating waivers: interactively with Calibre RVE and batch mode using the waiver_flow executable.

Figure 4-1. Calibre RVE Waiver Generation Flow



Creating Embedded Waiver Shapes Using Calibre RVE	75
Creating Non-Embedded Waiver Shapes Using Calibre RVE	81
Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE	86
Creating IP_MATCH Signature Checksums Using Calibre RVE	91

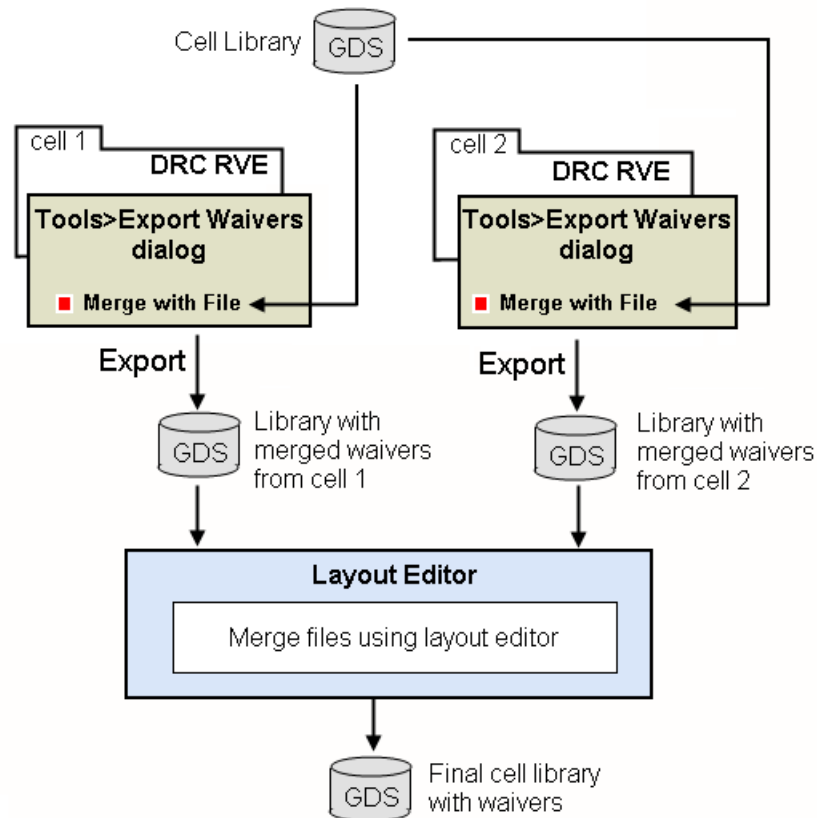
Creating Embedded Waiver Shapes Using Calibre RVE

If you are providing waiver-corrected design cells, such as an IP library, you can embed the waiver shapes in the design. The waivers are then automatically enforced in any design that uses the waiver-corrected cells.

If you do not want to embed waiver shapes in the main design, then see “[Creating Non-Embedded Waiver Shapes Using Calibre RVE](#)” on page 81.

You can embed waiver shapes in a GDS or OASIS format design using Calibre RVE for DRC and create the waiver criteria and waiver verification setup files needed to run Calibre Auto-Waivers using the waiver-corrected IP or cell blocks. An overview of the process is shown in the following figure for an IP library with GDS assumed as the layout format.

Figure 4-2. Creating Waiver-Corrected IP Using Calibre RVE



The Calibre RVE method is often preferable to using the waiver_flow tool because you can see the geometry as you are waiving the result. The waiver_flow tool is used for batch waiver generation.

The `drc_waiv_asc2gds` command enables batch conversion of a Calibre RVE *.waived* file to a geometric waiver database. This is useful if you already have Calibre RVE *.waived* files that you would like to convert to waiver geometry.

Prerequisites

- See “[Requirements for Waiver Export with Calibre RVE](#)” on page 71.

Procedure

1. Open the results database for the first library cell in Calibre RVE using one of the following menus, depending on your layout editor:

- Calibre DESIGNrev — **Verification > Start RVE**
- Other layout editors — **Calibre > Start RVE**

2. Review and highlight the results as needed.

See “[Using Calibre RVE for DRC](#)” in the *Calibre RVE User’s Manual*.

3. Waive the results using one of these methods:

- **Automatic** — If results are already waived in the results database for the cell from a previous Calibre RVE session, then the *database.waived* file is automatically loaded.
- **Import file** — Choose **Tools > Import Waivers** in Calibre RVE.

Use this method if another group provides the *database.waived* file for Calibre RVE.

Note



Calibre RVE automatically saves the file *database.waived* for each results database that has waived results.

- **Manual method** — Right-click each check or result to be waived and choose **Waive > Waive**. Alternatively, you can waive multiple results by using Shift- and Ctrl-click to select all results to be waived, then right-clicking the cluster.

Note



In general, it is best to waive an error at the lowest possible level of the hierarchy.

4. Choose **Tools > Export Waivers**.

Settings from your most recent Calibre RVE session are used to populate the fields in the Export Waived Results dialog box.



Note



The settings used to populate Calibre RVE for a waiver export are loaded from the ~/ .rvedb setup file.

5. Click the **Design** tab to specify the database to which you want to write the waiver shapes:
 - a. Select GDSII or OASIS for the Export format.

For the Virtuoso Export format, see “[Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE](#)” on page 86.
 - b. Specify the name of the final merged design file in the “Output file” text field. This file must already exist.
6. Specify a waiver setup file, merging options, and waiver cell placement as follows:
 - a. Click the **Options** tab.
 - b. (Optional) Read layer and datatype numbers for the waiver shapes and text annotations from a waiver setup file as follows:
 - i. Specify a waiver setup file. If the MGC_RVE_WAIVER_SETUP_FILE environment variable is used to specify the waiver setup file, then type \$MGC_RVE_WAIVER_SETUP_FILE in the text entry field.
 - ii. Click **Import**.


See “[Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE](#)” on page 93 for details on what settings are imported from the setup file.
 - c. Enable the “Merge from file” option to specify a merged waiver database.

This method is typically used when creating waivers for an IP cell library.

- d. Specify the database to merge the waiver shapes with the following options:
 - **Output file** — Adds the new waiver shapes to the file specified for Output file on the Design tab. The output file must already exist; it is overwritten when the waivers are exported.
 - **Other file** — Merges the new waiver shapes with the specified database. The merged database is saved to the new file specified by Output file on the Design tab. Output file is overwritten if it exists.
- e. (Optional) Enable the “Place waiver cells in intermediate cell” option to place waiver cells in an intermediate cell instead of the parent cell. See [“Intermediate Container Cells”](#) on page 219; also see ADD_WAIVER_HIERARCHY in the section [“Waiver Setup File Format for Waiver Verification”](#) on page 126.


This option is automatically enabled if Annotate waiver cells with IP_MATCH checksum text is selected on the **IP_MATCH** tab.

Note

 It is not necessary to specify “Output precision” or “Magnify coordinates by” if “Merge from file” is enabled. If “Merge from file” is enabled and the precision of the DRC results database differs from the precision of the layout database the waivers are being merged into, Calibre RVE prompts you to select an option to resolve the problem.

- f. Enable the remaining options on the **Options** tab as follows:
 - **Text Magnification** — Specify a magnification value for the generated text.
 - **Annotate waiver cells with checksum text** — (Recommended) Write a geometry checksum to new waiver cells.
 - **Annotate waiver cells with rule checksum text** — Write rule checksum text to new waiver cells using the rule file specified in the Rule File field. This ensures that the rule used to generate the result does not change when applying the waivers during a verification run. To see how to apply waivers with rule checksums in a verification run, refer to [“Using Rule Checksums in calibre - waiver”](#) on page 191.

Note

 To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See [“Calibre Environment Variables”](#) on page 25.

- **Annotate waiver cells with Calibre version text** — Annotate waiver cells with a text object indicating the Calibre version.

See [“Text Object Annotations”](#) on page 221.

7. (Optional) Click the **Criteria** tab and enable “Annotate waiver cells with waiver criteria text” if you want to annotate the waiver cells with the waiver criteria.

These criteria can later be extracted automatically with the EXTRACT option for WAIVER_CRITERIA in the waiver setup file. See “[Waiver Setup File Format for Waiver Generation](#)” on page 117 for a description of the setup file commands.

The following additional option is available:

- **Input waiver criteria file(s)** — By default, Calibre RVE annotates waiver cells with a global criteria of SINGLE 100 100. If non-default criteria values are desired, this dialog box may be used to specify the path to a waiver criteria file containing different criteria settings.


The waiver information is added to the waiver cell as text objects; see “[Text Object Annotations](#)” on page 221 for a description of the text annotations.

Note

 Ensure that the layout editor is configured to export text objects based upon layer and texttype.

8. (Optional) Click the **Layers** tab to define non-default layer and datatype settings for your exported waiver shapes.
 - a. Specify the Layer number and Datatype number for the waiver shapes. If you imported a waiver setup file in Step 6b, then these fields are populated from the waiver setup file.


Caution

 The layer and datatype number combination for the waiver shapes should not be used in the layout database the waivers apply to, nor should the combination be used in the rule file for layer mapping or mask results output.

9. Click **Export** to create the waiver geometry file specified in Step 5b.

See the section “[Waiver Cell Description](#)” on page 218 for information about the waiver cells.

Note

 Calibre RVE issues a warning if criteria is not defined for certain waiver shapes. See “[Waiver Criteria File Format](#)” on page 147 for information on writing waiver criteria.

10. (Optional) Choose **Tools > Create Waiver Report** to write a text report of the waivers in the output database.

This can be used when you write the waiver criteria file.

11. If you need to process waivers in another cell, open the results database for the next cell. In Calibre RVE, choose **File > Open Database**.

Proceed with applying waivers and creating a new waiver-embedded output file with this cell, as described in Steps 3 through 10.

12. If you output multiple cells when exporting waiver geometry, merge all output designs into one waiver-corrected IP design using your layout editor.

If you are applying waivers to a library, the objective is to produce a single design that contains all the waiver-corrected design cells in the library.

13. Write the waiver verification setup file if you do not already have one. See “[Writing the Waiver Criteria File for Waiver Verification](#)” on page 183.

If you annotated the waiver cells with waiver criteria text (Step 7), you can use the `EXTRACT pathname` option or the `APPEND pathname` option with the `WAIVER_CRITERIA` parameter. The `EXTRACT` option instructs Calibre Auto-Waivers to use the text annotations to construct the waiver criteria file. The `APPEND` option instructs Calibre Auto-Waivers to use the text annotations to append to an existing waiver criteria file.

14. Write the waiver criteria file if you did not do so as part of the waiver export process (Step 7) and you are not using the `EXTRACT` or `APPEND` options with the `WAIVER_CRITERIA` parameter in the waiver verification setup file. See “[Waiver Criteria File Format](#).”

If you created a waiver report for each cell in Step 10, you can use these reports to help write the waiver criteria file.

15. Ensure your rule file [Precision](#) and your [DRC Results Database Precision](#) match (they do by default). Otherwise, scaling can occur.
16. Hand off the waiver-corrected IP design (from Step 12), waiver criteria file (if a separate one is generated), and waiver verification setup file for verification.

Results

For single edge errors, Calibre RVE expands each edge by 0.005 user units in both directions to create a polygon. If the error output is an edge cluster, then the waiver output is converted to a polygonal region. If a waiver cell would contain only degenerate (zero area) polygons, the waiver cell is not created, and a warning message is displayed.

Note



If your waiver geometry output from Calibre RVE is produced from edge clusters from ENClosure, EXTERNAL, or INTERNAL layer operations, and those edge clusters could form non-Manhattan polygons, then you should change the corresponding layer operations in your rule file to use the `REGION` keyword before running DRC with waivers. Also see “[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)” on page 95.

The final output file embeds the waiver geometries in child cells placed in the applicable cell. See the section [Waiver Cell Description](#) for details about the waiver cell naming conventions, placement, and text annotations.

To validate this procedure, you can run Calibre with the waivers applied as described in [“Physical Verification with Waivers”](#) on page 180.

Related Topics

[Requirements for Waiver Export with Calibre RVE](#)

[Waiver Shape Generation Flow](#)

[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

[Waiver Options in Calibre RVE](#)

Creating Non-Embedded Waiver Shapes Using Calibre RVE

Export waived results to a separate GDSII or OASIS layout database using Calibre RVE. When using this method, waiver shapes are not embedded in the main design’s data. The waiver cells and shapes are kept separate, and are merged with the primary design at runtime by Calibre. The Calibre RVE method is generally preferable to using the `waiver_flow` tool because it is interactive.

If you want to embed waiver shapes in your main design, then see [“Creating Embedded Waiver Shapes Using Calibre RVE”](#) on page 75.

To perform waivers in batch mode, see [“Writing the Waiver Cells File for `waiver_flow`”](#) on page 101.

The `drc_waiv_asc2gds` command enables batch conversion of a Calibre RVE `.waived` file to a geometric format waiver database, which is useful if you already have `.waived` files that you would like to convert to waiver geometry.

Prerequisites

- See [“Requirements for Waiver Export with Calibre RVE”](#) on page 71.

Procedure

1. Open the results database for the layout design in Calibre RVE using one of the following menus, depending on your layout editor:
 - Calibre DESIGNrev — **Verification > Start RVE**
 - Other layout editors — **Calibre > Start RVE**
2. Review and highlight results as needed.


See “[Using Calibre RVE for DRC](#)” in the *Calibre RVE User’s Manual*.

3. Waive the results using one of these methods:

- **Automatic** — If results are already waived in the results database from a previous Calibre RVE session, then the *database.waived* file is automatically loaded.
- **Import file** — Choose **Tools > Import Waivers** in Calibre RVE.

Use this method if another group provides the Calibre RVE *.waived* file.

Note

 Calibre RVE automatically saves the file *database.waived* for each results database that is opened.

- **Manual method** — Right-click each check or result to be waived and choose **Waive > Waive**. Alternatively, you can waive multiple results by using Shift- and Ctrl-click to select all results to be waived, then right-clicking the cluster.

4. Choose **Tools > Export Waivers**.

Settings from your most recent Calibre RVE session are used to populate the fields in the Export Waived Results dialog box.



Note

 Settings used to populate Calibre RVE for a waiver export are loaded from the *~/ .rvedb* setup file.

5. Click the **Design** tab to specify the database to which you want to write the waiver shapes:
- a. Select GDSII or OASIS for the Export format. This specifies the format of the output waiver shape file. For the Virtuoso Export format, see “[Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE](#)” on page 86.
 - b. Specify the name of the exported waiver shape file in the “Output file” text field. If you specify %I, the tool uses the top cell name for the RDB.

6. (Optional) Read layer and datatype numbers for the waiver shapes and text annotations from a waiver setup file:

- a. Click the **Options** tab and specify a waiver setup file. If the environment variable `MGC_RVE_WAIVER_SETUP_FILE` is used to specify the waiver setup file, then type `$MGC_RVE_WAIVER_SETUP_FILE` in the text entry field.

See the section “[Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE](#)” on page 93 for details on what settings are imported from the setup file.

- b. Click **Import**.

7. Disable the “Merge from file” option to create a database with non-embedded waiver cells and shapes.

8. (Optional) Enable “Place waiver cells in intermediate cell” to place waiver cells in an intermediate cell instead of the parent cell.

See the sections “[Intermediate Container Cells](#)” on page 219 and `ADD_WAIVER_HIERARCHY` in the section “[Waiver Cell Description](#)” on page 218.

This option is automatically enabled if “Annotate waiver cells with IP_MATCH checksum text” on the **IP_MATCH** tab is selected.

9. (Optional) Scale exported waiver shapes.

This is useful if the precision of the DRC results database differs from the precision of the layout database, or the DRC results have a different magnification from the layout database. Scaling is selected with one of the following options:

- **Output precision** — Specifies the precision (*output precision*) for the output waiver geometry database. The output waiver shapes are scaled by (*output precision / results database precision*).
- **Magnify coordinates by** — Scale the waiver shape coordinates by the specified value. The precision of the output waiver geometry database is set to the DRC results database precision.


To correct for a precision difference between the DRC Results Database and the input layout database, the magnification factor should be (*results database precision / layout database precision*). For example, if the layout database precision is 1000 and the precision of the results database is 10,000, you should specify 0.1 for “Magnify coordinates by.”

10. Enable other options on the **Options** tab as desired.

- **Text Magnification** — Specify a magnification value for the generated text.
- **Annotate waiver cells with checksum text** — (Recommended) Write a geometry checksum to new waiver cells.

- **Annotate waiver cells with rule checksum text** — Write rule checksum text to new waiver cells using the rule file specified in the Rule File field. This ensures that the rule used to generate the result does not change when applying the waivers during a verification run. To see how to apply waivers with rule checksums in a verification run, refer to [“Using Rule Checksums in calibre -waiver”](#) on page 191.

Note

 To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See [“Calibre Environment Variables”](#) on page 25.

- **Annotate waiver cells with Calibre version text** — Annotate waiver cells with a text object indicating the Calibre version.

See the section [“Text Object Annotations”](#) on page 221.

11. (Optional) Click the **Criteria** tab and enable “Annotate waiver cells with waiver criteria text” if you want to annotate the waiver cells with the waiver criteria.


These criteria can later be extracted automatically with the EXTRACT option for WAIVER_CRITERIA in the waiver setup file. See [“Waiver Setup File Format for Waiver Generation”](#) on page 117 for a description of the setup file commands.

The following additional option is available:

- **Input waiver criteria file(s)** — By default, Calibre RVE annotates waiver cells with a global criteria of SINGLE 100 100. If non-default criteria values are desired, use this dialog box to specify the path to a waiver criteria file containing different criteria settings.

The waiver information is added to the waiver cell as text objects; see [“Text Object Annotations”](#) on page 221 for a description of the text annotations.


Note

 Ensure the layout editor is configured to export text objects based upon layer and texttype.

12. (Optional) Click the **Layers** tab and specify the layer and datatype numbers for the waiver shapes.

This defines non-default layer and datatype settings for your exported waiver shapes. If you imported a waiver setup file in Step 6, then these fields are populated from the waiver setup file.

Caution

 The layer and datatype number combination for the waiver shapes should not be used in the layout database the waivers apply to, nor should the combination be used in the rule file for layer mapping or mask results output.

13. (Optional) To create IP_MATCH signature checksums, click the IP_MATCH tab, enable Annotate waiver cells with IP_MATCH checksum text, and then provide the IP_MATCH input file.

See the section “[Creating IP_MATCH Signature Checksums Using Calibre RVE](#)” on page 91 for details.

14. Click **Export** to create the waiver geometry file specified in Step 5b.

See the section “[Waiver Cell Description](#)” on page 218 for important information on waiver cell naming conventions, placement, and text annotations.

15. (Optional) Choose **Tools > Create Waiver Report** to write an ASCII report of the waivers exported to the output database.

You can use this report to write the waiver criteria file.

16. Write the waiver verification setup file if you have not done so previously. See “[Writing the Waiver Criteria File for Waiver Verification](#).” If you chose a different name than the default in Step 5b, use that name in the WAIVER_DATABASE field of the waiver verification setup file.

If you annotated the waiver cells with waiver criteria text (as described in step 11), you can use the EXTRACT *pathname* option or the APPEND *pathname* option with the WAIVER_CRITERIA parameter. The EXTRACT option instructs Calibre Auto-Waivers to use the text annotations to construct the waiver criteria file. The APPEND option instructs Calibre Auto-Waivers to use the text annotations to append to an existing waiver criteria file.

17. Write the waiver criteria file if you have not done so previously (Step 11) and you are not using the EXTRACT or APPEND options with the WAIVER_CRITERIA parameter in the waiver verification setup file. See “[Waiver Criteria File Format](#).”


If you created a waiver report in Step 15, you can use these reports to help write the waiver criteria file.

18. Ensure your rule file [Precision](#) and your [DRC Results Database Precision](#) match (they do by default). Otherwise, scaling can occur.
19. Hand off the waiver geometry file, waiver criteria file, and waiver verification setup file for DRC verification.

Results

For single edge errors, Calibre RVE expands each edge by 0.005 user units in both directions to create a polygon. If the result output is an edge cluster, the waiver output is converted to a polygonal region. If a waiver cell would contain only degenerate (zero area) polygons, the waiver cell is not created, and a warning message is displayed.

Note

 If your waiver geometry output from Calibre RVE is produced from edge clusters from ENClosure, EXTernal, or INTernal layer operations, and those edge clusters could form non-Manhattan polygons, then it is recommended to change the corresponding layer operations in your rule file to use the REGION keyword before running DRC with waivers. Also see [“Waiver Export of Error Edge Clusters by Calibre RVE for DRC”](#) on page 95.

The output waiver geometry file includes the waiver geometries in waiver cells. See the section [“Waiver Cell Description”](#) for details about the waiver cell naming conventions, placement, and text annotations.

To validate this procedure, you can run Calibre with the waivers applied as described in [“Physical Verification with Waivers”](#) on page 180.

Related Topics

[Requirements for Waiver Export with Calibre RVE](#)

[Waiver Shape Generation Flow](#)

[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

[Waiver Options in Calibre RVE](#)

Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE

Calibre RVE can export waiver shapes to a design database that is open in Cadence Virtuoso.

Prerequisites

- See [“Requirements for Waiver Export with Calibre RVE”](#) on page 71.
- The design is open in Cadence Virtuoso and a socket connection to Calibre RVE is established; the socket connection is established automatically when Calibre RVE is started from Cadence Virtuoso. See [“Setting the Socket Port for Calibre RVE”](#) and [“Cadence Virtuoso”](#) in the *Calibre RVE User’s Manual*.
- The Cadence Virtuoso cell and waiver library must exist. Calibre RVE does not create a new Cadence Virtuoso library when exporting waiver shapes.

Procedure

1. From Cadence Virtuoso, choose **Calibre > Start RVE** to open the results database.
2. Review and highlight results as needed.

See [“Using Calibre RVE for DRC”](#) in the *Calibre RVE User’s Manual*.

3. Waive the results using one of these methods:

- **Automatic** — If results are already waived in the results database from a previous Calibre RVE session, then the *database.waived* file is automatically loaded.
- **Import file** — Choose **Tools > Import Waivers** in Calibre RVE.

Use this method if another group provides the Calibre RVE *.waived* file.

Note



Calibre RVE automatically saves the file *database.waived* for each results database that is opened.

4. Right-click each check or result to be waived, then choose **Waive > Waive**.
Alternatively, you can waive multiple results by using Shift- and Ctrl-click to select all results to be waived, then right-clicking the cluster.
5. Choose **Tools > Export Waivers** in Calibre RVE for DRC.

Settings from your most recent Calibre RVE session are used to populate the fields in the Export Waived Results dialog box.



Note



Settings used to populate Calibre RVE for a waiver export are loaded from the *~/ .rvedb* setup file.

6. Click the **Design** tab to specify the layout database to which you want to write your waivers:
- a. Select Virtuoso for the Export format and specify the design information:
 - **Cell library** — The library containing the top cell of the results database.
 - **View** — The name of the cell view for the top cell of the results database. Exported waiver cells are placed in this cell view.
 - **Waiver library** — Exported waiver cells are first saved to this library, then placed in the design.

- **Use streamin template** — If enabled, the specified streamin template file is used when writing the waiver cells to the waiver library. All settings in the streamin template file are used except for the input file and library name.
- 7. (Optional) To read layer and datatype numbers for the waiver shapes and text annotations from a waiver setup file, click the **Options** tab and specify a waiver setup file.
 - a. If the environment variable MGC_RVE_WAIVER_SETUP_FILE is used to specify the waiver setup file, then type \$MGC_RVE_WAIVER_SETUP_FILE in the text entry field.
 - b. Click **Import**.

See the section “[Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE](#)” on page 93 for details on what settings are imported from the setup file.

- 8. (Optional) Enable “Place waiver cells in intermediate cell” to add waiver cells in an intermediate cell instead of the parent cell. See “[Intermediate Container Cells](#)” on page 219; also see ADD_WAIVER_HIERARCHY under “[Waiver Cell Description](#)” on page 218.

This option is automatically enabled if Annotate waiver cells with IP_MATCH checksum text is selected on the IP_MATCH tab.

- 9. (Optional) Scale exported waiver shapes.

Scaling the shapes is useful if the precision of the DRC results database differs from the precision of the layout database, or the DRC results have a different magnification, and you have not enabled “Merge from file.” Scaling is selected with one of the following options:


- **Output precision** — Specifies the precision (*output precision*) for the output waiver geometry database. The output waiver shapes are scaled by (*output precision / results database precision*).
- **Magnify coordinates by** — Scale the waiver shape coordinates by the specified value. The precision of the output waiver geometry database is the same as the DRC results database precision.

To correct for a precision difference between the results database and the layout database, the magnification factor should be (*results database precision / layout database precision*). For example, if the layout database precision is 1000 and the precision of the results database is 10,000, you should specify 0.1 for “Magnify coordinates by.”

- 10. Enable other options on the **Options** tab as desired.
 - **Text Magnification** — Specify a magnification value for the generated text.

- **Annotate waiver cells with checksum text** — (Recommended) Write a geometry checksum to new waiver cells.
- **Annotate waiver cells with rule checksum text** — Write rule checksum text to new waiver cells using the rule file specified in the Rule File field. This ensures that the rule used to generate the result does not change when applying the waivers during a verification run. To see how to apply waivers with rule checksums in a verification run, refer to [“Using Rule Checksums in calibre -waiver”](#) on page 191.

Note

 To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See [“Calibre Environment Variables”](#) on page 25.

- **Annotate waiver cells with Calibre version text** — Annotate waiver cells with a text object indicating the Calibre version.

See [“Text Object Annotations”](#) on page 221 for a description of the text annotations.


11. (Optional) Click the **Criteria** tab and enable “Annotate waiver cells with waiver criteria text” if you want to annotate the waiver cells with the waiver criteria. These criteria can later be extracted automatically with the EXTRACT option for WAIVER_CRITERIA in the waiver setup file. See [“Waiver Setup File Format for Waiver Generation”](#) on page 117 for a description of the setup file commands.

The following additional option is available:

- **Input waiver criteria file(s)** — By default, Calibre RVE annotates waiver cells with a global criteria of SINGLE 100 100. If non-default criteria values are desired, use this field to specify the path to a waiver criteria file containing different criteria settings.


The waiver information is added to the waiver cell as text objects.

Note

 Ensure the layout editor is configured to export text objects based upon layer and texttype.

12. (Optional) Click the **Layers** tab to define non-default layer and datatype settings for your exported waiver shapes.
 - a. Specify the layer and datatype numbers for the waiver shapes. If you imported a waiver setup file in Step 7, then these fields are populated from the waiver setup file.

Caution

 The layer and datatype number combination for the waiver shapes should not be used in the layout database the waivers apply to, nor should the combination be used in the rule file for layer mapping or mask results output.

13. (Optional) To create IP_MATCH signature checksums, click the **IP_MATCH** tab, enable Annotate waiver cells with IP_MATCH checksum text, and then provide the IP_MATCH input file.

See the section “[Creating IP_MATCH Signature Checksums Using Calibre RVE](#)” on page 91 for details.

14. Click **Export** to create the waiver shapes and place them in the Cadence Virtuoso database.
15. (Optional) Choose **Tools > Create Waiver Report** to create a text report of the waived results.
16. Write the waiver verification setup file if you do not already have one.

See “[Writing the Waiver Criteria File for Waiver Verification](#).” If you chose a different name than the default in Step 5b, use that name in the WAIVER_DATABASE field of the waiver verification setup file.

If you annotated the waiver cells with waiver criteria text (Step 11), you can use the EXTRACT *pathname* option or the APPEND *pathname* option with the WAIVER_CRITERIA parameter. The EXTRACT option instructs Calibre Auto-Waivers to use the text annotations to construct the waiver criteria file. The APPEND option instructs Calibre Auto-Waivers to use the text annotations to append to an existing waiver criteria file.

17. Write the waiver criteria file if you have not done so previously (described in step 11) and you are not using the EXTRACT or APPEND options with the WAIVER_CRITERIA parameter in the waiver verification setup file.

See “[Waiver Criteria File Format](#)”. If you created a waiver report in Step 15, you can use these reports to help write the waiver criteria file.

Results

For single edge errors, Calibre RVE expands each edge by 0.005 user units in both directions to create a polygon. If the result output is an edge cluster, the waiver output is converted to a polygonal region. If a waiver cell would contain only degenerate (zero area) polygons, the waiver cell is not created, and a warning message is displayed.


Note



If your waiver geometry output from Calibre RVE is produced from edge clusters from ENClosure, EXTERNAL, or INTERNAL layer operations, and those edge clusters could form non-Manhattan polygons, then it is recommended to change the corresponding layer operations in your rule file to use the REGION keyword before running DRC with waivers. Also see “[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)” on page 95.

Waiver cells and shapes are written to the Waiver library specified on the Design tab, then placed in the correct library, cell, and view. See the section “[Waiver Cell Description](#)” for details about the waiver cell naming conventions, placement, and text annotations.

Note

 If you are missing waivers for results in cloned cells, ensure that the “Generate waivers in cloned cells” option is enabled in the Export Waived Results dialog box. This option cannot be used with IP_MATCH.

See the section “[Waiver Export to a Cadence Virtuoso Database for Calibre Auto-Waivers](#)” in the *Calibre RVE User’s Manual* for details about the waiver export process, such as handling of duplicate cell names and the use of a streamin template.

Settings in the Export Waived Results dialog box are saved when you exit Calibre RVE.

To validate this procedure, you can run Calibre with the waivers applied as described in “[Physical Verification with Waivers](#)” on page 180.

Related Topics

[Waiver Shape Generation Flow](#)


[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

[Waiver Options in Calibre RVE](#)

Creating IP_MATCH Signature Checksums Using Calibre RVE

You can create IP_MATCH cell signatures using Calibre RVE. IP_MATCH cell signatures allow the waivers in a cell to be applied to cells that have a different name but are otherwise the same.

Caution

 When IP_MATCH checksums are generated, they consider any waiver cells that are present in the specified IP_MATCH cells. The IP_MATCH input file should not contain any embedded waiver cells.

During a calibre -waiver run with RUN_IP_MATCH YES specified in the waiver setup file, any cell in the input design that matches the signature checksum of an IP_MATCH cell has the IP_MATCH cell waivers applied to it. The matching of a cell signature is independent of the cell name.

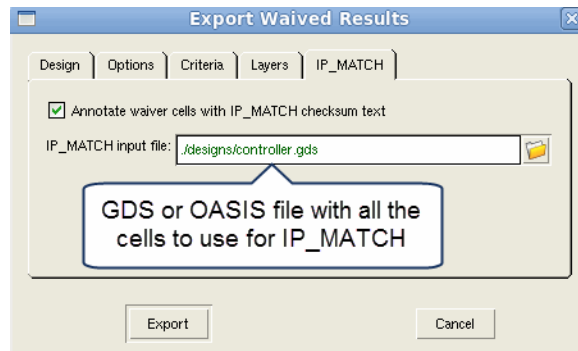
Prerequisites

- See “[Requirements for Waiver Export with Calibre RVE](#)” on page 71.

- An IP_MATCH input file in GDSII or OASIS format containing all cells to use for IP_MATCH operations.
- A results database open in Calibre RVE for DRC with waived results.

Procedure

1. Choose **Tools > Export Waivers** in Calibre RVE to open the Export Waived Results dialog box.
2. Click the **IP_MATCH** tab.




3. Enable the “Annotate waiver cells with IP_MATCH checksum text” option.
4. Specify the IP_MATCH input file.

This is a GDS or OASIS format file, which should contain all the cells to use for IP_MATCH.

If Calibre RVE was launched from Calibre Interactive, the IP_MATCH input file field is set to the value of the MGC_RVE_WAIVER_IP_MATCH_FILE environment variable, which stores the value of the Layout File specified in Calibre Interactive.

Note

 The “Place waiver cells in intermediate cell” checkbox on the Options tab is automatically enabled when “Annotate waiver cells with IP_MATCH checksum text” is enabled. This reduces the database size and the checksum verification runtime.

5. Enable other waiver export settings as instructed in either of these tasks:
 - [“Creating Non-Embedded Waiver Shapes Using Calibre RVE”](#) on page 81.
 - [“Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE”](#) on page 86.
6. Click **Export**.

Results

All waiver cells and shapes are placed in intermediate cells instead of the parent cell, as described in the section “[Intermediate Container Cells](#)” on page 219. See “[Waiver Cell Description](#)” on page 218 for a general discussion of the result of the waiver export process.

The IP_MATCH signature checksum is added to the intermediate container cells that correspond to the cells found in the IP_MATCH input file (Step 4). See the section “[IP_MATCH Checksum Annotations](#)” on page 226 for a description of the text annotation.

If waived results are found in a cell that does not exist in the IP_MATCH input file, Calibre RVE displays a warning but proceeds with the waiver export. The IP_MATCH signature is not generated for cells that do not exist in the IP_MATCH input file.

Related Topics

[Requirements for Waiver Export with Calibre RVE](#)

[Waiver Shape Generation Flow](#)

[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

[Waiver Options in Calibre RVE](#)

[IP_MATCH Checksum Annotations](#)

[Waiver Cells File Format](#)

Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE

Load a waiver setup file when exporting waivers using Calibre RVE. This is useful if you want to use the layer and datatype settings in the waiver setup file for the waiver shapes and text annotations, rather than the default settings.

You can also import other items from the waiver setup file. The complete list of imported settings can be found in [Table 4-1](#) on page 94.

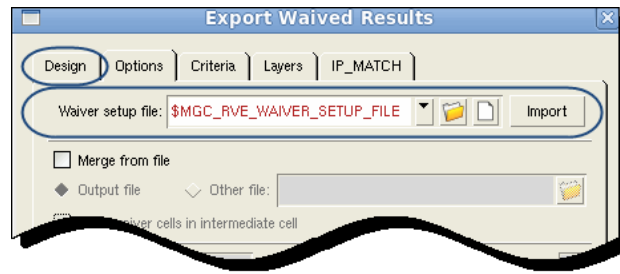
Prerequisites

- See “[Requirements for Waiver Export with Calibre RVE](#)” on page 71.
- A DRC results database open in Calibre RVE, with at least one waived result.
- A waiver setup file. See “[Waiver Setup File](#)” on page 116 for complete details.

Procedure

1. Choose **Tools > Export Waivers**.

The Export Waived Results dialog box opens, as shown.



- Specify a waiver setup file in the “Waiver setup file” field.

If the environment variable `MGC_RVE_WAIVER_SETUP_FILE` is used to specify the waiver setup file, then type `$MGC_RVE_WAIVER_SETUP_FILE` in the text entry field.

If you launched Calibre RVE from Calibre Interactive, the `MGC_RVE_WAIVER_SETUP_FILE` environment variable is set to the path of the waiver setup file used in Calibre Interactive.

- Click **Import** to load the layer and datatype settings from the waiver setup file.

See [Table 4-1](#) for a list of the settings are used by Calibre RVE.

- Specify the remaining settings in the Export Waiver Results dialog box as described in one of the following tasks:
 - “[Creating Embedded Waiver Shapes Using Calibre RVE](#)” on page 75.
 - “[Creating Non-Embedded Waiver Shapes Using Calibre RVE](#)” on page 81.
 - “[Exporting Waiver Shapes to a Cadence Virtuoso Database Using Calibre RVE](#)” on page 86.

Results

When you click **Import** for the waiver setup file, the tool reads the parameters listed in the following table from the waiver setup file.

Table 4-1. Waiver Setup File Parameters Used in Calibre RVE Waiver Export

Layer Setting	Datatype Setting
CALIBRE_VERSION_LAYER	CALIBRE_VERSION_TEXTTYPE
CHECKSUM_TEXT_LAYER	CHECKSUM_TEXTTYPE
COMMENT_TEXT_LAYER	COMMENT_TEXTTYPE
CRITERIA_TEXT_LAYER	CRITERIA_TEXTTYPE
LAYER_NUMBER	DATATYPE_NUMBER
DATE_TEXT_LAYER	DATE_TEXTTYPE

Table 4-1. Waiver Setup File Parameters Used in Calibre RVE Waiver Export

Layer Setting	Datatype Setting
RULE_CHECKSUM_TEXT_LAYER	RULE_CHECKSUM_TEXTTYPE
USERNAME_TEXT_LAYER	USERNAME_TEXTTYPE
	IPMATCH_TEXTTYPE

The following setup file statements are also imported from the waiver setup file:

- IP_MATCH
- RUN_RULE_CHECKSUM
- RULE_FILE
- TEXT_MAG
- TMP_DIR

All other waiver setup file parameters are ignored.

The waiver setup filename, layer, and datatype settings are automatically saved in the `~/.rvedb` file when you exit Calibre RVE and are used the next time you invoke Calibre RVE.

After a waiver setup file is loaded, you may change the settings for Layer number (LAYER_NUMBER) and Datatype number (DATATYPE_NUMBER) at any time by using the Export Waived Results dialog box. However, in order to change any of the remaining parameters in [Table 4-1](#) you must specify the parameters in a waiver setup file and import the waiver setup file. See the section “[Waiver Setup File Format for Waiver Verification](#)” on page 126 for a complete listing of the parameters and their defaults.

Related Topics

[Waiver Shape Generation Flow](#)

[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

[Waiver Options in Calibre RVE](#)

Waiver Export of Error Edge Clusters by Calibre RVE for DRC

When Calibre RVE exports edge results it displays a popup message titled “Warning: Exporting Edges” because the resulting waiver shape may not match the result shapes generated by Calibre.

See “[Waivers for Edge and Edge Cluster Outputs](#)” on page 61 for more information on this message and how Calibre Auto-Waivers handles edge clusters.

