SIEMENS EDA

Calibre® Release Notes

Software Version 2021.2
April 16, 2021
Document Revision 18

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
18	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes are listed in this document for all products in this release. Approved by Michael Buehler.	Released April 2021
17	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes are listed in this document for all products in this release. Approved by Michael Buehler.	Released January 2021
16	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes are listed in this document for all products in this release. Approved by Michael Buehler.	Released October 2020
15	Modifications to improve the readability and comprehension of the content. Approved by Lucille Woo. All technical enhancements, changes, and fixes are listed in this document for all products in this release. 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 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

Chapter 10

Chapter 1
Overview and Support

The following sections in this chapter provide useful information for this release:

Document Overview

These release notes provide release information for the following areas:

- "General Calibre Release Information" on page 13
- "Calibre Product Documentation" on page 17
- "Calibre Examples and Videos" on page 21
- "Calibre Platform and General Products" on page 29
- "Physical Verification" on page 39
- "Design for Manufacturability (DFM)" on page 135
- "Parasitic Extraction (PEX)" on page 163
- "Resolution Enhancement Technology (RET)" on page 169
- "Mask Data Preparation (MDP)" on page 259

Global Customer Support and Success

A support contract with Siemens Digital Industries Software is a valuable investment in your organization's success. With a support contract, you have 24/7 access to the comprehensive and personalized Support Center portal.

Support Center features an extensive knowledge base to quickly troubleshoot issues by product and version. You can also download the latest releases, access the most up-to-date documentation, and submit a support case through a streamlined process.

https://support.sw.siemens.com

If your site is under a current support contract, but you do not have a Support Center login, register here:

https://support.sw.siemens.com/register

Siemens Digital Industries Software Training

The Siemens EDA training catalog contains a large variety of instructor-led courses. Classes are available in Siemens Digital Industries Software worldwide training centers, onsite at your workplace, and online with live instructors.

Course descriptions and class schedules can be viewed at:

https://eda.learn.sw.siemens.com/training/course_categories/calibre

Corrected Defects and Known Issues

You can view a complete list of this release's corrected defects and known issues on Support Center.

1. Log on to Support Center and click the **Downloads** page:

   https://support.sw.siemens.com/product/852852053/downloads

2. Select a version from the dropdown list.

   The page updates to display the associated release(s).

3. Click the desired version.

   This displays a list of Release Documentation, including the Corrected Defects and Known Issues documentation, associated with the release.

Tracking Support Cases, DRs, and ERs

If you are a registered Support Center user, you can open a new support case and view the status of your own support cases, including the status of any attached Defect Report (DR) or Enhancement Request (ER). You can also view all support cases for your site, including all associated DRs and ERs.

https://support.sw.siemens.com/support-case

To be kept informed of the progress of a particular DR or ER, open a support case and indicate the DR or ER number you are interested in.

Chapter 2
General Calibre Release Information

The following sections provide general information about the Calibre release:

Platform Support and Roadmap

The Calibre Platform Support Overview and Roadmap provides the latest information on hardware architectures and operating systems that Calibre currently supports and intends to support in the upcoming year. The roadmap is accessible at:

`http://calibre.mentorcloudservices.com/docs/Calibre_OS_Roadmap.htm`

Note

While the Calibre Release Notes includes notices for the most recent platform discontinuations, it is highly recommended that you review the platform support and roadmap information at the URL shown above for detailed information regarding operating system and hardware platform support.

Additional information on supported platforms and running Calibre in a mixed operating system environment is available at "Supported Operating Systems and Hardware" in the *Calibre Administrator's Guide*.

RHEL 5 Discontinuation Notice

Support for the RHEL 5 operating system was deprecated starting with the 2017.1 release and is discontinued with the 2021.1 release.

Support for RHEL 8

The 2021.1 Calibre release includes support for Red Hat Enterprise Linux version 8 (RHEL 8). The Calibre software for the RHEL 8 operating system is packaged into the AOK executable.

The rsh (remote shell) program supported on previous RHEL operating system versions (7 and earlier) is not supported on RHEL 8, being superseded by the ssh program. The following list summarizes some of the changes that have already been made in Calibre products in preparation for support of RHEL 8 and ssh, in addition to recommended changes you should make to your system settings and scripts.

- Calibre products that run in a Calibre MTflex environment
    - Prior to the Calibre 2020.3 release, Calibre MTflex uses */usr/bin/rsh* by default.
    - Beginning with the Calibre 2020.3, Calibre MTflex first attempts to use */usr/bin/rsh*. If */usr/bin/rsh* is not found, then Calibre MTflex uses */usr/bin/ssh*.
    - Both prior to and after this change, the default Calibre MTflex behavior can be overridden to use ssh using either of these methods:
        - Set the CALIBRE_MTFLEX_LAUNCH environment variable to “RSH */user/bin/ssh*”. For example:

          ```
          setenv CALIBRE_MTFLEX_LAUNCH "RSH /usr/bin/ssh"
          ```

        - If using the -remotefile argument to specify a configuration file, set the RSH argument for the LAUNCH AUTOMATIC command to specify the path to the rsh or ssh program. For example:

          ```
          LAUNCH AUTOMATIC ... RSH "<path_to_rsh_or_ssh_program>"
          ```

- Calibre Cluster Manager (CalCM) and CalScope

  These products do not depend on rsh. However, if you have system settings or scripts that use rsh, you should change them to use ssh.

- Calibre Interactive

  Limited AOK support for the classic Tk version and the new Qt version is available by setting the CALIBRE_QA_CI_AOK environment variable to 1. For example:

  ```
  setenv CALIBRE_QA_CI_AOK 1
  ```

  There are existing GUI settings in Calibre Interactive that you can use to enable ssh instead of rsh.

- Calibre LFD

  With the Calibre 2020.3 release, the Calibre LFD Model-Based Hints (MBH) flow uses the ssh program instead of rsh when run using Calibre MTflex.

- Calibre MDPDefectAvoidance
    - In releases prior to 2020.3, rsh was used for index file generation for an Extended Job Deck that used remote machines in parallel. With the 2020.3 release, rsh was replaced by ssh.

- Calibre nmModelflow
    - Uses passwordless ssh by default in the example scripts, called “runscripts”, that Calibre nmModelflow uses to dispatch jobs to local network hosts.
    - You can selectively switch back to rsh. Refer to the instructions provided with the online help text for the runscript file.

Support for MGLS v2018_2 and FlexNet v11.16.0

With the 2019.2 release, the AOI and AOJ Calibre trees use MGLS v2018_2, which requires the FlexNet v11.16.0 licensing software.

Refer to the “Platform Support and Roadmap” on page 13 for information on accessing the “Calibre Platform Support Overview and Roadmap” on Support Center.

For more information about MGLS or FLEXnet, refer to *Licensing Mentor Graphics Software* (mgc_licen.pdf) or the *FLEXnet License Administration Guide* (flexnet_lic_admin.pdf). You can access the licensing documentation from the Calibre InfoHub. The licensing software is available on Support Center.

Google Chrome Browser Not Supported on RHEL 6

The Google Chrome browser is not supported on the RHEL 6 operating system. Using Chrome may cause core files to be written.

Chapter 3
Calibre Product Documentation

This chapter provides information on accessing product documentation, in addition to describing the 2021.2 changes and enhancements for Calibre documentation and related products.

Calibre Documentation Overview

The Calibre documentation includes user, reference, quick reference cards, and process documentation in both HTML and PDF formats to offer you the various features and benefits of each format. The actual content does not vary between the formats.

The Calibre documentation includes the InfoHub, a product information portal that provides links to all locally installed HTML and PDF documentation. The InfoHub includes a search interface for searching all locally installed HTML documents. The search interface also enables you to submit a search directly to Support Center.

The Siemens Software and Mentor Documentation System requires a browser and version 7 or higher of Adobe® Reader®. Browser requirements are described in "Browser Settings" in the *Siemens Software and Mentor Documentation System* manual.

For information on configuring and using the Siemens Digital Industries Software documentation, refer to the "Managing Calibre Documentation" chapter in the *Calibre Administrator's Guide*.

Calibre Documentation Installation

The Calibre documentation is packaged with the Calibre software tree for both the software CD and software download file. The installation process automatically installs the documentation at the same level as the Calibre software tree. The Calibre software tree contains a link to the documentation directory.

Refer to Configuration: Managing Calibre Documentation in the *Calibre Administrator's Guide* for information regarding the documentation software tree and installing the documentation.

New Product Documentation and Documentation Changes

The following documentation changes are effective with the Calibre 2021.2 release.

- **Calibre LSG for Synthetic Layout Generation User's Manual**

  A new manual documents the new Calibre LSG Synthetic Layout Generator (Calibre LSG) software tool. This tool is used in conjunction with Calibre WORKbench to create user-defined pattern definitions for randomly-generated layout clips. The layout clips are suitable for pattern analysis and other early design phase applications for specific layers and technologies.

A series of introductory and getting started videos are included in the Calibre documentation tree. The getting started videos are accessible from the *Getting Started with Calibre Video Series* document and the Calibre InfoHub. The Calibre documentation contains links to “how to” videos, which are also accessible from the Calibre InfoHub.

Known Problems and Workarounds

The documentation has some known problems and workarounds.

Broken Links in Documentation When Using Adobe Reader on Linux

When browsing documentation, some links between PDF manuals do not resolve when using Adobe Reader version 8 or 9 on Linux. If you encounter this issue, the workaround is to use Adobe Reader on Windows or Mac OS.

Linux Version of Adobe Reader No Longer Available

The Linux version of Adobe® Reader is no longer available for download from Adobe. The Calibre InfoHub and HTML documentation are included with the Calibre software tree as a solution for accessing and searching the user documentation in a Linux environment. The InfoHub and HTML documentation provide both global search and support of linking between documents. By default, the mgcdocs command and GUI Help menu items will invoke the Calibre InfoHub.

Broken Links in PDF Documentation and Benefits of HTML Manuals - (MG595892)

Due to enhanced security restrictions with web browser PDF plug-ins, some links do not function. Links in HTML documentation are fully functional.

Clicking a link within a PDF viewed in a web browser may result in no action, or it may load the title page of the current PDF manual (instead of the intended target in the PDF manual). The unresolved link behavior occurs in all web browsers on Windows® and Linux® platforms. Because of this behavior, the navigational experience of PDF manuals is compromised. PDF is ideal for printing because of its page-oriented layout.

Use the HTML manuals to search for topics, navigate between topics, and click links to examples, videos, reference material, and other related technical content.

For information on Adobe's discontinued support of Adobe Reader on Linux platforms and your available options, refer to Knowledge Article MG596568 on Support Center.

Providing Feedback

We welcome your feedback on the Calibre documentation. A link to a Documentation Feedback form is available at the end of the bookmark list for all PDF documents. For HTML documents, the link is at the end of the title page.

If interested, you can discuss your feedback with the technical writer responsible for the documentation for your product line.

Chapter 4
Calibre Examples and Videos

The following sections in this chapter provide information on Calibre examples:

Calibre Tutorials and Example Kits (eKits)

Tutorials and eKits contain design data, rule files, and instructions for running and learning more about the tool.

Note

The Calibre tutorials and eKits are now located on the **Documentation** tab in Support Center, rather than on the **Downloads** > **Related Downloads** tab.

To download from Support Center:

1. Select the **Documentation** tab.
2. Under Document Types, select "Getting Started Guide."

The link goes to Support Center, filtered for Getting Started Guide. If no eKits are listed, choose a release of 2019.3 or later in the "Restrict content to version" dropdown list.

The following tutorials and eKits are available, grouped by product area.

General and Physical Verification Tutorials and eKits (general eKits)

- **Try It: Calibre Tools Quick Start and Example Kit** — Provides a simple, but effective, introduction to the Calibre tool suite. The document and data are organized into seven modules, with each module taking approximately 30 minutes to perform. The IC data represents an ASIC that is designed and built in a 45nm non-proprietary process.
- **Try It: Calibre Layout Comparison Tutorial and Example Kit** — Demonstrates Calibre Layout Versus Layout (LVL) comparison tools and flows. Specifically, FastXOR, standard dual-database XOR, and DBdiff are used to perform LVL comparisons. Calibre 2010.4 or later is required.
- **Try It: Calibre Gray Box Tutorial and Example Kit** — Demonstrates how to perform block exclusion using different waiver criteria options.
- **Try It: Calibre 3DSTACK Debugging Shorts Tutorial and Example Kit** — Demonstrates how to find shorts using path isolation in Calibre 3DSTACK.
- **Try It: Calibre Advanced DRC (eqDRC) Tutorial and Example Kit** — Includes examples of Calibre equation-based DRC (eqDRC) and advanced DRC rule checks. The examples make use of DFM properties and DFM expressions to implement rule checks that are difficult to write with traditional DRC methods. The examples correspond to the examples in the "Advanced DRC Solutions" chapter of the *Calibre Solutions for Physical Verification* manual.

Design for Manufacturability Tutorials and eKits (DFM eKits)

- **Try It: Calibre Critical Area Analysis Tutorial and Example Kit** — Includes documentation and data for performing Critical Area Analysis (CAA) with Calibre

YieldAnalyzer. The procedures step you through running CAA on a chip, CAA with marker layers, CAA on a cell library, and CAA with No Defect Density (NDD).

- **Try It: Calibre Critical Feature Analysis Tutorial and Example Kit** — Includes documentation and data for performing Critical Feature Analysis (CFA) with the Calibre YieldAnalyzer and Calibre YieldEnhancer tools. This tutorial walks you through building an analysis database, assessing design quality, locating and prioritizing fixes for the worst problems, and locating and prioritizing fixes for areas most likely to fail.

- **Try It: Calibre CMPAnalyzer Tutorial and Example Kit** — Contains all of the files needed to run a batch CMP analysis using the Calibre CMPAnalyzer simulator flow. In addition, instructions are presented on how to calculate a user-defined hotspot, run a CMP batch bucketing flow analysis, and export CMP thickness data for parasitic extraction tools.

- **Try It: Calibre YieldEnhancer Via Tutorial and Example Kit** — Contains the documentation, scripts, and sample data for running the Calibre YieldEnhancer via enhancement flow. This flow is used to increase design robustness and reduce yield loss by improving the via transitions as part of the Calibre design for manufacturability (DFM) methodology. In the procedures, you use via doubling to insert second vias into a design and then you review the updated layout.

- **Try It: Writing Critical Feature Analysis Rule Checks Tutorial and Example Kit** — Includes the documentation and data for writing CFA rule checks. The procedures provide a hands-on introduction to the concepts described in the "Assessment Metrics" section in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

Resolution Enhancement Technology Tutorials and eKits (RET eKits)

- **Try It: Calibre OPCverify Tutorial and Example Kit** — Contains known working SVRF rule files, optical models, and a sample OASIS®[1] design database. Accompanying documentation explains each annotated rule file and highlights expected output. Users of this eKit can learn from the example code and conduct modifications to see the effects of their changes.

See the following topic for detailed download instructions:

Downloading Calibre Tutorials and Example Kits (eKits) From Support Center

The following steps describe how to download and install Calibre tutorials and eKits.

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

The Calibre tutorials and eKits are packaged as zip files and listed with the product documentation in the "Getting Started Guide" document type.

1. Log on to Support Center and go to the Calibre Documentation page:

   `https://support.sw.siemens.com/en-US/product/852852053/documentation`

2. Choose a release of 2019.3 or later in the "Restrict content to version" dropdown list.
3. In the left sidebar under Document Types, click "Getting Started Guide."
4. (Optional) Under Categories, select your product area.
5. Click the link for the tutorial or eKit you want to download and follow the instructions for your browser to save the file.
6. In a Linux terminal window, browse to the download directory.
7. Uncompress the zip file:

   ```
   unzip filename.zip
   ```

The documentation for performing the example or tutorial is packaged with the data.

Calibre Videos

The Calibre InfoHub and documentation includes links to the several getting started and how-to videos.

No new videos were added in the 2021.2 release.

The following videos were added in recent releases:

- **Getting Started with the Calibre Pattern Generator** — Calibre WORKbench supports the Calibre Pattern Generator (CPG) for the flexible creation of basic and complex test patterns for test layouts. This series of three videos includes an overview of the Calibre Pattern Generator and demonstrations of pattern and layout creation using the Pattern Creator and Layout Generator tools. (2021.1)
- **Fast Rigorous Modeling (FRM) Modes in Calibre nmModelflow** — The Calibre nmModelflow Flow Stage Wizard added support for the Fast Rigorous Modeling (FRM) resist model. This video shows you how to select and use the modes that include FRM models. (2020.4)
- **How to Review Calibre RealTime DRC Results** — You can use the Calibre RealTime Results Window to explore Calibre RealTime DRC results. This video guides you through a basic procedure to review and highlight results. (2020.3)
- **Calibre RET Flow Tool: Simulation Flow** — The Calibre RET Flow Tool v2.0 is now the way to perform basic simulation operations. This video was updated to show you how to run the new interface. (2020.2)

- **Pattern Match With Count Markers** — You can easily run pattern matching from an open layout and add count markers as a compile option. Count markers provide an accurate count of pattern matches and enable you to know if pattern matches overlap. (2020.2)

The Calibre InfoHub also includes links to the available videos on Support Center and includes a list of the videos available with the documentation.

Calibre Documentation Examples

The Calibre documentation is continually being enhanced with new examples.

New Examples

- "Resizing Polygons by a Percentage of the Area" in the *Calibre Solutions for Physical Verification* manual uses eqDRC and the DFM Size operation to resize polygons by a percentage of their area.

  This new example is included in the 2021.2 update of the Calibre Advanced DRC (eqDRC) Tutorial and Example Kit. See "Calibre Tutorials and Example Kits (eKits)" on page 22.

- Example 5 in DFM Fill demonstrates some special behavior of the ENVELOPE keyword with a multilayer FILLSTACK. In addition, the explanation of the ENVELOPE keyword was updated and includes a new figure.
- Example 17 in DFM Spec Fill uses INSIDE OF LAYER with and without the BY POLYGON keyword to demonstrate the effect of the keyword.
- Example 18 in DFM Spec Fill uses INSIDE OF LAYER and compares fill insertion with the INITIAL FILLREGION and INITIAL INSIDE OF LAYER keywords, which control the fill placement grid. In addition, the descriptions and figures for the INITIAL keywords were updated.

Examples Added in Previous Releases

- "Preventing Empty Fill Areas" in the *Calibre YieldAnalyzer and YieldEnhancer Reference Manual* has a new example that uses the NOEMPTY keyword in DFM Spec Fill Optimizer. (2021.1)
- The examples for DFM Spec Fill with FILLSTACK are new or updated. See "Example 6" and "Example 7" with DFM Spec Fill in the *Standard Verification Rule Format (SVRF) Manual*. (2021.1)
- "Finding the Length of a Net Path Connected to Gate: Effective Length Method" in the *Calibre Solutions for Physical Verification* manual provides a new method for finding the length of a net path. The method uses an equation to determine the effective length of the polygons in the net path. (2020.4)

- Example 8 for DFM Property Select Secondary demonstrates how to create a DMACRO for an operation that uses the :EDGE layer derivation notation. The :EDGE, :EXTENT, and :EXTENT_EDGE notations cannot be used within the DMACRO but must be passed in as an argument. Example 11 for DFM Property demonstrates a DMACRO using :EXTENT in a DFM Property operation. (2020.4)
- Two example kits and tutorials were released (2020.4):
    - **Try It: Calibre Gray Box Tutorial and Example Kit** — Demonstrates how to perform block exclusion using different waiver criteria options.
    - **Try It: Calibre 3DSTACK Debugging Shorts Tutorial and Example Kit** — Demonstrates how to find shorts using path isolation in Calibre 3DSTACK.

  See "Downloading Calibre Tutorials and Example Kits (eKits) From Support Center" on page 23.
- Several examples in the "Advanced DRC Solutions" section of the *Calibre Solutions for Physical Verification* manual were updated. (2020.3)
    - "Checking Maximum Via Coverage for Metal Overlap Regions" finds metal enclosure regions that contain fewer than the maximum number of vias based on the enclosure, width, and space constraints. The rule check now also outputs the percentage by which the via area is smaller than expected. The new output and updated error checking catches cases for which the number of vias is correct, but the vias are undersized. The derivation now uses the default INTERSECTING clustering method, which is more appropriate for the rule check intent.
    - "Finding Pad Edges Closest to the Chip Edge" now includes the pad edge to chip edge distance on the output markers. The rule check also uses a variable for the pad to chip enclosure distance, for easier rule file maintenance.
    - "Finding Unique Violations" outputs unique violations between a set of polygons. The DFM Classify operation is used to output only one marker polygon from a set of redundant interactions. (This example was previously titled "Finding Redundant Interactions.")
- Several examples in the *Calibre Solutions for Physical Verification* manual were updated. (2020.2)
    - "Finding Circular Polygons" now uses new methods introduced in the 2020.1 release. One example uses the new DFM Circle Analyze operation. The second example uses Inside Cell with the new CIRCLE keyword.
    - "Finding Polygon Extents for Special Configurations" demonstrates two different methods for finding extents of overlapping polygons. One uses the :EXTENT_EDGE layer derivation within the measurement function of DFM Property. The second method uses the :EXTENT edge layer derivation in DFM Property Select Secondary to output an unmerged polygon layer with the polygon extents.

- “Finding Length and Width Edges of Rectangles” now uses DFM Property Select Secondary with the :EDGE layer derivation to calculate the edge lengths and output the selected edges. The DRC_NE function is used to perform a not-equal comparison with a built-in tolerance. The example also handles squares.

- “Checking Equal Gate Pitch per Active Area” now has an optional layer derivation using DFM Property Select Secondary to output the width and spacing error layers. It also uses the DRC_NE function to perform a not-equal comparison with a built-in tolerance.

Chapter 5
Calibre Platform and General Products

This chapter provides information about updates to general Calibre products and operation.

Calibre FullScale Updates

This section describes the 2021.2 changes to the Calibre® FullScale™ platform.