Related Topics

[Requirements for Waiver Export with Calibre RVE](#)

[Waiver Shape Generation Flow](#)

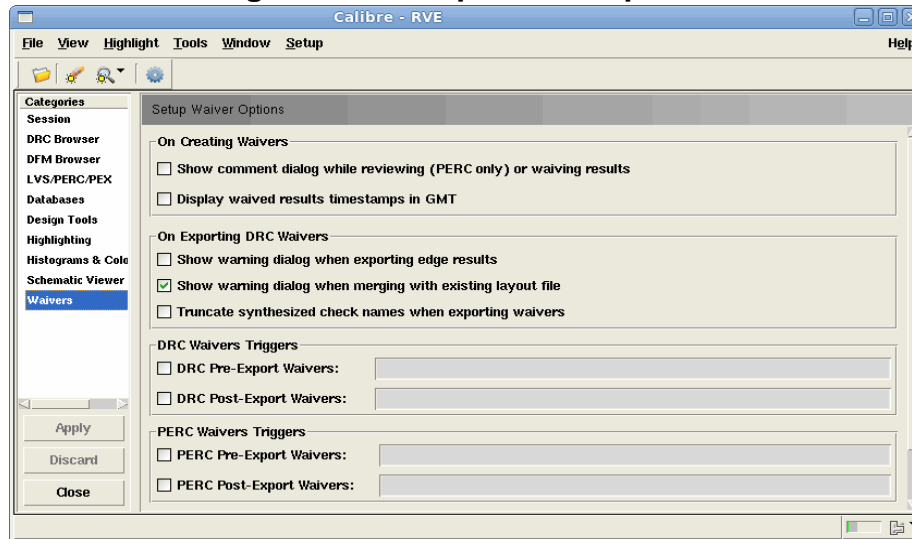
[Waiver Options in Calibre RVE](#)

Waiver Options in Calibre RVE

To access: In Calibre RVE, choose **Setup > Options**.

Sets additional waiver options for Calibre RVE.

Figure 4-3. Setup Waiver Options



Fields

Table 4-2. Setup Waiver Options Contents

Checkbox	Description
Show comment dialog while waiving results	Enables comment dialog in Calibre RVE when waiving results.
Display waived results timestamps in GMT	Sets all timestamps to GMT.
Show warning dialog when exporting edge results	Enables edge to region conversion warnings when exporting waivers. The warning informs you that edge results are converted into regions when waiver shapes are exported.
Show warning dialog when merging with existing layout file	Enables warnings when waivers are exported to an existing layout file.
Truncate synthesized check names when exporting waivers	Truncates checknames at the string "::<". Enable this option if Calibre RVE fails to write checksums correctly due to mismatches in the rule names.
DRC Waivers Pre and Post Triggers	Specifies a trigger function that runs before or after the waiver file is created. The tool calls the specified trigger function from the text field. The tool appends the filename of the waiver file to the list of parameters at execution time.

Usage Notes

Open the Setup Waiver Options panel by choosing **Setup > Options** from Calibre RVE and clicking the **Waivers** option under Categories. This panel enables you to set additional options that affect the display and export of waived items in Calibre RVE.

Related Topics

[Requirements for Waiver Export with Calibre RVE](#)

[Waiver Shape Generation Flow](#)


[Waiver Export of Error Edge Clusters by Calibre RVE for DRC](#)

Chapter 5

Waiver Shape Generation with the waiver_flow Tool

The waiver_flow executable generates waiver shapes in batch mode. The tool is multi-threaded, and for larger designs, that is the preferable run mode.

Caution

 Using Calibre RVE to generate waivers is often preferable to doing batch waivers because the waivers are applied while looking at the DRC results. See [“Creating Embedded Waiver Shapes Using Calibre RVE”](#) or [“Creating Non-Embedded Waiver Shapes Using Calibre RVE.”](#)

If you have legacy IP libraries to which you want to apply waivers, using waiver_flow can be an efficient way to add the waiver geometry.

If you have a Calibre RVE *.waived* file, the [drc_waiv_asc2gds](#) command enables batch conversion of the file to a GDS or OASIS format waiver database.

Note


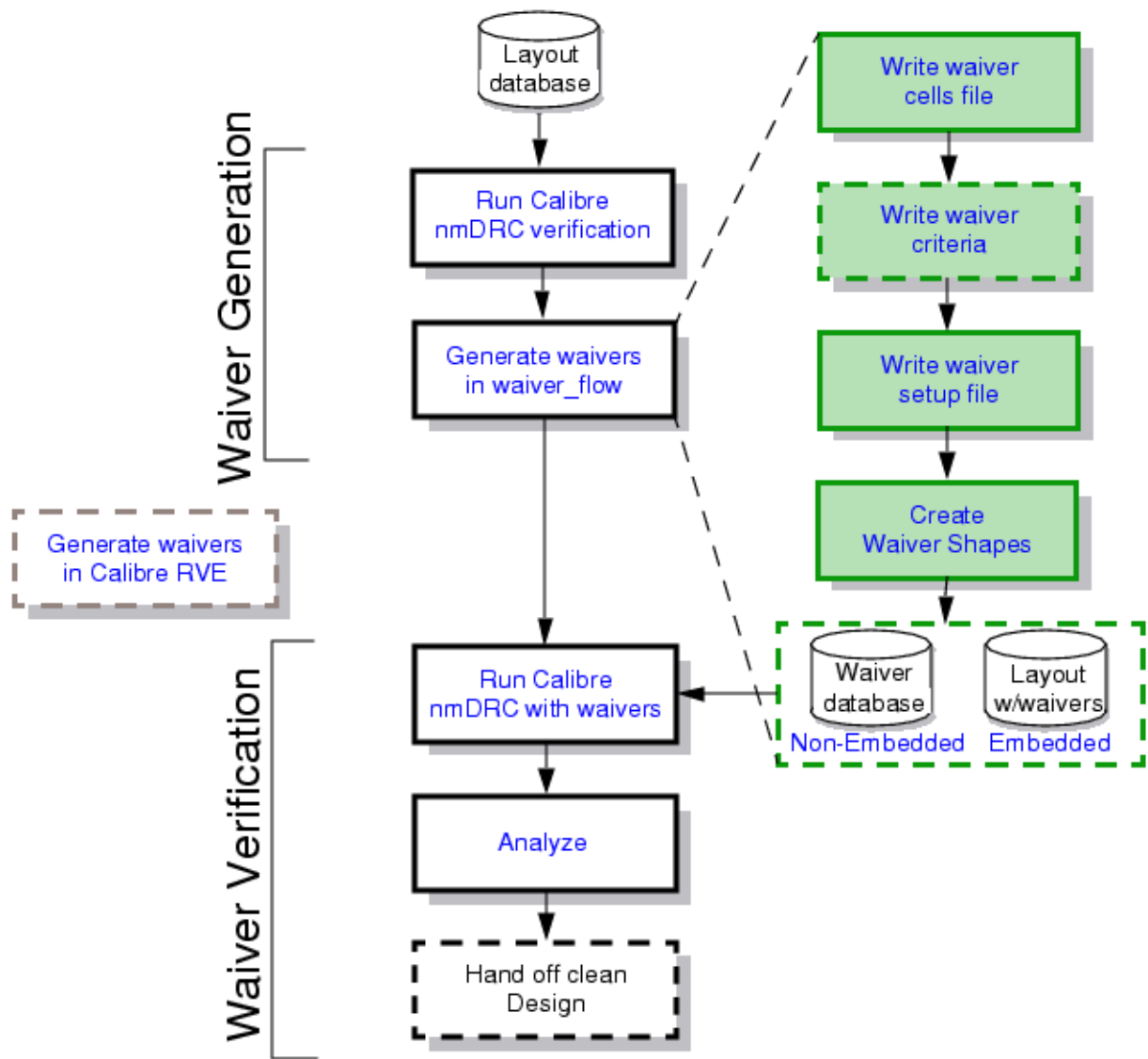
 As of 2019.2, waiver shape edges are always expanded by two DBU. If your waiver database was generated in 2019.1 and earlier, you may need to regenerate your waiver databases using 2019.2 and newer in order to ensure that waivers are correctly applied.

Figure 5-1. Waiver Shape Generation in Batch Mode Using waiver_flow



Writing the Waiver Cells File for waiver_flow 101

Writing the Waiver Criteria File for waiver_flow 101

Writing the Waiver Setup File for Waiver Generation 102

Generating Waivers with the waiver_flow Tool 104

waiver_flow..... 107

Creating Waiver Cell Rule Checksum Annotations with waiver_flow 109

Waiver Cell Precision and Magnification Considerations..... 110

Writing the Waiver Cells File for waiver_flow

In order to generate waivers with the waiver_flow tool, you must have a waiver cells file to define the cells from which waivers are generated. This file must be written manually and must be prepared before a waiver_flow run.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Knowledge of foundry waivers for error results in your design.
- Knowledge of “[Waiver Shape Interactions](#)” and the “[Waiver Cells File Format](#).”

Procedure

1. Examine your error results and the cells in which you want to waive results.
2. Write the waiver cells in a text file.

The waiver_flow tool uses the cells in this file to generate waiver shapes for the specified cells and rules.

Here is an example:

```
anlg      m2_width  m1_spacing
cellA     m2_width  m3_spacing
```

The first line means that the m2_width and m1_spacing rules can be waived for cell anlg. The second line means that the m2_width and m3_spacing rules can be waived for cell cellA.

3. If you are an IP library provider, give this file to your customers.

Related Topics

[Writing the Waiver Setup File for Waiver Generation](#)

[Generating Waivers with the waiver_flow Tool](#)

Writing the Waiver Criteria File for waiver_flow

The waiver criteria file is optional. If you do not create this file, SINGLE 100 100 is written to all waiver cells. Use this file to specify non-default criteria, such as SINGLE 98 98. The criteria text is used when applying waivers in a verification run. The criteria does not affect the waiver generation process.

Note



If you specify WAIVER_CRITERIA NONE in your waiver setup file, no criteria is written to the waiver cells.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Knowledge of foundry waivers for error results in your design.
- Knowledge of “[Waiver Shape Interactions](#)” and the “[Waiver Criteria File Format](#).”

Procedure

1. Determine if any rules require custom criteria for the waiver application process during a verification run. These values are written to the waiver shapes generated by the waiver_flow tool for the specified rule and cell combinations.
2. Write the waiver criteria in a text file. Each rule may only be listed once and only one primary criterion can be attached to a rule.

Here is an example:

```
m2_width          SINGLE 98 98
```

The specified criteria means that only single-waiver interactions are used for waivers applied to the m2_width rule. Both overlap criteria specify at least a 98 percent overlap (this is more relaxed than the default value).

3. (Optional) If you are waiving [Density](#) results windows inside the extent of a given cell, include waiver criteria such as the following:

```
density_rule WAIVE_DENSITY 100 WAIVE_EXTENT cell
```

If you are waiving Density results inside of a marker layer, use waiver criteria such as the following:

```
density_rule WAIVE_DENSITY 100 WAIVE_MARKER marker_layer
```

If you are an IP library provider, you give this file to your customers.

Related Topics

[Writing the Waiver Setup File for Waiver Generation](#)

[Writing the Waiver Cells File for waiver_flow](#)

[Generating Waivers with the waiver_flow Tool](#)

Writing the Waiver Setup File for Waiver Generation

In order to generate waiver geometry with waiver_flow, you must have a waiver generation setup file to define the inputs and outputs.

Waiver setup files can be edited using Calibre Interactive. See “[Starting a Waiver Run in Calibre Interactive](#)” in the *Calibre Interactive User’s Manual* for complete information.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Knowledge of “[Waiver Setup File Format for Waiver Generation](#).”

Procedure


1. Use the following file as a template to create your own setup file for waiver_flow.

```
INPUT_LIBRARY           <path to layout>
LAYOUT_SYSTEM           <GDS, OASIS, LEFDEF, OPENACCESS>
RULE_FILE               <path to foundry_rule_file>
WAIVER_CELLS            <path to waiver cells file>

WAIVER_CRITERIA         <optional path to waiver_criteria_file>
LAYER_NUMBER            <optional number not in INPUT_LIBRARY;
                        default 1234>
DATATYPE_NUMBER         <optional datatype number; default 5678>
```

The LAYER_NUMBER and DATATYPE_NUMBER combination should not be used in the design specified in the [Layout Path](#) statement of the rule file. The combination should also not be used in the rule file for layer mapping or mask results output.

Note

 For advanced nodes, if your rule file precision and input layout precision is different, specify PRECISION_CONVERSION YES in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

2. (Optional) Add this line if you intend to embed the waiver shapes in your INPUT_LIBRARY file:

```
MERGE YES
```

You can optionally specify a WAIVER_DATABASE statement to define the name of the output file.

If you are *not* embedding the waiver shapes in your INPUT_LIBRARY file, then specify this:

```
MERGE NO
WAIVER_DATABASE <external_waiver_file>
```

or do not add statements to the waiver generation setup file. In the latter case, you are opting to accept the default external waiver geometry filename, which is *waived.gds*.

3. (Optional) If you do not want custom waiver criteria values or the default SINGLE 100 100 criteria written to each waiver cell, then include the following statement in your waiver setup file:

```
WAIVER_CRITERIA NONE
```

4. If you are an IP provider, give the waiver generation setup file to your customers.

Related Topics

[Physical Verification with Waivers](#)

[Writing the Waiver Cells File for waiver_flow](#)

[Waiver Setup File Format for Waiver Generation](#)

[Writing the Waiver Criteria File for Waiver Verification](#)

Generating Waivers with the waiver_flow Tool

The waiver_flow executable generates waiver shapes in batch mode.

Note



If you run waiver_flow more than once, the tool overwrites files of the same name from an earlier run. For this reason, it is best to do DRC and ERC waiver generation in different directories.

Restrictions and Limitations

- The waiver_flow utility cannot be used to generate waivers for cloned cells. Use the drc_waiv_asc2gds utility or Calibre RVE.

Prerequisites

- See [“Requirements for Running Calibre Auto-Waivers”](#) on page 23.
- A waiver cells file. See [“Waiver Description Files”](#) on page 143 and [“Writing the Waiver Cells File for waiver_flow”](#) for instructions, if needed.
- A waiver generation setup file. See [“Writing the Waiver Setup File for Waiver Generation”](#) for instructions, if needed.
- The input layout should not have waiver shapes embedded in it that conflict with any design cell and rule check pair specified in the waiver description files. Failure to guarantee this can cause a warning in the waiver_flow run. Additional waivers from the conflicting rule check and cell pair *are not merged into the output design*.
- If using the -turbo option, you need sufficient licenses to run the job.

Procedure

1. Ensure your input files are all available and correct.

2. Open the waiver generation setup file and do the following:
 - Ensure the INPUT_LIBRARY, RULE_FILE, and WAIVER_CELLS paths are properly specified.
 - If you are embedding waiver shapes in your design, ensure the MERGE YES statement is specified in the waiver generation setup file.
3. Run the waiver_flow executable.

```
waiver_flow [-erc] [-turbo [N] [-hyper]] [-nowait | -wait n]  
waiver_setup_file |tee waiver_flow.log
```

Where *waiver_setup_file* is the waiver generation setup file described under “[Waiver Setup File Format for Waiver Generation](#)” on page 117.

The -turbo option causes Calibre to be called internally in MT mode. This option should be used when there are large cells in which waivers are applied.

4. After the run, check the following:
 - a. Open the *waiver.summary* (the name may be different if WAIVER_SUMMARY is specified in the setup file) file in a text reader and check the messages. A listing of messages is found in the section “[Calibre Auto-Waivers Runtime Messages](#)” on page 240.
 - b. Change directories to the *Waiver_results* directory.

This directory contains subdirectories with the names of cells from the INPUT_LIBRARY design in which waivers are applied. The primary files of interest in each subdirectory are as follows:

- A link to your INPUT_LIBRARY design called *chip.gds*
- A DRC summary report that ends in a *.report* suffix
- A DRC results database called *DRC_RES.db*
- A *calibre.log* file. The *calibre.log* file is useful to check if something appears abnormal in the run

If the run used the -erc option, then the files in this directory have the ERC string in them instead of DRC.

- c. If you find the waivers generally are not what you expect for a given rule check, then adjust your waiver cells file. See “[Waiver Cells File Format](#)” on page 173.
 - d. After making adjustments, run waiver_flow again and follow step 4 again.

- e. If you are embedding waiver shapes in your design (MERGE YES is specified in the setup file), then do the following:

- i. Return to your working directory and locate the WAIVER_DATABASE file (*waived.gds* by default).

The file contains your original design data merged with waiver shapes.

- ii. Open this file in your layout viewer and examine the cells in which waivers have been applied.


- iii. You can look for placements of the waiver geometry cells as seen in Step 1. See [“Physical Verification with Waivers”](#) on page 180 for a description of the waiver geometry output.

- f. If you are using non-embedded waiver shapes (MERGE NO), open the waiver geometry file (*waived.gds* by default; the WAIVER_DATABASE file, if specified) in your layout viewer and inspect the contents. You can look for placements of the waiver geometry cells as seen in Step 1.

This file is merged with your chip design in the calibre -waiver run to enforce the waivers.

- 5. If you are providing IP, give the waiver generation setup file and waiver description files to the IP users.
- 6. Perform design verification with waivers. See [“Writing the Waiver Criteria File for Waiver Verification.”](#)

Note

 As of 2019.2, waiver shape edges are always expanded by two DBU. If your waiver database was generated in 2019.1 and earlier, you may need to regenerate your waiver databases using 2019.2 and newer in order to ensure that waivers are correctly applied.

Related Topics

[Physical Verification with Waivers](#)

[Waiver Cells With Instances of Error Cells](#)

[Calibre Auto-Waivers Runtime Messages](#)

[Waiver Setup File Format for Waiver Generation](#)

[waiver_util](#)

waiver_flow

Access from: \$CALIBRE_HOME/bin/waiver_flow.

Use the waiver_flow tool to generate waiver shapes in batch mode.

Usage

waiver_flow [-erc] [-turbo [*number*] [-hyper]] [-nowait | -wait *n*] [-h] *waiver_setup_file*

Arguments

- *waiver_setup_file*
Required pathname to a waiver generation setup file. See “[Writing the Waiver Setup File for Waiver Generation](#)” on page 102 and “[Waiver Setup File Format for Waiver Generation](#).”
- -erc
Option that specifies the run performs waivers on ERC rule checks. By default, the [Pathchk](#) operation is supported for waivers. The Net Area family of layer operations are also supported for this option.
- -turbo [*number*]
Option that specifies to use multi-threading on a host with multiple CPUs. When used without *number*, the command uses the maximum number of CPUs available. When *number* is used, it specifies the number of CPUs to perform the run. The behavior is similar to a normal Calibre nmDRC run.
- -hyper
Option that specifies to run waiver_flow with hyperscaling to enhance the run time and scalability. This option must be specified with -turbo.
- -nowait
Option that causes Calibre to queue only briefly (approximately 10 seconds) before attempting to acquire appropriate licenses. This option is equivalent to specifying “-wait 0”.
- -wait *n*
Option that sets the maximum amount of time for Calibre to queue for appropriate licenses. If the license is unavailable after queueing for *n* minutes, the run aborts. The maximum value for *n* is 45000.
- -h
Option that displays a help message for this command. It may not be used with any other options.

Description

This command is used for batch waiver generation in Calibre Auto-Waivers. If using the -turbo option, you must ensure you have adequate licenses for the job.

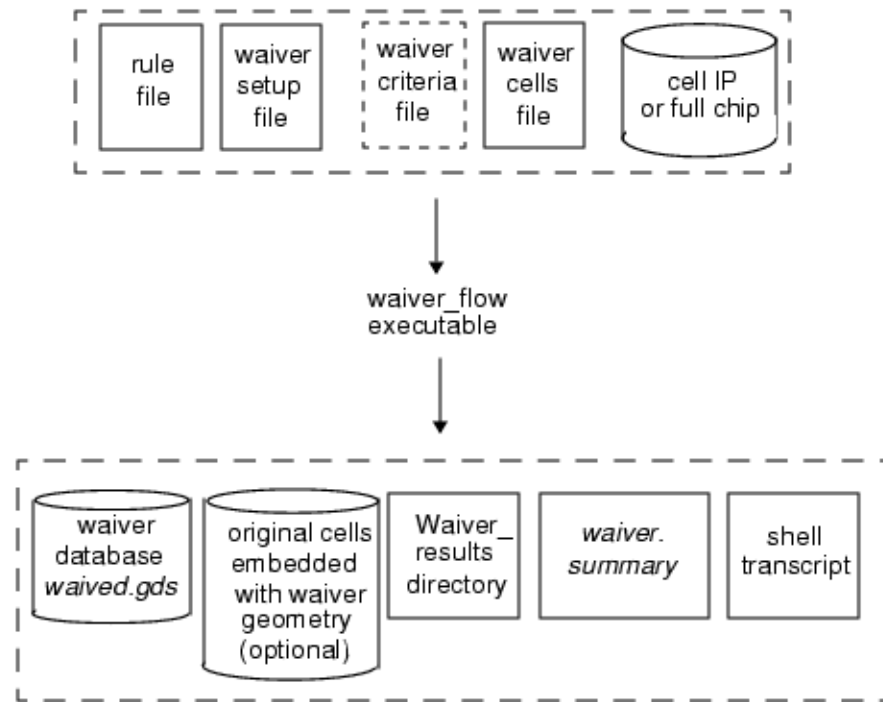
This command produces a *waiver.summary* file that contains runtime messages.

In order for waiver_flow to generate appropriate results, the [DRC Maximum Results](#) ALL or [ERC Maximum Results](#) ALL statement must be in the rule file. If the statement does not appear in the rule file, waiver_flow guarantees the statement is used in the run.

Derived edge results are expanded by 0.005 user units when forming waivers.

If an operation's output is an edge cluster (derived error layer), then the waiver output is converted to a polygonal region.

Figure 5-2. waiver_flow I/O



The *waived.gds* file is produced by default. A different name can be specified using the `WAIVER_DATABASE` statement. In that case, you are not embedding waiver shapes in your main design but are storing them in the `WAIVER_DATABASE`.

The *calibre-waiver.summary* file is the summary report file for the run.

Examples

This example runs the waiver_flow executable in MT mode with hyperscaling and acquires as many Calibre Auto-Waivers licenses and Calibre nmDRC/Calibre nmDRC-H license pairs as there are CPUs.

```
waiver_flow -turbo -hyper waiver.setup
```

This example shows a similar command for ERC waivers:

```
waiver_flow -erc -turbo waiver.setup
```

Related Topics

[Writing the Waiver Setup File for Waiver Generation](#)

[Writing the Waiver Cells File for waiver_flow](#)

Creating Waiver Cell Rule Checksum Annotations with waiver_flow

You can generate waiver shapes with rule checksum annotations using waiver_flow. In a verification run with waivers, any rule checksum in the waiver cell is compared to the rule in the specified rule file. If the checksum does not match the corresponding rule or rule filename, a warning is issued and waivers are not applied for that rule.

Note



To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See “[Calibre Environment Variables](#)” on page 25.

The following commands are used in the waiver setup file to control rule checksum generation:

- RULE_CHECKSUM_TEXT_LAYER
- RULE_CHECKSUM_TEXTTYPE
- RUN_RULE_CHECKSUM

For more details on rule checksums and limitations, refer to “[Rule Checksum Annotations](#)” on page 224 of Appendix A.

Prerequisites

- Layout files and associated rule file with rules that you want to waive
- A waiver criteria file and a waiver cells file for the design
- A supported layout viewer

Procedure

1. Review your waiver cells file and make any modifications as necessary.

2. Write a waiver setup file such as the following:

```
INPUT_LIBRARY design.gds
RULE_FILE rules
WAIVER_CELLS waiver_cells_file
LAYOUT_SYSTEM GDSII
MERGE NO
WAIVER_DATABASE waived.gds

## These checksum statements use the default layer settings ##
RULE_CHECKSUM_TEXT_LAYER 1234
RULE_CHECKSUM_TEXTTYPE 9
RUN_RULE_CHECKSUM YES
```

Save the file as *waiver_setup*.

3. Run the tool as follows:

```
waiver_flow waiver_setup_file
```

Waivers are generated based on the information in the waiver setup and description files. The embedded waiver shapes are saved, by default, to *waived.gds*.

4. Check the *waiver.summary* file to ensure there were no problems in the waiver_flow run.

Results

The waiver_flow tool computes the checksum for all rules associated with the generated waiver shapes. Checksum text objects appear inside the waiver geometry cells on layer 1234 and texttype 9.

In this example, the checksums for rules defined in the *rule* file are embedded in the waiver cell cells defined in the *waiver_cells* file. When performing a DRC run, the waiver cell rule checksums are compared with the DRC rule file.

Related Topics

[Using Rule Checksums in calibre -waiver](#)

[Rule Checksum Annotations](#)

[Waiver Setup File Format for Waiver Generation](#)

[Generating Waivers with the waiver_flow Tool](#)

Waiver Cell Precision and Magnification Considerations

If the rule and layout precisions are different, then the tool's internal view of a result may be different from the waiver shape, depending on how the waivers are saved and stored.

The handling of waiver shape precision and magnification is controlled by the PRECISION_CONVERSION statement in the waiver setup file.

If PRECISION_CONVERSION YES is specified (the default), then the rule file [Precision](#) is used to write the waiver geometry in waiver_flow. The waiver_flow tool does not observe the [DRC Results Database Precision](#) statement in the rule file in this case; it uses the rule file Precision. If the INPUT_LIBRARY precision does not match the rule file precision, then MERGE NO must be specified.

If PRECISION_CONVERSION YES is specified in a calibre -waiver run, Calibre does the following:

- If not already specified in the rule file with the values shown, these specification statements are set automatically and warnings are issued:

```
LAYOUT INPUT EXCEPTION SEVERITY METRIC_INPUT_FILE 0
LAYOUT INPUT EXCEPTION SEVERITY PRECISION_INPUT_FILE 0
LAYOUT INPUT EXCEPTION SEVERITY PRECISION_LAYOUT 0
```

- Both the primary Layout Path design and the WAIVER_DATABASE design are read in and adjusted according to the Precision in the rule file. Magnification is automatically applied to compensate for precision differences between the databases.

The PRECISION_CONVERSION YES setting keeps the waiver shapes at the Calibre internal precision and magnification. The YES setting is recommended for advanced process nodes where the rule file Precision is greater than the database precision.

If using Calibre RVE to generate your waiver shapes, the precision of the exported waiver shapes will match the results database precision. If the waiver shapes are embedded in the main design, then the precision of the exported waiver shapes must match the design in which they are embedded.

Related Topics

[Using Rule Checksums in calibre -waiver](#)

[Rule Checksum Annotations](#)

[Waiver Setup File Format for Waiver Generation](#)

[Generating Waivers with the waiver_flow Tool](#)

Chapter 6

Waiver File Descriptions

The waiver setup, waiver criteria, and waiver cells files control how waivers are generated and applied in batch mode. These files are also used to control Calibre Interactive and options in the Calibre RVE Export Waived Results window. The output files produced by Calibre Auto-Waivers depend on whether you are generating waivers or applying waivers and on the type of verification run.

Calibre Auto-Waivers File Usage	113
Waiver Setup File	116
Waiver Description Files	143

Calibre Auto-Waivers File Usage

The waiver setup file acts as a control file for the various commands, inputs, and outputs used in the waiver generation and verification flows. It can be specified in a batch run or in Calibre Interactive and Calibre RVE.

Input Files

- **Calibre rule file** — Defines rules for DRC, ERC, or DFM. A rule file must be specified to generate and apply waivers.
- **Waiver setup file** — Describes waiver input and output file settings for generating waiver shapes with the waiver_flow tool and applying waiver shapes for calibre -waiver runs. The waiver setup file can be written manually, or automatically generated with Calibre Interactive. A waiver setup file is not required for Calibre RVE, but it can be imported to override existing text layer settings in the Export Waived Results dialog box.
- **Waiver criteria file** — Defines waiver shape interaction criteria for waiveable rules. The statements specify the waiver type and overlap values that apply to each rule check. The criteria file is optional for waiver_flow. It is optional for calibre -waiver if criteria text are embedded in the design. Note, do not name the criteria file *waiver_desc*. This name is reserved for legacy description files.
- **Waiver cells file** — Defines the rules to waive by cell. The cells file can be written manually, or generated with Calibre Interactive. This file is only required for waiver_flow.

Output Files

- *calibre-waiver.summary* — A summary file used to debug your Calibre Auto-Waivers run. The waiver summary file contains a log of the physical verification run and the applied waiver operations. The file also lists any waivers that were not applied in the physical verification run. An intermediate summary file with the suffix “_original” is also written to your working directory. The intermediate file contains the physical verification run information before waivers are applied.
- *final.svrf* — A file generated during a verification run with waivers that contains a list of SVRF command operations used during the verification and waiver application process.
- Layout file with embedded waiver shapes (Optional) — A layout file containing original design data and waivers. If you specified MERGE YES in the waiver setup file for a waiver_flow run, then the generated waiver shapes are merged with your original layout file. The embedded waiver shapes in this database are applied during a calibre -waiver run.
- *waived.gds* — An output layout database file that contains waiver shapes generated by waiver_flow. The non-embedded waiver shapes in this database are applied during a calibre -waiver run. For waiver_flow, this file is only generated if MERGE NO is specified in the waiver setup file. The filename is specified with the WAIVER_DATABASE setup file command. In Calibre RVE, the default filename is **<cell> .waived.gds**.
- *unused_waiver.rdb* — A results database file generated from a calibre -waiver run. The results database contains the waivers that were not applied in the verification run. Use Calibre RVE to view the unused waivers. Any file with this name is deleted prior to a run with waivers.
- *used_waiver.rdb* — A results database file generated from a calibre -waiver run. The results database contains the original waivers that were successfully applied in the verification run. Use Calibre RVE to view the used waivers. If your criteria is SINGLE 100 100, these results are identical to the *waived.rdb* database. If non-default criteria were specified, the applied waivers in this database may be different from the original waiver shapes (such as when multiple waivers were applied to a single result region).
- *density_waived.rdb* — A results database file generated from a calibre -waiver run. Use Calibre RVE to view [Density](#) waivers in this database that were successfully applied in the physical verification run.
- *waived.rdb* — A results database file generated from a calibre -waiver run. The *waived.rdb* database contains check results that were successfully waived in the verification run. Use Calibre RVE to view the applied waivers. Any file with this name is deleted prior to a run with waivers.

Related Topics

[Waiver Setup File](#)


Waiver Description Files

Waiver Setup File

The waiver setup file controls waiver generation for the waiver_flow utility and specifies options in a verification run with calibre -waiver.

A single waiver setup file can be used for both operations, but it is highly recommended to separate the generation and verification waiver setup files. For information on waiver setup file statement usage for calibre -waiver and waiver_flow, see “[Waiver Setup File Statement Usage](#)” on page 233.

Note

 Calibre RVE is the recommended utility for generating waivers. Only use waiver_flow if batch mode is required and you are confident with its usage. For complete details on generating waivers from Calibre RVE, see “[Waiver Shape Generation with Calibre RVE](#)” on page 71. For more details on the waiver_flow tool, see “[Waiver Shape Generation with the waiver_flow Tool](#)” on page 99.

Waiver Setup File Format for Waiver Generation.....	117
Waiver Setup File Format for Waiver Verification	126

Waiver Setup File Format for Waiver Generation

Input for: waiver_flow, Calibre RVE

Specifies input and output parameters for waiver_flow.

Format

A waiver generation setup file must conform to the following restrictions:

- This file is parsed; fatal errors result from disallowed statements or syntax.
- The statement names and keywords are special and may not be used as user-defined argument names.
- The # character at the beginning of a line indicates a comment.
- Lines should not end with the ^M character (these are typically present in Windows-formatted text files).

The order of statements in the file is not important.

Environment variables are allowed in pathnames. For example:

```
RULE_FILE $DIR1/rules
INPUT_LIBRARY $lib
```

In this case, DIR1 and lib are expected to be defined in the environment and specify a directory and a file, respectively.

Paths must not contain the literal \$ character or spaces.

Table 4-1 in “[Importing Layer and Datatype Settings from a Waiver Setup File in Calibre RVE](#)” on page 93 lists the setup file commands that apply to Calibre RVE. The following table lists the supported commands.

Table 6-1. Waiver Generation Setup File Statements

Statement	Description
Required Commands	
INPUT_LIBRARY	Pathname of the input design
RULE_FILE	Pathname of a rule file
WAIVER_CELLS	Location of the waiver cells
Optional Commands	
ADD_WAIVER_HIERARCHY	Option that controls whether intermediate cells are written to contain waiver cells within a given parent cell
CALIBRE_VERSION_LAYER	Layer number for Calibre version text

Table 6-1. Waiver Generation Setup File Statements (cont.)

Statement	Description
CALIBRE_VERSION_TEXTTYPE	Text type number of Calibre version text
CHECKSUM_TEXT_LAYER	Layer to which checksum text objects are saved
CHECKSUM_TEXTTYPE	Text type number of checksum text objects
CRITERIA_TEXT_LAYER	Layer number for criteria text objects
CRITERIA_TEXTTYPE	Text type number of criteria text objects
DATATYPE_NUMBER	Datatype number for generated waiver shapes
GENERATE_CLONE_CELLS_WAIVERS	Option that causes drc_waiv_asc2gds to generate waivers in cloned cells.
IGNORE	Option that causes the layer operation not to have waiver geometry output
IPMATCH_TEXTTYPE	Text type number of IP_MATCH text objects
LAYER_NUMBER	Layer number for generated waiver shapes
LAYOUT_SYSTEM	Layout format of the design
MAGNIFY_TEXT_SPACING	Option that specifies vertical spacing between text in generated waiver cells
MERGE	Option that specifies whether waiver shapes get embedded in the input design
NOT_IGNORE	Option that causes the layer operation to have waiver geometry output
PRECISION_CONVERSION	Option that controls whether the rule file or INPUT_LIBRARY precision is used or writing waiver shapes
RULE_CHECKSUM_TEXT_LAYER	Layer number for rule checksum text objects
RULE_CHECKSUM_TEXTTYPE	Text type number of rule checksum text objects
RUN_RULE_CHECKSUM	Option that controls whether rule checksums are generated for waiver shapes
TEXT_MAG	Option that specifies the magnification level of the text when creating a GDSII waiver database
TMP_DIR	Path to a temporary files directory
WAIVER_CRITERIA	Location of the waiver criteria
WAIVER_DATABASE	Pathname of an output waiver geometry file
WAIVER_SUMMARY	Name of a waiver summary report file
WORKING_DIRECTORY	Path to the working directory

Parameters

- **INPUT_LIBRARY** *pathname*

Required pathname of the input design.

If the specified file has the .gz extension, then the file is decompressed in the /tmp directory and the run proceeds on the decompressed data. The gzip utility must be in your environment for this to occur.

If the LAYOUT_SYSTEM is LEFDEF, or OPENACCESS, the *pathname* must be specified in the same way as the pathname of the [Layout Path](#) statement in the rule file. If a cell/view combination cannot be found in the layout, the cell is skipped with a warning.

- **RULE_FILE** *pathname*

Required pathname of a rule file. Typically this is the tape-out rule file.

- **WAIVER_CELLS** *pathname* [*pathname* ...]

Required statement that defines the location of the waiver cells file. See “[Waiver Cells File Format](#)” on page 173 for a complete discussion of the file. More than one *pathname* may be specified.

- **ADD_WAIVER_HIERARCHY** {NO | YES}

Optional statement that controls whether intermediate cells are written to contain waiver cells within a given parent cell. The default is NO, which means container cells are not written. When YES is specified, then container cells are written as described in “[Intermediate Container Cells](#)” on page 219.

If IP_MATCH is specified in the waiver cells file, intermediate container cells are added regardless of the ADD_WAIVER_HIERARCHY statement setting.

The following table shows the results if MERGE YES is specified and various conditions exist during the run.

Table 6-2. Intermediate Cell Output Conditions with MERGE YES

ADD_WAIVER_HIERARCHY Setting	Intermediate container cell is present in INPUT_LIBRARY	Behavior
NO	NO	Any waiver cells that exist in the INPUT_LIBRARY are copied to the WAIVER_DATABASE.
NO	YES	Any waiver cell that is written appears directly under the parent cell. No intermediate container cells are written. Any generated waiver cell that collides with a preexisting waiver cell in the INPUT_LIBRARY does not overwrite the pre-existing waiver cell.

Table 6-2. Intermediate Cell Output Conditions with MERGE YES (cont.)

ADD_WAIVER_HIERARCHY Setting	Intermediate container cell is present in INPUT_LIBRARY	Behavior
YES	NO	Any waiver cell that is written appears under an intermediate container cell. Any generated waiver cell that collides with a preexisting waiver cell in the INPUT_LIBRARY does not overwrite the pre-existing waiver cell.
YES	YES	Any waiver cell that is written appears under an intermediate container cell. Any generated waiver cell or container cell that collides with a preexisting cell in the INPUT_LIBRARY does not overwrite the pre-existing cell.

- **CALIBRE_VERSION_LAYER** *layer*
Optional statement that defines the layer number of Calibre version text objects generated by the tool.
Default: LAYER_NUMBER specification
- **CALIBRE_VERSION_TEXTTYPE** *texttype*
Optional statement that defines the TEXTTYPE number of Calibre version text objects generated by the tool. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver generation setup file, including defaults.
Default: 7.
- **CHECKSUM_TEXT_LAYER** *layer*
Optional statement that defines the layer number of checksum text objects generated by the tool.
Default: LAYER_NUMBER specification
- **CHECKSUM_TEXTTYPE** *texttype*
Optional statement that defines the TEXTTYPE number of checksum text objects generated by the tool. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver generation setup file, including defaults.
Default: 1
- **CRITERIA_TEXT_LAYER** *layer*
Optional statement that defines the layer number of waiver criteria text objects generated by the tool.
Default: LAYER_NUMBER specification

- **CRITERIA_TEXTTYPE** *texttype*
Optional statement that defines the TEXTTYPE number of waiver criteria text objects generated by the tool. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver generation setup file, including defaults.
Default: 2
- **DATATYPE_NUMBER** *integer*
Optional datatype number to be used for waiver shapes. This number affects the WAIVER_DATABASE. The datatype is used together with LAYER_NUMBER.
Default: 5678
- **GENERATE_CLONE_CELLS_WAIVERS** {NO | YES}
Optional statement that causes drc_waiv_asc2gds to generate waivers in cloned cells. See “[Support For Waivers in Cloned Cells](#)” on page 230 for details. This option is not used by waiver_flow and cannot be used with IP_MATCH validation.
Default: NO
- **IGNORE** *layer_operation*
Optional statement that causes the *layer_operation* not to have waiver geometry output. The *layer_operation* parameter is the name of an SVRF layer operation anywhere in the rule file that has output to the global results database. Only primary keywords of layer operations may be specified. The *layer_operation* parameter is case-insensitive. Abbreviations like EXT may not be used.