For more information, see "Calibre FullScale Platform" in the *Calibre Post-Tapeout Flow User's Manual*.

2021.2 Updates

This release improves support for the following SVRF statements:

- DFM OR EDGE is supported.
- DFM PROPERTY can now use CORNER and produce numeric vector output.
- DFM PROPERTY SELECT SECONDARY is supported, except for NOMULTI, :EDGE, :EXTENT, :EDGE_EXTENT, and measurements on derived layers.
- HOLES can now use SINGULAR ALSO.

Previous Release Highlights

The 2021.1 release improves support for DFM PROPERTY, DFM SEGMENT, DFM SHIFT EDGE, and LAYOUT PROPERTY AUDIT. Additionally, it now reports GEDB as two values in the transcript and provides information on how PFSDB runs are adjusting load.

The 2020.4 release adds support for resuming a Calibre FullScale run after a hardware failure and some types of software failure. There are two new environment variables to enable this functionality: CALIBRE_FS_ENABLE_CHECKPOINTING and CALIBRE_FS_RESUME_SESSION.

The 2020.3 release adds support for DFM STRIPE, GROW, and SHIFT, INSIDE OF LAYER for supported LITHO and RET commands, various optional arguments for DFM commands. Additionally, the PFSDB statistics in the transcript have been expanded to include information on files and deletion.

The 2020.2 release adds the keyword “MFG” to Layout System. As of this release, rule files must use Layout System MFG to read VSB and MEBES input. See “Using MEBES and VSB Input” in the *Calibre Post-Tape Flow User’s Manual*. Prior to 2020.2, MEBES could be read in using Layout System OASIS but this now causes runs to exit with an error.

The 2020.2 release also adds three environment variables:

- CALIBRE_FS_RDB_DISABLE_INDEX controls how DFM RDB rule check names are written, affecting Calibre RVE presentation. By default, the output from a Calibre FullScale run groups the results for each section’s rule check individually. When this variable is set, the section identifier is removed in the RDB and Calibre RVE presents all the results of a rule check as a single group.
- CALIBRE_FS_MAX_READ_THREADS and CALIBRE_FS_MAX_WRITE_THREADS limit the number of concurrent input and output threads respectively.

Calibre Cluster Manager (CalCM) Updates

This section describes the 2021.2 changes to Calibre® Cluster Manager (CalCM).

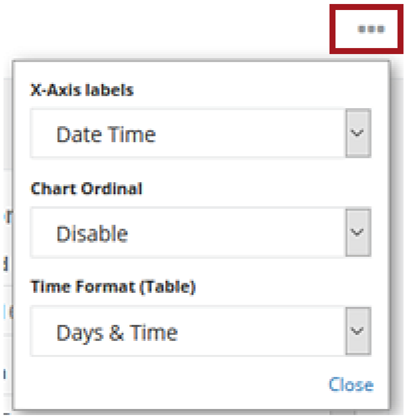
See the *Calibre Cluster Manager (CalCM) User’s Manual* for details.

2021.2 New Functionality

The 2021.2 release includes the following CalCM enhancements:

- A new time display option in the Resource Monitor and Resource Usage pages now enable you to choose how time is displayed in tables. You can choose between Hours or Days & Time.

Click the ellipsis icon in the upper right of the Resource Monitor or Resource Usage pages to access this setting.



X-Axis labels
Date Time

Chart Ordinal
Disable

Time Format (Table)
Days & Time

Patch Requirements for CalCM

See the "Calibre Release and Patch Requirements for CalCM" table in the *Calibre Cluster Manager (CalCM) User's Manual* for the recommended patch levels that support CalCM.

CalCM Supported Platforms

CalCM supports the IBM® Platform™ LSF® and Univa® Grid Engine® platforms by providing methods to integrate with different platform configurations. For CalCM platform customization to your environment, refer to the *Calibre Cluster Manager (CalCM) User's Manual* and your Siemens representative for assistance.

The default platform configuration for CalCM is to control Calibre jobs with a platform tool and remotes with remote shell. For remote shell users, CalCM uses the Linux secure shell (SSH) by default.

Note

CalCM runs with Calibre remotes should use an SSH authentication key instead of password access.

Previous Release Highlights

The 2021.1 release includes the following CalCM enhancements:

- (2021.1) The **License Usage** > **Group** page now supports Line and Area chart types.
- (2021.1) A new option JOB RULEFLAGS has been added to specify an argument string that is added to the Calibre invocation generated by CalCM. The specified argument is added to the command line just before the rule file name. It is typically used in the Defect Avoidance flow.

- (2021.1) The detailed Job page now shows notifications for all users by default. Previously, you had to log in to see notifications.
- (2021.1) The "Reorder or hide columns" dialog box has been improved to better display long lists of columns.

The 2020.4 release includes support for the following CalCM enhancements:

- (2020.4) Updated functionality in the Compare Jobs page (CalCM dashboard web interface):
  - Compare job plots in a new compact view format.
  - Change the sorting order of the job table and plots for efficient analyses of run data. For example, you can sort by jobs with longest runtimes, latest start/end times, or sort by job ID numbers.
  - Select one job as a baseline and compare job performance metrics for elapsed time, runtime, maximum number of remotes, number of allocated and active remotes.

The 2020.3 release includes support for the following CalCM enhancements:

- (2020.3) New commands:
  - check_cluster_limit *count*

    This optional command to the calcm_send_message script previews the result of adjusting the CLUSTERLIMIT value.
  - SCANJOBINFO = *flag*

    This optional command specifies whether to scan the job transcript file to gather job information and record it in the info_jobs table. The flag values are as follows:
    - 0 — No scanning is done. (default).
    - 1 — Scanning and recording are done.
- (2020.3) Updated commands:
  - EVENTFILTERS = *regexp*

    This command is updated with optional Tcl procedures "eval1Event", "eval1Filter", and "eval1FilterExpr", which are used to only report a notification once.
  - JOBFILTERS = *proc*
    - This command is updated with optional Tcl procedures "eval1Event", "eval1Filter", and "eval1FilterExpr", which are used to only report a notification once.
    - This command is updated with the optional variable "total", which is used for filtering the number of positive results that occur for a condition and not reset.

- (2020.3) Updated functionality (CalCM dashboard web interface):
    - Refresh chart data either manually or automatically to perform a live update of the Jobs detail pages and the System Utilization page:
        - Click the manual refresh icon in the upper-left corner of the page to refresh the page and chart data without reloading it.
        - Click the dropdown menu icon to the right of the refresh icon to see a list of automatic refresh intervals you can select from.
    - Keep selected check boxes and column search inputs when you click the browser back icon. Implemented only in the Active page and Finished job page.
    - Show, hide, and control select chart features:
        - Ctrl-click the **Zoom Out** icon to reset the zoom on all plots.
        - Ctrl-click the **Link Chart** icon to toggle chart linking globally, across all charts.
        - Ctrl-click the **Collapse** dropdown menu to show or hide all chart panes and tables.
        - Ctrl-click the chart legend to show or hide a plot line globally, across all charts.
        - Ctrl-double-click the chart legend to hide or show *all* plots lines globally, across all charts.

The 2020.2 release includes support for the following CalCM enhancements:

- (2020.2) Updated commands:
    - QUOTAOVERDRAFTLEVEL = *range*

      This command is enhanced to accept the level where the overdraft can be applied.
    - EVENTFILTERS = *regexp* …

      This command is enhanced to use with the evalFilterExpr Tcl procedure and the condition argument.
    - EVENTTRACKERS = *tracker_list*

      This command is enhanced with an optional condition argument that determines whether or not the event tracker is enabled.
    - JOBEVENTFILTERS = *regexp*

      This command is enhanced to filter events using JobAttr in the condition argument.
    - JOBEVENTTRACKERS = *tracker_list*

      This command is enhanced to track events using JobAttr in the condition argument.

- (2020.2) Updated functionality (CalCM dashboard web interface):
  - Jobs webpage enhanced functionality.
    - Select a common x- and y-axis.
    - Use a job comparison feature.
    - Display time and date, elapsed time, or epoch format.
    - Choose to display all running jobs in the Active Jobs and Finished Jobs pages.
  - System Utilization webpage enhanced functionality.
    - Show updated plot information for total and allocated RDS and job slots.
  - View MASTERFILTERS notifications in the Job detail webpage and Notifications webpage.

Calibre Cluster Manager Plus (CalCM+) Updates

This section describes the 2021.2 changes to Calibre® Cluster Manager Plus (CalCM+).

For further information on CalCM+, refer to the *Calibre Cluster Manager (CalCM) User's Manual* for details.

2021.2 New Functionality

The 2021.2 release includes the following CalCM+ enhancements:

- This release introduces the Calibre® System Monitoring Solution (CalScope) included with CalCM+.

  CalScope provides a system-level solution for monitoring hardware information and extracting job information for data collection and Calibre run analysis.

  With CalScope, administrators and end-users of Calibre jobs in high-performance computing environments gain insight to critical system information through the job analysis and hardware monitoring capabilities of this tool.

  - **Job Importing** — Provides job and key CPU performance information from the transcript of a completed Calibre job and saves it to a database. The collected information is used for run and post-run analysis, and job and hardware correlation. Analysis can identify job performance impact due to hardware-related issues.
  - **Hardware Monitoring** — Provides hardware information for identifying potential issues and saves it to a database. This functionality enables interaction with monitored hosts by providing status information, enabling add and remove of monitored hosts, and other actionable items.

- **Alert System** — Provides a built-in alert system configured to monitor alerts for job and hardware events. This functionality notifies CalScope end-users of problems occurring in the monitored cluster and enables mitigating action based on predefined criteria.

For more information, refer to "CalScope Usage and Reference" in the *Calibre Cluster Manager (CalCM) User's Manual*.

- You can now choose to view all jobs in the **Data Analysis > Cluster Utilization** page by checking **No Aggregate (view all jobs)**. It is recommended to choose a relatively small Period when using this feature to prevent too many jobs from loading.

Previous Release Highlights

The 2021.1 release was the first release of Calibre® Cluster Manager Plus (CalCM+).

CalCM+ is a new CalCM package that offers the CalCM standard features and also includes advanced features such as data analysis and visualization, efficient resource selection and planning, and a fully-integrated system monitoring solution.

Calibre Dynamic Resource Allocator (DRA) Updates

This section describes the 2021.2 changes to Calibre® Dynamic Resource Allocator (DRA).

2021.2 New Functionality

The 2021.2 release includes the following DRA enhancements:

- You now have the option in the calibre_dra command to specify how long to keep looking for controller information in the log file before exiting:

```
calibre_dra -waitcontroller <seconds>
```

  This can be used to avoid showing an error when the Calibre job has just started and is not yet ready to communicate with calibre_dra. The default is to not wait.

Previous Release Highlights

The 2021.1 release includes minor updates and bug fixes.

The 2020.4 release includes the following DRA enhancements:

- (2020.4) Calibre DRA Direct Connect Mode

  You now have the option to specify an IP address instead of a host name when adding or removing a resource using the calibre_dra command -add or -remove arguments. If the host# option is specified, the number (#count) of CPUs is added or removed from the remote host identified by the IP address. For example:

```
calibre_dra -log dra.log -add 123.456.78.910#4
```

```
calibre_dra -log dra.log -remove 123.456.78.910#3
```

The 2020.2 release includes the following DRA enhancements:

- (2020.2) Calibre DRA Direct Connect Mode

  The Calibre DRA Direct Connect mode allows you to start a Calibre job on a low number of remote CPUs, and at any time during the run, connect extra CPUs to the job. The Calibre DRA Direct Connect mode interfaces with common platform environments.

  Contact your Siemens representative for assistance with setting up the launch infrastructure in your job-scheduler environment.

About Calibre DRA

Calibre DRA is supported for Calibre versions as of 2018.4. It enables you to increase and decrease the number of CPU cores assigned to a Calibre job that is already running. The Calibre DRA command line interface also gives you access to job status information such as active and allocated remote resources, current operations, and resource demand.

- Calibre DRA resource changes are performed manually and on a per-job basis.
- Calibre jobs are launched within their existing infrastructure with minimal disruption to the existing job submission and launching processes.
- Calibre DRA requires the same license as Calibre CalCM, and the standard Calibre license consumption formula applies.
- The typical arguments are used for launching a Calibre MTflex job. However, the -turbo_litho argument is not supported for Calibre DRA.

Here is an example invocation of Calibre DRA from a Calibre command line by the -dra switch:

```
$MGC_HOME/bin/calibre -dra -remotefile mtflex.dra -drc -hier -turbo \
 rules.svrf >& calibre.log &
```

Calibre Encryption Updates

This section describes the 2021.2 changes to the Calibre® SVRFencrypt and TVFencrypt tools.

See the *Calibre SVRFencrypt User’s Manual* and the *Calibre TVFencrypt User’s Manual* for details.

Required Updates to Encryption Versions

The 2021.2 encryption updates are not compatible with older versions of Calibre products. Files created with the 2021.2 encryption tools cannot be used with earlier versions of Calibre. Files created with earlier versions of Calibre can still be used with Calibre 2021.2 and later.

Chapter 6
Physical Verification

This chapter describes the 2021.2 changes and enhancements for the Calibre Verification applications.

See “Corrected Defects and Known Issues” on page 12 for instructions on accessing the complete list of corrected defects and known issues for the release.

See “Calibre Utilities” on page 127 for updates to LEF/DEF and OpenAccess database handling.

Calibre Auto-Waivers

This section includes changes and enhancements to Calibre® Auto-Waivers™ for the 2021.2 release.

For additional information, refer to the *Calibre Auto-Waivers User’s and Reference Manual*.

Calibre Auto-Waivers Updates

This section describes updates to Calibre Auto-Waivers in the 2021.2 release.

- Include Cells in the Chip Context
- Support For Wildcards in the Suffix and Prefix Files

Include Cells in the Chip Context

The tool now 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.

This works inversely to the EXCLUDE_CELL statement in that you specify only those cells that you want to include in the run. Apply the new command as follows:

```
INCLUDE_CELL cell_name [cell_name …] [HALO halo_value]
  [BY_LAYER layer_name]
```

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.

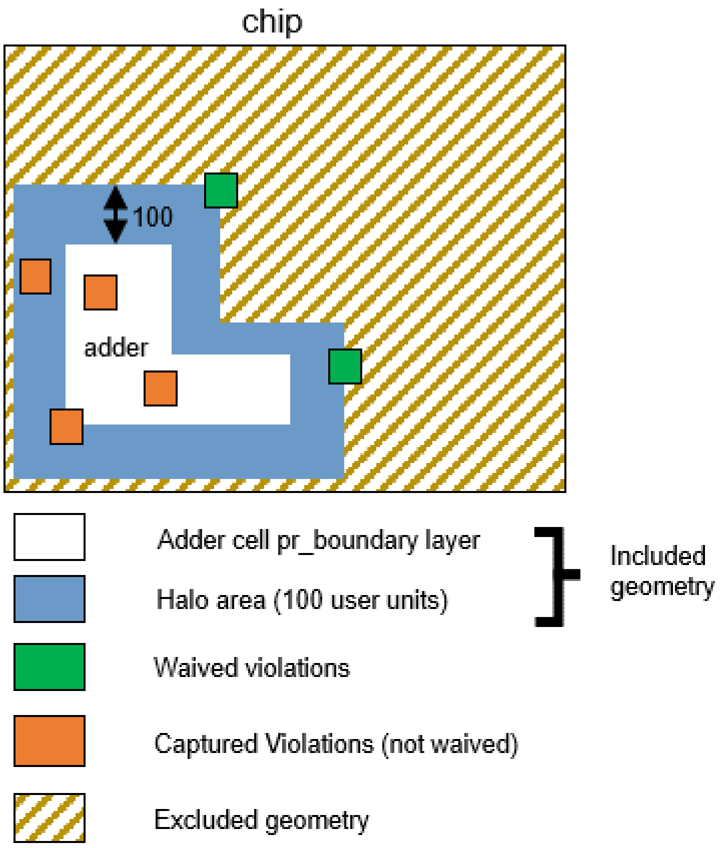
If you specify the HALO argument set, then the geometry that surrounds the cell by a specified positive *halo_value* distance is also included in the run. This enables you to check the interfacing regions around the cell. If you want to manually specify the extent of the cell, use the BY_LAYER argument 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.

The INCLUDE_CELL and EXCLUDE_CELL statements are mutually exclusive and both require a RealTime Digital license.

The following is an example:

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


chip
100
adder
Adder cell pr_boundary layer
Halo area (100 user units)
Included geometry
Waived violations
Captured Violations (not waived)
Excluded geometry

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":
```

Calibre Auto-Waivers Previous Release Highlights

Calibre Auto-Waivers introduced the following functionality in previous releases.

- 2021.1 Updates
- 2020.4 Updates
- 2020.3 Updates
- 2020.2 Updates

2021.1 Updates

- **Enhanced IP Match Waiver Application Behavior —** In previous releases, you could specify to apply waivers based on IP_MATCH checksums using the RUN_IP_MATCH YES waiver setup file statement. This statement considers waivers for application during a verification run based on IP_MATCH text checksums contained in the waiver cell. This is useful for matching waiver cells to design cells based on the cell geometry instead of *only* the cell name. The YES option would also still apply waivers if the following was true:

    - The waiver cells did not contain any IP_MATCH checksums.
    - The waiver cell name matched the cell in which the error result was found.

  As of this release, you can restrict the waiver verification run to only consider waiver cells with IP_MATCH checksum annotations.

  To enable this enhancement, the RUN_IP_MATCH waiver setup file statement supports a new YES_RESTRICTED option. This statement only considers waivers with IP_MATCH checksum annotations during a waiver verification run. Waivers that do not contain IP_MATCH checksums are ignored and not considered for application during a verification run. When a cell is ignored, the tool issues a warning message.

Calibre Interactive contains a new YES (restricted) option in the Verify IP_MATCH checksums dropdown list.

- **Automatically Ignore Waivers for Rules That Contain Preserved Layers —** 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 now automatically removes any applied waivers from the rule checks using the preserved layers that are the result of wildcard matching (for example, * SINGLE 100 100). Waiver criteria that is specified for explicit rules (for example, ruleA SINGLE 100 100) that interact with the preserved layers 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.
  ```

- **Enhanced Behavior When Specifying NONE with Other Primary Criteria —** In previous releases, if you specified the following statement in your waiver criteria file:

  ```
  RULE_1 NONE SINGLE 100 100
  ```

  The NONE keyword that specifies not to apply waivers was ignored and the tool used the SINGLE 100 100 criteria. The correct usage is the following:

  ```
  RULE_1 NONE
  ```

  As of this release, if you specify NONE with other primary criteria, as in the first example, the tool issues a syntax error similar to the following:

  ```
  ERROR: Unexpected token 'SINGLE' found after NONE.
  ```

- **Update to Documentation for Wildcard Matching Behavior —** To clarify the behavior on wildcard matching for secondary criteria, the "Waiver Criteria File Description" topic in the *Calibre Auto-Waivers User's and Reference Manual* has been updated.

2020.4 Updates

- **Control Rule File Precision in drc_waiv_asc2gds —** The drc_waiv_asc2gds utility includes a new -rp option to specify the precision of the rule file used to create results database. You must apply this option if your original rule file precision is different than the precision of the results in the results database or the *.waived* file. 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.

- **New Option to Report Additional Properties for Unwaived Results —** You can now optionally include additional properties in the DRC results database for unwaived results. These properties can help determine why waiver shapes were not applied. The

GENERATE_COMBINED_RESULTS waiver_setup file statement supports a new DEBUG option as follows:

```
GENERATE_COMBINED_RESULTS { NO | YES | DEBUG }
```

Specifying YES in this statement combines the waiver and DRC results, writes additional properties (USED, WAIVED, and others) to results, and combines the unwaived and waived waiver results databases in a *combined_results.rdb* file. Specifying the DEBUG keyword performs the same operation as YES, but the tool also writes the properties to unwaived results in the DRC results database file.

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.

Note

Open the DRC results database file in Calibre RVE to view these properties.

Calibre Interactive has also been enhanced to support the new DEBUG option.

- **Control Default IP_MATCH Setting for Waiver Cells in Calibre Interactive —** Calibre Interactive includes a new global option that enables you to specify whether the IP_MATCH checkbox for waiver cells is enabled by default. The new “Set IP_MATCH for added cells” checkbox can be configured from 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 that appears when you click the **Add** button.

  To provide additional control for GUI flows, you can set this option from a Calibre Interactive runset file by adding the following option:

  ```
  *cmnWaiverSetIPMatchForAllCells: 1
  ```

- **Messaging Enhancements for waiver_util —** In previous release, if waiver_util failed to generate an output file, the tool issued an informational message. As of this release, the tool now issues an error message for missing file output.

  Additionally, waiver_util now issues a warning message if any cell name in the prefix, suffix, or rename file is not found in the input layout. The warning message lists the cell names that do not exist in the layout design.

- **New Error For Non-ASCII Waiver Generation With Combined Results —** If you specify GENERATE_COMBINED_RESULTS YES or DEBUG and include the DRC Results Database statement with a non-ASCII output value (for example, OASIS), the tool now issues an error message and stops the run. You can only generate combined results with ASCII results databases.

2020.3 Updates

- **New OVERSIZE Option for Extent Waivers —** The WAIVE_EXTENT statement now includes an OVERSIZE option. This enables you to size the waiver area beyond the original bounding box of the cell's extents. This operation works similarly to the UNDERSIZE keyword. Apply the OVERSIZE keyword as follows:

```
WAIVE_EXTENT cell_name [cell_name …]  [FULL]
   [{UNDERSIZE | OVERSIZE} distance]
```

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 distance) and the extent size decreased by the halo value.

- **Output Combined DRC and Waived Results to a Single Results Database —** You can now optionally specify to output both the DRC results and the waivers (both used and unused) to the same Calibre results database. The following new waiver setup file statement controls this feature:

  ```
  GENERATE_COMBINED_RESULTS { NO | YES }
  ```

  The default value is NO, meaning that the results are not combined into one results database.