Note



If a rule includes an ignored layer operation, then the entire rule is ignored. This includes any layer operations that have output but are not ignored.

This statement also causes warnings for unsupported layer operations that appear in waived rule checks not to be issued. “[SVRF Command Support](#)” on page 27 discusses which layer operations are supported by waiver_flow.

This statement may be specified more than once. Each unique *layer_operation* is ignored. If the same *layer_operation* is specified in both an IGNORE and NOT_IGNORE statement, then the first instance of the *layer_operation* is used, and a warning is issued.

- **IPMATCH_TEXTTYPE** *texttype*
Optional statement that defines the TEXTTYPE number of IP_MATCH text objects generated by the tool. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver generation setup file, including defaults.
Note that using IP_MATCH is incompatible with MERGE YES.
Default: 6

- **LAYER_NUMBER** *integer*

Optional layer number to be used for generated waiver shapes during the run. This affects the WAIVER_DATABASE. This layer number together with the DATATYPE_NUMBER should not be used in the INPUT_LIBRARY file, nor should this layer and datatype combination be written to in the rule file. This layer number is applied to all text object annotations by default. See “[Text Object Annotations](#)” on page 221 for complete details.

Default: 1234

- **LAYOUT_SYSTEM** {GDS | OASIS | LEFDEF | OPENACCESS | OA}

Optional if the Layout System is GDS (the default). Required if the Layout System is any of the other choices. This parameter (including the default) must be consistent with the rule file [Layout System](#) statement. Geometric output file format is OASIS if OASIS is specified; otherwise, output is GDS.

Default: GDS

- **MAGNIFY_TEXT_SPACING** {NO | YES}

Optional statement that specifies whether the vertical spacing between text objects in generated waiver cells is magnified. The spacing is determined automatically based on the text magnification setting. If the [TEXT_MAG](#) statement is not applied, the default text magnification of 0.005 is used. If MAGNIFY_TEXT_SPACING YES is not applied, no vertical spacing is applied between text objects in the waiver cell. This means that multiple text objects, such as the criteria and checksums, can appear in the same location. This only affects the readability of the text.

Default: NO

- **MERGE** {NO | YES}

Optional statement that specifies whether waiver shapes get embedded in the original INPUT_LIBRARY data and the data is merged. The default is NO, which means waiver shapes are not embedded in the INPUT_LIBRARY design.

YES must be specified if you are embedding waiver shapes in your design using waiver_flow. If YES is specified, then the name of the merged output database is taken from the WAIVER_DATABASE statement or its default.

Note



If MERGE YES is specified and there are pre-existing waiver cells in the INPUT_LIBRARY that match waiver cells that could be written for a given parent cell, then a warning is issued and the pre-existing waiver cells are preserved.

The merged output database format is GDS unless LAYOUT_SYSTEM OASIS is specified.

Default: NO

- NOT_IGNORE *layer_operation*

Optional statement that causes the *layer_operation* to have waiver geometry output. The *layer_operation* parameter is the name of an SVRF layer operation that is unsupported by default and can be overridden by the NOT_IGNORE statement. Only primary keywords of layer operations may be specified. The *layer_operation* parameter is case-insensitive.

Currently, the only layer operations supported for this keyword are [Net Area](#), [Net Area Ratio](#), and [Net Area Ratio Accumulate](#).

This statement may be specified more than once. Each unique *layer_operation* generates output. If the same *layer_operation* is specified in both an IGNORE and NOT_IGNORE statement, then the first instance of the *layer_operation* is used and a warning is issued.

“[SVRF Command Support](#)” on page 27 describes which layer operations are supported by waiver_flow.

- PRECISION_CONVERSION {YES | NO}

Optional statement that controls how precision mismatches between the rule file and the layout are handled. The YES option means the rule file [Precision](#) is used when generating the waiver database. For advanced nodes, if your rule file precision and input layout precision is different, specify YES (this is the default behavior). Otherwise, you may generate unintended results.

When NO is specified, the INPUT_LIBRARY precision is used.

See “[Waiver Cell Precision and Magnification Considerations](#)” on page 110 for details.


Default: YES

- RULE_CHECKSUM_TEXT_LAYER *value*

Optional statement that specifies the layer to which rule checksum text objects are saved. The *value* is the layer number used to save a rule checksum text object.

Default: LAYER_NUMBER specification.

Note

 To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See “[Calibre Environment Variables](#)” on page 25.

- RULE_CHECKSUM_TEXTTYPE *value*

Optional statement that specifies the TEXTTYPE for rule checksum text objects. The *value* is the datatype used to save a checksum text object.

Default: 9

- RUN_RULE_CHECKSUM{NO | YES}

Optional statement that controls whether rule checksums are generated for waiver shapes.

Default: NO

The following layer operations are unsupported for checksums. If YES is specified and one of the following is detected, a warning is issued and the rule is skipped.

(Not) Net	TDDRC	TVF
DFM Analyze	Net Area Ratio family	DFM Critical Area
Any MDP operations	DFM Measure	Any RET operations

See “[Rule Checksum Annotations](#)” on page 224 and “[Creating Waiver Cell Rule Checksum Annotations with waiver_flow](#)” on page 109.

- **TEXT_MAG** *value*

Optional statement that specifies the magnification level of the text when creating a GDSII waiver database. For example, if the layout precision is 0.001, then the text object height is 5 dbu with the default TEXT_MAG value of 0.005. If the layout precision is 0.0001, then the text height is 50 dbu. TEXT_MAG is not used for OASIS layouts. To specify magnification of the vertical spacing between text, apply the [MAGNIFY_TEXT_SPACING](#) YES command.

Default: 0.005

- **TMP_DIR** *pathname*

Optional statement that defines the path to a temporary files directory. The *pathname* must be absolute.

Default: \$MGC_TMPDIR if it exists and is readable. If not, \$MGC_HOME/tmp if it exists and is readable. Otherwise, WORKING_DIRECTORY.

- **WAIVER_CRITERIA** {*pathname* [*pathname*...]} | NONE

Optional statement that defines the location of the waiver criteria file. See “[Waiver Criteria File Format](#)” on page 147 for a complete description of the file. More than one *pathname* may be specified. If this statement is not used, a default criteria of SINGLE 100 100 is applied to all cells. If you specify NONE, no criteria is written to the waiver cells.

- **WAIVER_DATABASE** *pathname*

Optional pathname of an output waiver geometry file. This statement may only be specified once for the waiver_flow executable run.

Note



The pathname must not collide with other existing filenames in the run directory or files that will be written to the run directory.

If MERGE NO is specified (the default), then the WAIVER_DATABASE file contains only waiver geometry. If MERGE YES is specified, then the WAIVER_DATABASE file contains waiver shapes embedded in the INPUT_LIBRARY design.

Compression by gzip is not performed if you supply a .gz extension. The file is written without the .gz extension.

The WAIVER_DATABASE format is GDS unless LAYOUT_SYSTEM OASIS is specified.

Default: *waived.gds*

- **WAIVER_SUMMARY** *pathname*

Optional statement that specifies the name of a waiver summary report file. A *calibre-waiver.summary* file is produced by default. If this statement is specified, then the *pathname* defines the summary report file location. Ensure that the pathname does not collide with other existing filenames in the run directory or that will be written to the run directory.

- **WORKING_DIRECTORY** *pathname*

Optional statement that defines the path to the working directory. Relative pathnames in the waiver generation setup file are resolved in relation to this setting. If the *pathname* is relative, then the paths in all generated files are relative to the working directory. Likewise, if the path is absolute, the paths in all generated files are absolute.

Default: Current directory in the shell

Examples

Embed waiver shapes into the INPUT_LIBRARY design using waiver_flow.

```
INPUT_LIBRARY      cell.gds
LAYOUT_SYSTEM     GDS
RULE_FILE          drc.rules
WAIVER_CELLS       waiver_cell_defs.cal
LAYER_NUMBER       1111
DATATYPE_NUMBER    2222
MERGE              YES
WAIVER_DATABASE    cell_waived.gds
```

Write an external waiver shape file using waiver_flow:

```
INPUT_LIBRARY      cell.gds
LAYOUT_SYSTEM     GDS
RULE_FILE          drc.rules
WAIVER_CELLS       waiver_cell_defs.cal
LAYER_NUMBER       1111
DATATYPE_NUMBER    2222
MERGE              NO
WAIVER_DATABASE    waivers.gds
```

Related Topics

[Physical Verification with Waivers](#)

[Writing the Waiver Cells File for waiver_flow](#)

[Waiver Setup File Format for Waiver Verification](#)

Waiver Setup File Format for Waiver Verification

Input for: calibre -waiver

Specifies runtime parameters for a calibre -waiver run.

Format

A waiver verification setup file must conform to the following restrictions:

- This file is parsed; fatal errors result from disallowed statements or syntax.
- The statement names and keywords are special and may not be used as user-defined argument names.
- The # character at the beginning of a line indicates a comment.
- Lines should not end with the ^M character (these are typically present in Windows-formatted text files).

The order of statements in the file is not important.

Environment variables are allowed in pathnames. For example:

```
RULE_FILE $DIR1/rules
INPUT_LIBRARY $lib
```

In this case, DIR1 and lib are expected to be defined in the environment and specify a directory and a file, respectively.

Paths must not contain the literal \$ character or spaces.

The following table lists the supported commands.

Table 6-3. Waiver Verification Setup File Statements

Statement	Description
Required Commands	
WAIVER_CRITERIA	Location of the waiver criteria
Optional Commands	
CALIBRE_VERSION_LAYER	Text layer from which Calibre Version text objects are read
CALIBRE_VERSION_TEXTTYPE	Text type number for Calibre version text objects
CALIBRE_WAIVER_NUMBER	Layers Calibre uses internally for enforcement of waivers during the run
CHECKSUM_TEXT_LAYER	Layer number from which checksum text objects are read

Table 6-3. Waiver Verification Setup File Statements (cont.)

Statement	Description
CHECKSUM_TEXTTYPE	Text type number for checksum text objects
COMMENT_TEXT_LAYER	Text layer from which comment text objects are read
COMMENT_TEXTTYPE	Text type number for comment text objects
CRITERIA_TEXT_LAYER	Text layer from which criteria text objects are read
CRITERIA_TEXTTYPE	Text type number for criteria text objects
DATATYPE_NUMBER	Datatype number where waiver shapes are found in the layout
DATE_TEXT_LAYER	Text layer from which date text objects are read
DATE_TEXTTYPE	Text type number for date text objects
DENSITY_WAIVER_RDB	Option that specifies whether the <i>density_waived.rdb</i> is generated
ENABLE_DYNAMIC_RESULTS_REPORTING YES	Specifies to display results in Calibre RVE as soon as they are available.
ENABLE_WAIVER_CELL_REPORTING	Specifies to write a property named WAIVER_CELL to the waived results database files (<i>waived.rdb</i> , <i>used_waiver.rdb</i> , and <i>unused_waivers.rdb</i>).
GENERATE_COMBINED_RESULTS	Specifies whether to write all results that would normally be in the DRC results database and the <i>used_waiver.rdb</i> and <i>unused_waiver.rdb</i> files to a new <i>combined_waivers.rdb</i> results database. This option only works when you specify ASCII output with the DRC Results Database statement.
IGNORE_DENSITY	Option that specifies whether to apply waivers for rule checks that include density operations
IPMATCH_TEXTTYPE	Text type number for IP_MATCH objects
LAYER_NUMBER	Layer number where waiver shapes are found in the layout
MERGE	Option that specifies whether all of the waiver shapes are embedded in the main design
OPTIMIZE_NONE_WAIVER_STATISTICS	Option that specifies whether to exclude statistics from the summary file for used, unused waivers, and waived results
PRECISION_CONVERSION	Option that controls how precision differences between the layout and the rule file are handled

Table 6-3. Waiver Verification Setup File Statements (cont.)

Statement	Description
RETAIN_RESULT_TYPE	Option that controls how waived shapes in the results database and applied waivers database (<i>waived.rdb</i>) are presented
RULE_CHECKSUM_TEXT_LAYER	Text layer from which rule checksum text objects are read
RULE_CHECKSUM_TEXTTYPE	Text type number for rule checksum text objects
RUN_CALIBRE_VERSION_CHECKSUM	Option that checks Calibre version text objects
RUN_DFM_GLOBAL	Option that controls how DRC and DFM waivers are applied in a -drc -hier and -dfm -hier run
RUN_IP_MATCH	Option that verifies IP_MATCH signature text objects
RUN_LAYOUT_CHECKSUM	Option that controls checksum verification of waiver cells
RUN_RULE_CHECKSUM	Option that controls whether rule checksums in waiver shapes should be compared to the associated rules in the verification rule file
TMP_DIR	Path to a temporary files directory
UNUSED_WAIVER_RDB	Option that specifies generation options for the <i>unused_waiver.rdb</i> results database
USED_WAIVER_RDB	Option that specifies generation options for the <i>used_waiver.rdb</i> results database
USERNAME_TEXT_LAYER	Text layer from which username text objects are read
USERNAME_TEXTTYPE	Text type number for username text objects
WAIVER_DATABASE	Option that specifies that some or all of your waiver shapes are not embedded in the main design. More than one WAIVER_DATABASE statement is allowed. You can also specify a directory of waiver databases.
WAIVER_RDB	Option that specifies generation options for the <i>waived.rdb</i> results database
WORKING_DIRECTORY	Path to the working directory

Parameters

- **WAIVER_CRITERIA** { *pathname* [*pathname* ...] | **EXTRACT** [*pathname*] | **APPEND** [*pathname*] }

Required statement that defines the location of the waiver criteria file. More than one *pathname* may be specified in the case where neither **EXTRACT** nor **APPEND** are specified.

If the waiver geometry is annotated with text objects containing stored waiver criteria (see “[Text Object Annotations](#)” on page 221), then the **EXTRACT** option may be used. If **EXTRACT** *pathname* is specified, then the waiver criteria file is constructed from the text objects and written to the *pathname*.

If the **EXTRACT** keyword alone is used (this is not a recommended practice), then a waiver criteria file is constructed, is written to your working directory, and is deleted after the run. If that location is not accessible, or if no criteria was found in the specified file, then no file is written and no waiving occurs.

The **APPEND** option causes the waiver criteria to be extracted from the layout text objects in a similar way to the **EXTRACT** option. However, in this case, the *pathname* specifies a path to an existing waiver criteria file. The extracted waiver criteria are appended to the specified waiver criteria file, and the concatenated file is used internally. The file is written to your working directory. If the **APPEND** keyword alone is used, the behavior is exactly the same as when the **EXTRACT** keyword alone is used.

If the specified waiver criteria file exists and is read-only, the waiver criteria file is written to your working directory and then deleted after the run.

If more than one *pathname* is specified with **EXTRACT** or **APPEND**, then only the first *pathname* is read and the remainder are ignored.

- **CALIBRE_VERSION_LAYER** *layer*

Optional statement that defines the layer number from which Calibre version text objects are read.

Default: LAYER_NUMBER specification

- **CALIBRE_VERSION_TEXTTYPE** *texttype*

Optional statement that defines the TEXTTYPE number from which Calibre version text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 7.

- **CALIBRE_WAIVER_NUMBER** {*integer* | AUTO}

Optional statement that specifies which layers Calibre uses internally for enforcement of waivers during the run. Specifying an *integer* causes Calibre to use that layer number for waiver shape assignments. Additionally, the four layer numbers immediately after the specified *integer* are used for internal processing. Default is 2000 if this statement is not

specified (which means 2001 through 2004 are also used). If the tool encounters a conflict, it chooses the layers to use and issues a warning.

The AUTO keyword specifies that Calibre should choose unused layers to use internally. Calibre uses the design specified in the [Layout Path](#) and [Layer Map](#) statements in the rule file to select the unused layers. This keyword is recommended.

Default: 2000

- **CHECKSUM_TEXT_LAYER** *layer*

Optional statement that defines the layer number from which checksum text objects are read.

Default: LAYER_NUMBER specification

- **CHECKSUM_TEXTTYPE** *texttype*

Optional statement that defines the TEXTTYPE number from which checksum text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 1

- **COMMENT_TEXT_LAYER** *layer*

Optional statement that defines the layer number from which user comment text objects are read.

Default: LAYER_NUMBER specification

- **COMMENT_TEXTTYPE** *texttype*

Optional statement that defines the TEXTTYPE number from which comment text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 5.

- **CRITERIA_TEXT_LAYER** *layer*

Optional statement that defines the layer number from which waiver criteria text objects are read.

Default: LAYER_NUMBER specification

- **CRITERIA_TEXTTYPE** *texttype*

Optional statement that defines the TEXTTYPE number from which waiver criteria text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 2

- **DATATYPE_NUMBER** *integer*

Optional datatype number, together with the LAYER_NUMBER, on which waiver shapes are found in the layout.

Default: 5678

- `DATE_TEXT_LAYER` *layer*

Optional statement that defines the layer number from which waiver date text objects are read.

Default: LAYER_NUMBER specification

- `DATE_TEXTTYPE` *texttype*

Optional statement that defines the TEXTTYPE number from which waiver date text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 4

- `DENSITY_WAIVER_RDB` {YES | NO}

Optional statement that specifies whether to generate the *density_waived.rdb* results database during a verification run with waivers.

Default: YES

- `ENABLE_DYNAMIC_RESULTS_REPORTING` {NO | YES}

Optional statement that specifies whether Calibre RVE immediately displays results as soon as they are available. To enable this feature, do one of the following:

- Specify `ENABLE_DYNAMIC_RESULTS_REPORTING YES` in the waiver setup file.
- Enable the “Enable dynamic results reporting” checkbox on the **Waivers** tab, **DRC Run** category tab in Calibre Interactive

Note



This feature does not work if the DRC Results Database Type is not ASCII, or if any rule check is directed to a non-ASCII output (through the DRC Check Map statement).

Default: NO

- `ENABLE_WAIVER_CELL_REPORTING` {NO | YES}

Specifies to write a property named `WAIVER_CELL` to the waived results database files (*waived.rdb*, *used_waiver.rdb*, and *unused_waivers.rdb*). The value of the property is the name of the cell from which the waiver shape was generated. This can help to determine where the waiver was originally generated, even if the waived DRC results were promoted to a higher level.

The `WAIVER_CELL` property appears in the right pane of Calibre RVE. You can also enable this option from Calibre Interactive. The **Inputs > Waivers** tab includes an “Enable waiver cell reporting” checkbox under the **Waiver RDB Files** pane.

Note

 If all of the RDB Control options in Calibre Interactive are set to NONE/GEOM/PROP, then the “Enable waiver cell reporting” checkbox will not be visible.

Default: NO

- **GENERATE_COMBINED_RESULTS** { NO | YES | DEBUG }

If you specify YES and you specify ASCII output with the DRC Results Database statement, the following occurs:

- The tool writes all results that would normally be in the *used_waiver.rdb* and *unused_waiver.rdb* files to a new *combined_waivers.rdb* results database. The tool generates the *combined_waivers.rdb* file in addition to the DRC results database.
- The properties are retained for all waivers, as set by the WAIVED_RDB waiver setup file statement.
- The tool writes properties as follows:
 - All results receive a new WAIVED property with a value of YES or NO. This indicates whether the error result was successfully waived. This does not apply to unwaived results.
 - All waivers receive a new USED property with a value of YES or NO. This indicates whether the waiver was applied (used waiver) or not (unused waiver) in the verification run.

If you specify DEBUG, the option behaves the same as YES, but also writes additional properties to the unwaived results in the DRC results database. These properties help determine why waiver shapes were not applied and are listed as follows:

- **result_cvg** or **max_result_cvg** — Results coverage properties for single or multi interaction criteria, respectively.
- **waiver_cvg** or **max_waiver_cvg** — Waiver coverage properties for single or multi interaction criteria, respectively.
- **waiver_count** — Total number of applied waivers.
- **WAIVER_COMMENT**
- **WAIVER_USERNAME**
- **WAIVER_DATE**

Note

 The additional properties are controlled by the WAIVER_RDB statement.

The value of the coverage properties is an integer number ranging from 0 to 100, where 0 means that there was no coverage to consider.

Figure 6-1. Without Debug Properties

Check / Cell	Results	Flat	Vertices
Check WAIVER_RDBS	1035	1035	1 388 4
✓ Check VIA4_SPACE	0	0	
✓ Check VIA4_SIZE	0	0	
✓ Check VIA3_SPACE	0	0	
✓ Check VIA3_SIZE	0	0	
✓ Check VIA2_SPACE	0	0	
✓ Check VIA2_SIZE	0	0	
✓ Check VIA1_SPACE	0	0	
✓ Check VIA1_SIZE	0	0	
✗ Check POLY_WIDTH	1	388	
✗ Check POLY_SPACE_TO_PDIFF	2	18	

Figure 6-2. With Debug Properties

Flat	Vertices	max_result_cvg	max_waiver_cvg	waived	WAIVER_COMMENT	waiver_count	WAIVER_DATE	WAIVER_USERNAME
1	4	0	0			0		
2	4	100	100	YES	can be ignored	1	Apr 15 06:13...	user

Default: NO

- IGNORE_DENSITY {NO | YES}

Optional statement that specifies whether to apply waivers for rule checks that include density operations. Specifying IGNORE_DENSITY YES in the verification setup file instructs Calibre Auto-Waivers to ignore all specified criteria for density rules and to not apply waivers for those rules.

Default: NO

- IPMATCH_TEXTTYPE *texttype*

Optional statement that defines the TEXTTYPE number from which IP_MATCH text objects are read. By default, the layer number is controlled by the LAYER_NUMBER argument. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 6

- LAYER_NUMBER *integer*

Optional layer number, together with the DATATYPE_NUMBER, on which waiver shapes are found in the layout. This layer number is applied to all text object annotations by default.

Default: 1234

- **MERGE {NO | YES | IGNORE_DESIGN_WAIVERS}**

Optional statement that determines whether all of the waiver shapes are embedded in the main design. If MERGE YES is specified, then WAIVER_DATABASE has no effect. If MERGE NO is specified, then the WAIVER_DATABASE files (or the default) are assumed to contain at least some of the waiver shapes. MERGE NO should be specified if any of your waivers are contained in separate files from your main design.

When you include the “MERGE IGNORE_DESIGN_WAIVERS” statement in the waiver setup file, then the only waivers that are applied during a physical verification are the waivers that are contained in the external waiver database specified in waiver setup file. Any embedded waivers are ignored. In other words, if there are waiver shapes that were merged with the original design, they are ignored and not applied during a verification run.

The IGNORE_DESIGN_WAIVERS option is not supported for waiver generation. If you include this option in a waiver setup file for waiver generation, then the tool issues a warning message and uses the MERGE NO option instead. This option is not supported using the old engine.

Default: NO

- **OPTIMIZE_NONE_WAIVER_STATISTICS {NO | YES}**

Optional statement that specifies whether to calculate statistics for used, unused waivers, and waived results and write them to the summary file. This option only affects the statistics in the summary file for waiver results database files (waived, used, and unused) that you specified not to generate. If you specify YES in this option and NONE in the WAIVER_RDB statements as follows:

```
OPTIMIZE_NONE_WAIVER_STATISTICS YES
UNUSED_WAIVER_RDB NONE
USED_WAIVER_RDB NONE
WAIVER_RDB NONE
```

The summary file includes the text “N/A” instead of the calculated statistics for used, unused, or waived metrics. For example:

```
RULECHECK RULE1 ... TOTAL Result Count = 0 (0)
    Number of waived Results           = N/A
    Number of unused waived Results    = N/A
    Number of used waived Results      = N/A
    TOTAL DFM RDB Result Count         = 1 (1)
```

Default: NO

- **PRECISION_CONVERSION {YES | NO}**

Optional statement that controls how precision differences between the layout and the rule file are handled. This statement has no effect if the precisions do not differ.

When the rule file [Precision](#) and the layout database precision differ, this option specifies how calibre -waiver adjusts the magnification of the waiver geometry. Specify YES (the default) when the ratio of the rule file Precision to the layout precision is less than 1 or non-integer.

The NO option causes vertices of candidate results to be moved to the appropriate grid by an internal application of the [Snap](#) operation. Vertices are snapped using the ratio of the rule file precision over the layout precision. If that ratio is less than 1, then a warning is issued, and you must use the YES option to process the waiver shapes.

For advanced process nodes, use the YES setting (the default), otherwise you may get unintended results. Waiver shapes should be generated using the same setting for such processes.

Default: YES

See “[Waiver Cell Precision and Magnification Considerations](#)” on page 110.

- **RETAIN_RESULT_TYPE {[NO](#) | YES}**

Optional statement that controls how waived shapes in the results database and applied waivers database (*waived.rdb*) are presented. By default, waived edge outputs are expanded into polygons and waived error clusters are converted into polygons. When YES is specified, waived edge and error results are not converted, but retain their usual forms in the results and *waived.rdb* databases. Unused waivers (*unused_waiver.rdb*) are not affected by this statement.

Default: NO

- **RULE_CHECKSUM_TEXT_LAYER *value***

Optional statement that specifies the text layer from which rule checksum text objects are read. The *value* is the layer number used to read a rule checksum text object.

Default: LAYER_NUMBER specification

- **RULE_CHECKSUM_TEXTTYPE *value***

Optional statement that specifies the TEXTTYPE number for rule checksum text objects. The *value* is the TEXTTYPE used to read a checksum text object.

Default: 9

- **RUN_CALIBRE_VERSION_CHECKSUM {[NO](#) | YES}**

Optional statement that specifies to check Calibre version text objects. When set to YES, verification of the Calibre version is performed. If the text object version string corresponds to the 20yy.q portion of the Calibre version, or newer, than the version used to write the object, then the waiver cell is quietly verified. For example, these Calibre versions are considered equivalent:

```
2011.4_15.12
2011.4_18.12
```

But these are not equivalent:

```
2011.4_15.12
2011.3_18.12
```


If the software version is older than the text object value, then a warning is issued, but the waivers in the parent cell are still applied. The same is true if the version text object is absent in a waiver cell, or if there are multiple version text objects in a waiver cell.

See “[Geometry Checksum Annotations](#)” on page 224.

Default: NO

- **RUN_DFM_GLOBAL {YES | NO}**

Optional statement that controls how DRC and DFM waivers are applied in a -drc -hier and -dfm -hier run. If a rule is specified like this:

```
rule {  
  L = DFM PROPERTY ...  
  DFM ANALYZE L ...  
  DFM RDB L dfm RDB  
}
```

then specifying YES or NO applies waivers in the same way. However, if the original rule file has something like this:

```
L = DFM PROPERTY ...  
rule { DFM RDB L dfm.rdb }  
rule1 { DFM ANALYZE L ...}
```

then the tool’s internal interpretation is the following when YES is specified:

```
L = // all unwaived results  
rule { DFM RDB L dfm.rdb }  
rule1 { DFM ANALYZE L ...}
```

Either rule or rule1 must have had waiver shapes generated for them.

When NO is specified, then the tool’s internal interpretation is this:

```
L = DFM PROPERTY ...  
rule { // local layer L has unwaived results  
  DFM RDB L dfm.rdb  
}  
rule1 { // local layer L has unwaived results  
  DFM ANALYZE L ...  
}
```

Both rule and rule1 must have had waiver shapes generated for them.

Default: YES

- **RUN_IP_MATCH {YES | YES_RESTRICTED | NO}**

Optional statement that specifies to verify IP_MATCH signature text objects (checksums) in waiver cells when applying waivers. The three options behave as follows:

- **YES** — Waivers are applied to all cells that match IP_MATCH signatures in the waiver cells that meet the specified criteria. Additionally, waiver cells that do not contain IP_MATCH checksum annotations can also be applied if the waiver cell name matches the parent cell name in which the error result is found.

Waivers are applied to all cells that match IP_MATCH signatures in the waiver cell that meet the specified criteria for the given results. Waiver cells that do not contain IP_MATCH checksum annotations are applied if the waiver cell name matches the parent cell name in which the error result is found. The tool does not consider waivers without IP_MATCH checksum text that have different parent cell names, even if the cell geometry is the same. If the parent cell name of the waiver database cell with IP_MATCH text does not exist in the layout design, the tool outputs the following warning message:

```
Warning: IP_MATCH of waiver cell <parent_cell_name> did not  
match with any cell in the input layout design. Excluding  
waivers for <parent_cell_name>
```

For waiver databases that do not have any cells with IP_MATCH checksums, the tool does not perform any IP_MATCH checksum matching. Waiver cells with the same name as the parent cell in which the result is found are considered for application during a verification run if the other checksum types match. The tool does not generate a warning message indicating that no IP_MATCH checksum matching is performed.

- YES_RESTRICTED — Waivers are only applied to all cells that match IP_MATCH checksums in the waiver cell that meet the specified criteria. Waiver cells that do not contain IP_MATCH checksums are not considered for application during the verification run.

Waivers are applied to all cells that match IP_MATCH signatures in the waiver cell that meet the specified criteria for the given results. Waiver cells that do not contain IP_MATCH checksum annotations are ignored and not considered for application in a verification run.

If the parent cell name of the waiver database cell with IP_MATCH text does not exist in the layout design, the tool outputs this message:

```
Warning: IP_MATCH of waiver cell <parent_cell_name> did not  
match with any cell in the input layout design. Excluding  
waivers for <parent_cell_name>
```

If the waiver database cell does not contain IP_MATCH text, the tool outputs this message:

```
Warning: No IP_MATCH for waiver cell <parent cell name>,  
Excluding waivers for <parent cell name>
```

For waiver databases that do not have any cells with IP_MATCH checksums, the tool does not consider any waivers for application. In this case, the tool issues a warning message for all waivers in the waiver database.

- NO — No checksum matching is performed. Waiver cells can only be applied if the waiver cell name matches the parent cell name in which the error result is found.

Specifying NO causes the signatures not to be verified.

Default: YES

- **RUN_LAYOUT_CHECKSUM** {YES | NO}

Optional statement that controls checksum verification of waiver cells. When set to YES (the default), verification of the contents of waiver cells is performed to see if they match the checksum. If there is a checksum mismatch, a warning is issued and the parent cell is processed. See “[Geometry Checksum Annotations](#)” on page 224. Checksum verification is performed by default. Checksum is conducted as follows:

- If MERGE YES is specified, then check all Layout Path designs in the rule file.
- If MERGE NO is specified and WAIVER_DATABASE is specified, then check all WAIVER_DATABASE designs that are present in addition to all Layout Path designs.
- If MERGE NO is specified and WAIVER_DATABASE designs are not found, then check *waived.gds* or *waived.oas* in addition to the Layout Path designs.
- If MERGE NO is specified, WAIVER_DATABASE designs are not found, and *waived.gds* designs are not found, then issue a warning and check just the Layout Path designs.

Default: YES

- **RUN_RULE_CHECKSUM** {NO | YES}

Optional statement that controls whether rule checksums in waiver shapes should be compared to the associated rules in the verification rule file.

Default: NO

The following layer operations are unsupported for checksums. If YES is specified and one of the following is detected, a warning is issued and the rule is skipped.

(Not) Net	TDDRC	TVF
DFM Analyze	Net Area Ratio family	DFM Critical Area
Any MDP operations	DFM Measure	Any RET operations

See “[Rule Checksum Annotations](#)” on page 224 and “[Creating Waiver Cell Rule Checksum Annotations with waiver_flow](#)” on page 109.

- **TMP_DIR** *pathname*

Optional statement that defines the path to a temporary files directory. The *pathname* must be absolute.

Default: \$MGC_TMPDIR if it exists and is readable. If not, \$MGC_HOME/tmp if it exists and is readable. Otherwise, WORKING_DIRECTORY.


- **UNUSED_WAIVER_RDB** {ALL | NONE | GEOM | USER | PROP}

Optional statement that sets generation options for the *unused_waiver.rdb* results database. The arguments that control the file generation are described as follows:

- **ALL** — Specifies to write all geometry coordinates, overlap, and user properties to the file. This is the default.
- **NONE** — Specifies not to generate the file. Note, HTML waiver reports are not available when you specify this option.
- **GEOM** — Specifies to only write geometry coordinates to the file.
- **USER** — Specifies to only write user properties to the file.
- **PROP** — Specifies to only write overlap properties to the file.

Default: ALL

Note

 If you do not want statistics to be calculated and written to the summary file for the unused waivers, specify NONE in this option and include the OPTIMIZE_NONE_WAIVER_STATISTICS YES statement.


- **USED_WAIVER_RDB** {ALL | NONE | GEOM | USER | PROP}

Optional statement that sets generation options for the *used_waiver.rdb* results database. The arguments that control the file generation are described as follows:

- **ALL** — Specifies to write all geometry coordinates, overlap, and user properties to the file. This is the default.
- **NONE** — Specifies not to generate the file. Note, HTML waiver reports are not available when you specify this option.
- **GEOM** — Specifies to only write geometry coordinates to the file.
- **USER** — Specifies to only write user properties to the file.
- **PROP** — Specifies to only write overlap properties to the file.

Default: ALL

Note

 If you do not want statistics to be calculated and written to the summary file for the used waivers, specify NONE in this option and include the OPTIMIZE_NONE_WAIVER_STATISTICS YES statement.

- **USERNAME_TEXT_LAYER** *layer*

Optional statement that defines the layer number from which username text objects are read.

Default: LAYER_NUMBER specification

- **USERNAME_TEXTTYPE** *texttype*

Optional statement that defines the TEXTTYPE number from which username text objects are read. Ensure that the specified *texttype* does not collide with any other layer and texttype pair that is used in the waiver verification setup file, including defaults.

Default: 3

- **WAIVER_DATABASE** *pathname*

Optional statement that is used if at least some of your waiver shapes are not embedded in your main design. Multiple WAIVER_DATABASE statements are allowed, but all of the specified layouts must use the same precision. You can also specify a directory of waiver databases instead of individual database files. In this case, all waiver database files in the directory are applied in a verification run.

If MERGE NO is specified, then the WAIVER_DATABASE file is used (default: *waived.gds* file). In this case, at least some waiver shapes are expected to be in the WAIVER_DATABASE and they are merged with the main design at runtime. More than one WAIVER_DATABASE statement may be specified if non-embedded waiver shapes are in more than one file.

If MERGE YES is specified, then WAIVER_DATABASE has no effect. In this case, the tool expects all waiver shapes to be embedded in the main design.

If the .gz extension is specified for a filename, then the file is uncompressed using gunzip when gunzip is available in the environment.

Default: *waived.gds*


- **WAIVER_RDB** {ALL | NONE | GEOM | USER | PROP}

Optional statement that sets generation options for the *waived.rdb* results database. The arguments that control the file generation are described as follows:

- ALL — Specifies to write all geometry coordinates, overlap, and user properties to the file. This is the default.
- NONE — Specifies not to generate the file. Note, HTML waiver reports are not available when you specify this option.
- GEOM — Specifies to only write geometry coordinates to the file.
- USER — Specifies to only write user properties to the file.
- PROP — Specifies to only write overlap properties to the file.

Default: ALL

Note

 If you do not want statistics to be calculated and written to the summary file for the unused waivers, specify NONE in this option and include the OPTIMIZE_NONE_WAIVER_STATISTICS YES statement.

- **WORKING_DIRECTORY** *pathname*

Optional statement that defines the path to the working directory. Relative pathnames in the waiver verification setup file are resolved in relation to this setting. If the *pathname* is relative, then the paths in all generated files are relative to the working directory. Likewise, if the path is absolute, the paths in all generated files are absolute.

The following files are generated in the **WORKING_DIRECTORY**:

- All waiver files (*waived.rdb*, *used_waiver.rdb*, *unused_waiver.rdb*, *density_waived.rdb*)
- Waiver summary file (*calibre-waiver.summary*)
- Intermediate rule files (*waiver_dfm_rules.svrf*, *waiver_file_temp*)
- Intermediate Calibre YieldServer script (*ys_waiver_script.tcl*)
- Waiver shapes debug file (*debug_waiver.rdb*)
- DFM database (*waiver.dfmdb*), if you did not specify the DFM database in the rules file

The following files are generated in your current (run) directory, regardless of the **WORKING_DIRECTORY** setting:

- DRC summary file (both the DRC summary file and the file appended with “_original”)
- Results database file
- RDB files generated from the rule file and not waived
- DFM database, if defined in the rule file. If it is not defined in the rule file, the tool names the file *waiver.dfmdb* and generates it in the specified working directory

Default: Current directory in the shell

Examples

Here is an example file for when all waiver shapes are embedded in the main design. These examples show the **EXTRACT** option, which extracts waiver definitions from the input design.

```
WAIVER_CRITERIA EXTRACT  waiver_defs.cal
LAYER_NUMBER             1111
DATATYPE_NUMBER          2222
MERGE                    YES
CALIBRE_WAIVER_NUMBER    AUTO
```

Here is an example file for when any waiver shapes are in a separate file from the main design.

```
WAIVER_CRITERIA EXTRACT  waiver_defs.cal
LAYER_NUMBER      1111
DATATYPE_NUMBER   2222
MERGE             NO
WAIVER_DATABASE    waivers.gds
WAIVER_DATABASE    more_waivers.gds
CALIBRE_WAIVER_NUMBER AUTO
```

Related Topics

[Writing the Waiver Criteria File for Waiver Verification](#)

[Waiver Setup File Format for Waiver Generation](#)

Waiver Description Files


Waiver description files are used to define the waivers that you intend to enforce. There are two waiver description files: the waiver criteria and the waiver cells files.

The waiver criteria file statements specify the [Waiver Shape Interaction Types](#) that apply to each rule and the waiver cells file specifies the rules by cell in the design to which the waivers are applied.

If you specify in Calibre RVE to annotate the output waiver geometry file with waiver criteria text objects, then you do not need to write the waiver criteria file. In this case, the waiver verification setup file should include the WAIVER_CRITERIA statement's EXTRACT *pathname* option. This causes a waiver criteria file to be extracted and used during a calibre -waiver run. See "[Waiver Setup File Format for Waiver Verification](#)" on page 126 and "[Text Object Annotations](#)" on page 221.

If you use the waiver_flow tool to generate your waiver geometry (see "[Generating Waivers with the waiver_flow Tool](#)"), then these files must be written beforehand.