  The Calibre Interactive tool also includes a new **Generate combined results** dropdown list on the **Inputs** > **Waivers** > **Waiver RDB Files** pane. This controls whether the results are combined into one database and is set to NO by default.

- **Support for IP_MATCH Waivers, Waiver Comments, and Waiver Hierarchy in drc_waiv_asc2gds —** The drc_waiv_asc2gds utility now supports the following new features:

  - IP_MATCH waivers for *.waived* input files.
  - The ADD_WAIVER_HIERARCHY YES statement for *.waived* files.
  - Comments in generated waivers that include custom text loaded from a file, the current date, and the username. Specify comments from a file using the new -cm option as follows:

    ```
    drc_waiv_asc2gds -i input_results_database \
       -o output_database \
       [ -cm comment_file ]
    ```

2020.2 Updates

- **Support For Waivers in Cloned Cells —** Calibre Auto-Waivers now supports the generation of waiver shapes for results in cloned cells. Cloned cells can be generated by statements in your rule file, such as Layout Clone Rotated Placements. In previous releases, waivers could not be generated inside cloned cells.

As of this release, you can optionally generate waivers in cloned cells by exporting waivers from Calibre RVE or with the drc_waiv_asc2gds utility. This capability is not enabled by default.

The cloned cell waiver enhancement 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.

The “Generate waivers in cloned cells” checkbox is new for the Calibre RVE Export Waived Results window. When enabled, this option enables waiver generation for cloned cells.

- **Change to Waiver Checksums —** The method used to calculate checksums for cells has been changed for a very limited number of corner cases in order to provide compatibility with newer versions of VCO. If waiver cells generated by older versions of Calibre Auto-Waivers are not applied correctly, you must update the cell checksums or regenerate the waiver cells using the latest version.

Calibre DESIGNrev

This section includes changes and enhancements to Calibre® DESIGNrev™ for the 2021.2 release:

For more information, refer to the *Calibre DESIGNrev Layout Viewer User's Manual* and the *Calibre DESIGNrev Reference Manual*.

Calibre DESIGNrev Updates

The following updates were made to Calibre DESIGNrev for the 2021.2 release.

- Enhanced Support for Opening Compressed Files
- New Ability to Hide Cells in the GUI
- Miscellaneous Enhancements for Layout Filemerge

Enhanced Support for Opening Compressed Files

In previous releases, Calibre DESIGNrev would generate an error when opening a compressed file that did not contain a .gz extension, or opening an uncompressed file that contained the .gz extension. With the 2021.2 release, when opening a file with the incorrect extension using **File** > **Open Layout Files**, a command line invocation, or a script using layout create, Calibre DESIGNrev now generates a warning that the extension is incorrect and proceeds to open the file. The original filename is preserved when using **File** > **Save**. (ER VIEWER-15392)

New Ability to Hide Cells in the GUI

The Calibre DESIGNrev GUI now includes the ability to hide cells in the Cells Browser. To hide a cell, you can right-click on a cell name in the Cells Browser and choose **Hide Cell**, or you can choose **View** > **Show/Hide Cells** to display the Show/Hide Cells in Palette dialog box and select the cells you want to hide. Upon hiding a cell, the Cells Browser updates to display a hidden icon indicating there are hidden cells.

Refer to "Cells Browser" in the *Calibre DESIGNrev Layout Viewer User's Manual* for more information. (ER VIEWER-16728)

Miscellaneous Enhancements for Layout Filemerge

Outputting Compressed (.gz) Files

The layout filemerge command now supports the ability to output a gzipped file when using the -oasisout or -gdsout argument and specifying the .gz extension. (ER VIEWER-16898)

Writing OASIS Files in Non-Strictmode Format

When strictmode is disabled for layout filemerge (-strictmode 0), Calibre DESIGNrev now writes CELLNAME records at the beginning of the output OASIS file. The CELLNAME records are still written at the end of the layout when strictmode is enabled. (ER VIEWER-16923)

Calibre DESIGNrev Previous Release Highlights

The following updates were made in previous releases.

2021.1 Release Highlights

Change to OASIS File Output

When saving an OASIS file using **File** > **Save** or **File** > **Save As**, Calibre DESIGNrev now saves OASIS files as follows:

- The output is saved with CBLOCKs even when the input does not contain CBLOCKs. (ER VIEWER-15119)
- The output is saved in strict mode format, regardless of the strict mode input format. (ER VIEWER-16676)

To save an OASIS file without CBLOCKs or in non-strict mode format, you must use **Export** > **Layout**.

Change to Ruler Measurements

You can now measure off-grid results, such as simulation output, using rulers. To measure off-grid results, set the ruler units to “um” or “nm” and set the precision to %.4f or higher. When exporting a ruler, the ruler file format exports floating point values instead of integers into an XML file. When reading a 2021.1 XML ruler file with older versions of Calibre DESIGNrev, the floating point values are truncated to integers. To enable integer-based rulers, you must set the ruler units to “dbu”.

Refer to “Rulers Palette” and “Preferences Dialog Box - Rulers Tab” in the *Calibre DESIGNrev Layout Viewer User’s Manual*. (ER VIEWER-16661)

Layout Filemerge Enhancements

The layout filemerge batch command is updated to include the following new enhancements:

- Ability to specify the GDS format as the output file using the -gdsout argument.
- Ability to ignore geometries (-ignoreGeometries) or references (-ignoreReferences) when using the -smartdiff argument.
- Ability to specify the information returned (cells, layers, undefined cells) from the layout filemerge operation using the -peek argument.

The new syntax is:

```
layout filemerge -in {...}
   {{-out … | {-gdsout output_file}}
   [-smartdiff {... | [-ignoregeometries] | [-ignorereferences]}]
   [-peek {[-cells] [-layers] [-undefcells]}]
```

Refer to the layout filemerge command in the *Calibre DESIGNrev Reference Manual* for more information. (ERs VIEWER-16762, VIEWER-16807, and VIEWER-16808)

Change to Layout Create -ignoreInsts Behavior

When using the layout create (GDS or OASIS file) command with the -ignoreInsts argument, Calibre DESIGNrev now ignores any child cells of instances (ER VIEWER-16777).

Support for Character Shapes

When creating character shapes in Calibre DESIGNrev using **Object** > **Character Shapes**, you can now specify parentheses in addition to alphanumeric characters. Refer to "Creating Character Shapes" in the *Calibre DESIGNrev Layout Viewer User's Manual*. (ER VIEWER-16786)

Changes to Console (Terminal) Window Behavior

The Calibre DESIGNrev console (terminal) window now supports the following command history and auto-completion capabilities: (ER VIEWER-16731)

- Press the up and down arrow keys to display the command history list. Use "!" (exclamation point) followed by the command number in the history list to re-execute a specific command.
- Press the left and right arrow keys to move the cursor position on the command line.
- Press the tab key to auto-complete file and directory names. A backslash (/) is automatically appended to auto-completed directory names.

Layer Filters Enhancements

The Layer Filters now support the sorting of layers. To sort in ascending or descending order, right click on the **Filters** tab to display a popup menu with options to sort in increasing or decreasing order.You can also click on a filter name and then drag and drop it in the desired location

The "All Layers" filter always displays at the top of the filter list regardless of the selected sorting state. The **Filters** tab displays the order in which the filters are sorted (increasing or decreasing). When the layer filters is sorted, the ordering of the list is maintained and updated to include or remove any filters. When the layer filters is unsorted, the filter list is preserved when adding a new filter, which is placed at the bottom of the filter list. To retain filters, you must save layer filters to a layer properties file by selecting **Layer** > **Save Layer Properties**. (ER VIEWER-16702)

When using the calibredrv -dl or -l arguments to load a layer properties file, Calibre DESIGNrev now loads the layer filters in the same order as defined in the layer properties file. (ER VIEWER 16763)

Support for Merging Shapes

Calibre DESIGNrev now supports the merging of selected paths and polygons when the selected shapes are on the same layer and in the same cell. To merge polygons or paths, select the shapes, right click to display the popup menu and select **Merge Shapes** or select the **Edit** > **Merge Shapes** menu. (ER VIEWER-16809)

Calibre Interactive

This section includes information on the changes and enhancements to Calibre® Interactive™ for the 2021.2 release.

For additional information, refer to the *Calibre Interactive User's Manual*.

Note

Calibre Interactive and Calibre RVE are now documented in separate manuals. The smaller manuals make it easier to find information.

Calibre View generation is covered in the *Calibre Interactive User's Manual*.

Calibre Interactive General Updates

The following general update was made to Calibre Interactive in the 2021.2 release.

New Batch Command Switch for Converting Classic Runsets

When running the Calibre command line in the new Calibre Interactive GUI, you can now convert classic runsets using the switch "-output_runset" with a runset name argument. You can also use this switch to combine multiple runsets; if multiple are specified, they are combined into the new runset, with the last runset taking precedence.

File Viewing Enhancements Added

In the new Calibre Interactive GUI, you can now visualize files in the Include Rules Files and Waiver Criteria Files sections of the **Options** and **Waivers** pages, respectively.

Added Distributed MTflex Support

In the new Calibre Interactive GUI, you can now run Calibre using Distributed (MTflex) when running on Other Cluster, which was previously called Cluster Command. Specify this option in the **Run Control** page.

Calibre Interactive nmDRC Updates

The following updates were made to Calibre Interactive nmDRC in the 2021.2 release.

New Options for Hierarchical Databases

There is a new option to control the behavior of Distributed (MTflex) runs by specifying when to connect to remote hosts in HDB (Hierarchical Database) construction. In the **Run Control** page, you can specify to connect at later stages of HDB construction, which uses the "-hdbflex" command line switch, or to connect in parallel with HDB construction, which uses the "-hdbflex_acquire" command line switch.

In the new Calibre Interactive GUI, you can also specify to save a new RHDB (Reusable Hierarchical Database) file or restore an RHDB file with a new option in the **Outputs** page. An RHDB is useful in some flows to save time by avoiding the overhead of repeatedly constructing a hierarchical database.

New createReport.sh Script Enhancement

In the new Calibre Interactive GUI, if you enable "View report" when creating the report, you can now include an optional switch "-open" when running the *createReport.sh* script in order to open the report after it has been created.

Calibre Interactive nmLVS Updates

The following updates were made to Calibre Interactive nmLVS in the 2021.2 release.

Additional Short Isolation Statement Support

In the new Calibre Interactive GUI, you can now specify the LVS SI Select Connects statement in the **Options** page when performing a Calibre nmLVS Reconnaissance Short Isolation run. This statement selects Connect and Sconnect operations to participate in stand-alone short isolation.

Enhanced Layer Sorting

In the **Options** page of the new Calibre Interactive GUI, the LVS SI Select Connects option's Layers dropdown menu now displays the layers in stack order.

Calibre Interactive xACT Updates

The following updates to Calibre Interactive xACT were made in the 2021.2 release.

New Probes Page Added

A **Probes** page has been added to Calibre Interactive xACT for specifying probe points, which are used to verify timing from specific points on each net. On this page, you can add and load probe points, set their locations, and configure other options.

New PEX Netlist Keywords Added

The CALIBRATED and RES_EMIR keywords are now supported in Calibre Interactive xACT and can be enabled in the **Options** page when working with DSPF netlists.

The CALIBRATED keyword is only available if either LAYERMAP is selected or the netlist compatibility mode is set to TOTEM. When used with LAYERMAP, the CALIBRATED keyword specifies to only list calibrated layers in the LAYER_MAP section of the extracted netlist. When used with TOTEM, it specifies to only netlist calibrated layers in the LAYER_MAP section of the netlist.

The RES_EMIR keyword specifies to include the PCDEF and PEX ignore internal resistance dimensions in the extracted DSPF netlist for the EM/IR analysis flow.

Calibre Interactive Previous Release Highlights

Important changes in Calibre Interactive from previous releases are summarized in the following sections.

Calibre Interactive General Previous Updates

The following general updates were made to Calibre Interactive in previous releases.

AOK Support Added

(2021.1) Support for remote hosts running with AOK has been added. You can now specify AOK entries in the Remote MGC_HOME and Remote MGC_LIBPATH tables, and remote scripts now have entries for AOK.

New Version of Calibre Interactive Now Available

(2020.4) This new version brings features such as a search option to easily located controls, more flexible GUI customization, and simplified views for DRC and LVS applications. Many features from classic Calibre Interactive are still available, including integration with all design tools supported by classic Calibre Interactive. For more information, refer to the *Calibre Interactive (New GUI) User's Manual*.

Updated SSH Verification

(2020.3) Calibre Interactive now uses SSH_ASKPASS for ssh authentication.When using ssh, ensure that you have the openssh-askpass package installed; the package comes installed with ssh.

New OpenAccess Version Specification Hierarchy

(2020.3) If you set the OpenAccess version in the rule file with the Layout System statement, Calibre Interactive now includes the OA version setting in the Layout System statement written to the control file. Previously it was necessary to use the environment variable MGC_FDI_OA_VERSION in order to specify the OA version used for a Calibre Interactive run.

Previous Calibre Interactive nmDRC and DFM Updates

The following updates to Calibre Interactive nmDRC, FastXOR, and DFM were made in previous releases.

- New IP_MATCH Option in Waiver Cell Creation
- New Waiver Creation Option
- Support for Calibre Reconnaissance Mode Recipe Editing
- Calibre Incremental DRC is Obsolete

New IP_MATCH Option in Waiver Cell Creation

(2020.4) When working under the Waiver Cells File dialog box, you can now specify the "Set IP_MATCH for added cell" to automatically enable IP_MATCH when adding new cells. For more information, see "Calibre Auto-Waivers Updates" on page 40.

New Waiver Creation Option

(2020.3) Enable "Generate combined results" (GENERATE_COMBINED_RESULTS YES) to generate a combined waivers results database with both the used and unused waiver shapes. The entries in the combined database include additional properties to indicate whether the waiver was applied or unused.

Support for Calibre Reconnaissance Mode Recipe Editing

(2020.2) The Check Selection Recipe Editor now supports using checks selected in Calibre Reconnaissance mode and its inverse mode. Using these expression categories requires a license for Calibre RealTime Digital.

Calibre Incremental DRC is Obsolete

Calibre Incremental DRC is obsolete starting with the 2018.1 release and the functionality has been removed from the Calibre software. Calibre Incremental DRC was deprecated in the 2017.2 release. The functionality has been superseded by Calibre RealTime Custom and Calibre RealTime Digital.

Contact your Siemens representative or customer support for help in migrating to Calibre RealTime. Documentation regarding Incremental DRC has been removed from the manuals.

Previous Calibre Interactive nmLVS Updates

The following updates to Calibre Interactive nmLVS were made in previous releases.

New Automatic Hcell Generation Option

(2021.1) When automatically generating an hcell list, you can now specify the “Use default thresholds as hcell selection criteria” option.

Remote Comparison Option Added

(2021.1) In the new Calibre Interactive GUI for LVS, you can now specify remote LVS comparison options. Configure these options in the Run Calibre LVS Compare on Remote Host section on the **Run Control** page.

Support for Recon SI Runs Added

(2021.1) Recon short isolation runs are now supported. You can specify to run power ground short isolation, IO short isolation, or both. A simplified view for Recon SI is also available.

New Hcell Option

(2020.4) Under the H-Cells tab on the Inputs page, you can now select the “Expand H-Cells to resolve false discrepancies” option to specify to examine LVS discrepancies and, in certain cases, to automatically perform additional comparison runs with fewer hcells. This option uses the LVS EXPAND ON ERROR statement.

Previous Calibre Interactive xACT Updates

Calibre Interactive xACT had the following updates in previous releases:

- (2020.4) Configuration Editor Now Available
- (2020.4) Platform LSF and Distributed (MTflex) Options Now Available
- (2020.4) Point To Point Option Added
- (2020.4) NOINSTANCEX Support Increased
- (2020.3) New ERC Options for LVS Runs
- (2020.3) Environment Page Updates
- (2020.2) Formatter Options Added
- (2020.2) Custom Tcl Files Supported
- (2020.2) Transcript Scroll Lock Button Added
- (2020.2) Runset Loading Default Improved

(2020.4) Configuration Editor Now Available

You can use the configuration editor to easily customize your Calibre Interactive GUI by adding and hiding pages, moving options to different pages, hiding options, and performing other changes.

(2020.4) Platform LSF and Distributed (MTflex) Options Now Available

You can now specify to run Calibre xACT on Platform LSF, and to run using Distributed (MTflex) by using your existing computer network as a distributed computing platform. Configure these options in the Run Control page.

(2020.4) Point To Point Option Added

You can now select the "CALIBREVIEW" option under the PEX Report Point2Point section in the Outputs page to include the CALIBREVIEW keyword as part of the PEX REPORT POINT2POINT statement.

(2020.4) NOINSTANCEX Support Increased

The NOINSTANCEX keyword is now supported when using a netlist of type CALIBREVIEW.

(2020.3) New ERC Options for LVS Runs

New options related to ERC statements, such as LVS Execute ERC and ERC Cell Name, are now configurable in the LVS options page.

(2020.3) Environment Page Updates

The rule file variables displayed in the Environment page now automatically update when the page is loaded. Additionally, the order and sizes of the columns has been adjusted to ease readability.

(2020.2) Formatter Options Added

You can now configure the xACT formatter in the **Outputs** pane:

- Ground all coupling capacitors — Adds the -g option.
- Display formatter warnings — Adds the -fmt_warnings option.
- Display formatter information — Adds the -fmt_info option.

(2020.2) Custom Tcl Files Supported

You can now load Tcl customization files when launching Calibre Interactive xACT through the command line. Use the -custom *custom_file* argument, or the -customgui *custom_file* argument when in batch mode. See "Calibre Interactive xACT Command Line" in the *Calibre Interactive User's Manual* for more information.

(2020.2) Transcript Scroll Lock Button Added

You can now lock the transcript view to prevent the transcript from snapping to the end whenever new text is added. This button is located at the bottom of the scroll bar.

(2020.2) Runset Loading Default Improved

Upon invoking the Calibre Interactive xACT GUI, if the environment variable MGC_CALIBRE_XACT_RUNSET_LIST has been used to pre-populate the list of runsets, the runset defined by MGC_CALIBRE_XACT_RUNSET_FILE is now selected by default.

Previous Calibre Interactive PEX Updates

Calibre Interactive for PEX had the following updates in previous releases:

New Extraction Options

(2021.1) You can now specify to perform in-context extraction, extracting a cell's parasitic information with reference to structures outside the cell boundary. Select In-Context when specifying the Extraction Type on the **Outputs** page.

You can also specify an extraction accuracy mode of MEMS, or to run at the highest accuracy needed for MEMS designs. Select this option when specifying the Extraction Mode on the **Outputs** page.

Enhanced MinCap Reduction Option

(2021.1) When performing MinCap reduction, you can now specify “No Remove” (PEX REDUCE MINCAP REMOVE NO) to disable the default reduction of small capacitance between pairs of nets.

Calibre View Pin Delimiter Now Configurable

(2021.1) When configuring Calibre View options, you can now specify the PINDELIM option to specify the pin delimiter from a colon (“:”) to other characters (“_” or “.”). The pin delimiter character separates device names and pin names in the Calibre View netlist.

Point To Point Option Added

(2020.4) You can now select the “CalibreView” option under Point To Point in the Outputs page to include the CALIBREVIEW keyword as part of the PEX REPORT POINT2POINT statement.

NOINSTANCEX Support Increased

(2020.4) The NOINSTANCEX keyword is now supported when using a netlist of type CALIBREVIEW.

Calibre RealTime Custom and Calibre RealTime Digital

This section includes information on changes and enhancements to Calibre® RealTime Custom and Calibre® RealTime Digital for the 2021.2 release.

Documentation for Calibre RealTime Custom is found in the following locations:

Calibre RealTime Custom User's Manual

Calibre RealTime Integration to Synopsys Laker quick reference card

Calibre RealTime Integration to Cadence Virtuoso quick reference card

Pyxis Layout User's Manual

Documentation for Calibre RealTime Digital is found in the following location:

Calibre RealTime Digital User's Manual

Calibre RealTime Digital in Cadence Innovus quick reference card

See the following topics for information on changes and enhancements:

Calibre RealTime Custom Updates

There were no major updates made to Calibre® RealTime Custom in the 2021.2 release.

Calibre RealTime Digital Updates

There were no major updates made to Calibre® RealTime Digital in the 2021.2 release.

See the *Calibre RealTime Digital User's Manual* for more information on the features available in Calibre RealTime.

Calibre RealTime Custom Previous Release Highlights

The following updates were made to Calibre RealTime Custom in previous releases.

The following updates were made in previous releases.

- New Summary File Enhancements
- New Environment Variable for Rule File Loading
- New Highlight All Results Option
- New Tanner L-Edit Integration
- Support for Disabling Environment Variable Files
- Run Status Updates Now Supported
- Additional Layer Map Support
- Reinstall Process Improved

New Summary File Enhancements

(2021.1) The summary file written during a RealTime Digital run now includes the total number of executed checks and empty checks. These values can be found on the Selected Checks and Empty Checks lines.

New Environment Variable for Rule File Loading

(2021.1) You can now set the environment variable MGC_REALTIME_AUTO_LOAD_RULE_FILES to “none” to disable automatic loading of rule files in all configurations.

New Highlight All Results Option

(2020.4) In the Options dialog box, you can now choose to automatically highlight DRC results from all configurations after a DRC run. When this option is enabled, highlights are preserved when switching configurations in both single and serial runs, and the options “Clear existing highlights” and “Highlight after DRC” are ignored. Additionally, you can only cross-highlight results from the layout editor to the Results window if they are from the current configuration.

This feature is not supported in the Calibre DESIGNrev or Pyxis™ Layout integrations.

New Tanner L-Edit Integration

(2020.3) Calibre RealTime Custom is now integrated with the Tanner™ L-Edit™ layout design tool. The Calibre RealTime toolbar and options are automatically available when you install the L-Edit tool, and it uses the same setup and communication as other integrations.