Note

 If the deprecated WAIVER_DESCRIPTION setup file command is specified in the waiver setup file, Calibre Auto-Waivers translates the file into the correct format. Alternatively, you can use waiver_util -splitdesc to split pre-2012.3 description files. The new files are saved in the working directory with the following filenames:

- *translated_waiv_criteria*
- *translated_waiv_cells*

Waiver Criteria File Description	144
Waiver Criteria File Format.....	147
Standard Form Criteria	150
EXCLUDE_CELL	154
INCLUDE_CELL.....	156
WAIVE_EXTENT	158
WAIVE_MARKER	161
WAIVE_PATTERN.....	164
WAIVE_DENSITY	167
P2P	171
Waiver Cells File Format	173
Waiver Cells File Description.....	174
Batch Mode File Usage	176

Waiver Criteria File Description

The waiver criteria file is optionally used by the `waiver_flow` executable to generate waiver shapes with custom criteria. It is used during calibre -waiver runs to enforce waivers. It can be imported into Calibre RVE to define waiver criteria.

See “[Waiver Criteria File Format](#)” on page 147 for details on the syntax of the waiver criteria file. The following table summarizes the keywords that are supported in the waiver criteria file:

Table 6-4. Waiver Criteria File Statements

Criteria	Description
<code>rule_name waiver_type waiver_overlap result_overlap</code> (Standard Form Criteria)	The standard form criteria specifies the waiver interaction type and the overlap values used to determine whether the waiver should be applied to the result.
<code>EXCLUDE_CELL</code>	Excludes all geometries from specified cells from the waiver run.
<code>WAIVE_EXTENT</code>	Applies waivers based upon the extent of a cell and the <i>rule_name</i> .
<code>WAIVE_MARKER</code>	Applies waivers based upon a marker layer and the <i>rule_name</i> .
<code>WAIVE_PATTERN</code>	Applies waivers based upon the pattern definition parameters and waiver criteria.
<code>WAIVE_DENSITY</code>	Applies waivers for Density results on a per-window basis, such as the results in a Density RDB.

For the standard form criteria, the *rule_name* parameters are expected to match rule check names in the rule file. If they do not match, a warning is issued.

The *rule_name* parameter is used with the associated cell name in the [Waiver Cells File Format](#) to generate waiver cell names as discussed under “[Waiver Cell Description](#)” on page 218.

The waiver criteria parameters are used to generate text objects in the waiver cells as discussed under “[Text Object Annotations](#)” on page 221.

Wildcard Matching in Criteria

The asterisk character (*) may be used in rule check names in the waiver criteria file. All unique pairings of rule names to check names are matched. Matching of names is case-sensitive. If there are multiple waiver criteria matches that could apply due to wildcards, one is chosen and a warning is issued.

An exact match of a rule name and check name takes priority over a wildcard match. For example, suppose these lines are in the waiver criteria file:

```
rule1 MULTI 90 90
rule* SINGLE 98 98
```


If the first statement exactly matches rule1, then the MULTI 90 90 setting is used. A warning is issued because the wildcards in the second line could also match this rule name, and the waiver criteria are different in the second line.

Name matching is done in this order of priority:

1. Exact match of a rule name.
2. Longest wildcard expression match of a rule name.

Suppose we have these lines in the waiver criteria file (line numbers are for illustration only):

```
1 rule* SINGLE 98 98
2 rule1* MULTI 100 90
```

For the rule1, rule10, rule11, and so forth, line 2 is used because that is the longest rule name match. For rule2, rule20, rule 21, and so forth, line 1 is used. Warnings are issued in both cases because multiple differing waiver criteria could be applied due to the possible wildcard matches.

Additional Considerations for Secondary Criteria

The following cases describe wildcard matching for secondary criteria:

Case 1

Consider the following criteria in the waiver criteria file:

```
rule* SINGLE 100 100
rule1 SINGLE 98 98
```

In this case, rule1 is assigned a criteria of SINGLE 98 98, as this is the primary criteria. The most specific criteria (rule1) overrides the general match (rule*)

Case 2

Consider the following criteria in the waiver criteria file:

```
rule* WAIVE_MARKER marker1
rule1 WAIVE_MARKER marker2
```

In this case, rule1 is assigned *both* marker layers, as these are secondary criteria (the same behavior applies to WAIVE_EXTENT and WAIVE_MARKER). Multiple secondary criteria statements are accumulated from all criteria that match a rule (instead of being overridden as in Case 1).

Explicitly Specify Not to Apply Waivers

The NONE keyword ensures that no waivers are applied for a specified rule. For example, a typical waiver criteria statement may look like this:

```
metall* SINGLE 100 100
```

If you do not want to apply waivers for a particular rule that matches the example wildcard statement, then you could specify the following criteria statement:

```
metall_spacing NONE
```

If you specify NONE with other primary criteria (for example, metall_a NONE SINGLE 98 98), the tool issues a syntax error similar to the following:

```
ERROR: Unexpected token 'SINGLE' found after NONE.
```

Waiver Criteria Type and Overlap Restriction

For each **rule_name**, the *waiver_type*, *waiver_overlap*, and *result_overlap* values must only be specified once. If this is not the case, an error is issued at Calibre Auto-Waivers run time.

WAIVE_DENSITY cannot be specified under a rule with a primary *waiver_type*, *waiver_overlap*, and *result_overlap*.

Resolution of Rule Conflicts in the Waiver Criteria File

When two or more statements in the waiver criteria file conflict, Calibre Auto-Waivers chooses the final specification statement based on the following order of precedence:

1. Specified rule
2. Wildcard rule
3. SINGLE 100 100

If the conflicts are at the same level, then the last item listed takes precedence.

Related Topics

[Waiver Shape Interaction Types](#)

[Waiver Cell Description](#)

[Waiver Description Files](#)

[Waiver Setup File Format for Waiver Generation](#)

[Waiver Setup File Format for Waiver Verification](#)

Waiver Criteria File Format

Input for: [waiver_flow](#), [calibre -waiver](#), and Calibre RVE

This file is optional for waiver generation in [waiver_flow](#) and verification in [calibre -waiver](#). It is used in [calibre -waiver](#) to specify criteria for waiver shapes.

The file can be written manually, or it can be extracted from waiver shapes with embedded criteria text using the `WAIVER_CRITERIA EXTRACT` setup file command. You may optionally import a waiver criteria file into Calibre RVE to apply non-default waiver criteria during waiver shape export.


The waiver criteria file is used to specify which rules can be waived and how the generated waiver shape interacts with a result region. See “[Waiver Shape Interactions](#)” on page 53 for complete details.

To see how waiver criteria values affect waiver shape interactions, view the following video:



The waiver criteria file is parsed at run time. Fatal errors are flagged as a result of disallowed statements or syntax errors.

Caution

 Do not name the waiver criteria file *waiver_desc*. Calibre Interactive assumes any file with this name is a legacy description file and automatically converts it to a waiver cells and waiver criteria file when you click on the Waivers tab. In this case, Calibre Interactive issues a syntax error if the file is a valid waiver criteria file.

Format

The waiver criteria are specified by rule and each rule must only be listed once, with the criteria written in-line after the rule. The default criteria for all rules are `SINGLE 100 100`.

The syntax of each line in the file depends on the type of criteria you want to apply.

The file must conform to the following format:

- For each rule, only one primary criteria (*waiver_overlap* and *result_overlap*) can be applied. The `WAIVE_EXTENT`, `WAIVE_MARKER`, and `WAIVE_PATTERN` options are specified on the same line.

- The file is not case-sensitive.
- The # character at the beginning of a line indicates a comment.
- Lines in the file may not end in ^M characters (typically found in Windows-formatted text files).

Parameters

No Waiver Criteria

To specify not to waive a rule, use the NONE keyword with no other arguments:

```
rule_name NONE
```

Standard Form Criteria

A [Standard Form Criteria](#) rule consists of the **rule_name** followed by the **waiver_type**, **waiver_overlap**, and **result_overlap** values.

```
rule_name waiver_type waiver_overlap result_overlap
```

Exclude Cell Waiver Criteria

```
EXCLUDE_CELL cell_name [cell_name ...] [HALO value] [BY_LAYER layer_name]  
[PRESERVE layer]
```

Include Cell Waiver Criteria

```
INCLUDE_CELL cell_name [cell_name ...] [HALO halo_value]  
[BY_LAYER layer_name]
```

Extent Waiver Criteria

```
rule_name WAIVE_EXTENT cell [cell ...]  
{[UNDERSIZE size_value] | [OVERSIZE size_value]}  
[waiver_type waiver_overlap result_overlap]
```

Marker Waiver Criteria

```
rule_name WAIVE_MARKER marker_layer [marker_layer] [UNDERSIZE size_value]  
[INSIDE cell [cell ...]] [waiver_type waiver_overlap result_overlap]  
[FULL]
```

Pattern Waiver Criteria

```
rule_name WAIVE_PATTERN pattern dmacro_file library_name  
input_layer_count input_layer [input_layer ...] ["options_string"]  
output_layer_count [(output_layer_list)]}  
[waiver_type waiver_overlap result_overlap]
```

Density Waiver Criteria

You can specify a density rule for the **rule_name** argument in the above mentioned standard, extent, marker, and pattern waiver usages, or you can waive a density rule by a density percentage using the following syntax:

```
density_rule WAIVE_DENSITY result_overlap
```

P2P Waiver Criteria

P2P:*experiment_name* **P2P** [*tolerance* | {*tolerance_above* *tolerance_below*}]

Standard Form Criteria

Waiver criteria file statement.

The standard form criteria specifies the waiver interaction type and the overlap values used to determine whether the waiver should be applied to the result.

Usage

Standard Form Criteria

rule_name waiver_type waiver_overlap result_overlap

No Waiver Criteria

rule_name NONE

Arguments

- *rule_name*

Required name of the rule check to which the waiver applies. A *rule_name* with matching criteria may only be specified once.

The asterisk (*) wildcard is allowed and matches zero or more characters. Rule names can use characters that are not allowed as parts of cell names in GDS format. The allowed characters for GDS cell names are alphanumeric characters and the `_`, `?`, and `$` characters. If a disallowed character for GDS is used in a *rule_name*, then the generated waiver cell name substitutes this for the disallowed character:

`?$ASCII_code?`

Here, *ASCII_code* is the representation of the disallowed character. See “[Waiver Cell Description](#)” on page 218.

Most Calibre nmDRC layer operations are supported, but not all. In the case of unsupported layer operations, if you can rewrite the rule to derive the layer outside of a rule check, and then output the layer inside a rule check using a [Copy](#) or [DFM Copy](#) operation, then the rule check may be used. See “[SVRF Command Support](#)” on page 27 for a complete description of layer operation support.

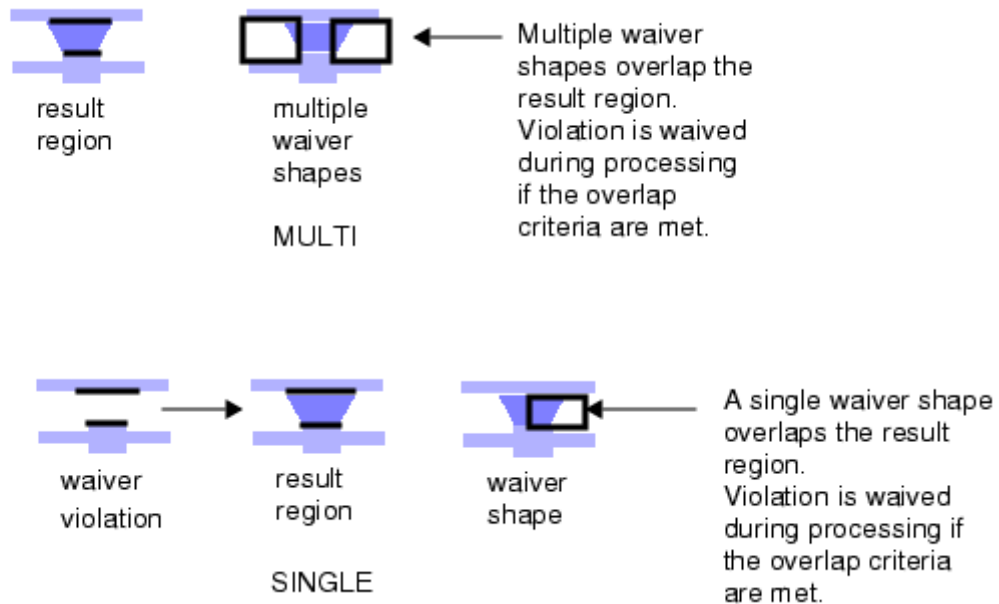
Unsupported layer operations are listed under “[Unsupported Layer Operations in waiver_flow](#)” on page 29.

- *waiver_type*

Required argument that specifies the number of waiver interactions that are allowed. See “[Waiver Shape Interaction Types](#)” on page 54 for additional background. The following types are available.

- SINGLE — Specifies that only one waiver shape may overlap a result region to waive the error result.
- MULTI — Specifies that one or more waiver shapes may overlap a result region. For all waiver shapes that meet the *waiver_overlap* value for a given result region,

the overlap areas from each shape with the result region are added together (intersections between these areas are counted once). The sum of these areas is then used to determine the *result_overlap*. Only waiver shapes generated from the same rule check are considered.



Again, for any given *rule_name*, the *waiver_type*, *waiver_overlap*, and *result_overlap* must only be specified once. For instance:

```
metal_rule SINGLE 99 99 WAIVE_EXTENT cellA SINGLE 98 98
```

This example is incorrect because two primary criteria are applied to the same rule. To prevent a conflict, only one primary criteria can be applied. Either of these statements will correct this example:

```
metal_rule SINGLE 99 99 WAIVE_EXTENT cellA
```

or

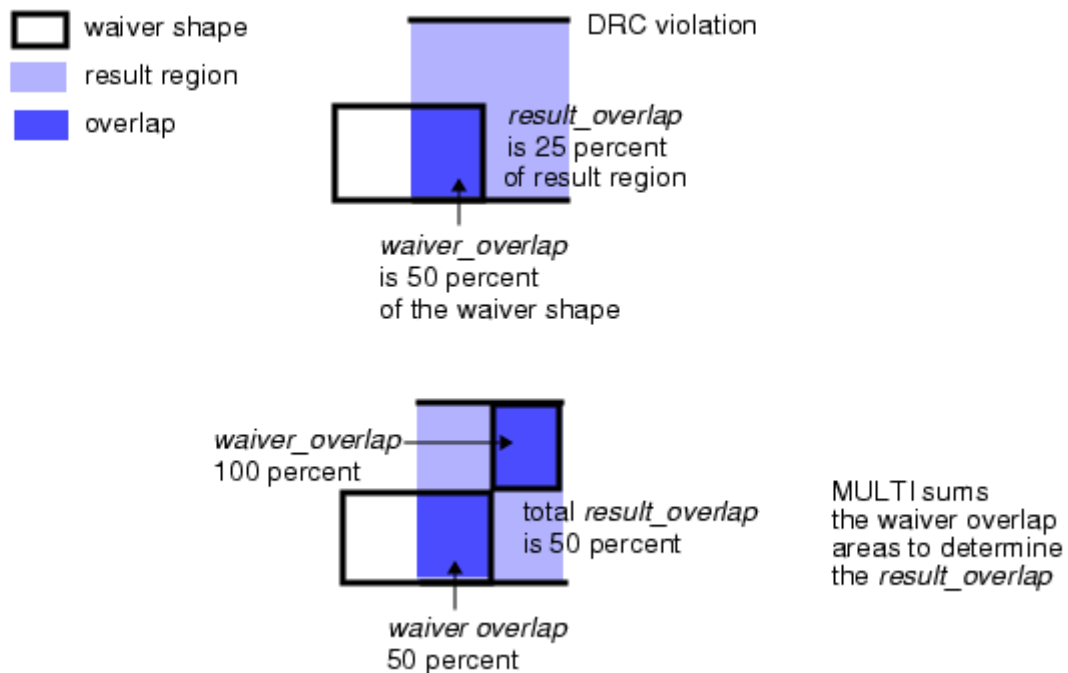
```
metal_rule WAIVE_EXTENT cellA SINGLE 98 98
```

- ***waiver_overlap***

Required argument that specifies the *minimum* percent of the overlap area of a waiver shape with a result region. This is an integer value from 0 to 100, interpreted as percent. This parameter is specified but has no effect in WAIVE_EXTENT or WAIVE_MARKER statements. The *waiver_overlap* value must only be specified once per rule.

If 100 is specified, then the overlap must be 100 percent. If a number from 1 to 99 is specified, then the overlap must be greater than or equal to the value. If 0 is specified, then there must be at least one square database unit of overlap. If a waiver shape meets the

waiver_overlap value, then the shape is used to determine if the *result_overlap* criterion is met.




This value is used for the measurement of polygons. However, for results reporting, the value is rounded to the nearest integer.

- ***result_overlap***

Required argument that specifies the *minimum* percent of the overlap area of waiver shapes with a result region. The *result_overlap* value must only be specified once per rule. This is an integer value from 0 to 100, interpreted as percent.

If 100 is specified, then the overlap must be 100 percent. If a number from 1 to 99 is specified, then the overlap must be greater than or equal to the value. If 0 is specified, then there must be at least one square database unit of overlap.

Note

 If you have no waiver shapes and you are waiving only using secondary criteria, such as WAIVE_EXTENT, WAIVE_PATTERN, and WAIVE_MARKER, it is recommended that you set the criteria to “{SINGLE | MULTI} N 100”

where *N* is the waiver coverage, which has no effect on the waiving. The 100 results coverage criteria enables higher performance. If you relax the results coverage (anything other than 100) then the tool uses the less optimized flow.

If SINGLE is specified, then a single waiver shape that meets the *waiver_overlap* criterion is used to determine the *result_overlap*. If MULTI is specified, then all waiver shapes that meet the *waiver_overlap* criterion are used to determine the *result_overlap* coverage area.

The percent of overlap is then based upon the aggregate overlap of these waiver shapes with the result region, for example:

pdiff.12	SINGLE	98	98
m2.1	MULTI	50	100

For the first line, waivers are applied for the specified rule if all of the following is true:

- Only one waiver shape is present
- 98 percent of its area intersects the result region
- 98 percent of the result region is covered by the waiver shape

For the second line, waivers are applied to the specified rule if all of the following is true:

- At least one waiver shape is present
- At least 50 percent of its area intersects the result region
- The result region is entirely covered by all of the waiver shapes that interact with it that meet the 50 percent criterion

If WAIVE_EXTENT is specified, then the waiver shape is the extent of the specified *cell* and *result_overlap* is applied accordingly. Similarly, if WAIVE_MARKER is specified, then the waiver shape is taken as the extent of a *marker_layer* shape. If WAIVE_PATTERN is specified, the waiver shape is taken from the output layers.

- NONE

Optional argument that specifies that no waivers for the ***rule_name*** are applied in calibre -waiver or generated in waiver_flow. NONE must *not* be specified with other primary criteria.

Description

Examples

```
m1_rule SINGLE 100 100
```

EXCLUDE_CELL

Waiver criteria file statement.

Optional parameter set that excludes all geometries from specified cells from the waiver run. This option is useful if you want to remove certain cells (blocks) from your run that can impact performance and introduce errors that can be ignored.

Note



A single Calibre Real-Time Digital license is required for jobs with EXCLUDE_CELL in the criteria file, regardless of the number of CPUs.

Usage

EXCLUDE_CELL {*cell_name* [*cell_name* ...]} [HALO *value*] [BY_LAYER *layer_name*]
[PRESERVE *layer* ...]

Arguments

- *cell_name*
Required name of the cell to exclude from the run. You must specify at least one cell. Without any additional arguments, the tool removes the entire cell from processing.
- HALO *value*
Optional halo distance that forms an outside-facing region of the block to *include* in the run. This enables you to remove the contents of a cell, but still check the interacting regions. The distance *value* is specified in user-units.
- BY_LAYER *layer_name*
Optional argument that excludes the cell based on the extents of the specified *layer_name*. This option is useful for non-rectilinear cells.
- PRESERVE *layer_name* ...
Optional argument that includes the specified *layer_name* in the verification run. Multiple layers can be specified. This is useful for excluding incomplete blocks, but including certain layers from the block that are known to be ready for verification.

If you choose to exclude a cell from a design using the EXCLUDE_CELL keyword, but also specify the PRESERVE option to keep certain layers in the verification run, the tool automatically removes any applied waivers from the rule checks using the preserved layers that are the result of wildcard matching.

For example, if you specify the following criteria:

```
* SINGLE 100 100
ruleA SINGLE 100 100
```

The first statement uses a wildcard to apply criteria to all rules. Any of these rules that use preserved layers as a part of their derivation tree will not be waived. However, waiver criteria that is specified for explicit rules (without wildcards) that interact with the preserved

layers are still applied. Since the second statement explicitly specifies criteria for ruleA, then those waivers for ruleA are still applied. When the tool automatically ignores a rule on a preserved layer, the following warning message is issued:

```
WARNING: Rulecheck 'rule' will not be waived as it is using a  
preserved layer in its derivation tree.
```

Description

By combining the EXCLUDE_CELL statement and WAIVE_EXTENT (for rectangular blocks) and WAIVE_MARKER (for rectilinear blocks) with the UNDERSIZE keyword, you can remove geometries from a cell and waive specified rules from the overlapping region. For example:

```
EXCLUDE_CELL block_A HALO 100  
  
m1_space WAIVE_EXTENT block_A UNDERSIZE 50
```

When a layer is used in both the EXCLUDE_CELL BY_LAYER keyword and in the WAIVE_MARKER criteria, the generated marker layer will be the difference between the marker layer (sized down by UNDERSIZE) and the BY_LAYER layer (sized down by the HALO). If a cell is used in both EXCLUDE_CELL and WAIVE_MARKER statements, then the HALO value must be greater than or equal the UNDERSIZE value.

Note



If you use the BY_LAYER option with EXCLUDE_CELL, you cannot specify the WAIVE_EXTENT criteria with the same cell name used in the EXCLUDE_CELL statement. In these cases, you should use WAIVE_MARKER instead.

Examples

See “[Block-Level Waivers \(Gray Box\)](#)” on page 62 for an example.

INCLUDE_CELL

Waiver criteria file statement.

Optional parameter set that includes all geometries from the specified cells in the waiver run. All other cells are ignored.

Note



A single Calibre Real-Time Digital license is required for jobs with INCLUDE_CELL in the criteria file, regardless of the number of CPUs.

Usage

INCLUDE_CELL *cell_name* [*cell_name* ...] [HALO *halo_value*]
[BY_LAYER *layer_name*]

Arguments

- ***cell_name***
Required name of the cell to include in the run. You must specify at least one cell. Without any additional arguments, the tool includes only the specified cells in the run based off of their cell extents.
- **HALO *value***
Optional halo distance that forms an outside-facing region of the cell to *include* in the run. This enables you to include both the contents of a cell and the interacting regions around the cell. The distance *value* is specified in user-units.
- **BY_LAYER *layer_name***
Optional argument that includes the cell based on the extents of the specified *layer_name*. This option is useful for non-rectilinear cells.

Description

The geometry that belongs to all cells listed by this command are loaded during a verification run and any other geometry from the layout file is ignored. This improves performance and results debugging. This capability is useful for block designers that need to validate their blocks in the context of a full chip without expending time checking the rest of the design.

If you specify the HALO argument set, then geometry that surrounds the cell by a specified positive *halo_value* distance is also included in the run. This enables you to check the interfacing regions around the cell. If you want to manually specify the extent of the cell, use the BY_LAYER argument with a layer that represents the extents of the cell. This is useful for rectilinear cells.

Any violation determined to be outside or touching the cell's extents is automatically waived by default. You can add criteria to cells and rules as usual to adjust the waiving operation. You can use the NONE criteria format to disable waiving.

The INCLUDE_CELL and EXCLUDE_CELL statements are mutually exclusive.

Examples

See “[Include Cells in the Chip Context](#)” on page 68 for an example.

WAIVE_EXTENT

Waiver criteria file statement.

Specifies that waivers are applied based upon the extent of the cell and the rule name. Results that intersect the cell's extent and meet the result overlap criterion are waived.

Usage

WAIVE_EXTENT *cell* [*cell* ...] {[UNDERSIZE *size_value*] | [OVERSIZE *size_value*]}
[*waiver_type* *waiver_overlap* *result_overlap*] [FULL]

Arguments

- *cell*

Required argument that specifies the cell to which the WAIVE_EXTENT operation is applied. You must specify at least one cell.

The *cell* values may use wildcards, but the value must be contained within quotes. For example, WAIVE_EXTENT "nand*" matches any cells that begin with the string nand.

- UNDERSIZE *size_value*

Optional argument set that specifies a value by which to shrink the extent of the cell.

The *size_value* is a positive floating-point number in user units that shrinks the results of the extent cell operation. The tool applies the shrinking operation to all cells after WAIVE_EXTENT keyword and before the UNDERSIZE keyword. UNDERSIZE is mutually exclusive with OVERSIZE.

- OVERSIZE *size_value*

Optional argument set that specifies a value by which to grow the extent of the cell. The *size_value* is a positive floating-point number in user units that expands the results of the extent cell operation. The tool applies the expansion operation to all cells after WAIVE_EXTENT keyword and before the OVERSIZE keyword. OVERSIZE is mutually exclusive with UNDERSIZE.

Note



The OVERSIZE keyword can only be used with EXCLUDE_CELL when you specify the HALO keyword. In this case, the generated waiver layer is the area between the cell's extent (as increased by the OVERSIZE *size_value*) and the extent size decreased by the halo value.

- *waiver_type* *waiver_overlap* *result_overlap*

Optional argument set that specifies the waiver criteria for the *rule_name*. See [“Standard Form Criteria”](#) on page 150 for details.

If a standard form waiver statement specifies the *waiver_type*, *waiver_overlap*, and *result_overlap* and a WAIVE_EXTENT statement follows in-line, then the WAIVE_EXTENT statement does not need to specify these parameters as they are taken

from the standard form statement. Otherwise, the *waiver_type*, *waiver_overlap*, and *result_overlap* must be specified.

Note that the *waiver_overlap* parameter is specified with a *result_overlap*, but the former parameter has no effect if you are only using secondary criteria and have no waiver shapes. A *result_overlap* of 100 achieves higher performance and is recommended.

- **FULL**

Optional argument that generates a marker layer over the entire area in which to waive all top-level results. If this option is not specified, there can still be connectivity, density, or other violations due to omitting all of the geometries in the extent.

Description

Waivers using cell extents are often applied when a DRC rule does not apply to a certain cell. Here is an example of an extent waiver statement:

```
m1_width WAIVE_EXTENT analog_block SINGLE 1 50
```

Any single result from the m1_width rule that shares at least 50 percent of its area with the extent of the analog_block cell is waived. The value “1” is arbitrary and does not have an effect for WAIVE_EXTENT. Wildcards are also supported for cell names, as shown in this example:


```
m1_* WAIVE_EXTENT "analog*" SINGLE 1 50
```

The quotes (“”) are required when using wildcards (*) for cells in WAIVE_EXTENT.

If a WAIVE_EXTENT statement is specified without the *waiver_type*, *waiver_overlap*, and *result_overlap* parameters, but after a **WAIVE_DENSITY** or standard form statement for the same *rule_name*, then the **WAIVE_DENSITY** or primary criteria are applied to the extent of the *marker_layer* during a calibre -waiver run.

You can optionally adjust the size of cell extent waivers when using WAIVE_EXTENT criteria with the **UNDERSIZE** or **OVERSIZE** keywords.

Note

 In waiver_flow, WAIVE_EXTENT generates waiver cells at the top level that contain only text object annotations. No waiver shapes are generated, so that the **WAIVER_CRITERIA** **EXTRACT** or **APPEND** options can be used. Whether these cells are present or not, calibre -waiver applies waivers according to the cell, rule, and result overlap specified.

See “[Waiver Criteria File Description](#)” on page 144 for additional details.

Examples

UNDERSIZE Example

The following usage waives results inside the full extent of cell1:

```
rule SINGLE 0 0 WAIVE_EXTENT cell1
```

The following usage waives results inside both cell1 and cell2 that are 0.3 user units smaller than the extent in all dimensions:

```
rule SINGLE 0 0 WAIVE_EXTENT cell1 cell2 UNDERSIZE 0.3
```

To apply different UNDERSIZE values to multiple cells, apply multiple WAIVE_EXTENT statements as follows:

```
rule SINGLE 0 0 WAIVE_EXTENT cell1 UNDERSIZE 0.2 WAIVE_EXTENT cell2 UNDERSIZE 0.3
```

The WAIVE_EXTENT UNDERSIZE keyword can be combined with EXCLUDE_CELL for rectangular blocks.

WAIVE_MARKER

Waiver criteria file statement.

Specifies that waivers are applied based upon the marker layer and the rule name. The marker layer is defined in the rule file. Results that intersect the marker layer and meet the result overlap criterion are waived.

Usage

WAIVE_MARKER *marker_layer* [*marker_layer ...*] [UNDERSIZE *size_value*]
 [INSIDE *cell* [*cell ...*]] [*waiver_type* *waiver_overlap* *result_overlap*] [FULL]

Arguments

- ***marker_layer*** [*marker_layer ...*]


Required argument that specifies a marker layer that is defined in the rule file. Results that intersect the ***marker_layer*** and meet the *result_overlap* criterion are waived.

- *waiver_type* *waiver_overlap* *result_overlap*

Optional argument set that specifies the waiver criteria for the ***rule_name***. See “[Standard Form Criteria](#)” on page 150 for details.

If a standard form waiver statement specifies the *waiver_type*, *waiver_overlap*, and *result_overlap* and a WAIVE_MARKER statement follows in-line, then the WAIVE_MARKER statement does not need to specify these parameters as they are taken from the standard form statement. Otherwise, the *waiver_type*, *waiver_overlap*, and *result_overlap* must be specified.

Note

 The *waiver_overlap* parameter is specified with a *result_overlap*, but the former parameter has no effect if you are only using secondary criteria and have no waiver shapes. A *result_overlap* of 100 achieves higher performance and is recommended.

- **INSIDE** *cell*

Optional argument set that specifies a cell in which the marker layer applies. The marker layer only applies to the specified list of cells. This can be useful if you want to apply marker layers for rectilinear blocks within certain cells. For example, the following command uses a marker layer inside cells a, b, and c.

```
rule_2 WAIVE_MARKER marker_layer INSIDE a b c
```

If the marker exists in cell d, the waiver will not be applied.

- **UNDERSIZE** *size_value*

Optional argument set that specifies a value by which to shrink the marker layer. The *size_value* is a positive floating-point number in user units that shrinks the layer in all dimensions. The usage is similar to WAIVE_EXTENT and it can be combined with EXCLUDE_CELL.

The WAIVE_MARKER UNDERSIZE statement can be combined with EXCLUDE_CELL for rectilinear blocks.

- FULL

Optional keyword that generates a marker layer over the entire area in which to waive all top-level results. If this option is not specified, there can still be connectivity, density, or other violations due to omitting all of the geometries in the marker area.

See “[Waiver Criteria File Description](#)” on page 144 for additional details.

Description

The WAIVE_MARKER statement must be specified in-line with a primary *rule_name* or *density_rule*.

In waiver_flow, WAIVE_MARKER generates waiver cells at the top-level that contain only text object annotations. No waiver shapes are generated, so that the WAIVER_CRITERIA EXTRACT or APPEND options can be used. Whether these cells are present or not, calibre -waiver applies waivers according to the layer, rule, and result overlap specified.

Note



The WAIVE_MARKER INSIDE option can be applied with the EXCLUDE_CELL and BY_LAYER options; any marker layer outside the cells specified by the INSIDE argument set are not considered in the WAIVE_MARKER operation.

Waivers using a marker layer are often applied when the DRC rules do not apply to blocks such as memories and analog elements. The marker layer is defined in the rule file. Here is an example of a marker waiver command:

```
m1_width WAIVE_MARKER analog_mkr SINGLE 1 50
```

Any single result from the m1_width rule that shares at least 50 percent of its area with the analog_mkr layer is waived, no matter in which cell the result may occur. The value “1” is arbitrary and does not matter for WAIVE_MARKER.

If a WAIVE_MARKER statement is specified without the *waiver_type*, *waiver_overlap*, and *result_overlap* parameters, but after a WAIVE_DENSITY or standard form statement for the same *rule_name*, then the WAIVE_DENSITY or primary criteria are applied to the extent of the *marker_layer* during a calibre -waiver run.

You can optionally adjust the size of marker layer waivers when using WAIVE_MARKER criteria. The “UNDERSIZE *size_value*” argument set specifies a value by which to shrink the marker layer of the cell during waiver application.

Derived layers are *not* supported for marker layers if any of the following conditions occur:

- WAIVE_MARKER is used with the INSIDE option. The tool issues an error during waiver application.

- EXCLUDE_CELL is used with the BY_LAYER option for the same cell. The tool issues an error during waiver application.
- If the marker layer of different cells is abutting and the UNDERSIZE keyword is used.

Examples

```
m1_width WAIVE_MARKER analog_mkr SINGLE 1 50
```

WAIVE_PATTERN

Waiver criteria file statement.

Specifies that waivers are applied based upon the pattern definition parameters and waiver criteria. Only Topological Edge Matching (TEM) patterns are supported.

Usage

WAIVE_PATTERN *pattern_dmacro_file library_name input_layer_count input_layer*
[*input_layer ...*] [*“options_string”*] *output_layer_count* [(*output_layer_list*)] [*waiver_type*
waiver_overlap result_overlap]

Arguments

- ***pattern_dmacro_file***
Required path to a pattern DMACRO file.
- ***library_name***
Required name of a library in the *pattern_dmacro_file*.
- ***input_layer_count***
Required number of input layers in the pattern CMACRO call.
- ***input_layer***
Required input layer for the pattern. At least one input layer must be specified.
- ***“options_string”***
Optional arguments for the DMACRO. If the pattern DMACRO has no options, then this parameter is not required. If the pattern DMACRO has the option string enabled and you want to add runtime CMACRO options, the options should be specified as the *“options_string”* with the enclosing quotes. Typical usage does not require this parameter.
- ***output_layer_count***
Required number of output layers in the pattern.
- **(*output_layer_list*)**
Optional list of integers delimited by commas followed by whitespace and enclosed in parentheses. The integers correspond to the ordinal sequence of marker layers in the pattern DMACRO. If not specified, all marker layers are used for generating waiver shapes. If specified, then the *output_layer_list* defines which marker layers are used for waiver shapes, based upon their order in the CMACRO call. For example, if (1, 2) is specified, then only the first and second marker layers in the pattern are merged together and used for waiver shapes.
- ***waiver_type waiver_overlap result_overlap***
Optional argument set that specifies the waiver criteria for the ***rule_name***. See [“Standard Form Criteria”](#) on page 150 for details.

If a standard form waiver statement specifies the *waiver_type*, *waiver_overlap*, and *result_overlap* and a WAIVE_PATTERN statement follows in-line, then the WAIVE_PATTERN statement does not need to specify these parameters as they are taken from the standard form statement. Otherwise, the *waiver_type*, *waiver_overlap*, and *result_overlap* must be specified.

Note that the *waiver_overlap* parameter is specified with a *result_overlap*, but the former parameter has no effect if you are only using secondary criteria and have no waiver shapes. A *result_overlap* of 100 achieves higher performance and is recommended.

Description

In *waiver_flow*, WAIVE_PATTERN generates waiver cells at the top level that contain only text object annotations. No waiver shapes are generated. Whether these cells are present or not, calibre -waiver applies waivers according to all of the specified parameters except *waiver_overlap*.

See the [Calibre Pattern Matching User's Manual](#) for information about patterns.

DRC rules that employ Calibre Pattern Matching can be waived.

A simple way to implement these types of waivers is to use the MULTI 0 0 waiver criteria, for example:

```
pattern_rule MULTI 0 0
```

A more precise solution is to use the WAIVE_PATTERN statement. Here is an example of a pattern waiver command:

```
m1_rule WAIVE_PATTERN pattern_lib.svrf pattern_lib_m1 1 m1 1 SINGLE 2 98
```

These are the parameter definitions:

- **m1_rule** — Name of the rule that receives waivers.
- **pattern_lib.svrf** — Pattern DMACRO library file.
- **pattern_lib_m1** — Pattern library name.
- **1** — Number of input layers.
- **m1** — Layer in the design on which to search for a pattern.
- **1** — Number of output layers for the pattern.
- **SINGLE 2 98** — Waiver criteria; the 2 is arbitrary.

Any single result from m1_rule that shares at least 98 percent of its area with the output layer from pattern_lib_m1 is waived.

If a WAIVE_PATTERN statement is specified without the *waiver_type*, *waiver_overlap*, and *result_overlap* parameters, but after a WAIVE_DENSITY or standard form statement for the same **rule_name**, then the WAIVE_DENSITY or primary criteria are applied to the extent of the pattern during a calibre -waiver run.

Examples

The following example shows typical waiver criteria file statements used to set the waiver shape criteria for specified rules.

```
#####Standard form description rules#####
rule1 SINGLE 90 100
rule2 MULTI 75 75
#####WAIVE_DENSITY description rules#####
rule3 WAIVE_DENSITY 75
rule4 WAIVE_DENSITY 50 WAIVE_EXTENT cellA
####Wild card support###
rule3* MULTI 98 98
*DEN* WAIVE_DENSITY 100
###Multiple parameter waiver description rules###
rule7 SINGLE 98 98 WAIVE_EXTENT cellA cellB* WAIVE_MARKER exclude \
exclude1 WAIVE_PATTERN p1 p1 2 M1 M2 " " 3 (2,3)
##Rules for which you do not want to generate or apply waivers
rule3_a NONE
DENSITY_rule16 NONE
```

WAIVE_DENSITY

Waiver criteria file statement.

Specifies that the waivers are applied for Density results on a per-window basis, such as the results in a Density RDB.

Usage

density_rule { **WAIVE_DENSITY** [*result_overlap*] } | NONE

Arguments

- *result_overlap*
Optional argument that specifies the overlap percentage of the density result with the waiver. **WAIVE_DENSITY** is not specified with a *waiver_type* or *waiver_overlap* parameter. The *result_overlap* parameter is applied per-window, not per-merged-result as is done with a standard form waiver statement.
- NONE
Optional argument that specifies that no waivers for the specified *density_rule* are applied in calibre -waiver or generated with waiver_flow. NONE must not be specified with other primary criteria.