For more information on this integration, see the Calibre RealTime chapter in the *Tanner L-Edit User's Manual*.

Support for Disabling Environment Variable Files

(2020.2) The new environment variable MGC_REALTIME_DISABLE_FILE_SETTINGS_ENVS prevents files defined by other environment variables, such as the rule, runset, and CTO files, from being applied upon startup.

Run Status Updates Now Supported

(2020.2) In the Cadence Virtuoso integration, detailed processing status updates can now be displayed in the shell during a Calibre RealTime run. This feature is disabled by default and can be configured in the Options dialog box.

Additional Layer Map Support

(2020.2) The Cadence Virtuoso and Synopsys Custom Compiler integrations now support Synopsys Custom Compiler layer map files.

Reinstall Process Improved

(2020.2) In the Cadence Virtuoso integration, reinstalling the RealTime Custom client and server now resets all processes without having to open or close Virtuoso.

Calibre RealTime Digital Previous Release Highlights

The following updates were made to Calibre RealTime Digital in previous releases.

The following updates were made in previous releases.

- New Summary File Enhancements
- New Environment Variable for Rule File Loading
- New Highlight All Results Option
- New API Commands for Error Descriptions
- Run Status Updates Now Supported
- Support for Disabling Environment Variable Files
- Ignore Undefined Cells

New Summary File Enhancements

(2021.1) The summary file written during a RealTime Digital run now includes the total number of executed checks and empty checks. These values can be found on the Selected Checks and Empty Checks lines.

New Environment Variable for Rule File Loading

(2021.1) You can now set the environment variable MGC_REALTIME_AUTO_LOAD_RULE_FILES to "none" to disable automatic loading of rule files in all configurations.

New Highlight All Results Option

(2020.4) In the Options dialog box, you can now choose to automatically highlight DRC results from all configurations after a DRC run. When this option is enabled, highlights are preserved when switching configurations in both single and serial runs, and the options "Clear existing highlights" and "Highlight after DRC" are ignored. Additionally, you can only cross-highlight results from the layout editor to the Results window if they are from the current configuration.

New API Commands for Error Descriptions

(2020.3) You can use the new API commands GetResultInfo and GetCheckInfo to retrieve check information based on a given result or check name. Additionally, the GetResultCoordinates command now has the argument -output_layer to specify whether to also return the mapped output layout number and data type associated with the check.

Run Status Updates Now Supported

(2020.2) Detailed processing status updates can now be displayed in the shell during a Calibre RealTime run. This feature is disabled by default and can be configured in the Options dialog box.

Support for Disabling Environment Variable Files

(2020.2) The new environment variable MGC_REALTIME_DISABLE_FILE_SETTINGS_ENVS prevents files defined by other environment variables, such as the rule, runset, and CTO files, from being applied upon startup.

Ignore Undefined Cells

(2020.2) In the Synopsys IC Compiler II integration of Calibre RealTime Digital, you can now choose to ignore undefined cells when generating DRC results. This option is disabled by default and can be configured in the Options dialog box.

Calibre RVE

This section includes information on the changes and enhancements to Calibre® RVE™ for the 2021.2 release.

For additional information, refer to the *Calibre RVE User's Manual*.

Note

Calibre Interactive and Calibre RVE are now documented in separate manuals. The smaller manuals make it easier to find information.

Calibre RVE for DRC Updates

The following updates were made to Calibre RVE for DRC in the 2021.2 release.

Waiver Report Generation Batch Command Option Added

You can now generate a waiver report using the Calibre RVE command line with the -waiver_report switch. This option creates a text report of the results that are waived using Calibre RVE. See the *Calibre RVE User's Manual* for more information on this switch and additional optional arguments.

Support for Multiple Initial Waiver Comments Added

You can now specify multiple initial waiver comments when waiving results. After multiple comments are defined in the **Options** tab, you can select one of them upon waiving a result.

Ability to Disable Unwaiving Results with Keyboard Shortcut Added

Typically, you can unwaive results using the W keyboard key, which is also used to waive results. You can now disable this unwaiving ability in the **Options** tab; this feature prevents accidental unwaiving of results, which removes waiver comments. When this feature is enabled, the W key performs no operation if any selected results are waived.

DRC HTML Reporting Updates

The following updates were made to DRC HTML Reporting in the 2021.2 release.

Property Filter Options Added

A new key value AndPropertyFilter has been added, which includes results that meet the specified property expression and are combined using AND logic when multiple property expressions are used. In addition, in the AndPropertyFilter and OrPropertyFilter key values, you can specify to include results for which a property does not exist using the NotExists value.

New createReport.sh Script Enhancement

If you enable "Open report after it is created" when creating a Calibre DRC HTML report, you can now include an optional switch "-open" when running the *createReport.sh* script in order to open the report after it has been created.

New Histogram Export Options

When exporting histograms, you can now choose to save them as comma-separated value (CSV) files as well.

Calibre RVE for DFM and DFM HTML Reporting Updates

The following updates were made to Calibre RVE for DFM and DFM HTML Reporting in the 2021.2 release.

New Histogram Export Options

Histograms can now be saved as a chart, a comma-separated value (CSV) file, and as a txt file. In the configuration file, when working with the ByErrorHistogram, ByWindowHistogram, and ByCellHistogram report types, you now specify the new HistogramOutputFormat key value to specify either a chart, csv, or txt output. The default behavior is to export as a chart.

When exporting histograms in the GUI, you can also save them as these three output formats as well.

Calibre RVE for LVS Updates

There following updates were made to Calibre RVE for LVS in the 2021.2 release.

Rotation and Reflection Extended to Shorts

When rotating and reflecting highlighted shapes in Calibre RVE for LVS, shorts are now affected as well.

Calibre RVE for PERC Updates

The following updates were made to Calibre RVE for PERC in the 2021.2 release.

Support for Multiple Initial Waiver Comments Added

You can now specify multiple initial waiver comments when waiving results. After multiple comments are defined in the **Options** tab, you can select one of them upon waiving a result.

Waiver Description File Name Replaceable Support Added

When creating a waiver description file, you can now specify environment variables, which are replaced with their values, or %T, which is replaced with the topcell name.

Calibre RVE for PEX Updates

The following updates were made to Calibre RVE for PEX in the 2021.2 release.

CSV Export Option Added

You can now export parasitic information as a CSV (comma-separated value) file. Select **File** > **Export Parasitics** > **Export to CSV** to export an open net parasitics table or detailed parasitics view.

Calibre RVE Previous Release Highlights

Important changes to Calibre RVE from previous releases are summarized here.

Calibre RVE Previous General Updates

The following general updates were made to Calibre RVE in previous releases.

Enhanced Filter Loading with Multi-Level Configuration Management

(2021.1) When loading multiple configuration files with the MGC_RVEDB_DIR environment variable, filters found in all directories are now applied, with the filters in the leftmost directories appended to the ones in the rightmost directories. If there is a naming conflict, the filter in the leftmost directory takes precedence.

New Environment Variable for Terminal Control

(2020.3) Set the environment variable MGC_CALIBRE_DISABLE_GUI_TERMINAL to disable the terminal pane. When set, the controls **View** > **Terminal** and **Setup** > **Options** > **Session** > **Show Terminal Pane** are disabled, and the cmnShowCommandPrompt runset option is not honored.

Previous Updates to Calibre RVE for DFM and DFM HTML Reporting

The following updates were made to Calibre RVE for DFM and DFM HTML Reporting in previous releases.

New Options for Size Control of Colormaps and Histograms

(2020.3) You can use new key values to set the minimum or maximum pixel dimensions of colormaps and histograms in DFM HTML reports. For histograms, use SetHistogramMinSize and SetHistogramMaxSize, and for colormaps, use SetColormapMinSize and SetColormapMaxSize. You cannot specify both for colormaps.

Calibre RVE for DRC Previous Updates

Several updates were made to Calibre RVE for DRC in previous releases.

- Updated Editing Option for Histograms
- New Command Line Option for Saving CSV Results
- Batch Filtering Available
- Support for Waivers in Clone Cells

Updated Editing Option for Histograms

(2021.1) After opening a histogram, you can now edit the Divisions field directly in addition to using the spin control to increment the value.

New Command Line Option for Saving CSV Results

(2020.4) When using the command line to save DRC results in CSV format, include the new option -write_top_coords to add both cell transformation details and the transformed results expressed in top level coordinates to the output.

Batch Filtering Available

(2020.2) The command line now supports filtering results in batch mode. You can filter results based on their properties, waived states, locations in their cells, and other criteria, then save them to new results databases.

Full usage is given by calibre -rve -filter -help. See "calibre -rve Command Line for Calibre RVE Batch Filtering" in the *Calibre RVE User's Manual* for more information.

Support for Waivers in Clone Cells

(2020.2) The Calibre RVE Export Waived Results dialog box includes a new option to support waivers in cloned cells. See "Calibre Auto-Waivers Updates" on page 40 for more information.

Previous Updates to DRC HTML Reporting

Several updates were made to DRC HTML Reporting in previous releases.

- New Default Sorting for Color Maps
- Bare Report Images Added

New Default Sorting for Color Maps

(2020.2) Color maps in HTML reports are now sorted alphanumerically by their property names.

Bare Report Images Added

(2020.2) Reports with layout snapshots, such as color maps, can now be rendered without any additional graphics, such as grids or frames. Set the key value ShowLayoutScale = 0 to disable these graphics.

Previous Updates to Common Tools in Calibre RVE for LVS, PERC, and PEX

Several updates were made to the common tools in Calibre RVE for LVS, PERC, and PEX in previous releases. Previous updates to the Calibre RVE SPICE File Viewer are included here.

New Highlighting Option

(2020.3) You can now highlight connectivity objects by their corresponding layout names. When enabled, these objects are highlighted in the connected schematic viewer using their corresponding layout names. This option is only available when the option to highlight objects in the schematic design tool is enabled.

Access this option with **Setup** > **Options**, then choose the **Highlighting** category and expand the "LVS/PERC/PEX Highlighting" section. This option is enabled automatically by Calibre Interactive-nmLVS in Netlist vs Netlist mode.

Calibre RVE for LVS Previous Updates

Several updates were made to Calibre RVE for LVS in previous releases.

New Hotkeys for Highlighting Short Polygons

(2021.1) You can now use various keyboard shortcuts to traverse through and highlight polygons in Calibre RVE for LVS.

Keyboard key	Action
h, H	Highlight current polygon
Left arrow, p, P	Highlight previous polygon
Right arrow, n, N	Highlight next polygon
Up arrow	Move to previous selection
Down arrow	Move to next selection

Reload Feature Enhanced

(2021.1) When **File** > **Reload Database** is selected while an RDB is open, such as one with ERC results, the entire LVS database is now reloaded, not just the currently selected RDB.

Errors and Warnings Display Order Updated

(2020.3) In the LVS Runtime Errors and Warnings dialog box, errors now appear before warnings.

Option to Open ERC Databases Added

(2020.2) You can now choose to automatically open ERC databases upon starting up Calibre RVE. This option is available in the Setup Database Filters and Triggers Options pane and is disabled by default.