Description

The tool applies **WAIVE_DENSITY** waivers where a [Density](#) window's area intersects a waiver shape, and the ratio of the area of intersection to the area of the Density window exceeds the *result_overlap* percentage. For example, if your *result_overlap* is 50, then a result window is waived when greater than 50 percent of its area intersects the waiver shape. The only exception to this is when a *result_overlap* of 100 is used. In that case, 100 percent intersection is required for a waiver to apply.

If the *density_rule* is specified in a standard form statement and a **WAIVE_DENSITY** statement exists, then the **WAIVE_DENSITY** statement is applied by default unless the rule has no Density operation in it. In the latter case, the corresponding standard form statement is applied and a WARNING is issued.

The *density_rule* check can have only one layer operation that has output to the DRC Results Database when **WAIVE_DENSITY** is used, otherwise a warning is issued and no waivers occur.

WAIVE_DENSITY statements are applied in these ways:

- **waiver_flow** — Waiver shapes are always generated for the final layer operation in the rule (the rule with output to the DRC Results Database), whether that operation is a Density operation, or not.
- **calibre -drc -waiver** — Waivers are applied to the Density operation in the rule, if that operation is deriving a layer, or that operation is the final one in the rule. If the Density

operation is deriving a layer, that layer is manipulated by any subsequent layer operations in the rule. If this occurs, DRC results can intersect waiver shapes, which can lead to non-intuitive results. In such cases, use a standard form waiver statement, or a WAIVE_EXTENT or WAIVE_MARKER statement in conjunction with the WAIVE_DENSITY statement.

Density waivers may be applied using the standard form waiver criteria syntax; however, most users want to waive [Density](#) results on a per-window basis rather than on a per-merged-result basis. To implement per-window waivers, the WAIVE_DENSITY statement is used.

For this waiver rule:

```
density_rule WAIVE_DENSITY 99
```

a waiver_flow run generates waiver shapes for all ***density_rule*** results windows. In a subsequent calibre -waiver run, any ***density_rule*** result window that is completely overlapped (99 percent coverage) by a waiver shape is waived. See “[Density Window Waivers](#)” on page 42 for information about this flow.

The output of the calibre -drc -waiver run includes *density_waived.rdb* and *unused_waiver.rdb* files, which can be loaded into Calibre RVE.

If your Density results are false errors due to an offset of the cell-level Density window grid versus the top-level window grid, you can calculate the threshold *result_overlap* value for waiving all Density results in the cell, as follows. Assume these definitions:

x-offset — Difference in the +x direction between the cell-level Density window grid and the top-level window grid.

wx — Length of the Density window in the x-direction.

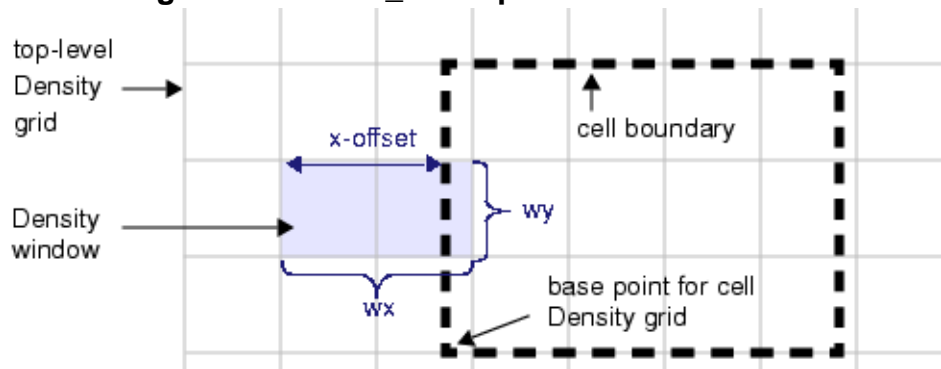
wy — Length of the Density window in the y-direction.

area_offset = x-offset * wy

area_window = wx * wy

threshold percent = ((1- area_offset/area_window) * 100) - 1

Figure 6-3. result_overlap Waiver Threshold



A *result_overlap* that is less than or equal to the threshold percent waives all results in the cell. Calculations for y-direction offsets are similar. If there is an offset in both the x- and y-directions, the lesser threshold percent serves as the threshold for both directions.

The preceding threshold calculation is cumbersome because of layout dependencies. An easier solution to waiving all Density results in a cell is to do this:

```
density_rule WAIVE_DENSITY 100 WAIVE_EXTENT cell
```

This combination of statements causes all *density_rule* results to be waived when they are completely overlapped (100 percent coverage) by the extent of the cell. See “[Using Cell Extent Waivers](#)” on page 46 for information about using the EXTENT keyword.

Another common usage is to use a marker layer to waive Density results when a cell is placed in the main design. To do this, you can have the following:

```
density_rule WAIVE_DENSITY 100 WAIVE_MARKER marker
```

This causes all *density_rule* results to be waived when they are completely overlapped by the marker layer in the cell.

If you have a WAIVE_DENSITY rule check that contains supported layer operations other than Density, this can have side effects.

Consider this rule:

```
Check1{
X = DENSITY M1 < 0.4 WINDOW 100 100 STEP 10
SIZE X BY 100
}
```

The waiver_flow command generates the waiver shapes from the [Size](#) operation. Assume this waiver criteria statement is used: Check1 WAIVE_DENSITY 100. During the calibre -waiver run, the initial results are those Density windows outside of the waiver shapes. These results then have the Size operation applied to them. It is possible that after sizing, there are results that overlap a waiver shape. This is probably not what is desired; you may prefer to use a standard

waiver statement, a WAIVE_MARKER statement, or a WAIVE_EXTENT statement in conjunction with WAIVE_DENSITY.

Here is another case:

```
Check2{  
X = DENSITY M1 < 0.4 WINDOW 100 100 STEP 10  
EXT X < 50  
}
```

The waiver_flow command generates the waiver shapes from the [External](#) operation. You would expect these waiver shapes to abut layer X shapes, but they are not inside layer X. If you use just WAIVE_DENSITY for this rule, there probably will be no waivers applied in the calibre -waiver run because there is no interaction between the waiver shapes and the Density windows. You may prefer to use a standard waiver statement, a WAIVE_MARKER statement, or a WAIVE_EXTENT statement in conjunction with WAIVE_DENSITY.

Here is another case:

```
Check3{  
X = DENSITY M1 < 0.4 WINDOW 100 100 STEP 10  
X AND (EXTENT NOT RAM)  
}
```

In this case, the waiver_flow output waiver shapes will be a subset of the data from layer X. Using WAIVE_DENSITY for this rule will likely cause the waivers to be applied by calibre -waiver in matched locations.

As a general guideline, if downstream layer operations reduce the Density output to a subset of shapes that Density generates, then WAIVE_DENSITY results will likely be acceptable. If downstream operations go outside of the Density shapes, then there is risk of unexpected results. If you have such rules, you must decide how to manage the waivers: use a standard waiver statement, or use WAIVE_MARKER or WAIVE_EXTENT in combination with your WAIVE_DENSITY statement.

See “[Waiver Criteria File Description](#)” on page 144 for additional details.

Examples

```
dens_1 WAIVE_DENSITY 99
```

P2P

Waiver criteria file statement.

Specifies that Calibre PERC LDL P2P resistance waivers are applied based upon a tolerance range around the waived resistance value.

Usage

P2P:*experiment_name* **P2P** [*tolerance* | {*tolerance_above* *tolerance_below*}]

Arguments

- *experiment_name*
A required name of an experiment defined by [perc_ldl::design_p2p_experiment](#). The “*” wildcard is supported and matches zero or more characters.
- **P2P**
A required keyword.
- *tolerance*
An optional floating-point number between 0 and 100 interpreted as a percentage. For a waived resistance value R , $R \pm R * tolerance/100$ constitutes the waived values. The default is 10. May not be specified with other tolerance arguments.
- *tolerance_above* *tolerance_below*
Optional floating-point numbers between 0 and 100 interpreted as percentages. For a waived resistance R , waived values are on the closed interval $[R - R * tolerance_below/100, R + R * tolerance_above/100]$. The default is 10 10. May not be specified with *tolerance*.

Description

The P2P statement only applies to LDL P2P resistance waivers. The default tolerance is 10%, which allows minor changes in the extracted parasitic resistance without affecting the status of the waiver. The layout path between a source and a sink may be altered (without altering the source and sink locations) and the waiver may still apply if the parasitic resistance is within the tolerance. This statement does not apply to waiver_flow.

To allow simplified application of waivers from Calibre RVE, you can define P2P tolerances in a waiver criteria file. You can load the waiver criteria file in the **Export Waived Results** dialog box’s **Criteria > Input waiver criteria file(s)** field before exporting waivers, and the annotations in your configuration will be available for extraction during a verification run. You can then use the **WAIVER_CRITERIA EXTRACT** statement in your setup file referenced by the [PERC LDL Waiver Path](#) statement in the rules, and your tolerances will be applied in an LDL run.

See “[LDL Geometry Waiver Application](#)” in the *Calibre PERC User’s Manual* for related information.

Examples

This statement waives P2P resistances from 0 ohms to 110% of a waived resistance for experiment rule_1:

```
P2P:rule_1 P2P 10 100
```

Waiver Cells File Format

Input for: waiver_flow

Specifies the rules that you want to waive by cell when generating waiver shapes with the waiver_flow tool.

Format

The waiver cells file must conform to the following conditions.

- For each cell you want to apply waivers to, specify the rules to be waived. Each cell can only be specified once. The waiver cells file takes the following format:
 - *cell rule* [*rule ...*]
- IP_MATCH description rules are mentioned in this file with the following format:
 - *cell IP_MATCH rule* [*rule ...*]
- WAIVE_EXTENT, WAIVE_PATTERN, and WAIVE_MARKER keywords should not be specified in this file. Cells that these keywords apply to are not specified in this file.
- This file is not case-sensitive, except for cell and rule names. Cell and rule name parameters must match the corresponding case of the layout cell and rule check names.

Parameters

- *cell*

The name of a design cell to which the waiver applies. A *cell* may only be specified once. The asterisk (*) wildcard is allowed for all waiver definitions that support the *cell* parameter. The * matches zero or more characters.
- *rule*

The name of a rule to which the waiver applies. A *rule* may be specified for multiple *cell* and **IP_MATCH** statements. The asterisk (*) wildcard matches zero or more characters.
- **IP_MATCH**

Specifies to generate IP_MATCH signatures during a waiver_flow run. One or more *cell* parameters must be specified. The *cell* is a cell name in the INPUT_LIBRARY design. The * wildcard may be used to match zero or more characters.

In waiver_flow, **IP_MATCH** generates text object annotations (checksum signatures) on texttype 6 in the WAIVER_DATABASE (see “[Waiver Setup File Format for Waiver Generation](#)” on page 117 for setup file details). These objects appear in intermediate hierarchy cells that correspond to the specified *cell* parameters. When generating waiver shapes, waiver geometry must be generated in order for **IP_MATCH** to apply. Otherwise, a warning is issued and no **IP_MATCH** signature is generated for the *cell*.

If **IP_MATCH** is specified, then the MERGE YES statement in the waiver verification setup file may not be specified. The WAIVER_DATABASE statement, or its default, is used.

During a calibre -waiver run where RUN_IP_MATCH YES is specified in the waiver setup file, any cell in the input design that matches the signature checksum of an IP_MATCH cell has the IP_MATCH cell waivers applied to it. The matching of a cell's signature is independent of the cell's name. Note that **IP_MATCH** does not need to be in the waiver cells file for a calibre -waiver run.

See “[IP_MATCH Waivers](#)” on page 175 for additional details.

Waiver Cells File Description

The waiver cells file is used by the waiver_flow tool to generate waiver shapes for specified cells. It is used in Calibre RVE to specify **IP_MATCH** waivers during waiver export. It is ignored in calibre -waiver runs.

The **cell** parameters are expected to match cells in the input design. If they do not, a warning is generated. Because of how Calibre processes hierarchy, it is best to pick the lowest possible cell in the hierarchy to waive errors.

The **rule** parameters are expected to match rule check names in the rule file. If they do not, a warning is issued.

The asterisk character (*) may be used in rule check names and design cell names in the waiver cells file. All unique pairings of rule names to check names are matched. Matching of names is case-sensitive. If there are multiple matches that could apply due to wildcards, one match is chosen, and a warning is issued.

An exact match of a rule name and check name takes priority over a wildcard match. For example, suppose these lines are in the waiver cells file:

```
nand2 rule1 top_metal_rules
nand* space_rule* *poly*
```

If the first statement exactly matches nand2, then the waivers for the specified rules are applied to that cell. A warning is issued because the wildcards in the second line could also match these in the same cell, but the rules are different in the second line. Name matching is done in this order of priority:

1. Exact match of cell name and rule name
2. Exact match of cell name, longest wildcard expression match of rule name
3. Longest wildcard expression match of cell name, exact match of rule name
4. Longest wildcard expression matches of cell name and rule name


Suppose you have these lines in the waiver cells file (line numbers are for illustration only):

```
1  nand* rule*
2  nand* rule1*
```

For the cell nand2 and rule1, rule10, rule11, and so forth, line 2 is used because that is the longest rule name match. For rule2, rule20, rule 21, and so forth, line 1 is used. Warnings are issued in both cases because multiple differing waiver criteria could be applied due to the possible wildcard matches.

The **rule** and **cell** parameters are used to generate waiver cell names as discussed under “[Waiver Cell Description](#)” on page 218.

Note


 WAIVE_EXTENT, WAIVE_PATTERN, and WAIVE_MARKER keywords must only be specified in the waiver criteria file. Cells that these keywords apply to are not specified in the waiver cells file.

IP_MATCH Waivers

IP_MATCH is used when there are cells (call this set of cells C) for which you want to apply C’s waivers to any other cells in the design that geometrically match cells in C. The cells that geometrically match C may or may not be known. For example, if you want waivers for a given standard cell to apply to any other cell in the design that is the same (the cells’ names are not considered in the matching process) as the standard cell, then the **IP_MATCH** statement is useful.

To implement **IP_MATCH** cell matching, Calibre generates a cell signature checksum text object. This text object contains information (a signature) about a cell’s configuration that enables matching of some other cell to that signature’s characteristics. The signature text object is stored on texttype 6 in a waiver cell in the WAIVER_DATABASE. See “[IP_MATCH Checksum Annotations](#)” on page 226 for more information.

Caution

 The cells in the INPUT_LIBRARY for which you want to generate IP_MATCH checksums must not contain any existing waiver cells. The waiver cells are considered when the checksums are generated.

In the calibre -waiver run, cell signatures are used when RUN_IP_MATCH YES is specified in the setup file, which it is by default.

The following discussion assumes MERGE YES is not specified in the waiver verification setup file.

When using `waiver_flow`, the signature text objects are generated for cells that are matched by the **IP_MATCH** commands. For example, this set of statements:

```
cell1 IP_MATCH ruleA
inv*_x IP_MATCH ruleB
```

generates signature text objects for `cell1` and `inv1_1x`. The text objects appear in waiver cells in the `WAIVER_DATABASE` that correspond to the `INPUT_LIBRARY` cells.

When running `calibre -waiver`, the **IP_MATCH** cell signatures are read from the `WAIVER_DATABASE`, and the layout is scanned for cells that match the signatures. This match does not consider cell names, only geometry and text. For cells in the layout that match the specified **IP_MATCH** cell signatures, the **IP_MATCH** cell waivers are applied.

Using the results of the previous example above, all cells in the layout that match the signatures of `cell1` and `inv1_1x` will have the specified waivers applied.

If **MERGE YES** is specified in your waiver verification setup file, then it is anticipated waiver shapes are embedded in the main design. If you have two otherwise identical cells, one with embedded waivers and one without, then **IP_MATCH** can cause waivers that are applied to the former cell to be applied to the latter cell also. But these cells are not truly identical, because only one contains the waiver shapes. For this reason, **MERGE YES** and **IP_MATCH** are incompatible.

The following example is a small waiver cells file.

```
cellabc rule1 check22 po.r.2 m1*
celld* rule5 metal_rules
cell_inv IP_MATCH rule1 rule2 rule3*
```

The first line specifies that verification results for `rule1`, `check22`, `po.r.2`, and any rule beginning with the string, `m1`, should be waived for `cellabc`. The second line indicates that results from `rule5` and `metal_rules` can be waived for any cell beginning with the string, `celld`. The third line specifies that any cell that geometrically matches `cell_inv` can waive results from `rule1`, `rule2`, and rules beginning with the string, `rule3`.

Related Topics

[Waiver Setup File](#)

[Waiver Description Files](#)

Batch Mode File Usage

The waiver criteria and the waiver cells file are used differently depending on whether you are generating or applying waivers.

For calibre -waiver

- A waiver cells file is not required. The WAIVER_CELLS statement is ignored in the waiver setup file.
- One or more waiver criteria files are required. The WAIVER_CRITERIA statement must appear at least once in the waiver setup file.

Alternatively, you can extract waiver criteria from the waiver shapes, if the cells are annotated with criteria text. In this case, you must use the WAIVER_CRITERIA EXTRACT statement.

- The EXTRACT and APPEND setup file keywords only apply to the files specified by the WAIVER_CRITERIA statement.

For waiver_flow

- One or more waiver cells files are required. The WAIVER_CELLS statement must appear at least once in the waiver setup file. Calibre performs verification runs on each cell specified in the waiver cells files.
- Any matching results from listed rules are exported as waiver geometry.
- The waiver criteria values are used from the specified WAIVER_CRITERIA files. The criteria values are exported as waiver text to a GDS or OASIS database.
- If WAIVER_CRITERIA is not specified in the waiver setup file, a default value of SINGLE 100 100 is applied to all rules.

Related Topics

[Waiver Setup File](#)


[Waiver Description Files](#)

Chapter 7

Running a Physical Verification with Waiver Shapes

In order to correctly match and waive allowed error results, you must set up and run a physical verification with waivers.

Tip

 For a high-level Calibre Auto-Waivers flow overview, see “[Getting Started With Calibre Auto-Waivers](#)” on page 33.

Physical Verification with Waivers	180
Performing a Waiver Run.....	184
Performing a DRC Waiver Run Using Multiple Waiver Criteria Files.....	189
Using Rule Checksums in calibre -waiver.....	191
Differences in Generated SVRF Rules When Running TVF Rules with Waivers	193

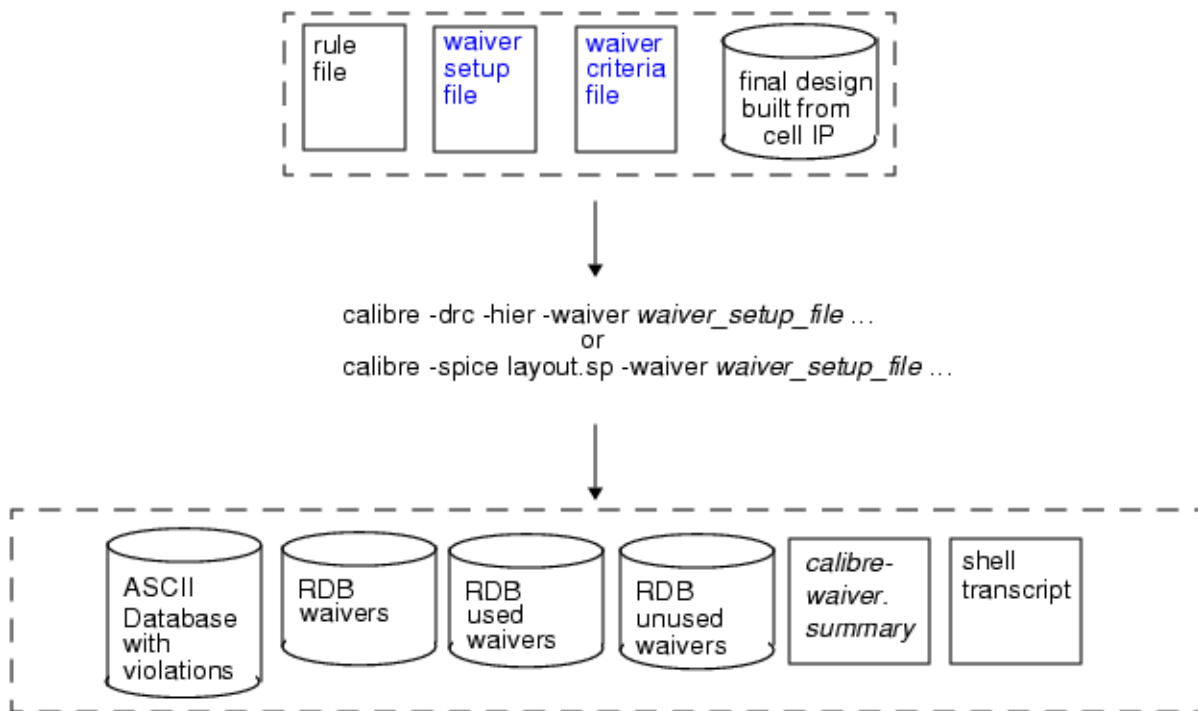
Physical Verification with Waivers

The verification flow for DRC, DFM, and ERC waivers is similar. The calibre -waiver option causes waivers to be applied for which there are waiver shapes (either in the input design or merged with it at runtime) that correspond to the waiver criteria file and waiver verification setup file specifications.

The only exceptions for the necessity of waiver shapes being present are the WAIVE_DENSITY, WAIVE_EXTENT, and WAIVE_PATTERN specifications in the waiver criteria file.

The following figure shows the inputs and outputs for this portion of the DRC or ERC flow:

Figure 7-1. I/O for Waiver Verification in DRC and ERC



The geometric output results of the Calibre run are in these categories:

- Violations that are not waived (ASCII Results Database format)
- Waived violations (*waived.rdb* in ASCII RDB format)
- Successfully applied waiver shapes (*used_waivers.rdb* in ASCII RDB format)
- Unused waivers (*unused_waiver.rdb* in ASCII RDB format)
- Density waivers (*density_waived.rdb* in ASCII RDB format; output only if WAIVE_DENSITY is used)

Unused waivers are violations that might be waived, but have not been because of how the waiver criteria files are written.

The *calibre-waiver.summary* is the summary report file for the run.

In a Calibre DFM waiver run (*calibre -dfm -hier -waiver*), a DFM Database is produced. The RDB files shown in [Figure 7-1](#) are not produced.

Writing the Waiver Setup File for Waiver Verification	181
Writing the Waiver Criteria File for Waiver Verification.....	183

Writing the Waiver Setup File for Waiver Verification

Waiver setup files are generally written manually in a text editor, but you can also create and edit them with Calibre Interactive.

See “[Starting a Waiver Run in Calibre Interactive](#)” in the *Calibre Interactive User’s Manual* for complete information.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Knowledge of “[Writing the Waiver Criteria File for Waiver Verification](#)” and “[Waiver Setup File Format for Waiver Verification](#).”

Procedure

1. (Optional) If you want to reproduce the waiver descriptions that were used to produce the waiver geometry, then enter this line in the waiver setup file:

```
WAIVER_CRITERIA EXTRACT <pathname>
```

The waiver criteria file is extracted from text object annotations in the waiver cells and written to the pathname.

Performing this step assumes the waiver geometry has text object annotations that include the waiver criteria (see “[Text Object Annotations](#)”). It also assumes that no separate waiver criteria file is to be used.

2. Use the following file as a template to create your own setup file.

```
WAIVER_CRITERIA      <path to waiver_criteria_file,
                      or use EXTRACT option>
LAYER_NUMBER         <optional number not in INPUT_LIBRARY;
                      default 1234>
DATATYPE_NUMBER      <optional datatype number; default 5678>
CALIBRE_WAIVER_NUMBER AUTO
RUN_LAYOUT_CHECKSUM  <YES or NO>
```

- The LAYER_NUMBER and DATATYPE_NUMBER combination should not be used in the design specified in the [Layout Path](#) statement of the rule file. The combination should also not be used in the rule file for layer mapping or mask results output.
- RUN_LAYOUT_CHECKSUM specifies whether a checksum is performed on the waiver cells during a calibre -waiver run. When a checksum is performed, a warning is issued if the waiver cell has been altered from its original configuration. The default for this setting is YES. If you do not want layout checksum validation, you should specify NO. See “[Text Object Annotations](#)” on page 221 for details.

Note



For advanced nodes, if your rule file precision and input layout precision is different, specify PRECISION_CONVERSION YES in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

3. If all of your waiver shapes are embedded in your Layout Path file(s), then specify this:

```
MERGE YES
```

If *at least some* of your waiver shapes are external to the Layout Path file, then specify this:

```
MERGE NO  
WAIVER_DATABASE <external_waiver_file>
```

or do not add statements to the waiver verification setup file. In the latter case, you are opting to accept the default external waiver geometry filename, which is *waived.gds*. You can specify more than one WAIVER_DATABASE statement if there is more than one external file with waiver shapes in it.

4. If you are an IP provider, give the waiver verification setup file to your customers.

Tip



When waivers generated from edge clusters are applied in a verification run, waived edge outputs are expanded into polygons and waived error clusters are converted into polygons in the results databases. If you specify RETAIN_RESULT_TYPE YES in the waiver verification setup file, the original edge clusters are written to the DRC results database and the *waived.rdb* database.

Related Topics

[Physical Verification with Waivers](#)

[Writing the Waiver Setup File for Waiver Generation](#)

Writing the Waiver Criteria File for Waiver Verification

You can specify custom criteria for waiver shapes that are applied during verification if the default values do not meet your requirements. The tool writes all waiver criteria applied to rule checks as a DRC Check Comment. These comments are visible in Calibre RVE.

If your waiver shapes contain embedded criteria text, or the default criteria of SINGLE 100 100 meets your requirements, proceed to “[Writing the Waiver Setup File for Waiver Verification](#)” on page 181.

In order to apply *non-default* criteria text to waiver shapes that do not contain embedded waiver criteria text objects, you must write a waiver criteria file for calibre -waiver. Use the `WAIVER_CRITERIA pathname` statement to specify a path to your criteria file in the waiver setup file.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Embedded or non-embedded waiver shapes generated by Calibre RVE or waiver_flow.
- Knowledge of “[Waiver Shape Interactions](#)”, and “[Waiver Criteria File Format.](#)”

Procedure

1. Examine your DRC results to determine the criteria you want to apply to waiveable rules. If waivers are not applied as expected, the default global criteria of SINGLE 100 100 may be too strict.
2. Write the waiver criteria in a text file. Each rule may only be listed once and only one primary criteria can be attached to a rule.

Here is an example:

```
## standard form criteria ##
m2_width      SINGLE 98 98
m2_space      MULTI 100 100

## waiveable rule without criteria (global default is applied)##
m1_space
```

The first line means only single waiver interactions are used for waivers applied to the m2_width rule. Both overlap criteria specify at least a 98 percent overlap (this is a good value to begin testing). The m1_space rule does not have any specified criteria, so the default of SINGLE 100 100 is applied to this rule and a warning is issued.

3. If you are waiving [Density](#) results windows inside the extent of a given cell, you can include waiver criteria such as the following:

```
density_rule WAIVE_DENSITY 100 WAIVE_EXTENT cell
```

If you are waiving Density results inside of a marker layer, you can use waiver criteria such as the following:

```
density_rule WAIVE_DENSITY 100 WAIVE_MARKER marker_layer
```

4. If you are an IP library provider, you give this file to your customers.

Related Topics

[Waiver Criteria File Format](#)

[Waiver Setup File Format for Waiver Verification](#)

[Writing the Waiver Setup File for Waiver Verification](#)

[Performing a DRC Waiver Run Using Multiple Waiver Criteria Files](#)

Performing a Waiver Run


Waiver shapes are applied to error results in a Calibre verification run.

If you are using IP from a vendor that provided you with a waiver verification setup file and a waiver criteria file, then use those files along with the foundry rule file in a `calibre -waiver run`. Some vendors may not provide the waiver criteria if they have embedded the waiver criteria into the IP using text objects. See “[Text Object Annotations](#)” on page 221 for details.

If you are using IP from different vendors, and they each provide you with waiver verification setup files and waiver criteria files, then you perform the steps listed under “[Performing a DRC Waiver Run Using Multiple Waiver Criteria Files](#)” on page 189.

If using ERC waivers, frequently, the intent is to apply the waivers when the blocks are ERC verified outside of the context of the complete chip design.

Note


 Be aware that if you run `waiver_flow` for DRC and ERC waivers in the same directory, the later run will overwrite files of the same name from the earlier run. For this reason, it is best to do DRC and ERC waiver generation in different directories.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Waiver shapes are either embedded in the design or maintained in separate files.
- No unsupported layer operations or rule file constructs are used for waivers. See “[Unsupported Layer Operations in `waiver_flow`](#)” on page 29.
- If using the ERC automatic waivers, a block design that has waiver shapes embedded in it, or a larger design comprised of such blocks.

- Waiver verification setup file. See “[Writing the Waiver Setup File for Waiver Verification](#)” and “[Waiver Setup File Format for Waiver Verification](#).”
- Waiver criteria file. See “[Writing the Waiver Criteria File for Waiver Verification](#).” Alternatively, a design that has waiver criteria embedded in it and a waiver verification setup file that contains a `WAIVER_CRITERIA` statement using the `EXTRACT pathname` option.
- If your waiver geometry output from Calibre RVE is produced from edge clusters from ENClosure, EXternal, or INTernal layer operations, and those edge clusters could form non-Manhattan polygons, then it is recommended to change the corresponding layer operations in your rule file to use the `REGION` keyword before running Calibre with waivers.

Note

 For advanced nodes, if your rule file precision and input layout precision is different, specify `PRECISION_CONVERSION YES` in the waiver setup file (this is the default behavior). Otherwise, you may generate unintended results.

Procedure

1. Do one of the following:

- If *all* of your waiver shapes are embedded in your Layout Path design(s), ensure the waiver verification setup file contains this statement:

```
MERGE YES
```


Use `MERGE NO` (the default) if there are any non-embedded waiver geometry files.

- If at least some of your waiver shapes are not embedded in your Layout Path design(s), either specify `MERGE NO` or accept the default of `NO`. Ensure that there is a statement like this:

```
WAIVER_DATABASE <path_to_non_embedded_waiver_file>
```

for every external waiver geometry file that you have.

Note

 To provide information in the results about the cell in which a waiver was applied, include the `ENABLE_WAIVER_CELL_REPORTING YES` statement in your waiver setup file before a verification run with waivers. This option creates a `WAIVER_CELL` column in the results database.

2. Execute one of the following commands:

```
calibre -drc -hier -waiver waiver_setup_file [DRC_options] rules
calibre -dfm -hier -waiver waiver_setup_file [DFM_options] rules
calibre -spice layout.sp -waiver waiver_setup_file [LVS_options]
rules
```

3. The **rules** file should be a tapeout rule file. It overrides the rule file specified in the **waiver_setup_file**. The **-lvs -hier** options can be used without **-spice** if there is a [Mask SVDB Directory](#) statement in the rule file.

Results

These results files are produced for DRC and ERC runs:

- An ASCII results database that contains design rule errors.
- A *waived.rdb* file that contains all the original DRC results that have been waived.
- A *used_waivers.rdb* file that shows the waiver shapes that were successfully applied during the run.
- An *unused_waiver.rdb* file that shows waivers that were candidates for being used, but were not applied.
- A *calibre-waiver.summary* file containing runtime messages and results statistics.
- If [DRC Summary Report](#) or [ERC Summary Report](#) YES is specified in the rule file, the corresponding report is generated. Using these statements causes the *calibre-waiver.summary* file to report waivers per rule check.

The end of the transcript includes information about the run time. For example:

```
--- DFM CPU TIME = 0 + 3 REAL TIME = 1 LVHEAP = 8/14/14 SHARED = 0/1 MALLOC = 63/63/63
    ELAPSED TIME = 5
--- YIELD SERVER CPU TIME = 7 REAL TIME = 19 LVHEAP = 1153/1170/1170 MALLOC = 432/432/472
    ELAPSED TIME = 24
--- TOTAL RULECHECKS EXECUTED = 60
--- TOTAL RESULTS GENERATED = 0 (0)
--- TOTAL DFM RDB RESULTS GENERATED = 213845 (286887)
--- DRC RESULTS DATABASE FILE = error.db (ASCII)
```

Where the CPU times are calculated as follows:

```
DFM CPU TIME = <DFM_CPU_time for local machine> + <DFM_CPU_time for remotes>
YIELD SERVER CPU TIME = <YS_CPU_time for local machine>
Total cpu time = <DFM_CPU_time for local machine> + <YS_CPU_time for local machine>
```

The YieldServer time for remote CPUs is not included.

To evaluate DRC or ERC results, do the following:

- Review the *calibre-waiver.summary* report and the DRC or ERC Summary Report (if specified) to see the runtime messages. If WARNING messages are present, be sure you understand what they mean. See [“Calibre Auto-Waivers Runtime Messages”](#) for more details.
- Open your design in a layout viewer and the ASCII results database in Calibre RVE for DRC and review the errors.
- Open the *waived.rdb* file in Calibre RVE. The results open in a new tab.

- Choose **View > Results Options > Details**.
- The results open in a tabular form with various properties at the top of each column. Which columns you see depends upon the settings in the waiver criteria file for the current rule check.
- Sort the columns by clicking the headers. This can be useful when looking for outliers among the waived results. The properties are based upon the number of valid waiver shapes (waivers that meet the specified criteria in the waiver criteria file). Here are the column header descriptions:

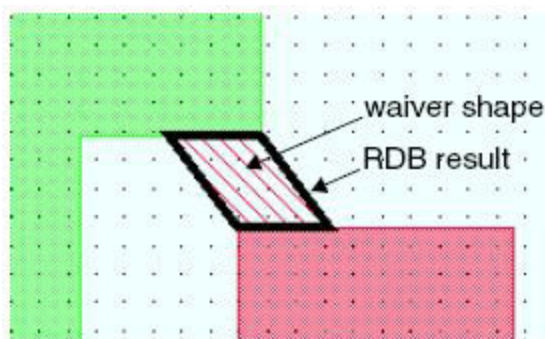
Table 7-1. Calibre RVE waived.rdb Properties

Property Name	Description
max_result_cvg	Maximum percentage of any result region's area intersected by waiver shape area. This is reported for MULTI results. These values are rounded in the RDB, so they may not correspond exactly to what occurs in the layout.
max_waiver_cvg	Maximum percentage of total waiver shape area intersected by the result. This is reported for MULTI results. These values are rounded in the RDB, so they may not correspond exactly to what occurs in the layout.
result_cvg	Percentage of the result area intersected by the valid waiver shapes. These values are rounded in the RDB, so they may not correspond exactly to what occurs in the layout.
waiver_cvg	Percentage of the valid waiver shape's area intersected by the result. This is reported for SINGLE results. These values are rounded in the RDB, so they may not correspond exactly to what occurs in the layout.
waiver_count	Number of valid waiver shapes that intersect the result region. This number is 1 if SINGLE is specified.
Coordinates	Number of vertices of a waiver shape, followed by a series of x,y coordinates.
WAIVER_COMMENT	If a user comment was applied when the waiver geometry was exported in Calibre RVE, then this property shows the user comment of the associated text object.

Table 7-1. Calibre RVE waived.rdb Properties (cont.)

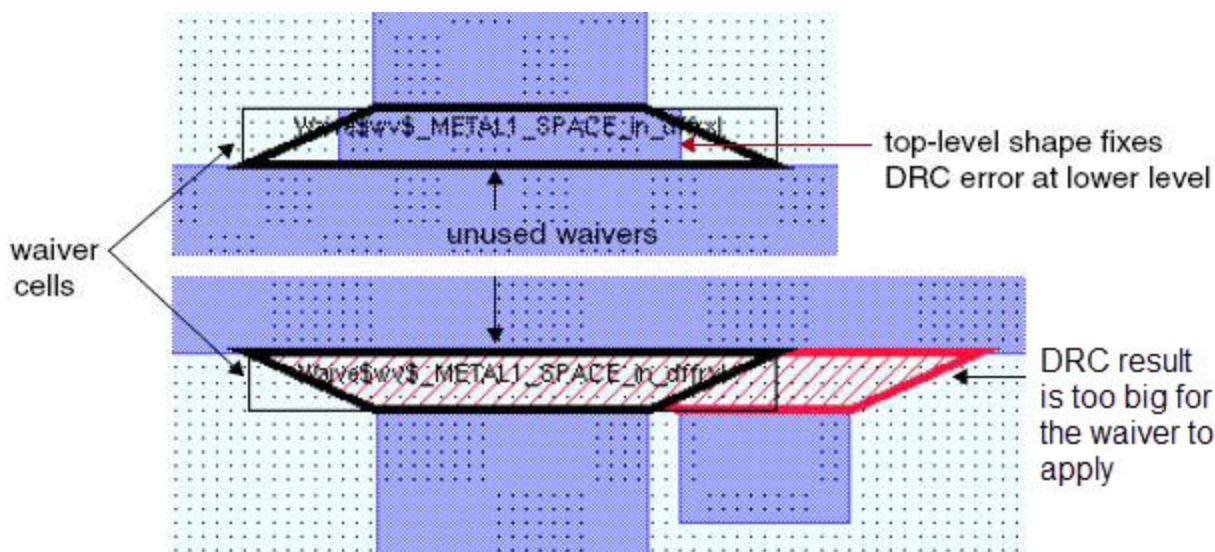
Property Name	Description
WAIVER_DATE	If a waiver comment was applied when the waiver geometry was exported in Calibre RVE, then this property shows the date of the associated text object.
WAIVER_USERNAME	If the waiver comment was applied when the waiver geometry was exported from Calibre RVE, then this property shows the username of the associated text object.

- Scan the waived results to see if they meet expectations. Here is an example of a waiver shape and RDB result that indicates a waived error:



Check the waivers thoroughly, especially after a tapeout run.

- Open the *unused_waiver.rdb* file in Calibre RVE. Scan the results to determine why the waivers were not applied in these cases. This is done by comparing the relevant entry in the waiver criteria file to the geometry associated with the waiver. Here are examples:



There are three criteria that cause an unused waiver:

- The waiver overlap criterion was not met
- The results overlap criterion was not met
- The SINGLE waiver type is specified and there are multiple waiver shapes interacting a result region

Consult with the foundry if you have questions about the status of any unused waivers you think might be waiveable.

- If there are design errors, fix them, run calibre -waiver again and then repeat the Results task validation steps.

These results are produced from a Calibre DFM run:

- DFM Database
- A *calibre-waiver.summary* file containing runtime messages and results statistics.

Waivers are applied automatically. When you highlight Chip Summary tab results in Calibre RVE (**View > Browse Hierarchy Errors**), the waived results do not appear in the list.

Related Topics

[Waiver Shape Interaction Types](#)

[Waiver Shape Generation with Calibre RVE](#)

[Calibre Auto-Waivers Runtime Messages](#)

Performing a DRC Waiver Run Using Multiple Waiver Criteria Files

Use multiple waiver-corrected IP libraries, where the IP vendors each provide you with waiver setup and waiver criteria files. Most likely, the IP libraries come from a foundry or a non-foundry vendor.

Prerequisites

- See “[Requirements for Running Calibre Auto-Waivers](#)” on page 23.
- Completed design using waiver-corrected libraries.
- Waiver verification setup files from each IP vendor.
- Knowledge of the “[Waiver Criteria File Format](#).” and “[Writing the Waiver Criteria File for Waiver Verification](#).”
- Waiver criteria files from each IP vendor.

Procedure

1. Do one of the following:
 - If the waiver verification setup files have LAYER_NUMBER and DATATYPE_NUMBER settings that differ, then the waiver shapes will be on different layers. Calibre Auto-Waivers does not work if the layers and data types are not consistent for waiver shapes. If there is an inconsistency, then do the following:
 - i. Choose one of the waiver verification setup files and use its LAYER_NUMBER and DATATYPE_NUMBER settings for *all* waiver shapes in *all libraries*. Edit the libraries as appropriate to guarantee consistency.
 - ii. Build your design with the libraries, where consistent waiver layer and datatype numbers are used for all of the libraries.
 - iii. If your design is already built, then change the layer and datatype numbers for waiver shapes so that they match one of your waiver verification setup files.
 - If the IP libraries contain identical cell names, then do the following:
 - i. Rename the identically-named cells so they no longer match any other cell name.
 - ii. For waiver cells having the form Waive\$wv\$_rulecheck_in_cellname, where you have changed the name of the parent cell, change the *cellname* string to match the parent cell.
 - iii. If you needed to rename cells, then make corresponding changes to cell names in the waiver cells files.
2. Compare the waiver criteria files. Adjust the criteria as necessary.
3. If the foundry waiver criteria for a given rule and cell name differ from another IP vendor's, choose the foundry settings.
4. Concatenate the waiver criteria files. Remove any blank lines.
5. Choose the waiver verification setup file that matches the layer and datatype numbers you chose for waivers in your design. Ensure the WAIVER_CRITERIA setting refers to your concatenated waiver criteria file. Ensure CALIBRE_WAIVER_NUMBER is set to AUTO.
6. Execute the following command:

```
calibre -drc -hier -waiver waiver_setup_file [usual_drc_options]  
rules
```

Results

These results files are produced:

- A DRC results database that contains DRC errors.
- A *waived.rdb* file that contains all the original DRC results that have been waived.

- A *used_waivers.rdb* file that shows the waiver shapes that were successfully applied during the run. This will be identical to the *waived.rdb* file if SINGLE 100 100 criteria was used.
- An *unused_waiver.rdb* file that shows waivers that were candidates for being used, but were not applied.

Follow the steps under “Results” in “[Performing a Waiver Run](#)” on page 184 to check your output.

Related Topics

[Waiver Shape Generation with Calibre RVE](#)

[Waiver Shape Interaction Types](#)

[Calibre Auto-Waivers Runtime Messages](#)

Using Rule Checksums in calibre -waiver

Apply calibre -waiver to verify rule checksums from waiver cells in a physical verification run. Rule checksums ensure that the rule code used to generate waiver cells matches the rule code used in the verification run.

Note



To use rule checksums, you must enable the legacy Calibre Auto-Waivers engine by setting the CALIBRE_WAIVER_OLD_ENGINE environment variable. See “[Calibre Environment Variables](#)” on page 25.

Note



Rule checksums are generated from the rule filename and the rule code associated with each rule. If a rule name is changed from waiver generation to waiver verification, the waiver is still applied if the rule code remains unchanged.

For information on enabling rule checksums in Calibre Interactive, refer to “[Generating Waiver Shapes with Calibre Interactive](#)” in the *Calibre Interactive User’s Manual*.

Prerequisites

- Waiver shapes with embedded rule checksum text generated in a prior run. Refer to the procedures under the section “[Waiver Shape Generation Flow](#)” on page 74 to generate rule checksums with Calibre RVE. Refer to “[Creating Waiver Cell Rule Checksum Annotations with waiver_flow](#)” on page 109 for information on creating rule checksums in *waiver_flow*.
- A supported layout viewer.

Procedure

1. Specify the layout database containing the waiver shapes using one of the following procedures.

- If you have non-embedded waiver geometry files that are external to the main design, then ensure statements like this appear in the waiver setup file:

```
MERGE NO  
WAIVER_DATABASE external_file1.gds  
WAIVER_DATABASE external_file2.gds
```

- If all of your waiver geometry is embedded in the Layout Path design(s), then include this in the waiver setup file:

```
MERGE YES
```

2. Set the desired checksum settings and ensure that the RUN_RULE_CHECKSUM YES statement is included in the waiver setup file:

```
RULE_CHECKSUM_TEXT_LAYER 1234  
RULE_CHECKSUM_TEXTTYPE 9  
RUN_RULE_CHECKSUM YES
```

3. Run the following command:

```
calibre -drc -hier -waiver waiver_setup_file [drc_options] rules
```

Results

During the calibre -waiver run, rule checksums are read from waiver shape and compared to the verification rule file. The rule that generated the waiver shape has its checksum generated during the run. If the two checksums do not match, the waiver is not applied and a warning message is sent to the transcript.

Related Topics

[Creating Waiver Cell Rule Checksum Annotations with waiver_flow](#)

[Waiver Setup File Format for Waiver Generation](#)

[Waiver Setup File Format for Waiver Verification](#)

[Generating Waivers with the waiver_flow Tool](#)

[Waiver Shape Generation Flow](#)

[Rule Checksum Annotations](#)

Differences in Generated SVRF Rules When Running TVF Rules with Waivers

If you perform a verification run with waivers and you choose to generate an SVRF file from the TVF input with the -E option, then the tool adds statements to the generated SVRF file.

For example, the following command performs a waiver verification run using a TVF rules file named *calibre.tvf* file. The command also creates the *generated_rules.svrf* file from the TVF input.

```
calibre -drc -hier -turbo -E generated_rules.svrf \  
-waiver waiver_setup_file rules.tvf
```

Because Calibre Auto-Waivers performs internal DFM operations during waiver verification, additional statements are included at the beginning of the *generated_rules.svrf* file that are not present in the original TVF rules, such as the following:

```
DFM SELECT CHECK L1.Width.Check  
DFM UNSELECT CHECK COPY_L1  
DRC ICSTATION YES
```

Similarly, additional lines are appended to the end of the *generated_rules.svrf* file, such as the following:

```
DFM SUMMARY REPORT "output.rep"  
DFM DATABASE waiver.dfmdb OVERWRITE REVISIONS [ORIGINAL]
```

If you generate an SVRF file from the TVF input before running with waivers, then the generated SVRF file does not include the additional statements mentioned above. For example:

```
calibre -E generated_rules.svrf rules.tvf  
calibre -drc -hier -turbo -waiver waiver_setup_file generated_rules.svrf
```


Chapter 8

Calibre Auto-Waivers Utilities

Calibre Auto-Waivers includes several utilities to help manage your waiver flow.

drc_waiv_asc2gds.....	196
checksum_util.....	203
waiver_util	208

drc_waiv_asc2gds

Batch ASCII DRC results database and Calibre RVE *.waived* file translator.

Access from: `$CALIBRE_HOME/bin/drc_waiv_asc2gds`

Translates an ASCII DRC results database or a Calibre RVE *.waived* file to a GDS or OASIS waiver database suitable for use in Calibre Auto-Waivers.

Note



As of 2019.2, waiver shape edges are always expanded by two DBU. If your waiver database was generated in 2019.1 and earlier, you may need to regenerate your waiver databases using 2019.2 and newer in order to ensure that waivers are correctly applied.

Usage

```
drc_waiv_asc2gds {-i {input_results_database | input_directory}...} -o output_database
  [-d datatype]
  [-c waiver_cells_file ... [-e waiver_criteria_file...]] | [-f waiver_description_file ...]
  [-ip IP_MATCH_input_file]
  [-l layer]
  [-p database_precision]
  [-rp rule_file_precision]
  {[-cm comment_file] | [-r]}
  [-s setup_file]
  [-t {gdsii | oasis}]
```

Arguments

- **-i {*input_results_database* | *input_directory* } ...**

Required argument set that specifies the pathname to a file or directory of files. The target files must be ASCII DRC results databases or Calibre RVE *.waived* files. You can specify multiple files or directories, but all the files must be of the same type (either *.waived* or results database files).

If you specify *.waived* files, then the *-r* option must also be used. If you specify more than one file or directory, the path must be preceded by *-i*.

- **-o *output_database***

Required argument set that specifies the pathname of an output waiver database file.

- **-d *datatype***

Optional argument set that specifies the datatype of the waiver layer output, where *datatype* is a non-negative integer. The default is 5678. This option overrides the setup file if *-s* is used.

- **-c *waiver_cells_file***

Optional argument set that specifies the pathname of a waiver cells file. When this option is present, the specified file is parsed for cell names and *IP_MATCH* statements. This

information is used for generating output. More than one *waiver_cells_file* may be specified.

- *-e waiver_criteria_file*

Optional argument set that specifies a pathname of a waiver criteria file. When this option is present, the specified file is parsed for waiver criteria. More than one *waiver_criteria_file* may be specified. If a waiver criteria file is not specified, the default criteria of SINGLE 100 100 are used.

Note

This option is required if you specify a *.waived* file with *-r* and a *waiver_cells_file*.

- *-f waiver_description_file*

Optional argument set that specifies the pathname of a legacy waiver description file. This argument set is provided for backwards compatibility only. When this option is present, the specified file is parsed for waiver cells and criteria. This information is used for generating output. More than one *waiver_description_file* may be specified.

The waiver description file is deprecated as of the 2012.3 release. Use the *-c* and *-e* switches instead.

- *-ip IP_MATCH_input_file*

Optional argument set that specifies the pathname of an IP_MATCH input file. The cells in the IP_MATCH input file are used to generate IP_MATCH checksums in the waiver cells.

When you apply this argument set and the *drc_waiv_asc2gds* utility encounters a cell listed in the *waiver_cells_file* that contains the IP_MATCH statement, a hierarchical cell is created for the matched cell with an IP_MATCH checksum.

See “[Waiver Cells File Format](#)” on page 173 for a description of IP_MATCH.

- *-l layer*

Optional argument set that specifies the waiver layer number of the output, where *layer* is a non-negative integer. The default is 1234. This option overrides the setup file if *-s* is used.

- *-p database_precision*

Optional argument set that specifies the database precision of the output, where *database_precision* is a positive floating-point number. The precision is taken from the input database when this option is not used.

- *-rp rule_file_precision*

Optional argument set that specifies the precision of the rule file used to create the results, where *rule_file_precision* is a positive floating-point number. You must specify this option if the original rule file precision is different than the input results database or *.waived* precision. If you do not specify this option and the precisions are different, the tool can generate incorrect waivers for some coincident edges and small polygons.

- **-cm *comment_file***

Optional argument set that adds comments to generated waiver shapes. The comments include custom text loaded from a file, the current date, and the username. The *comment_file* argument specifies the path to a file that contains a list of rule names and associated comments. See “[Comment File Format for -cm](#)” on page 200 for details on the *comment_file*.

- **-r**

Optional argument that specifies the *input_results_database* is a Calibre RVE *.waived* file. If this argument is not supplied, then the *input_results_database* is expected to be a DRC results database.

You must specify a waiver criteria file with -e if you are translating a *.waived* file and you specify a *waiver_cells_file* with -c.

- **-s *setup_file***

Optional argument set that specifies the pathname of a waiver verification setup file. When this option is used, the waiver layer number, datatype, waiver annotation texttype, and checksum format settings (including defaults) are taken from the setup file. The *setup_file* settings for LAYER_NUMBER and DATATYPE_NUMBER can be overridden by -l and -d.

- **-t {*gdsii* | *oasis*}**

Optional argument set that specifies the layout format for the output. The default is *gdsii* if this option is absent.

Description

This command creates waiver geometry output that can be merged with your main design at runtime to enforce waivers. The *output_database* is a single waiver geometry file based upon the results in the *input_results_database*. The format is dependent upon which -t option is used.

The output layer is 1234 datatype 5678 by default. If the -s option is used, then the settings in the waiver verification setup file are used for layer, datatype, and texttype values. The -d and -l options take precedence over the -s option.

The following actions occur if the -c *waiver_cells_file* is not specified:

- Cell-specific waiver geometry output is generated for all cells in the *input_results_database*.
- The Waive\$wv\$_cell present in the waiver geometry file is annotated with text objects for checksum, waiver criteria (SINGLE 100 100 is used) and a user comment, if present. See “[Text Object Annotations](#)” on page 221 for details.

If the `-c waiver_cells_file` and `-e waiver_criteria_file` are specified, the following occurs for any cell name and rule check name combination present in the **input_results_database** that is also present in the *waiver_cells_file* and the *waiver_criteria_file*:

- Cell-specific waiver geometry output is generated for cells from the *waiver_cells_file* that are matched in the **input_results_database**.
- The Waive\$wv\$_cell present in the waiver geometry file is annotated with text objects for checksum, waiver criteria (taken from the matched line in the *waiver_criteria_file*) and a user comment, if present.

The precision of the output files is 1000 by default. If `-p database_precision` is specified and the *database_precision* does not match the precision of the **input_results_database**, then the output waiver shapes are magnified by this factor:

$$\text{database_precision} / \text{input_results_database precision}$$

The `-r` option specifies the **input_results_database** is a Calibre RVE .waived file instead of a results database.

If error-directed edge clusters are encountered in the ASCII results input file (such as are generated with dimensional check operations), then these clusters are converted to polygons.

The cell names in the waiver geometry files are of this form:

Waive\$wv\$_*checkname_in_cellname*

Where *checkname* is the name of the rule check and *cellname* is the name of the original cell that contains the DRC result. There can be "\$39\$_" strings inserted in the waiver cell names if either the *checkname* or the *cellname* contain the string "_in_".

Note



If you convert an ASCII results database file to a layout file and there is already a corresponding .waived file in your working directory that uses the same base file name as the ASCII RDB, then the utility does the following:

- Includes the user name, date, and comment properties in the output database, if they exist in the .waived file.
 - Converts all the results in the ASCII RDB file to a layout, regardless of the results in the .waived file. In other words, if you waive some rule check results using Calibre RVE, then the .waived file contains fewer rule check results than the RDB. However, the utility converts the entire ASCII RDB file regardless of the .waived file contents; it only uses the properties from the .waived file.
-

Comment File Format for -cm

The *comment_file* specified by the -cm option is formatted as follows:

```
check_name1 {comment}  
check_name1 {comment}  
...
```

Comments are specified for each rule check and the comment must begin on the same line as the *check_name*. The *check_name* can include wildcards (*) if you want to apply the same comment to multiple rules. The following are examples of valid comments:

```
m1_space_1 Waiver approved by JD  
v7_c2* These c2 results can be waived  
v7_c2_le HP rule only
```

In the case of wildcard matching, the most specific match to a rule will receive the comment if there are multiple matching comments. For example, if you specify these comments:

```
m1* Comment #1  
m1_space Comment #2
```

The tool assigns “Comment #2” to waiver shapes for the m1_space rule, even though the first line also matches the rule name. If multiple, equally matching waiver comments apply to the same rule name, the tool chooses one comment arbitrarily and issues a warning.

If the comment spans more than one line, then you must enclose the comment in { } brackets. For example:

```
rule_1 {Waiver comment  
that spans multiple  
lines}
```

Note



You cannot include brackets { } within the text of your comment. This conflicts with the brackets used to denote multi-line comments. For example, this is *not* allowed:

```
rule_1 Comment for {DRC} waiver flow
```

Cell Coordinate Checking

If [DRC Cell Name](#) or [ERC Cell Name](#) YES has been specified, then drc_waiv_asc2gds checks the cell coordinates of results to see if they are consistent with the cell. If not, the following error occurs:

```
ERROR: Unable to write waiver shape for cell '%s'. Specify coordinates in  
cell space.
```

The DRC or ERC Cell Name CELL SPACE and XFORM keywords are both recommended for waiver generation to avoid this error.

Examples

Example 1

Suppose you have a waiver criteria file called *waiver.criteria* with the following lines:

```
rule1 SINGLE 90 90
rule2 SINGLE 90 90
```

and you have a waiver cells file called *waiver.cells* with the following lines:

```
cellA rule1
cellB rule2
```

Further, suppose you have an ASCII Calibre RVE waiver file named *drc_results_database.waived* and it contains results that match the cell and rule pairings shown in the previous two examples.

You then execute this command:

```
drc_waiv_asc2gds -i drc_results_database.waived -o waivers.gds \
-c waiver.cells -e waiver.criteria -r
```

The *-r* option processes the *drc_results_database.waived* file. The file *waivers.gds* has these characteristics:

```
Waiver cells: Waive$wv$_rule1_in_cellA, Waive$wv$_rule2_in_cellB
Waiver layer: 1234.5678 (default layer and datatype)
Precision: 1000 (default)
Format: GDS (default)
Text objects: checksum at coordinates (0,0) on layer 1234.1 and waiver
criteria 0 90 90 at coordinates (0,10) on layer 1234.2
```

The output waiver geometry files can be specified in *WAIVER_DATABASE* statements in your waiver verification setup file. You can also manually merge the output waiver geometry files into a single *WAIVER_DATABASE* file.

Example 2

The following example demonstrates how to generate *IP_MATCH* checksums for cells using the *drc_waiv_asc2gds* utility. Assume you have a *waiver_cells* file that contains the following statements:

```
cellA IP_MATCH rule1
inv_x1 rule2
```

Your *waiver_criteria* file contains the following statements:

```
rule1 SINGLE 95 90
rule2 SINGLE 95 90
```

Both the `cellA` and `inv_x1` cells are contained in the `cells.gds` file and previously generated DRC results (from the rules file that contains rule1 and rule2) are saved in the `drc.results` ASCII database.

```
drc_waiv_asc2gds -i drc.results -o waivers.gds \  
-c waiver_cells -e waiver_criteria \  
-s waiver_setup \  
-ip cells.gds
```

Example 3

You can specify multiple input files with the following syntax:

```
drc_waiv_asc2gds [-r] [-i {input_directory | input_file} ...] \  
-o output_database
```

The input results database files must be of the same type (either ASCII or *.waived*). If you specify *.waived* files, you must include the `-r` option.

The following example loads multiple *.waived* files from two directories (each directory may contain multiple files):

```
drc_waiv_asc2gds -r -i my_waived_dir_1 -i my_waived_dir2 -o out.gds
```

The following example loads multiple *.waived* files:

```
drc_waiv_asc2gds -r -i database_a.waived -i database_b.waived -o out.gds
```

The following example loads multiple results database files:

```
drc_waiv_asc2gds -i database_a.rdb -i database_b.rdb -o out.gds
```

Related Topics

[Creating Embedded Waiver Shapes Using Calibre RVE](#)

[Physical Verification with Waivers](#)

[Writing the Waiver Cells File for waiver_flow](#)

checksum_util

Layout design waiver checksum report generator.

Path: `$CALIBRE_HOME/bin`

This command takes a GDSII or OASIS input file and calculates checksum values that conform to the command's parameters.

Usage

checksum_util -i *input_database_file*

[-c]
[-f *ignore_file*]
[-k {TEXT | INST | LAYERNO}]
[-o *summary_file*]
[-p *primary_cell*]
[-t {gdsii | oasis}]
[-v]

Arguments

- **-i *input_database_file***

Required parameter set that specifies the pathname of a GDSII or OASIS layout file. If the specified file has the .gz extension, then the file is decompressed and the run proceeds on the decompressed data. The gzip utility must be in your environment for this to occur. The run is faster if the file is decompressed externally.

- **-c**

Option that specifies to use CRC32 encoding. By default MD5 encoding is used, which should be accepted unless you are familiar with these encodings and have reasons for preferring CRC32.

- **-f *ignore_file***

Optional parameter set that specifies an ignore file (format discussed in the Description section). The file specifies cells or layers to ignore during the run.

- **-k {TEXT | INST | LAYERNO}**

Optional parameter set that specifies to ignore certain elements in the *input_database_file* for the purpose of computing a checksum. If the TEXT keyword is used, then all text objects are ignored. If the INST keyword is used, all instances in a cell are ignored. If LAYERNO is used, then all numerals of layers (including text layers) are ignored. Note, LAYERNO does not ignore the layers themselves, just the numerals. The -f option controls ignoring of actual layers. More than one of the optional arguments may be specified independently in the same command line, each preceded by -k.

- **-o *summary_file***

Optional parameter set that specifies the pathname of the summary report for the run. By default, the report is written to *checksum.summary*.

- **-p *primary_cell***
Optional parameter set that specifies the top-level cell from which to begin checksum calculations. If not specified, the top-level cell is determined from the *input_database_file*.
- **-t {*gdsii* | *oasis*}**
Optional parameter set that specifies the layout format. Default is *gdsii*.
- **-v**
Optional parameter that produces a verbose summary report. By default, only checksums for cells (annotated C) appear in the report, and runtime messages are written to STDOUT only. When this option is specified, checksums for geometry layers (G), text layers (T), and instances (I) are also reported. These checksums can be filtered using the **-f** or **-k** options. Also, warnings are returned to STDOUT and to the summary report file when **-v** is specified.

Description

This command generates a report of the checksums of cells throughout the hierarchy of the *input_database_file*. The name of the report is *checksum.summary* by default. The output filename is controlled using the **-o** option. Output is sorted by cell name in the summary report.

Checksums are generated only for cells by default. If the **-v** option is used, checksums for layer geometry, text objects, and instances in each cell are also reported.

You can control whether text objects, instances, or layer numerals contribute to the calculation of checksums by using the **-k** option. These options are useful when text objects, cell instantiations, or layer number assignments are expected to change for the input design.

When **-k LAYERNO** is not used, layer numerals contribute to the checksum. Therefore, two identical polygons on different layers have different checksums by default. When **-k LAYERNO** is used, identical polygons have identical checksums.

The **-f** option specifies an ignore file, which is used to designate cells or layers to ignore during the checksum run. The syntax of the commands in the file is as follows:

```
CELL <cell_name>  
LAYER <number>
```

The *cell_name* is the name of a cell. The *cell_name* may contain * wildcards, which match 0 or more characters. The *number* is an integer corresponding to a layer number in the *input_database_file*. Each of the commands can be specified any number of times in an ignore file. Entries need not be unique. Blank lines are allowed in the file. There is no comment character.

When you specify a cell as ignored, then that cell does not participate in checksum verification. The cells present as instances in the hierarchy of the ignored cell do participate in checksum verification (and are written to summary file) only if they are present in any other cell's hierarchy that is not ignored.

The -f and -k options can affect a computed cell checksum.

If more than one primary cell is found, the first cell in the database is chosen and the others are ignored. The -p option enables you to choose which cell to consider as the primary cell. Checksums are calculated moving down the hierarchy from the primary cell.

The default layout input format is GDSII. You can choose OASIS using the -t option.

The default format of the summary file is as follows:

```
=====
=== CALIBRE::CHECKSUM SUMMARY REPORT
===
Execution Date/Time:  <day> <month> <time> <year>
Checksum Version:    <calibre_version> <day> <month> <time> <year>
Layout System:       <layout_type>
Layout File Pathname: <path>
Layout Primary Cell:  <topcell>
Algorithm:            <checksum_algorithm>
Current Directory:    <path>
User Name:            <username>
-----
--- RUNTIME WARNINGS
---
-----
--- IGNORED CELL // Specified if any cell is ignored. ---
-----
--- CELL CHECKSUM DETAIL
---
=====
CELL                                CHECKSUM
=====
<cell_name>                        <checksum>
<cell_name>  IP_MATCH key value is <n>##<m>##<k>##<checksum> <alg_version>
...
-----
--- SUMMARY
---
TOTAL Cells Checksummed: <number>
```

If you use the -v option, then the checksum portion of the report looks like this:

```
=====
CELL                                LAYER      CHECKSUM                                TYPE
=====

<cell_name>                        <layer>     <checksum>                                {G | I | T}
...
=====
<cell_name>                        <checksum>                                C
<cell_name> IP_MATCH key value is <n>##<m>##<k>##<checksum> <alg_version>
=====
...

// Definitions
<cell_name> : name of cell.
<layer> : number of a layer.
<checksum> : checksum code. if no object is available, the value is 0.
G : checksum for geometry layer.
I : checksum for an instance.
T : checksum for text layer.
C : checksum is for cell.
```

Examples

Here is an example command invocation and summary report file using the defaults:

```
checksum_util -i layout.gds
cat checksum.summary
=====
=== CALIBRE::CHECKSUM SUMMARY REPORT
===
Execution Date/Time: Fri Oct 19 16:05:14 2012
Checksum Version: v2012.4 Wed Oct 17 14:23:01 GMT 2012
Layout System: GDS
Layout File Pathname: layout.gds
Layout Primary Cell: TOP
Algorithm: MD5
Current Directory: /home/workarea
User Name: jdoe
-----
--- IGNORED CELL
---
=====
CELL                                CHECKSUM
=====

aoi21xl                            <checksum text>
aoi22xl                            <checksum text>

-----
--- SUMMARY
---
TOTAL Cells Checksummed: 2
```

Related Topics

[Text Object Annotations](#)

[waiver_util](#)

[checksum_util Runtime Messages](#)

waiver_util

Batch waiver cell update utility.

Path: `$CALIBRE_HOME/bin`

This command takes a GDSII or OASIS input file and performs a specified waiver operation on the file.

Usage

```
waiver_util
{
    { -checksum [-g | -v]}
    | { -mergedesc [-d waiver_desc_file ... [-e] [-l layer]]
      { { [-s suffix_file] [-p prefix_file]} | [-r rename_cells_file]}
    | { -splitdesc -i waiver_desc_file -cells waiver_cells_file -criteria waiver_criteria_file }
    | -extract
}
-i input_file
-o output_file
[-c [-vtt version_texttype]]
[-t texttype] } | { -rulechecksum rule_file [rule_name]}
[-h]
```

Arguments

- **-checksum**
Specifies to generate or to validate checksum text objects for waiver cells in the *input_file*. Either the -g or -v option must also be specified with **-checksum**. May not be specified with other primary command options like **-mergedesc** and so forth.
- -g
Specifies to generate checksum text objects for in the *input_file* and write them to the *output_file*. May only be specified with **-checksum**. May not be specified with -v.
- -v
Specifies to validate checksum text objects for waiver cells in the *input_file*. May only be specified with **-checksum**. May not be specified with -g. By default, checksum text objects are assumed to have texttype 1. The -t option is used to specify a different texttype. By default, output is to STDOUT. Output to a file is specified through the -o parameter set.
- **-mergedesc**
Specifies to add or to update the waiver criteria text objects in waiver cells in the *input_file* and write the text objects to the *output_file*. At least one of the -d or -s options must also be specified with **-mergedesc**. May not be specified with other primary command options like **-checksum**.

- **-d *waiver_desc_file***

Specifies the pathname of a waiver description file. The waiver criteria in the file are used to generate waiver criteria text objects in the ***output_file***. More than one *waiver_desc_file* may be specified. May only be specified with **-mergedesc**.

Note, the waiver description file is deprecated as of the 2012.3 release. Use the -e switch to indicate that the waiver description file is a waiver criteria file, as described under “[Waiver Criteria File Format](#)” on page 147.

- **-e**

Specifies that the *waiver_desc_file* argument is a waiver criteria file.

- **-l *layer***

Optional parameter set that specifies the layer (without datatype) that is used when writing waiver criteria text object annotations in the ***output_file***. The *layer* must be a layer on which waiver shapes appear or a warning is issued and no text objects are written. This option can only be specified if -d is also specified.

- **-s *suffix_file***

Optional parameter set that specifies the pathname of a cell suffix file. This option is used to change the suffixes of waiver cell names, the hierarchical cell name (if it exists), and the names of the parent cells of the waiver cells. If you apply the waiver_util -s option on a layout file that has no waiver cells, the cells in the layout are still renamed to include the suffix. This option can only be specified with **-mergedesc** and cannot be specified with -r.

Wildcards are supported in the suffix file. See “[Support For Wildcards in the Suffix and Prefix Files](#)” on page 216.

- **-p *prefix_file***

Optional parameter set that specifies the pathname of a cell prefix file. This option is used to change the prefixes of waiver cell names, the hierarchical cell name (if it exists), and the names of the parent cells of the waiver cells. If you apply the waiver_util -p option on a layout file that has no waiver cells, the cells in the layout are still renamed to include the prefix. This option can only be specified with **-mergedesc** and cannot be specified with -r.

See “[Apply a Prefix to Waiver Cells](#)” on page 215 for an example. Wildcards are supported in the prefix file. See “[Support For Wildcards in the Suffix and Prefix Files](#)” on page 216.

- **-r *rename_cells_file***

Optional parameter set that specifies the pathname of a file that contains cell renaming instructions. Use this option to change the cell name, (including the hierarchical cell name, if it exists), and the waiver cell name for the specified cell. This option can only be specified with **-mergedesc** and cannot be specified with either -p or -s.

See “[Rename Waiver Cells](#)” on page 214 for an example.

Note



One-to-many and many-to-one mapping is not supported. For example, these groups of statements will result in error messages:

```
#Incorrect one-to-many mapping
cellA INV_A
cellA INV_A_2

#Incorrect many-to-one mapping
cellA INV_A
cA INV_A
```

- **-splitdesc**

Use this option to split a pre-2012.3 waiver description file into the corresponding waiver cells and waiver criteria file formats. Use the **-cells** and **-criteria** parameters to specify the output filenames. You may specify a waiver cells file, a waiver criteria file, or both using the **-criteria** and **-cells** keywords. At least one file type must be specified for description file translation. May not be specified with other primary command options like **-checksum**.

- **-cells *waiver_cells_file***

Specifies the pathname of a waiver cells file to export using the **-splitdesc** option. Use only with the **-splitdesc** option.

- **-criteria *waiver_criteria_file***

Specifies the pathname of a waiver criteria file to export using the **-splitdesc** option. Use only with the **-splitdesc** option.

- **-extract**

Specifies to extract the waiver criteria from waiver criteria text objects in the *input_file* and write them to an ASCII *output_file*. The output file can serve as a waiver criteria file. By default, waiver criteria text objects are assumed to be on texttype 2. The **-t** option is used to specify a different texttype. May not be specified with either **-checksum** or **-mergedesc**.

- **-i *input_file***

Required parameter set that specifies a GDSII or OASIS design in which there are waiver cells. If the file ends in a *.gz* suffix, it is uncompressed if the gunzip utility is available in your environment. If **-splitdesc** is specified, the **input_file** is a waiver description file.

- **-o *output_file***

Specifies the name of an output file. This parameter set must be specified if **-mergedesc**, **-checksum -g**, or **-extract** is specified. For **-mergedesc** or **-checksum -g**, the layout format of the *output_file* matches the format of the *input_file*. For **-extract**, the waiver criteria in the *input_file* are extracted and written to an ASCII *output_file*. This parameter set is optional if **-checksum -v** is specified. If **-v** is specified, checksum data is written to an ASCII *output_file* rather than to STDOUT.

- **-c**
Specifies to validate the Calibre version for waiver cells that contain Calibre version text objects. If the Calibre version used for the run is earlier than the 20yy.q portion of a Calibre version text object (on layer 1234 texttype 7 by default) in the *input_file* design, then a WARNING is issued. This option does not generate Calibre version text objects.
- **-vtt *version_texttype***
Specifies the texttype from which to read Calibre version text objects. By default, such text objects are on texttype 7. This option may only be specified with -c.
- **-t *texttype***
Specifies a texttype. The *texttype* is a non-negative integer. When this option is used for text-object-generating modes of the utility, all output text objects are written to the specified *texttype*. Text objects generated by this option should not share texttypes with any other Calibre Auto-Waivers text object.

If **-checksum -v** is specified with -t, then checksum text objects are read from the specified *texttype* instead of texttype 1. If **-extract** is specified with -t, then text objects are read from the specified *texttype* instead of texttype 2.
- **-rulechecksum *rule_file* [*rule_name*]**
Specifies to output checksum values in the transcript for the text of rule checks in the *rule_file*. Checksum values are calculated for all rule checks when *rule_name* is not used. The *rule_name* is the name of a rule check. When *rule_name* is used, only the checksum for that rule's text is generated. May not be specified with other primary command options like **-checksum**.
- **-h**
Optional parameter that specifies to print a usage message.

Description

The command can do any of the following:

- Add waiver criteria text objects to waiver cells based upon the contents of waiver criteria and waiver cells files.
- Update waiver criteria text objects in waiver cells based upon the contents of waiver criteria and waiver cells files.
- Add checksum annotation text objects to waiver cells.
- Validate checksum annotation text objects in waiver cells.
- Add suffixes or prefixes to waiver cell names, the hierarchical cell name (if it exists), and the names of the parent cells of the waiver cells.
- Extract a waiver criteria file from a layout.
- Calculate checksums for rule check text.

See “[Text Object Annotations](#)” on page 221 for details about text object annotations.

This command reads the required *input_file* and performs the functions specified by other arguments to the command. For **-mergedesc**, **-**, and **-checksum**, the format of the input file may be either GDS or OASIS, which the tool detects automatically. For **-splitdesc**, the *input_file* is a legacy waiver description file.

If no changes are made to the data read from the *input_file*, then the *output_file* is not generated.

One of **-checksum**, **-mergedesc**, or **-** must be specified.

If **-checksum** is specified, this causes checksum text objects either to be generated (-g option) or validated (-v option).

The -g option causes waiver cells from the *input_file* to receive hexadecimal checksum text annotations with texttype 1 on the waiver geometry layer. The waiver cells and text objects are written to the *output_file*. If an input cell has a checksum text object, then that object’s value is overwritten by the command for the output, and a warning is issued. If the waiver cell has no shapes in it, then 0xFFFFFFFF is written to the output as the checksum value. The -t option can be used to specify the texttype of the output text objects. See “[Geometry Checksum Annotations](#)” on page 224 for information about checksum text objects.

If the -v option is specified, then checksum text annotation objects in the *input_file* are validated. The validity of checksum annotations is written per-waiver-cell to STDOUT by default. If a waiver cell has no checksum text object, then a warning is issued. If -o is also specified, then all checksum validation messages are written to the *output_file*. Cells that are not waiver cells are not validated. The -t option is used to validate checksum text objects that are not on texttype 1.

If **-mergedesc** is specified, this causes waiver criteria text objects to be written to the *output_file* (-d option) and changes the suffixes or prefixes of waiver cells and their parent cells with the -s or -p options, respectively.

If the -d option is used (with or without -e), then the *waiver_desc_file* specifies that the waiver cells (that is, Waive\$wv\$... cells) are updated with waiver criteria text object annotations. The waiver criteria in the *waiver_desc_file* are mapped to the waiver cells. By default, the text objects are written to the layer of the final waiver shape record in the database for a given waiver cell using texttype 2. If -l is used, then the text objects are written to the specified *layer*. If -t is used, then the text objects are written to the specified texttype. See “[Waiver Description Files](#)” on page 143 and “[Text Object Annotations](#)” on page 221 for additional information about waiver description files and waiver cells.

If the -d option is used (with or without -e), and there are waiver criteria text objects in the *input_file*, and those text objects conflict with what the waiver_util command generates based upon the *waiver_desc_file*, then a warning is issued. In this case, the criteria from the

waiver_desc_file are output. However, if the new text objects are to be written to a layer that differs from the text objects in the *input_file*, then the text objects in the *input_file* are output in addition to the text objects generated by *waiver_util*.

The *-s* option may be used to specify a file that defines suffixes to add to waiver cell names and the names of their parent cells when they are written to the *output_file*. The format of the lines in the *suffix_file* are as follows:

```
cell_name suffix
```

The **-splitdesc** option is used to translate legacy waiver description files into the [Waiver Criteria File Format](#) and [Waiver Cells File Format](#) format. In this case, the **-i** parameter must point to the pathname of a waiver description file. You may specify a waiver cells file, a waiver criteria file, or both using the **-criteria** and **-cells** keywords. At least one file type must be specified for description file translation. The **-splitdesc** keyword is specified as follows:

```
waiver_util -splitdesc -i desc_file -cells cells_file -criteria crit_file
```

The **-extract** is used to extract waiver criteria from the *input_file*. The extracted file may be used for a waiver criteria file. The **-t** option is used when waiver criteria text objects do not have *texttype 2*.

The **-c** and **-vtt** options can be used for Calibre version *texttype* verification in any of the three primary modes of operation. Only text object verification is performed; text objects are not written by these options.

When the **-rulechecksum** option is used, the text of rule checks is read from the *rule_file*. Checksum output in the transcript appears per-check:

```
Rule      : rule1
Checksum  : 2f4a4b483649333737494759334b4145495a553d3c51212122
```

Examples

Annotate Waiver Cells With Criteria

For this example, assume your input design *layout.gds* has waivers as follows:

```
Cell: nand
Waiver_cell: Waive$wv$_met_501_in_nand
Waiver_LAYER_NUMBER: 1234
Waiver criteria text annotations: none
```

Your *waiver_criteria_file* has the following entry:

```
met_501 SINGLE 98 98
```

Assume you run this command:

```
waiver_util -mergedesc -i layout.gds -o layout_mod.gds \
-d waiver_criteria_file -e -l 1234
```

The *layout_mod.gds* file will have the following characteristics:

```
Cell: nand
Waiver_cell: Waive$wv$_met_501_in_nand
Waiver_LAYER_NUMBER: 1234
Waiver criteria text annotations: CRITERIA:0 98 98 on layer 1234 texttype
2
```

Generate Checksum Text Annotations for Waiver Cells

This example shows how to generate checksum text annotations for waiver cells. Assume you run this command:

```
waiver_util -checksum -i layout.gds -o layout.cksum.gds -g
```

The run transcript would appear as follows:

```
Running Waiver checksum on layout.gds

Layout File layout.cksum.gds is generated.
Checksum generation completed.
```

You can then run the following command to validate the checksums:

```
waiver_util -checksum -i layout.cksum.gds -v
```

The run transcript would appear as follows:

```
Running Waiver checksum on layout.cksum.gds
VALID CELL "Waive$wv$_poly_in_nand"
Checksum validation completed.
```

Rename Waiver Cells

Similar to the suffix or prefix feature, you can rename waiver cells with the -r option.

Assume that your waiver database includes cellA and cellB as follows:

```
cellA
  Waive$wv$_rule1_in_cellA_?0?
cellB
  Waive$wv$_rule1_in_cellB_?0?
```

You must manually create a cell renaming text file that includes a cell name followed by a space and a new name for the cell on each line, for example:

```
cellA newCellA
cellB INVB
```

When the cell renaming file is complete, enter the following command:

```
waiver_util -mergedesc -i waivers.gds -o mo.gds -r cell_rename_file
```

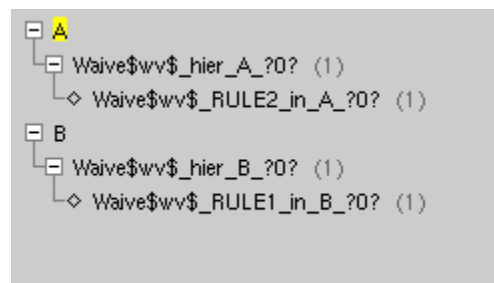
For the given example renaming file, the waiver database is modified as follows:

```
newCellA
  Waive$wv$_rule1_in_newCellA_?0?
INVB
  Waive$wv$_rule1_in_INVB_?0?
```

Apply a Prefix to Waiver Cells

You can add a suffix or prefix to a waiver cell name with the -s and -p options, respectively. The following example demonstrates how to add a prefix to certain cells. The flow for adding suffixes is the same.

Assume that your waiver database includes cells A and B as follows:



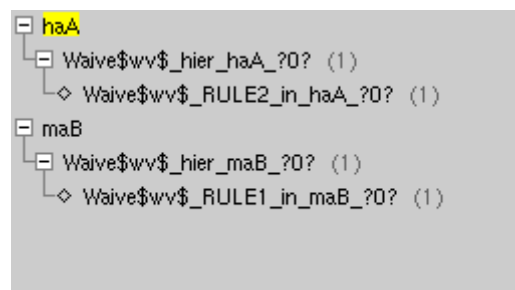
You must manually create a prefix text file that includes a cell name followed by a space and a prefix for the cell on each line. The following is an example:

```
A ha
B ma
```

When the prefix file is complete, enter the following command:

```
waiver_util -mergedesc -i waivers.gds -o mo.gds -p my_prefix_file
```

For the given example prefix file and database, the cell names in the waiver database are modified as follows:



Note that the tool adds the prefix “ha” to the waiver cell and its parent hierarchy. Similarly, it adds “ma” to cell B. If you apply the waiver_util -p option on a layout file that has no waiver cells, the cells in the layout are still renamed to include the prefix.

Support For Wildcards in the Suffix and Prefix Files

The suffix and prefix files used in the `waiver_util` application support wildcards (*) for waiver cells. For example:

```
waiver_util -mergedesc -p prefix.txt -i waived.gds -o out.gds
```

where *prefix.txt* contains the following:

```
T*   r
TO*  x
```

If the *waived.gds* layout file in this example contains a waiver cell with name TOP, then the tool uses the prefix value with the most specific cell name (TO* in this case). The resulting renamed cell name in the *out.gds* layout will be xTOP rather than rTOP.

The wildcard can be placed at any location in the cell name. For example:

```
T*P   r
```

One-to-many wildcards are not supported. For example, the following is *not* allowed:

```
T*   Q_
T*   X_
```

This rule applies across both prefix and suffix files.

If you specify a wildcard that does not match any cell in the layout, the tool issues the following warning:

```
WARNING: The following Cell Name(s) are not found in the input design file
"layout":
```

Related Topics

[Text Object Annotations](#)

[waiver_util Runtime Messages](#)

[checksum_util](#)

Appendix A

Waiver Cell Description and Text Annotations

The format and objects within the waiver cells are controlled by the waiver setup file.

Waiver Cell Description	218
Text Object Annotations	221
Waiver Cells With Instances of Error Cells	228
Support For Waivers in Cloned Cells	230

Waiver Cell Description

Waiver cell format, naming conventions, cell hierarchy, and annotations written to the waiver shapes are described.

When waiver shapes are generated either using Calibre RVE or `waiver_flow`, the waiver shapes that meet the criteria from the [Waiver Criteria File Format](#) and [Waiver Cells File Format](#) are embedded in waiver cells.

The name of the waiver cells is usually of this form:

Waive\$wv\$_checkname_in_cellname_?count?

Where *checkname* is the name of the rule check, *cellname* is the name of the parent cell in which the waiver was applied, and *count* is an integer that increments for each new waiver cell that is created. However, if the string “_in_” appears in either the *checkname* or the *cellname*, then the name of the waiver cells is this:

Waive\$wv\$_checkname_\$39\$_in_\$39\$_cellname_?count?

These waiver cells are either embedded in the design or are saved into a separate waiver geometry database that is merged with the main design at run time.

The waiver cell names must conform to the rules for GDSII Structure names when GDS is exported. The allowed characters for such names are represented by the regular expression `[A-Za-z0-9_?$]+`. If the name of a rule check includes a character that is not allowed in GDS, that character is replaced by the following:

?\$ASCII_code_of_character\$

For example, if check MT1.1 is specified for Cell1, the waiver cell name would be this:

Waive\$wv\$_MT1?46\$1_in_Cell1_?0?


because the “.” is disallowed in GDS and is represented as 46 in ASCII.

OASIS output conforms to this convention for consistency.

During a calibre -waiver run, the *checkname* portion of the waiver cell name is checked against the rule file. If a rule check name cannot be matched, warnings are issued and waivers are not applied using the waiver cell.

If the `?$ASCII_code_of_character$` naming convention appears in a waiver cell name during a calibre -waiver run, the ASCII code is replaced by the character the code represents.

Note

 If replacement occurs because the GDS design’s original cell name used the name replacement convention, this can lead to unexpected results. Use cell names that do not use the replacement syntax convention.

You should not change waiver cell names. The only exception might be if you change the name of the parent cell. In that case, the *cellname* portion of the waiver cell should be changed to match the parent cell. Waiver cell geometry should not be edited by hand as this can cause inconsistent waivers during verification. If you specify RUN_LAYOUT_CHECKSUM YES (the default) in the waiver verification setup file, then cells that fail the checksum due to cell modification do not have waivers applied.

See “[Waiver Setup File Format for Waiver Verification](#)” on page 126 for more information about the waiver verification setup file.

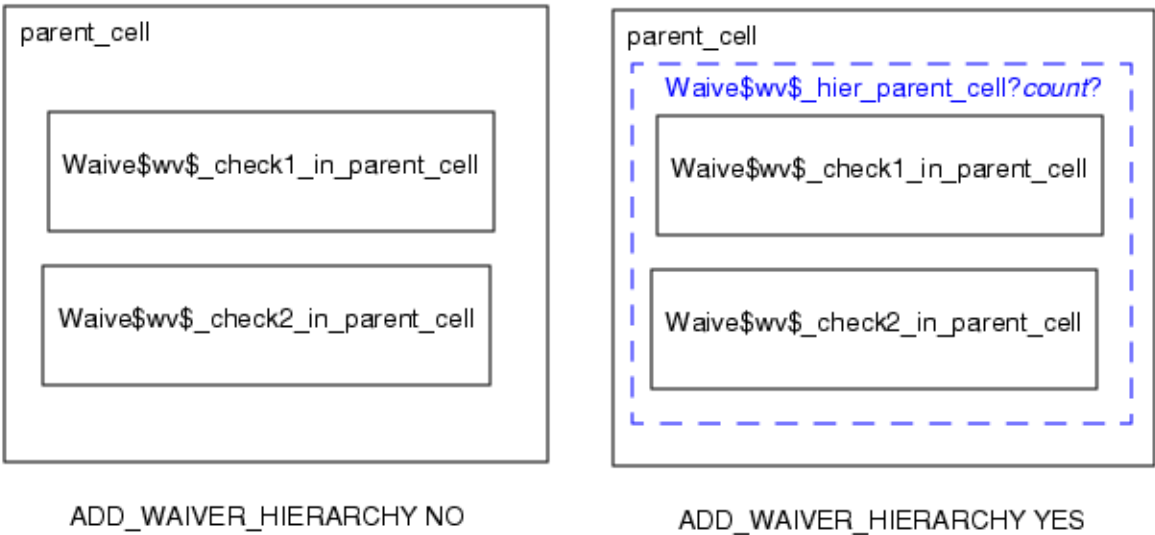
Intermediate Container Cells 219

Intermediate Container Cells

Waiver shape cell hierarchy can be controlled to improve maintenance and readability of the waiver cells.

Waiver cells are instantiated in their parent cells as discussed under “[Waiver Cell Description](#)” on page 218. In certain cases, it may be desirable to have all waiver cells in a given parent cell be instantiated into an intermediate cell as shown in [Figure A-1](#). Intermediate cells for waivers help organize the waiver cells in the layout. They are especially useful when waivers are created for multiple rules. This improves the ease of maintenance when you need to modify the parent cell or the waivers because all of the waivers for the cell are contained in one intermediate cell instead of individual waiver cells for each rule.

Figure A-1. Intermediate Container Cell



By default, the intermediate container cell shown on the right is not added by Calibre Auto-Waivers tools. Output of the intermediate cell can be specified in the Export Waivers dialog box of Calibre RVE. If ADD_WAIVER_HIERARCHY YES is specified in the waiver generation

setup file, then the container cell is output by `waiver_flow`. (The `drc_waiv_asc2gds` utility ignores this statement). See “[Waiver Setup File Format for Waiver Generation](#)” on page 117 for more information about the waiver generation setup file.

The name of the hierarchical intermediate container cell is of this form:

Waive\$wv\$_hier_in_cellname_?count?

Where *cellname* is the name of the parent cell and *count* is a value that increments for each waiver export of the same rule and cell combination to avoid possible conflicts.

If `IP_MATCH` is specified in the waiver cells file, waivers are always written to intermediate container cells, regardless of the `ADD_WAIVER_HIERARCHY` setting. `IP_MATCH` waivers always use intermediate cells because the `IP_MATCH` text is written to the hierarchical waiver cell, instead of each individual waiver cell in the parent cell. This reduces the database size and significantly improves run time when `IP_MATCH` is checked in verification runs.

If you specify with Merge YES, and you add a new waiver for the same rule and cell to a layout that already has waivers instantiated into an intermediate hierarchy, the `IP_MATCH` text is checked and the following actions are taken:

- **IP_MATCH text is different** — Waiver cell is written to an intermediate hierarchical waiver cell of the same name but with a different *count*.
- **Waiver does not contain IP_MATCH text** — Waiver cell is written to an intermediate hierarchical cell of the same name but with a different *count*.
- **IP_MATCH text is the same** — Waiver cell is added to the existing intermediate hierarchical waiver cell.

Note



The *?count?* number does not increment for hierarchical waiver cells in Cadence Virtuoso databases.

Related Topics

[Text Object Annotations](#)

[Waiver Cells With Instances of Error Cells](#)

Text Object Annotations

Text objects can be annotated to the waiver shapes to control how waivers are applied during a verification run.

A waiver cell can contain text object annotations that are used for various purposes. The layer number and texttype used for text annotations is specified in the Calibre RVE and Calibre Interactive user interfaces if using those tools to generate waivers.

If using waiver_flow, the layer number and datatype can be specified in the waiver generation setup file, otherwise default values are used. The texttypes can also be configured in the waiver generation setup file, along with some of the text object layers. The following table shows the default settings.

Table A-1. Text Object Annotations

Form and Layer	Description
<p><geometry_checksum></p> <p>Same value as the LAYER_NUMBER statement in the waiver setup file if it is set. Otherwise, the default layer is 1234 with texttype 1.</p> <p>Location (0, 0) of the waiver cell.</p>	<p>The checksum is a hexadecimal string computed from the polygons in the cell. The checksum ensures that a modified waiver cell (after the checksum is applied) is flagged in a Calibre run with waivers when RUN_LAYOUT_CHECKSUM YES is specified in the setup file. This text object is always generated by waiver_flow. If “Annotate waiver cells with checksum text” is enabled in Calibre RVE, then this text object is output.</p> <p>See “Geometry Checksum Annotations.”</p>

Table A-1. Text Object Annotations (cont.)

Form and Layer	Description
<p>Waiver Criteria Settings</p> <p>Typical primary criteria:</p> <p><type> <waiver_overlap> <result_overlap></p> <p>For WAIVE_DENSITY objects:</p> <p><type> <result_overlap></p> <p>For WAIVE_EXTENT objects:</p> <p><type> <cell></p> <p>For WAIVE_MARKER objects:</p> <p><type> <marker></p> <p>For WAIVE_PATTERN objects:</p> <p><type> <pattern> \$</p> <p>Multiple WAIVE_PATTERN criteria are separated by the \$ symbol.</p> <p>Same value as the LAYER_NUMBER statement in the waiver setup file if it is set. Otherwise, the default layer is 1234 with texttype 2. Location (0, 0.1) of the waiver cell.</p>	<p>The values are integers that correspond to criteria that would appear in a waiver description file.</p> <p><type> corresponds to the waiver type:</p> <p>0 = SINGLE 1 = MAX 2 = MULTI 3 = WAIVE_DENSITY 4 = XOR 5 = WAIVE_PATTERN 6 = WAIVE_MARKER 7 = WAIVE_EXTENT</p> <p><waiver_overlap> corresponds to the waiver overlap number.</p> <p><result_overlap> corresponds to the results overlap number.</p> <p>This text object is always generated by waiver_flow. If “Annotate waiver cells with checksum text” is enabled in Calibre RVE, then this text object is output.</p> <p>For WAIVE_EXTENT, and WAIVE_MARKER, or WAIVE_PATTERN text objects, see “Dummy Waiver Cells with Text Annotations” for details.</p> <p>These text objects are for annotation purposes only. They are not verified in the flow.</p>
<p><userid></p> <p><date></p> <p><comment></p> <p>The text objects are placed on vertices of the waiver shape, starting at the lower-left and going counter-clockwise, with these texttypes:</p> <p><userid>: 3 <date>: 4 <comment>: 5</p>	<p>If the waiver comment is specified by the user in Calibre RVE, then these text objects are output per polygon. Newlines are replaced by space characters.</p> <p>Not all polygons will have these objects. The <comment> object may be absent if there is no user comment associated with the rule check.</p> <p>These text objects are not generated by waiver_flow by default.</p> <p>These text objects are for annotation purposes only. They are not verified in the flow.</p>

Table A-1. Text Object Annotations (cont.)

Form and Layer	Description
<p><ip_match_signature></p> <p>Same value as the LAYER_NUMBER statement in the waiver setup file if it is set. Otherwise, the default layer is 1234 with texttype 6. The text object appears in an intermediate hierarchy cell.</p>	<p>When IP_MATCH is specified in a waiver generation run, a signature text object for cell matching is output. This is used to match a layout cell to a waiver-corrected IP cell, even if the names are different.</p> <p>See “IP_MATCH Checksum Annotations.”</p>
<p><rule_checksum></p> <p>Same value as the LAYER_NUMBER statement in the waiver setup file if it is set. Otherwise, the default layer is 1234 with texttype 9.</p> <p>Location (0, 0) of the waiver cell.</p>	<p>The checksum is a hexadecimal string computed from the rule code and rule filename that generated the waiver.</p> <p>Rule checksums guarantee that the rule code associated with the rule name has not been changed since the waiver shape was last generated.</p> <p>During a verification run, rules from the rule file are checksummed and compared to the waiver cell checksums when RUN_RULE_CHECKSUM YES is specified in the waiver setup file.</p> <p>See “Creating Waiver Cell Rule Checksum Annotations with waiver_flow” and “Using Rule Checksums in calibre -waiver.”</p>
<p><version_string></p> <p>Same value as the LAYER_NUMBER statement in the waiver setup file if it is set. Otherwise, the default layer is 1234 with texttype 7.</p>	<p>The version string is derived from the Calibre version used during the waiver generation run. The string ensures that a waiver cell is verified by the same Calibre version, or later, as the one used to generate the cell. This string is verified when RUN_CALIBRE_VERSION_CHECKSUM YES appears in the setup file. Note that only the 20yy.q portion of the text object version string is compared to the Calibre version.</p> <p>See “Geometry Checksum Annotations.”</p>

If a waiver cell has multiple waiver criteria or geometry checksum text objects in it, that cell is ignored in a calibre -waiver run. In either case, a WARNING is issued.

See “[waiver_util](#)” on page 208 for information about adding text object annotations using this batch utility.

Geometry Checksum Annotations	224
Rule Checksum Annotations.....	224
Calibre Version Text Annotations	225

IP_MATCH Checksum Annotations	226
Dummy Waiver Cells with Text Annotations	227

Geometry Checksum Annotations

Geometry checksums can be annotated to the waiver shapes to ensure that cells are not geometrically modified.

Geometry checksum text objects are written to waiver cells in hexadecimal format by Calibre Auto-Waivers tools as described in [Table A-1](#).

The geometry checksum text objects (texttype 1 by default) can be removed from waiver cells to prevent checksum verification, but this is not a recommended practice. Multiple geometry checksum text objects cause a waiver cell to be ignored during verification.

If RUN_LAYOUT_CHECKSUM YES is specified or accepted as the default in the waiver verification setup file during a calibre -waiver run, and the waiver cell has been geometrically altered causing a checksum mismatch, then a warning is issued at run time and waivers in the call are not applied. Specifying RUN_LAYOUT_CHECKSUM NO disables checksum verification. See “[Waiver Setup File Format for Waiver Verification](#)” on page 126 for complete details.

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

Rule Checksum Annotations

Rule checksums are used to ensure that the rule being waived has not been modified. Waivers generated in Calibre Auto-Waivers are based on the rule name. Rule checksums guarantee that the rule code associated with the rule name has not been changed since the waiver shape was last generated.

The following layer operations are supported:

- All DRC operations.
- DFM RDB, DFM Analyze, DFM Property, DFM Property Merge, and DFM Copy.

The following rule checksum operations behave as described:

- Filenames are compared verbatim. If *drc.rules* and *./drc.rules* are specified, different checksums are generated.
- Encrypted rule check operations are not checksummed. The use of encrypted expressions and constraints will not affect the checksum.

- The values depend on the precision of the rule file. If the precision of the rule file is modified, the checksum will change.

Checksums for Derived Layers

The checksum of a custom layer definition for a rule is generated from the following criteria:

- The name of the layer operation.
- The checksum of the input layer, or the handle of the input layer if it is an original layer.
- Any additional options and constraints.

The scope of the derived layer (whether it is global or local to a rule check) has no impact on the rule checksum.

Checksums for Rule Checks

The checksum for a rule check is generated from the following criteria:

- If there is one output operation, then the checksum is created from the output operation.
- If there are multiple output operations, the checksum is independent of the order of output operations.
- Rule checksums are generated from the rule filename and the rule code associated with each rule. If a rule name is changed from waiver generation to waiver verification, the waiver is still applied if the rule code remains unchanged.

To generate rule checksums using Calibre RVE, refer to “[Waiver Shape Generation Flow](#)” on page 74. To generate rule checksums in waiver_flow refer to “[Creating Waiver Cell Rule Checksum Annotations with waiver_flow](#)” on page 109.

To verify rule checksums, refer to “[Using Rule Checksums in calibre -waiver](#)” on page 191.

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

Calibre Version Text Annotations

Version checksums can be annotated to the waiver shapes to ensure that the same version of Calibre is used during generation and verification.

Calibre version text objects are written to waiver cells by Calibre Auto-Waivers tools as described in [Table A-1](#). The Calibre version text objects (texttype 7 by default) should not normally be modified.

If RUN_CALIBRE_VERSION_CHECKSUM YES is specified in the waiver verification setup file during a calibre -waiver run, then waiver cells are checked for Calibre version text objects.

If such objects do not exist, or the 20yy.q portion of the Calibre version is older than the version value for the generated text object, then a warning is issued but the waivers are still applied. The same is true if there are multiple version text objects in a waiver cell.

See “[checksum_util](#)” on page 203 and “[waiver_util](#)” on page 208 for information about batch utilities that can validate checksums in waiver cells. The latter utility can also add checksum text objects to waiver cells.

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

IP_MATCH Checksum Annotations


IP_MATCH checksums can be annotated to the waiver shapes when you want to produce waivers for IP blocks.

When you specify IP_MATCH in the waiver description file for generating text objects (texttype 6 by default), the signatures contained in these text objects are used to match other design cells by geometry and text, but not necessarily by name. These text objects appear in cells of the WAIVER_DATABASE file. The design cells that contain such objects must not be modified.

IP_MATCH checksum text objects are always written to intermediate hierarchy cells as discussed under “[Intermediate Container Cells](#)” on page 219.

IP_MATCH checksum text objects do not respond to RUN_LAYOUT_CHECKSUM settings in the waiver verification setup file.

Note


 If you are using waivers with IP_MATCH checksums from Calibre 2019.1 and earlier, it is recommended that you regenerate the waiver database using the 2019.2 version of Calibre. There was an internal change to how IP_MATCH checksums were generated in 2019.2, so using waiver databases generated with older versions of Calibre Auto-Waivers may result in a different checksum calculation.

IP_MATCH Checksum Generation From Calibre Interactive

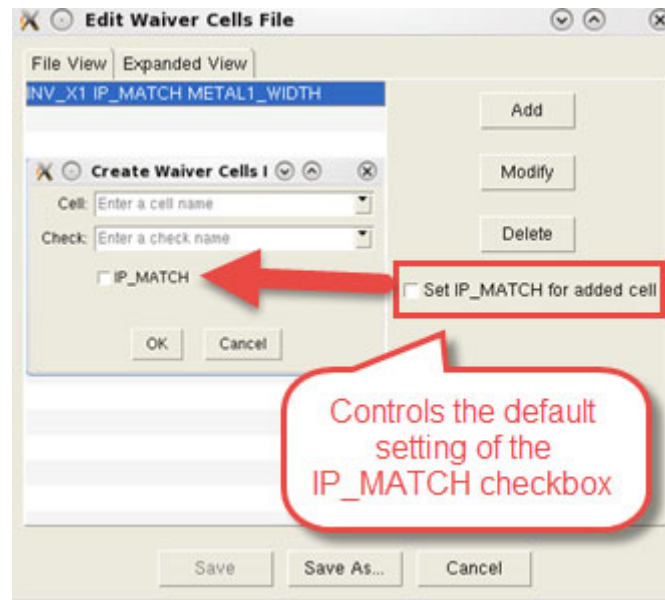
You can generate IP_MATCH checksums for waiver cells from Calibre Interactive using the Edit Waiver Cells File window. To set IP_MATCH behavior from Calibre Interactive, do the following:

1. Invoke Calibre Interactive nmDRC:

```
calibre -gui -drc
```

2. Click **Inputs** and choose “Autowaiver Creation” from the Run dropdown list.
3. Click **Waiver Creation** and click the **Edit the waiver cells** button .
4. Click **Add** and specify a cell and rule.
5. Enable the IP_MATCH checkbox to generate checksums for the specified waiver cell.

To set the IP_MATCH checkbox globally for all newly added waiver cells, enable the “Set IP_MATCH for added cells” checkbox on the Edit Waiver Cells File window. The value of this checkbox is applied to the IP_MATCH checkbox on the Create Waiver Cells File Record dialog box.



In other words, this new option behaves as a default setting for the IP_MATCH checkbox. If the “Set IP_MATCH for added cell” checkbox is enabled, then the IP_MATCH checkbox is also enabled.

You can set this option from a Calibre Interactive runset file by adding the following option:

```
*cmnWaiverSetIPMatchForAllCells: 1
```

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

Dummy Waiver Cells with Text Annotations

Text can be annotated to empty waiver cells generated for WAIVE_EXTENT, WAIVE_MARKER, or WAIVE_PATTERN waivers.

When waivers are generated with `waiver_flow` or Calibre RVE for `WAIVE_EXTENT`, `WAIVE_MARKER`, or `WAIVE_PATTERN` types of waivers, empty cells are generated at the top level of the output design. These empty, or dummy, cells are used to store text object annotations (texttype 2 by default) for the waivers. No geometry is generated for these types of waivers. These waiver annotations can be extracted when the `WAIVER_CRITERIA EXTRACT` or `APPEND` options are used during a `calibre -waiver` run. The encoded text follows the format shown in [Table A-1](#). For example, if this statement appears in your waiver criteria file:

```
rule1 WAIVE_MARKER M1 M2 M3
```

it is embedded in the waiver cell as the following text:

```
rule1 6 3 M1 M2 M3
```

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

Waiver Cells With Instances of Error Cells

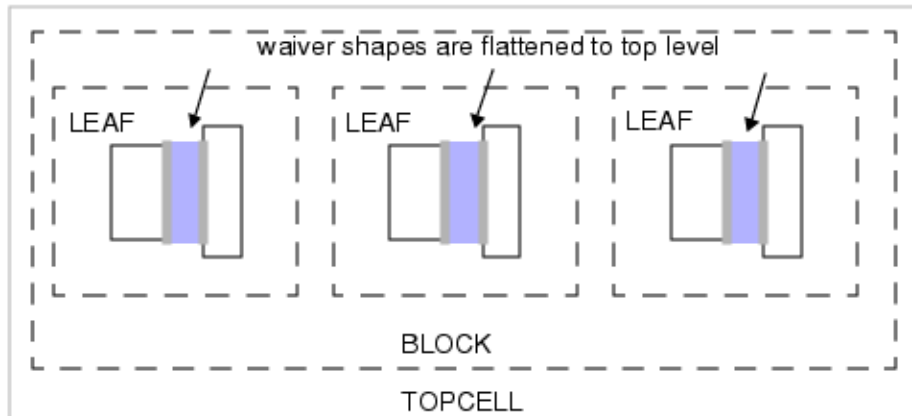
Cells with waive-able errors that contain instances of cells with waive-able errors must be treated carefully in order to have waivers applied properly throughout the design.

In a typical hierarchical run, an error result is reported once per cell, in the lowest, left-most placement of that cell. This is for efficiency, because reporting the result in every instance of the cell would be wasteful. Once the cell is fixed, all instances are presumably correct. This behavior is discussed under “[Calibre nmDRC-H Results Data Storage](#)” in the *Calibre Verification User’s Manual*.

If a design cell that has waive-able errors contains multiple instances of a lower-level cell, and that lower-level cell contains waive-able errors, this situation must be treated carefully. It would not suffice for only the lowest, left-most placement of the lower-level cell to have its waiver applied.

In order to apply the waivers to all instances of the lower-level cell, Calibre RVE and the `waiver_flow` executable automatically flatten the results output of the rule check that produced the waiver shape. This flattening produces the correct waiver geometry in the output design for all instances of the lower-level cell.

Figure A-2. Flattening of Waiver Shapes



In [Figure A-2](#), the LEAF cell is a lower-level cell that has a violation. The LEAF cell is instantiated in the BLOCK cell multiple times. The BLOCK cell is declared in the waiver cells file. The waiver shapes from the BLOCK cell are flattened to the top level by default to apply the waivers in all instances of the LEAF cell inside the BLOCK cell.

If you encounter this behavior, you must decide whether you want errors in the lower-level cell waived wherever they may occur, or only in the context of a parent cell. If the decision is that waivers in the parent cell are what is desired (BLOCK cell in [Figure A-2](#)), then the waiver cells and criteria files should be written accordingly. For example, if you want the waivers in the LEAF cell to be local to the BLOCK cell, you could have this in the waiver cells file:

```
BLOCK space_rule
```

and this in the waiver criteria file:

```
space_rule SINGLE 98 98
```

However, if your intent is to waive errors in the lower-level cell no matter where it appears in the design, then you should place the lower level cell (LEAF cell in [Figure A-2](#)) in the waiver cells file:

```
LEAF space_rule
```

and this in your waiver criteria file:

```
space_rule SINGLE 98 98
```

Related Topics

[Text Object Annotations](#)

[Waiver Cell Description](#)

Support For Waivers in Cloned Cells

Calibre Auto-Waivers optionally supports the generation of waiver shapes inside cloned cells.

Cloned cells can be generated by statements in your rule file, such as [Layout Clone Rotated Placements](#).

You can optionally generate waivers in cloned cells when exporting waivers from Calibre RVE or with the `drc_waiv_asc2gds` utility. This feature is not enabled by default.

The flow described in this topic only affects waiver generation; there is no difference in Calibre Auto-Waivers for waiver verification.

Note



IP_MATCH validation is not supported for this feature and should be disabled if your waiver database contains waivers in cloned cells.

If IP_MATCH is specified for cloned cells in the waiver cell file for waiver generation, a warning message is generated stating that IP_MATCH checksums will not be generated for the cloned cells.

- [waiver_flow Usage](#)
- [drc_waiv_asc2gds Usage](#)
- [Calibre RVE Export Waived Results Usage](#)

waiver_flow Usage

The `waiver_flow` utility cannot be used to generate waivers for cloned cells. Use the `drc_waiv_asc2gds` utility or Calibre RVE.

drc_waiv_asc2gds Usage

- **ASCII Results Database** — To generate waivers in cloned cells from an ASCII database, you must include the following statement in your waiver setup file:

```
GENERATE_CLONE_CELLS_WAIVERS YES
```

Setting this statement to YES causes Calibre Auto-Waivers to generate waivers in cloned cells. You can then generate waivers as follows:

```
drc_waiv_asc2gds -i input_rdb -o out.gds -s waiver_setup_file
```

If you specify a waiver cells file and the file contains original (non-cloned) cell names, the tool searches for corresponding cloned cells.

- **Calibre RVE .waived File** — To generate cloned cells in the *.waived* file, you must do the following in Calibre RVE:
 - a. Choose **Setup > Options** and click the **DRC Browser** category.
 - b. Enable the “Preserve cloned cells for output layout” checkbox.

When there are cloned cell names in the generated *.waived* file, the `drc_waiv_asc2gds` utility generates waivers with cloned cell names without requiring additional command-line arguments, as shown in the following example:

```
drc_waiv_asc2gds -r -i database_with_cloned_cells.waived -o out.gds
```

- **Waiver Cells File** — Additionally for the *.waived* file, the tool handles cloned cells in the waiver cells and waiver criteria files as follows:
 - If a cell with its cloned name exists in the *.waived* file, the tool searches for its original non-cloned name in the cells file and, if found, generates waivers using the existing cloned names. The tool issues a warning message indicating that additional waiver cells may be generated other than the ones specified in the cell file.

Apply the following command in this context:

```
drc_waiv_asc2gds -i results_with_clones.waived -o out.gds \
  -c waiver.cells -e waiver.criteria -r
```

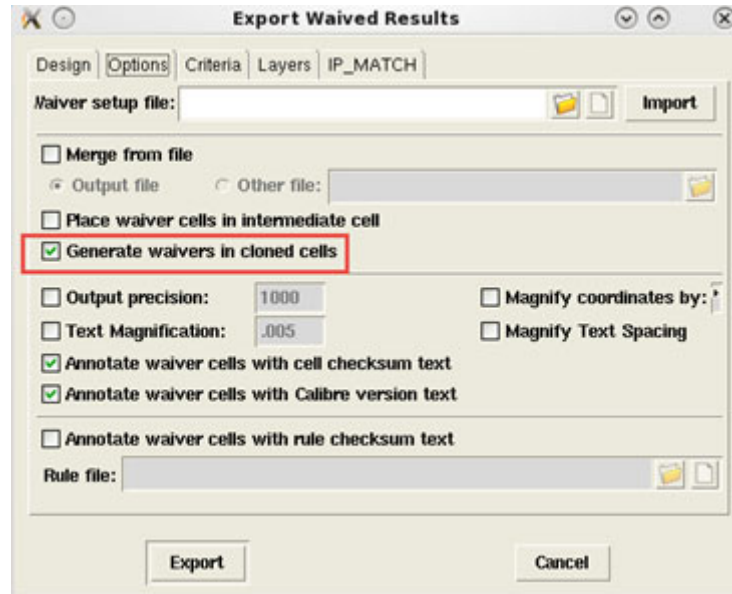
Note



The waiver criteria file is required.

Calibre RVE Export Waived Results Usage

The “Generate waivers in cloned cells” checkbox enables waiver generation for cloned cells in the Calibre RVE Export Waived Results window.



Appendix B

Waiver Setup File Statement Usage

Waiver setup file commands are applied differently for waiver_flow and calibre -waiver.

See “[Waiver Setup File Format for Waiver Generation](#)” on page 117 for the list of setup file statements used with waiver_flow.

See “[Waiver Setup File Format for Waiver Verification](#)” on page 126 for the list of setup file statements used with calibre -waiver.

Statement Usage..... 233

Statement Usage

It is possible to have a single setup file for both waiver_flow and calibre -waiver, but usually most users prefer to use separate files.

The following table shows a summary of the waiver setup file statement behaviors based upon tool usage.

Table B-1. Waiver Setup File Statements Usage

Statement	waiver_flow	calibre -waiver
INPUT_LIBRARY <i>pathname</i>	Required	Ignored
RULE_FILE <i>pathname</i>	Required	Ignored
WAIVER_CRITERIA { <i>pathname</i> EXTRACT [<i>pathname</i>] APPEND [<i>pathname</i>] }	Optional Default criteria: SINGLE 100 100	Required
WAIVER_CELLS <i>pathname</i>	required	Ignored
LAYER_NUMBER <i>integer</i>	Optional Default: 1234	Optional Default: 1234
DATATYPE_NUMBER <i>integer</i>	Optional Default: 5678	Optional Default: 5678
GENERATE_CLONE_CELLS_WAIVERS { <u>NO</u> YES}	Ignored Supported in drc_waiv_asc2gds and Calibre RVE export only.	Ignored Not required in waiver verification.

Table B-1. Waiver Setup File Statements Usage (cont.)

Statement	waiver_flow	calibre -waiver
GENERATE_COMBINED_RESULTS { <u>NO</u> YES DEBUG}	Ignored	Optional Default: NO
LAYOUT_SYSTEM {GDS OASIS LEFDEF OPENACCESS OA}	Optional Default: GDS	Ignored
MERGE { <u>NO</u> YES IGNORE_DESIGN_WAIVERS}	Optional Default: NO IGNORE_DESIGN_WAIVERS not supported	Optional Use YES if all of your waiver shapes are embedded in the main (Layout Path) design otherwise, specify NO or accept the default.
WAIVER_DATABASE <i>pathname</i>	Optional may only be specified once Default: <i>waived.gds</i>	Optional may be specified multiple times with MERGE NO. Default: <i>waived.gds</i>
DENSITY_WAIVER_RDB {YES NO}	Ignored	Optional Default: YES
USED_WAIVER_RDB { <u>ALL</u> NONE GEOM USER PROP}	Ignored	Optional Default: ALL
UNUSED_WAIVER_RDB { <u>ALL</u> NONE GEOM USER PROP}	Ignored	Optional Default: ALL
WAIVER_RDB { <u>ALL</u> NONE GEOM USER PROP}	Ignored	Optional Default: ALL
CALIBRE_WAIVER_NUMBER { <i>integer</i> AUTO}	Ignored	Optional Default: 2000
IGNORE <i>layer_operation</i>	Optional	Ignored
IGNORE_DENSITY { <u>NO</u> YES}	Ignored	Optional Default: NO
NOT_IGNORE <i>layer_operation</i>	Optional	Ignored
ADD_WAIVER_HIERARCHY { <u>NO</u> YES}	Optional Default: NO	Ignored

Table B-1. Waiver Setup File Statements Usage (cont.)

Statement	waiver_flow	calibre -waiver
RUN_DFM_GLOBAL { <u>YES</u> NO}	Ignored	Optional Default: YES
RETAIN_RESULT_TYPE { <u>NO</u> YES}	Ignored	Optional Default: NO
RUN_LAYOUT_CHECKSUM { <u>YES</u> NO}	Ignored	Optional Default: YES
PRECISION_CONVERSION { <u>YES</u> NO}	Optional Default: YES	Optional Default: YES
TEXT_MAG <i>value</i>	Optional Default: 0.005	Ignored
MAGNIFY_TEXT_SPACING { <u>NO</u> YES}	Optional Default: NO	Ignored
CHECKSUM_TEXT_LAYER <i>layer</i>	Optional Default: LAYER_NUMBER	Optional Default: LAYER_NUMBER
RUN_RULE_CHECKSUM { <u>NO</u> YES}	Optional Default: NO	Optional Default: NO
RULE_CHECKSUM_TEXT_LAYER <i>value</i>	Optional Default: LAYER_NUMBER	Optional Default: LAYER_NUMBER
RULE_CHECKSUM_TEXTTYPE <i>value</i>	Optional Default: 9	Optional Default: 9
CHECKSUM_TEXTTYPE <i>texttype</i>	Optional Default: 1	Optional Default: 1
CRITERIA_TEXT_LAYER <i>layer</i>	Optional Default: LAYER_NUMBER	Optional Default: LAYER_NUMBER
CRITERIA_TEXTTYPE <i>texttype</i>	Optional Default: 2	Optional Default: 2
USERNAME_TEXT_LAYER <i>layer</i>	Ignored	Optional Default: LAYER_NUMBER

Table B-1. Waiver Setup File Statements Usage (cont.)

Statement	waiver_flow	calibre -waiver
USERNAME_TEXTTYPE <i>texttype</i>	Ignored	Optional Default: 3
DATE_TEXT_LAYER <i>layer</i>	Ignored	Optional Default: LAYER_NUMBER
DATE_TEXTTYPE <i>texttype</i>	Ignored	Optional Default: 4
COMMENT_TEXT_LAYER <i>layer</i>	Ignored	Optional Default: LAYER_NUMBER
COMMENT_TEXTTYPE <i>texttype</i>	Ignored	Optional Default: 5
RUN_IP_MATCH { <u>YES</u> YES_RESTRICTED NO}	Ignored	Optional Default: YES
IPMATCH_TEXTTYPE <i>texttype</i>	Optional Default: 6	Optional Default: 6
OPTIMIZE_NONE_WAIVER_STATISTICS { <u>NO</u> YES}	Ignored	Optional Default: NO
RUN_CALIBRE_VERSION_CHECKSUM { <u>NO</u> YES}	Ignored	Optional Default: NO
CALIBRE_VERSION_LAYER <i>layer</i>	Optional Default: LAYER_NUMBER	Optional Default: LAYER_NUMBER
CALIBRE_VERSION_TEXTTYPE <i>texttype</i>	Optional Default: 7	Optional Default: 7
WAIVER_SUMMARY <i>pathname</i>	Optional Default: waiver.summary	Ignored Report: <i>calibre-waiver.summary</i>
TMP_DIR <i>pathname</i>	Optional Default: \$MGC_TMPDIR	Optional Default: \$MGC_TMPDIR

Table B-1. Waiver Setup File Statements Usage (cont.)

Statement	waiver_flow	calibre -waiver
WORKING_DIRECTORY <i>pathname</i>	Optional Default: current shell directory	Optional Default: current shell directory

Appendix C

Calibre Auto-Waivers Messages

Error and warning messages are presented at Calibre Auto-Waivers run time. These messages appear in the terminal window or in the log files. Use this chapter to debug errors, warnings, and status updates received during a run with Calibre Auto-Waivers or its utilities.

Calibre Auto-Waivers Runtime Messages..... 240

Calibre Auto-Waivers Runtime Messages

Runtime messages issued by Calibre Auto-Waivers applications are prefixed by an error, warning, or note string to indicate the severity of the message.

Table C-1. Calibre Auto-Waivers Error Messages

Message	Description
ERROR: Cannot determine the LAYOUT PATH from the rule file <file>.	The Layout Path statement is missing from the rule file.
ERROR: Cannot determine the LAYOUT PRIMARY from the rule file <file>.	The Layout Primary statement is missing from the rule file.
ERROR: Could not open waiver description file <file>.	The WAIVER_DESCRIPTION file specified in the waiver setup file is not accessible. Note, the waiver description has been deprecated. Use waiver cells and waiver criteria files instead.
ERROR: Conflicting primary criteria found for the RULE “<rule_name>”	Conflicting primary criteria was found for a rule. You may only specify one primary criteria per rule.
ERROR: Error <layer_number> on line <line_number> of <path>/final.svrf - this layer definition name is also defined as an original layer: error_layer.	This error occurs when a layer in your rules file uses the name “ERROR_LAYER”. The “ERROR_LAYER” layer name is reserved internally and cannot be specified in your rules file.
ERROR: GLOBAL_DEFAULT declared more than once in WAIVER_CRITERIA file <waiver_criteria_file>:<line_number>.	The GLOBAL_DEFAULT can only be specified once in the WAIVER_CRITERIA file.
ERROR: Invalid CALIBRE_WAIVER_NUMBER: <string> at line <line>.	The specified string in the waiver setup file is not valid. It must be a layer number or the AUTO keyword.
ERROR: Invalid DATATYPE_NUMBER: <string> at line <line>.	The specified string in the waiver setup file is not valid. It must be a datatype number.
ERROR: Invalid format while parsing waiver description file at line <file>:<line>.	The waiver description file has a syntax problem. Note, the waiver description has been deprecated. Use waiver cells and waiver criteria files instead.
ERROR: Invalid format while parsing WAIVER_CRITERIA file at line <file>:<line>.	The criteria file has a syntax error or the file is a legacy waiver description file.
ERROR: Invalid LAYER_NUMBER: <string> at line <line>.	The specified string in the waiver setup file is not valid. It must be a layer number.

Table C-1. Calibre Auto-Waivers Error Messages (cont.)

Message	Description
ERROR: Invalid LAYOUT_SYSTEM: <value> at line <line>	Invalid value for the LAYOUT_SYSTEM specified. Only OASIS, GDSII, GDS, LEFDEF, OA, or OPENACCESS can be specified as the layout system.
ERROR: Invalid statement <statement_type> at line <line_count>	There is unknown statement type in the waiver setup file.
ERROR: Invalid waiver type defined at file <description file> : <line>.	The waiver type may only be SINGLE or MULTI.
ERROR: <value> on line <line_number> of <path> - superfluous or invalid input object: *.	This error may result from a WAIVE_EXTENT statement that uses a wildcard (*) in the <i>cell</i> value that is not enclosed in quotes.
ERROR: Invalid <value>: <waiver_setup> at line <line_number>.	Invalid value specified for the TEXT_MAG option.
ERROR: Reading of waiver description file failed on line <line>.	There is a problem in the waiver description file on the specified line that causes a read failure. Note, the waiver description has been deprecated. Use waiver cells and waiver criteria files instead.
ERROR: Result overlap entry must be an integer between 0 and 100 inclusive at file <file>:<line>.	The waiver criteria file has an invalid result_overlap parameter in the indicated file and line number.
ERROR: Statement <statement_type> specified more than once at line <line>	Statement should be mentioned only once in the waiver setup file.
ERROR: The combination of layer and text type are the same for more than one text in the waiver setup file.	The waiver setup file has a layer and texttype pair that is duplicated.
ERROR: This file name or path name parameter is too long: <file>	The filename or pathname is too long either before or after evaluation of embedded environment variables.
ERROR: TVF encryption not supported.	Encrypted TVF may not be used in the rule file.
ERROR: Undefined or empty environment variable: <file>.	A filename or pathname cannot be resolved in the shell environment.
ERROR: WAIVER_DESCRIPTION file not specified.	Waiver setup file must contain the WAIVER_DESCRIPTION statement. Note, the waiver description has been deprecated. Use waiver cells and waiver criteria files instead.

Table C-1. Calibre Auto-Waivers Error Messages (cont.)

Message	Description
ERROR: Waiver overlap entry must be an integer between 0 and 100 inclusive.	The waiver criteria file has an invalid waiver_overlap parameter.
NOTE: Converting WAIVER_DESCRIPTION file(s) to WAIVER_CELLS file "<file_name>" and WAIVER_CRITERIA file "<file_name>".	During a waiver_flow or calibre -waiver run, if a waiver description file is specified, it is internally converted to a waiver cells and waiver criteria file.
NOTE: Treating WAIVER_DESCRIPTION file "<file_name>" as WAIVER_CRITERIA for extraction.	If the EXTRACT keyword is specified with the WAIVER_DESCRIPTION statement in the waiver_setup file, the WAIVER_DESCRIPTION file is extracted as a waiver criteria file.
WARNING: Ignoring conflicting criteria texts for RULE <rule_name> while extraction.	Conflicting primary criteria for a rule is ignored during extraction as it will lead to an error during the parsing of that waiver_criteria file.
WARNING: IP_MATCH for cell "<cell_name>" created with version "<algo_version>". The IP_MATCH algorithm uses version "<algo_version>".	An IP_MATCH checksum for a cell was created with an old algorithm, but the IP_MATCH checksum validation uses a new algorithm. IP_MATCH checksum validation will fail.
WARNING: IP_MATCH cannot be performed as unable to create file in the directory "<dir_name>".	An IP_MATCH file cannot be created because a temporary directory cannot be created. In this case, IP_MATCH will not be performed.
The following messages are issued by calibre -waiver only.	
ERROR: Invalid value for RUN_LAYOUT_CHECKSUM: <value> at line <number>.	Improper value is specified for the RUN_LAYOUT_SYSTEM statement in the waiver setup file. Only permissible values are YES and NO. Default is YES.
ERROR: LAYOUT ALLOW DUPLICATE CELLS should be set to YES.	The rule file must specify Layout Allow Duplicate Cell YES with the current run settings.
ERROR: LAYOUT INPUT EXCEPTION SEVERITY DUPLICATE_CELL should be set to 3 or 4.	The Layout Input Exception Severity specification statement must have the settings indicated by the message given the current run settings.
The following messages are issued by waiver_flow only.	

Table C-1. Calibre Auto-Waivers Error Messages (cont.)

Message	Description
ERROR: Both SOURCE SYSTEM and LAYOUT SYSTEM specified as “SPICE” in the SVRF file <file>.	The -erc option is specified and the Layout System is set to SPICE. For waiver verification, the Layout System must be geometric.
ERROR: Could not open INPUT_LIBRARY file <file>.	The INPUT_LIBRARY file specified in the waiver setup file cannot be opened.
ERROR: DRC results not generated. No waivers created.	There has been a problem with generating DRC results. Check the DRC Results Database.
ERROR: ERC results not generated. No waivers created.	There has been a problem with generating ERC results. Check the ERC Results Database.
ERROR: INPUT_LIBRARY not specified.	The INPUT_LIBRARY statement is not specified in the waiver setup file.
ERROR: INPUT_LIBRARY <file> has incorrect GDSII OASIS format.	The INPUT_LIBRARY format does not match the LAYOUT_SYSTEM.
ERROR: INPUT_LIBRARY <file> is not accessible.	The INPUT_LIBRARY file in the waiver setup file is not accessible.
ERROR: Invalid magic byte string in the INPUT_LIBRARY.	LAYOUT_SYSTEM OASIS is specified in the waiver setup file but INPUT_LIBRARY is not in standard OASIS format.
ERROR: More than one WAIVER_DATABASE specified at line <line>.	The WAIVER_DATABASE statement in the setup file can only specify one output. “Waiver Setup File Format for Waiver Generation” on page 117.
ERROR: Multiple waiver setup files specified.	Only one waiver setup file may be specified on the command line.
ERROR: No waiver shape generating rule check specified in the rule file <file>.	There is no rule active in the rule file that can generate a waiver shape. Check the rule file for DRC or ERC Select Check and Unselect Check statements. ERC must use an ERC Select Check Statement in order to run.
ERROR: Precision of output database different from INPUT_LIBRARY. Specify MERGE as NO.	The precision of the output database is different from the input database. WAIVER_DATABASE must be used in this case. See “Waiver Cell Precision and Magnification Considerations” on page 110.

Table C-1. Calibre Auto-Waivers Error Messages (cont.)

Message	Description
ERROR: RULE_FILE <file> is not accessible.	The RULE_FILE in the waiver setup file is not accessible.
ERROR: RULE_FILE <file> not specified.	The RULE_FILE statement is not specified in the waiver setup file.
ERROR: Unable to find unused CALIBRE_WAIVER_NUMBER.	The AUTO keyword is specified with CALIBRE_WAIVER_NUMBER in the waiver setup file, but no free layer can be found for this purpose.
ERROR: Unknown switch specified: <switch>	An unknown option is specified with waiver_flow.
ERROR: Waiver flow Calibre run failed. Exiting.	The waiver_flow run terminated abnormally.
FATAL ERROR	There is an internal error detected by the tool.
The following errors are issued by Calibre RVE.	
ERROR: Error while parsing layout file "<file_name>".	There is a format error in the layout.
ERROR: Cannot create file: <file_name> for output.	There is a problem generating the output file. Check permissions and disk space.
ERROR: Unable to open input geometry file: <file_name> for reading.	There is a problem with the input file. Check path and permissions.
ERROR, Waiver criteria mismatch for the cell "<cell_name>".	There is a mismatch between the input files in the waiver criteria values for the given cell.
ERROR: IP_MATCH checksum mismatch for the cell "<cell_name>".	IP_MATCH checksum is different for the given cell in the input database and waiver database.
The following errors are issued by Calibre Interactive.	
Problem while splitting waiver description file: "<filename>" Internal error while parsing waiver description file "<path>/waiver_desc" Conversion aborted	Calibre Interactive automatically attempts to convert any file named <i>waiver_desc</i> from the legacy description format to a waiver cells and waiver criteria file. If you named any file <i>waiver_desc</i> that is not a legacy description file, this error occurs.

Table C-2. Calibre Auto-Waivers Warning Messages

Message	Description
WARNING: Both MERGE and OUTPUT_LIBRARY Statements cannot be mentioned simultaneously. Ignoring OUTPUT_LIBRARY statement.	OUTPUT_LIBRARY is deprecated. Use MERGE YES instead.
WARNING: Could not open layout file <file>	A layout file could not be opened.
WARNING: Multiple waiver description rules found with different waiver criteria for CELL "<cell>" and RULE "<rule>".	The waiver criteria and cells file(s) contain multiple lines that could apply to the given cell and rule. One line is chosen using internal heuristics. See " Waiver Description Files " on page 143.
WARNING: Rule <rule> is node preserving, therefore not modifying it.	The rule is specified in a DFM Select Check NODAL statement. This is not supported.
WARNING: Multiple standard waiver description rules found with different waiver criteria for the RULE "<rule>".	The waiver criteria file contains multiple waiver description statements of the specified type with differing waiver criteria for the same rule. Waiver criteria are applied according to the priority: standard form or WAIVE_DENSITY, WAIVE_PATTERN, WAIVE_MARKER, then WAIVE_EXTENT.
Multiple WAIVE_DENSITY waiver description rules found with different waiver criteria for the RULE "<rule>".	
Multiple WAIVE_EXTENT waiver description rules found with different waiver criteria for the RULE "<rule>".	
Multiple WAIVE_MARKER waiver description rules found with different waiver criteria for the RULE "<rule>".	
Multiple WAIVE_PATTERN waiver description rules found with different waiver criteria for the RULE "<rule>".	
WARNING: Rule <rule> is defined under WAIVE_DENSITY category but has no DENSITY operation. Applying GLOBAL_DEFAULT criteria and ignoring WAIVE_DENSITY criteria.	The <rule> is specified in the WAIVE_DENSITY statement form, but there is no Density operation. The global default criteria, specified by GLOBAL_DEFAULT, is applied instead.
WARNING: RULE <rule> has operation not supported in waiver flow, therefore not modifying it.	There is an unsupported layer operation in the given rule check specified in the waiver criteria file. No action is taken for that rule.

Table C-2. Calibre Auto-Waivers Warning Messages (cont.)

Message	Description
WARNING: Rule <rule> is defined in the WAIVE_DENSITY category but it has multiple output operations. Applying GLOBAL_DEFAULT criteria and ignoring WAIVE_DENSITY criteria.	A WAIVE_DENSITY rule check can contain only one layer operation with output to the DRC Results Database. The global default criteria, specified by GLOBAL_DEFAULT, is applied instead.
WARNING: Rule file has operation <operation_type> that is unsupported for rule checksumming. Rule checksum ignores it.	RUN_RULE_CHECKSUM YES is specified and an unsupported layer operation has been detected. The rule is skipped.
WARNING: Statement <statement_type> at line <line> is deprecated now.	A deprecated statement is used in the waiver setup file. Waiver setup file parser does not give error only to provide legacy support.
WARNING: Waiver setup command CHECKSUM_HEX mentioned at line <line_num> has been deprecated. HEX checksum generation is the default now.	Checksums for waivers are written in hexadecimal format only. No other checksum text formats are supported.
The following messages are issued by calibre -waiver only.	
WARNING: CALIBRE_WAIVER_NUMBER <number> is already in use. Calculating a non conflicting CALIBRE_WAIVER_NUMBER.”	CALIBRE_WAIVER_NUMBER is set to an integer value, either explicitly or by default. There is a layer in the design, or a mapped layer, that conflicts with this setting. The tool chooses a different layer automatically.
WARNING: Calibre version mismatch for the waiver CELL “<waiver_cell>.”	RUN_CALIBRE_VERSION_CHECKSUM YES is specified in the setup file and the Calibre version used for the verification run is older than the version used for generating Calibre version text objects in the waiver cell. Such cells are still validated.
WARNING: Calibre version text missing in the waiver CELL “<waiver_cell>.”.	RUN_CALIBRE_VERSION_CHECKSUM YES is specified in the setup file, but the specified waiver cell lacks a version checksum text object. Such cells are not validated.
WARNING: Could not extract WAIVER_CRITERIA file <file>.	WAIVER_CRITERIA EXTRACT is specified in the waiver setup file, but the extraction failed due to an internal error.

Table C-2. Calibre Auto-Waivers Warning Messages (cont.)

Message	Description
WARNING: No criteria was found to extract. Criteria extraction will be skipped.	WAIVER_CRITERIA EXTRACT is specified in the waiver setup file, but the file did not contain any criteria to extract. If this case, no waivers are applied. Check to make sure that the specified waiver database contains criteria.
WARNING: Could not open layout file <file>.	The layout file cannot be opened when checksum validation is requested in the waiver setup file.
WARNING: Could not perform checksum validation on layout file <file>.	The waiver setup file specifies RUN_LAYOUT_CHECKSUM YES, but there are no checksum text objects.
WARNING: Could not validate waiver CELL "<cell>", as checksum not present.	The waiver setup file specifies RUN_LAYOUT_CHECKSUM YES but there are no checksum text objects in the given cell.
WARNING: Excluding waiver cell <cell>.	An IP_MATCH cell was not geometrically matched. The corresponding cell in the design has no waivers applied.
WARNING: Ignoring Calibre version validation for the CELL "<waiver_cell>" since multiple Calibre version texts present in cell.	RUN_CALIBRE_VERSION_CHECKSUM YES is specified in the setup file, but the specified waiver cell has multiple version text objects. Such cells are not validated.
WARNING: Ignoring checksum validation for the cell "<waiver_cell>" since multiple checksums present in the cell.	Multiple checksum text objects exist in a waiver cell. Such cells are not validated.
WARNING: Ignoring waiver criteria for the cell "<waiver_cell>" since multiple criteria texts present in cell.	Multiple waiver criteria text objects exist in a waiver cell. Such cells are not validated.
WARNING: Invalid checksum for the cell "<cell>"	The cell has been altered since the checksum text object was inserted.
WARNING: IP_MATCH <cell> also present in the design, but did not match with the corresponding design cell, excluding its waiver cells.	An IP_MATCH cell is present in the WAIVER_DATABASE using the same name in the main design, but the cells are geometrically different.
WARNING: IP_MATCH <cell> did not match with any design cell.	An IP_MATCH cell in the WAIVER_DATABASE matches no cell in the main design.

Table C-2. Calibre Auto-Waivers Warning Messages (cont.)

Message	Description
WARNING: IP_MATCH not allowed when MERGE is set to YES.	IP_MATCH is ignored when MERGE YES or OUTPUT_LIBRARY is specified.
WARNING: LAYOUT INPUT EXCEPTION SEVERITY METRIC_INPUT_FILE is not 0. Setting it to 0.	The precision of the Layout Path design is different from the WAIVER_DATABASE waiver design. To accommodate the difference, this setting is adjusted at runtime. See “Waiver Cell Precision and Magnification Considerations” on page 110.
WARNING: LAYOUT INPUT EXCEPTION SEVERITY PRECISION_INPUT_FILE is not 0. Setting it to 0.	The precision of the Layout Path design is different from the WAIVER_DATABASE waiver design. To accommodate the difference, this setting is adjusted at runtime.
WARNING: LAYOUT INPUT EXCEPTION SEVERITY PRECISION_LAYOUT is not 0. Setting it to 0.	The precision of the Layout Path design is different from the WAIVER_DATABASE waiver design. To accommodate the difference, this setting is adjusted at runtime.
WARNING: Neither MERGE YES nor WAIVER_DATABASE is specified. The “waived.gds” is also not present in the working directory.	WAIVER_DATABASE is not specified and the default filename is not present in the working directory.
WARNING: No waiver rule present for CELL “<cell>” and RULE “<check>”.	The waiver criteria and cell(s) file contains a line that lacks waiver criteria for the given cell and rule.
WARNING: Precision mismatch could not be handled since (rule_file_precision)/(layout_precision) must be an integer and > 1. Set PRECISION_CONVERSION to YES.	The ratio of the rule file precision to the layout precision is less than 1. Use the PRECISION_LAYOUT YES setting in the setup file. This must be used with MERGE NO.
WARNING: Checksum verification failed for rule “rule_name”. No waivers applied to this rule.	This warning indicates that the rule checksum from the waiver cell does not match the rule checksum from the specified rule file. This indicates that the rule used to generate the waiver is different from the rule in the verification file.
WARNING: Waiver description file not specified.	A waiver description filename is not specified when using the EXTRACT or APPEND keywords in the waiver setup file.
WARNING: Waiver Type MAX has been deprecated. Please update your description file. Using SINGLE for this run.	The MAX keyword in the waiver description file was deprecated in the 2010.2 release. The SINGLE keyword is used instead.

Table C-2. Calibre Auto-Waivers Warning Messages (cont.)

Message	Description
The following messages are issued by waiver_flow only.	
WARNING: All the waiver cells present under the hierarchy CELL "<cell_name>" already present in INPUT_LIBRARY. Not merging hierarchical CELL.	Waiver cells already exist in the INPUT_LIBRARY for the given parent cell_name. The waiver cells are not written in this case.
WARNING: Checksum already present in cell <cell_name>, overwriting it.	A checksum text object is already present in the waiver cell "cell_name". The existing checksum is overwritten.
WARNING: DRC MAXIMUM RESULTS should be set to ALL, modifying it.	DRC Maximum Results ALL must be used for a waiver_flow run.
WARNING: Ignoring conflicting "<IGNORE/NOT_IGNORE>" specification for SVRF check "<DRC operation>" at line <line number>. It is specified more than once in the waiver setup file.	There are IGNORE and NOT_IGNORE statements for the same DRC operation in the waiver setup file. The run proceeds as if neither were specified.
WARNING: Multiple WAIVER_DATABASE statements found. Considering the first output waiver database <WAIVER_DATABASE> as valid output waiver database.	If more than one WAIVER_DATABASE statement appears in the waiver setup file, only the first instance is accepted.
WARNING: No waiver geometry present for CELL "<cell>" and RULE "<check>".	The given cell and rule are specified in the waiver description file, but there is no DRC result to waive for this combination.
WARNING: No waiver shape found in the waiver file <file_name>. Output not generated.	While creating merged output waiver database, there is no waiver shape present in the file. Hence, output is not copied to the generated database.
WARNING: No waiver shape found in the WAIVER_DATABASE file. Output file not generated.	No waiver shape found in the WAIVER_DATABASE file (or its default). Hence a merged output database is not generated. If the rule file is an ERC rule file, then the -erc option must be specified.
WARNING: Rule <rule_name> mentioned in waiver description file but not found in rule file.	The rule check specified in the waiver description file is absent from the rule file.

Table C-2. Calibre Auto-Waivers Warning Messages (cont.)

Message	Description
WARNING: Waiver CELL "<cell>" already present in INPUT_LIBRARY, therefore not merging.	There is a waiver cell present in the INPUT_LIBRARY specified in the waiver setup file and an additional DRC result is found for that cell. The additional result is not merged into the output. See " Waiver Cell Description " on page 218.
The following warning messages are issued by Calibre RVE.	
WARNING: Calibre version mismatch for the waiver CELL "<cell_name>".	There are Calibre version text objects present in a cell that conflict with the Calibre version of the tool.
WARNING: Merging not done since there is no waiver cell present in the waiver file "<file_name>".	The waiver file has no waiver cells.

Table C-3. Calibre Auto-Waivers Note Messages

Message	Description
NOTE: DATATYPE_NUMBER not specified, setting it to 5678.	The DATATYPE_NUMBER is not specified in the waiver setup file. The default datatype is used.
NOTE: LAYER_NUMBER not specified, setting it to 1234.	The LAYER_NUMBER is not specified in the waiver setup file. The default datatype is used.
NOTE: LAYOUT_SYSTEM not specified, setting it to GDSII.	The LAYOUT_SYSTEM is not specified in the waiver setup file. The default system is used.
NOTE: WAIVER_DATABASE not specified, setting it to waived.gds	The WAIVER_DATABASE is not specified in the waiver setup file. The default filename is chosen.
NOTE: Waiver setup command OUTPUT_LIBRARY mentioned at line <line number> has been deprecated. Use MERGE YES NO instead.	OUTPUT_LIBRARY is deprecated and should be replaced by MERGE in the setup file.
NOTE: Waiver setup command OUTPUT_WAIVER_DATABASE mentioned at line <line number> has been deprecated and renamed to WAIVER_DATABASE.	OUTPUT_WAIVER_DATABASE is deprecated and should be replaced by WAIVER_DATABASE in the setup file.
NOTE: Waiver setup file is <i>filename</i> .	The waiver setup file is specified as the <i>filename</i> .

Table C-3. Calibre Auto-Waivers Note Messages (cont.)

Message	Description
The following messages are issued by calibre -drc -waiver only.	
NOTE: CALIBRE_WAIVER_NUMBER not specified, setting it to 2000.	The CALIBRE_WAIVER_NUMBER is not specified. Layer 2000 is being used for internal processing of waiver shapes.
NOTE: CALIBRE_WAIVER_NUMBER set to <number>.	The AUTO keyword is specified and the <number> is the selected layer.
NOTE: Comment layer set to <number>.	Text object annotations for comments are on the specified layer.
NOTE: Date layer set to <number>.	Text object annotations for date are on the specified layer.
NOTE: Extracting WAIVER_CRITERIA file <file_name>.	The WAIVER_CRITERIA EXTRACT option is specified. The file is extracted as specified in the note.
NOTE: IP match required checksumming of <number> cells of the design <layout>.	IP_MATCH has been specified and the number of cells that had to be checked for a match are shown.
NOTE: IP_MATCH <cell> matched with design cell <cell>.	A design cell has been matched to an IP_MATCH cell.
NOTE: RUN_LAYOUT_CHECKSUM not specified, setting it to YES.	Checksum is performed by default. If checksum fails, other warnings are produced.
NOTE: Running checksum validation on <i>filename</i>	Checksum validation is being run on the file specified as <i>filename</i> .
NOTE: Username layer set to <number>.	Text object annotations for username are on the specified layer.
The following messages are issued by waiver_flow only:	
NOTE: Merging the individual waiver file to <file>	Individual waivers are being written to the <i>filename</i> .
NOTE: Merging the individual waiver file to <WAIVER_DATABASE>	Indicates where the WAIVER_DATABASE (or its default) is written.
NOTE: No waiver shapes are generated because the statements in waiver description file lack a waiver-producing statement.	The waiver description file has no statement that produces waiver shapes. Check the cell and rule names.

Table C-3. Calibre Auto-Waivers Note Messages (cont.)

Message	Description
NOTE: Parsing WAIVER_DESCRIPTION file.	Indicates the waiver description file has been detected and is being examined. Note, the waiver description has been deprecated. Use waiver cells and waiver criteria files instead.

drc_waiv_asc2gds Runtime Messages

The following table describes the runtime messages from the drc_waiv_asc2gds tool.

Table C-4. drc_waiv_asc2gds Messages

Message	Description
ERROR: No waiver cell written for CELL <cell> and RULE <rule> as shapes with vertex count > 8191 cannot be written in GDSII format.	The vertex count exceeds 8191 for a polygon in the layout database.
ERROR: Unable to write waiver shape for cell <cell>. Specify coordinates in cell space.	The DRC or ERC Cell Name YES statement was used without the CELL SPACE or XFORM keywords when the results database was generated. Use the CELL SPACE or XFORM keywords with YES.
ERROR: File <filename>.waived does not exist.	This message can occur if you incorrectly specify a .waived file with the -i option. If you want to translate a .waived file, you must specify your DRC results database with -i and a .waived file must exist in the same location with the extension .waived.
ERROR: Cannot open waiver comment file [<file_name>] for reading.	The waiver comment file is not accessible.
ERROR: Reading of waiver comment file failed at line <line_number>.	There is a problem in the waiver comment file that caused a read failure.
ERROR: Invalid format while parsing waiver comment file at line <line_number>.	The waiver comment file has a syntax error.
WARNING: No matches were found for <rule_check_name_or_filter>.	A rule in the comment file does not match any rule from the specified input file.
WARNING: Multiple comments were found for <rule_check_name_or_filter>. A random comment is chosen.	The waiver comment file contains multiple comments for the same rule. One comment is chosen randomly. This can occur if multiple wildcards match the same rule.

Table C-4. drc_waiv_asc2gds Messages (cont.)

Message	Description
WARNING: No comment is written for <rule_check_name_or_filter>.	The rule has an empty comment. You must specify the rule name and comment on the same line. Use brackets { } for multi-line comments.

Related Topics

[Calibre Auto-Waivers Runtime Messages](#)

[checksum_util Runtime Messages](#)

[waiver_util Runtime Messages](#)

checksum_util Runtime Messages

The following table documents the runtime messages from the checksum_util tool.

Table C-5. checksum_util Messages

Message	Description
ERROR: Failure to open input file <file> for read access.	The input layout database cannot be opened.
ERROR: Failure to determine primary cell within the input layout database.	The top-level cell cannot be determined.
ERROR: File <file_name>: Size exceeds 32 bits space, use checksum_util64.	The 64-bit version is needed for the input file.
ERROR: Invalid ignore type.	The -k option is specified with an invalid argument.
ERROR: Invalid file type.	The input file is not a valid format for this command.
ERROR: Option -<option> requires an argument.	An argument to an option is missing.
ERROR: Specified primary cell <cell> is not located within the input layout database.	The cell specified with the -p option cannot be located in the layout.
ERROR: Specified primary cell <cell> is an ignored cell.	The indicated primary cell from the -p option is also in the -f option file.
ERROR: Unknown option <option>.	Invalid option.
WARNING messages:	
WARNING: Failure to open ignore file <file> for read access.	File specified with the -f option cannot be opened.

Table C-5. checksum_util Messages (cont.)

Message	Description
WARNING: Failure to read ignore file at line <number>. Bad format.	File specified with the -f option has a problem at the returned line number. Reading of the file failed.
WARNING: Line <number> skipped in ignore file.	The indicated line in the -f option file has a format problem. Line is skipped and reading the file continues.
The following warnings are issued only when the -t oasis option is used. The warnings apply to the specified structures or records in the database.	
WARNING: Degenerate (vertex count = <number>) boundary at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed boundary object has been ignored.
WARNING: Degenerate (vertex count = <number>) polygon at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed polygon object has been ignored.
WARNING: Non-orientable boundary at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A boundary object has non-orientated vertex coordinates and has been ignored.
WARNING: Non-orientable polygon at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A polygon object has non-orientated vertex coordinates and has been ignored.
WARNING: Non-closed boundary at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed boundary object has been ignored.
WARNING: Zero-area boundary at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A boundary object has no area and has been ignored.
WARNING: Zero-area polygon at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A polygon object has no area and has been ignored.
WARNING: Non-simple boundary at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed boundary object has been ignored.
WARNING: Non-simple polygon at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed boundary object has been ignored.

Table C-5. checksum_util Messages (cont.)

Message	Description
WARNING: Degenerate (vertex count = <number>) path at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed path object has been ignored.
WARNING: Non-orientable path at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A path object has non-orientable coordinates and has been ignored.
WARNING: Non-simple path at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed path object has been ignored.
WARNING: Large vertex (>1024 vertices) path at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A path has too many vertices and has been ignored.
WARNING: Path of zero width at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A path has no width and has been ignored.
WARNING: 2-point boundary of zero length or width at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A 2-point boundary object has no length or width and has been ignored.
WARNING: Rectangle of zero length or width at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A rectangle object has no length or width and has been ignored.
WARNING: Circle of radius zero at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A circle object has no radius and has been ignored.
WARNING: CTRAPEZOID of zero area at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A CTRAPEZOID record has zero area and has been ignored.
WARNING: Degenerate CTRAPEZOID at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed CTRAPEZOID record has been ignored.
WARNING: TRAPEZOID of zero area at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A TRAPEZOID record has no area and has been ignored.

Table C-5. checksum_util Messages (cont.)

Message	Description
WARNING: Degenerate TRAPEZOID at location <x y> in cell <name> on layer <number> datatype <number> dropped.	A badly constructed TRAPEZOID record has been ignored.
NOTE messages:	
NOTE: INSTANCES are ignored in checksumming.	Cell instance checksums are not computed. Issued when the -k INST option is used.
NOTE: TEXT is ignored in checksumming.	Text layer checksums are not computed. Issued when the -k TEXT option is used.

Related Topics

[drc_waiv_asc2gds Runtime Messages](#)

[Calibre Auto-Waivers Runtime Messages](#)

[waiver_util Runtime Messages](#)

waiver_util Runtime Messages

The following tables document the error, warning, and note messages from the waiver_util tool.

Table C-6. waiver_util Messages

Message	Description
ERROR: Cell suffix file -s <file> does not exist.	The specified file is not present.
ERROR: Could not open layout file <file>.	The -i input file is not accessible.
ERROR: Incorrect GDSII/OASIS format in file <file>.	There is a problem with the layout format.
ERROR: Invalid format while parsing waiver description file at line <N>.	The waiver description file has a syntax problem.
ERROR: Invalid OASIS/GDS character “<character>” found in the suffix “<suffix>” for cell “<cell>” at file <filename>:<line>	There is an invalid character in the -s suffix file. This character cannot be used for cell names.
ERROR: Invalid waiver type defined at file <description file> : <line number>.	The waiver type may only be SINGLE or MULTI.
ERROR: Output file <file> not generated.	The input file may not have any of the specified cells for the run, or no waiver criteria have been updated for the input design. Review the transcript for other WARNING messages.

Table C-6. waiver_util Messages (cont.)

Message	Description
ERROR, Result overlap entry must be an integer between 0 and 100 inclusive at file <name>:<line>.	The waiver criteria file uses an incorrect number for the result overlap parameter.
ERROR: Specify proper layer number with -l switch.	The -l option is improperly specified. It must have a valid layer number as an argument.
ERROR: Waiver description file -d <file> does not exist.	The specified file is not present.
ERROR: Waiver overlap entry must be an integer between 0 and 100 inclusive at file <name>:<line>.	The waiver criteria file uses an incorrect number for the waiver overlap parameter.
WARNING: Ignoring cell <cell> since input layer <layer> is not a waiver layer for this cell.	No waiver shape is present on the specified layer for this cell.
WARNING: No conflicting criteria found in the input file "<file>" at the waiver layer. No criteria have been updated.	These are the possible causes: 1. The criteria file waiver criteria match what is found in the input design. 2. The -l layer is not a waiver layer for any cell.
WARNING: No waiver cell in the input file "<file>" matched with the entries in the waiver description file(s) "<description file>". No criteria have been updated.	These are the possible causes: 1. No waiver cell found in the input design. 2. No cell mentioned in the description file is found in the input design.
WARNING: Skipping dummy cell <cell>.	Empty cells with embedded waiver criteria that are generated by waiver_flow or Calibre RVE are not updated.
WARNING: Waiver description criteria found in the cell <cell> at layer <layer>. Using waiver description file(s) criteria instead.	Waiver criteria text objects exist in a waiver cell. The criteria from the waiver description file are written instead.
WARNING: -x switch has been deprecated. HEX checksum generation is default now.	The -x is no longer used because checksums are only formatted in hexadecimal.
NOTE messages:	
NOTE: Cell suffix file is <file>.	The -s option is used and the file is read.
NOTE: Merging layout file <input_layout_file> with waiver description file <desc_file>.	Waiver criteria in the input design are being merged with information in the waiver description file.

Table C-6. waiver_util Messages (cont.)

Message	Description
NOTE: Output file <file> generated.	The -o file has been generated.
NOTE: Updated waiver criteria for cell <cell> at layer <layer>.	Waiver criteria from the waiver criteria file have been added to the specified cell.
NOTE: Ensure texttypes do not collide among all the text objects.	The texttypes of text objects should be different from each other.

Related Topics

[drc_waiv_asc2gds Runtime Messages](#)

[checksum_util Runtime Messages](#)

[Calibre Auto-Waivers Runtime Messages](#)

— Symbols —

`[]`, 31
`{}`, 31
`|`, 31

— B —

Bold words, 31

— C —

`calibre -drc -waiver` command line, 186
Cell signature, 175
`checksum_util`, 203
Checksums for derived layers, 225
Command lines
 DRC waiver run, 186
 `drc_waiv_asc2gds`, 196
 `waiver_flow`, 105, 107
Command syntax, 31
Comment text, 222
Courier font, 31

— D —

Double pipes, 31
DRC or ERC verification with waivers, 38
DRC run for library IP, 183
DRC run with multiple waiver definitions files, 189
DRC waiver flow, 13
Dummy waiver cells, 228

— E —

Error result region, 18

— F —

Files
 Calibre rule file, 113
 `calibre-waiver.summary`, 114
 `density_waived.rdb`, 114
 input files, 113
 output files, 114
 `unused_waiver.rdb`, 114

`used_waiver.rdb`, 114
 `waived.gds`, 114
 `waived.rdb`, 116
 waiver cells file, 114
 waiver criteria file, 113
 waiver description files, 143
 waiver setup file, 113

Font conventions, 31

— G —

Geometry checksum, 221

— H —

Heavy font, 31

— I —

Intermediate cells, 219
`IP_MATCH`, 173, 175, 220
`IP_MATCH` checksum, 223
Italic font, 31

— L —

Library waiver generation, 36
Library waiver verification, 180

— M —

Minimum keyword, 31

— P —

Parentheses, 31
Permitted rule file elements, 27
Pipes, 31
Precision, 111
Primary criteria, 147, 222

— Q —

Quotation marks, 31

— R —

Requirements, 23
Rule checksum, 223
Rule conflicts, 146

— S —

Slanted words, [31](#)
Square parentheses, [31](#)
Standard form criteria rule, [148](#), [150](#)
Syntax conventions, [27](#), [28](#), [29](#), [30](#)

— T —

Terminology definitions, [17](#)

— U —

Underlined words, [31](#)
Usage models, [20](#)
Usage syntax, [31](#)

— V —

Version, [223](#)

— W —

Waiver criteria file
 WAIVE_MARKER, [161](#)
Waiver, [17](#)
Waiver cell, [18](#), [218](#)
 text object annotations, [221](#)
Waiver cell naming, [218](#)
Waiver cells file, [173](#)
Waiver criteria file, [147](#), [183](#)
 input_layer, [164](#)
 input_layer_count, [164](#)
 library_name, [164](#)
 marker_layer, [161](#)
 MULTI, [151](#)
 NONE, [150](#)
 options_string, [164](#)
 output_layer_count, [164](#)
 output_layer_list, [165](#)
 pattern_dmacro_file, [164](#)
 result_overlap, [152](#)
 SINGLE, [151](#)
 WAIVE_DENSITY, [167](#)
 WAIVE_EXTENT, [158](#)
 waiver_overlap, [151](#)
 waiver_type, [150](#)
Waiver flow, [13](#)
Waiver setup file, [116](#)
Waiver setup file commands, [117](#), [126](#)
Waiver shape, [17](#)
Waiver shape interaction examples, [54](#)

Waiver text, [221](#)
Waiver type summary, [60](#)
waiver_flow command line, [105](#)

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.