Calibre RVE for PERC Previous Updates

Several updates were made to Calibre RVE for PERC in previous releases.

- Reload Feature Enhanced
- Filtering by Categories Added
- Visibility Enhancements
- Updated Names for Sources and Sinks
- Transformed Schematics Now Displayed

Reload Feature Enhanced

(2021.1) When selecting **File** > **Reload Database** while an RDB is open, such as one with LDL results, the entire PERC database is now reloaded, not just the currently selected RDB.

Filtering by Categories Added

(2020.3) You can now filter results in Calibre RVE for PERC with a CTO file. After importing a CTO file, click the Filter button and select a category.

Visibility Enhancements

(2020.3) Excessively long lines in the RVE window are now automatically truncated. Additionally, menus now show a maximum of 50 design objects.

Updated Names for Sources and Sinks

(2020.2) When viewing Calibre PERC P2P results in Calibre RVE, the column headers for sources and sinks now have more detailed names that include information such as device type and port nets.

Transformed Schematics Now Displayed

(2020.2) After using PERC LOAD XFORM to generate transformed schematic data, both the original and the transformed representation of the schematic are viewable in Calibre RVE.

Calibre RVE for PEX Previous Updates

Several updates were made to Calibre RVE for PEX in previous releases.

Find Nets and Coupling Capacitance Control Bar Enhancements

(2020.3) Under the Parasitics tab, the Find Nets and Coupling to: Specified Nets search fields now have a dropdown menu populated with all of the layout nets. You can select a net or enter a space-separate list of nets as before.

BSPEF API Support Added

(2020.2) Calibre RVE for PEX can now read BSPEF netlist files produced by Calibre xACT runs. Point-to-point inductance calculation is not supported.

Calibre RVE Design Tool Integrations Previous Updates

Several updates were made to Calibre RVE design tool integrations in previous releases

Line Width Specification in Calibre DESIGNrev

(2020.2) You can now configure highlight line width in the Calibre DESIGNrev integration for Calibre RVE. Adjust this setting under **Verification** > **RVE/CI Setup**.

Calibre nmDRC and Calibre nmDRC-H

This section includes information on changes and enhancements to Calibre® nmDRC™ and Calibre® nmDRC-H™ for the 2021.2 release.

For additional information, refer to the *Standard Verification Rule Format (SVRF) Manual* and the *Calibre Verification User's Manual*.

See "Calibre Utilities" on page 127 for updates to LEF/DEF and OpenAccess database handling.

Try It!	**Calibre Tutorials and Example Kits**
	Tutorials and Example Kits (eKits) contain design data, rule files, and instructions for running and learning more about the tool. Go to this page on Support Center to see a list of physical verification eKits (Documentation tab, Document Types=Getting Started Guide). The link goes to the latest release.

Calibre nmDRC and nmDRC-H Updates

The following updates were made to Calibre nmDRC and Calibre nmDRC-H for the 2021.2 release.

- Enhanced Check Selection in Calibre nmDRC Reconnaissance
- Support for Calibre RVE Priority in DRC Analyze
- DFM DV: Limit Number of Results
- New Keyword for [Not] Inside Cell Statement
- Miscellaneous Updates
- Calibre Advanced DRC (eqDRC) Tutorial and Example Kit

Enhanced Check Selection in Calibre nmDRC Reconnaissance

Calibre nmDRC Recon has enhanced methods for selecting the rule checks that are executed in the run.

- **New statements to select checks by layer**

  The new statements DRC Recon Add Check By Layer and DRC Recon Remove Check By Layer enable and disable checks according to a specified original layer, similar to the DRC Select Check By Layer and DRC Unselect Check By Layer statements. A check is added or removed from the run if the specified layer names are required anywhere in the rule checks' layer derivation trees, including connectivity derivations.

- **Predefined rule check groups**

  You can now enable and disable groups of rule checks using new keywords for the existing statements DRC Recon Add Check and DRC Recon Remove Check. For example, RECON_DEFAULT selects the default set of checks, and RECON_CONNECT selects all checks that require connectivity.

  You can customize your check selection using the new predefined rule check groups. This example executes the default checks plus the m1_netarea check, minus all density checks:

  ```
  DRC RECON ADD CHECK RECON_DEFAULT
  DRC RECON ADD CHECK m1_netarea
  DRC RECON REMOVE CHECK RECON_DENSITY
  ```

  See "Selecting Checks in Calibre nmDRC Recon" in the *Calibre Solutions for Physical Verification* manual for an example.

- **Change to DRC Recon Add Check**

  Previously DRC Recon Add Check *added* the specified rule check to the set of checks executed by Calibre nmDRC Recon. Now, *only* the specified rule checks and/or predefined rule check groups are executed. However, the new rule check group keywords enable you to get the same behavior as in the prior release.

  Note

  This change requires a rule file update in order to get similar behavior to the prior release.

  For example:

  ```
  DRC RECON ADD CHECK m1_netarea
  ```

  ○ **2021.1 Result** — Executes m1_netarea *plus* default Calibre nmDRC Recon checks.

  ○ **2021.2 Result** — Executes *only* m1_netarea.

  To get the same behavior as in 2021.1, do the following:

  ```
  DRC RECON ADD CHECK m1_netarea RECON_DEFAULT
  ```

Support for Calibre RVE Priority in DRC Analyze

The DRC Analyze flow now reads the priority from the Calibre RVE "@ RVE Priority" rule check comments. If no priority is found in the rule check, then the default is priority is high.

DFM DV: Limit Number of Results

The new MAXIMUM keyword in the DFM DV operation specifies the maximum number of results that are output. This is useful at the beginning of the design process when the design has many errors, but should not be used if the output of DFM DV is processed by a later operation that expects to see all errors (such as an operation that applies waivers).

New Keyword for [Not] Inside Cell Statement

The [Not] Inside Cell statements have a new NOT WITH LAYER keyword. You can use this keyword to specify a layer, causing the operation to limit selection to cells specified by the [Not] Inside Cell statement that do not have any geometry on the specified layer at the primary levels of the specified cells.

Refer to Inside Cell and Not Inside Cell in the *Standard Verification Rule Format (SVRF) Manual* for more information.

Miscellaneous Updates

- DFM Read is now supported in dual-database mode. (ER 1365567)
- The [NOT] Inside Cell WITH MATCH and Extent Cell WITH MATCH statements cannot be used in dual-database mode.

Calibre Advanced DRC (eqDRC) Tutorial and Example Kit

Introduced in 2021.1, the example kit (eKit) is designed to assist users of Calibre eqDRC and those writing advanced DRC rule checks. The examples correspond to the examples in the “Advanced DRC Solutions” chapter of the *Calibre Solutions for Physical Verification* manual. The eKit contains the data and rule files necessary to run all the examples in the chapter.

Try It!	Go to this page on Support Center to download the complete eKit. The link goes to the latest release (Documentation tab, Document Types=Getting Started Guide, Categories=IC Verification & Signoff).

Calibre nmDRC Previous Release Highlights

Important changes from previous releases are summarized here.

- 2021.1
- 2020.4
- 2020.3
- 2020.2

2021.1

DFM DV Override Enhancements

(2021.1) Several enhancements were made to DFM DV Override:

- Input layer priority specified with the '>' character
- New DVPARAMS argument set, similar to DVPARAMS in DFM DV
- New SELECTED and INVALID keywords to output a subset of the specified input layer

DFM Circle Analyze Update for Annular Polygons

(2021.1) DFM Circle Analyze now has enhanced capability for analyzing annular polygons. An additional property, ANNULUS_OUTRADIUS, is now generated and the property INRADIUS is no longer set to zero for an annulus.

Note

Any derivation that detects circles and relies on the property INRADIUS being set to zero for an annulus should instead use the property ANNULUS_INRADIUS to determine if a polygon is an annulus or circle.

New Command Line Arguments for Calibre nmDRC-H (-rhdb and -hdbflex_acquire)

(2021.1) The Calibre nmDRC-H command line now includes the -rhdb and -hdbflex_acquire arguments. The syntax is:

```
calibre -drc -hier … [ -rhdb {save | restore} filename ]
…
[ {-hyper [ connect ] [ remote]} | -hdbflex | -hdbflex_acquire ]
…
```

- The -rhdb argument specifies a file in which to save or restore an RHDB (Reusable Hierarchical Database) file. An RHDB file is useful in some flows to save time by avoiding the overhead of repeatedly constructing a hierarchical database.
    - save *filename* — Saves a proprietary image of the hierarchical database, generated from all input files in the rule file, as an RHDB file to the specified *filename*. The RHDB file is saved with an embedded checksum. Calibre exits once the RHDB file is saved.
    - restore *filename* — Restores an RHDB file from the specified *filename*. A checksum is generated from all input files and compared with the checksum in the specified *filename*. A mismatch between the checksums generates an error.

  The following statements and command line argument are not supported with -rhdb:

    - DFM Read
    - Layout Place Cell

- Layout Property Audit
- Layout Property Text
- Layout Property Text OASIS
- Inside Cell … [NOT] WITH PROPERTY
- Not Inside Cell … [NOT] WITH PROPERTY
- -hdbflex and -hdbflex_acquire

With the implementation of the -rdhb argument, Calibre now generates the following errors when unable to open an RDHB:

- ERROR: Cannot open RHDB file <*filename*>.

  When saving or restoring an RHDB, the specified <*filename*> could not be opened.
- ERROR: RHDB is not supported with LAYOUT PLACE CELL.

  When saving or restoring an RHDB, the Layout Place Cell statement was encountered in the rule file.
- ERROR: Premature EOF reached when reading RHDB file <*filename*>.

  When restoring an RHDB from <*filename*>, an EOF was prematurely encountered in <*filename*>.
- ERROR: RHDB and HDBFLEX are not supported with properties.

  When saving or restoring an RHDB or running -hdbflex, a DFM Read, Layout Property Audit, [NOT] Inside Cell … [NOT] With Property, or Layout Property Text statement was encountered in the rule file.

Refer to "Calibre File Input Output Errors" in the *Standard Verification Rule Format (SVRF) Manual* for more information.

- The -hdbflex_acquire argument acquires resources in parallel with the HDB (Hierarchical Database) construction process. This can be particularly useful if remote acquisition takes a long time. The -hdbflex_acquire argument differs from the -hdbflex argument, which does not connect to remotes until the latter stages of HDB construction.

Refer to "Calibre nmDRC and Calibre nmDRC-H Command Line" in the *Calibre Verifications User's Manual* for more information on the -rhdb and -hdbflex_acquire arguments.

New Layout Input Exception Severity

(2021.1) The Layout Input Exception Severity statement includes the new TEXT_LENGTH exception. This exception applies to both GDS and OASIS text objects and text in TEXTSTRING records that are excessive in length. The severities are:

0 — Quietly process the text object.
1 — Warn and process the text object. (default)

2 — Warn, truncate the text object to 16383 characters, and process the text object.
3 — Fatal error.

2020.4

New DRC Maximum Results Density Statement

The DRC Maximum Results Density statement specifies a default value for the MAXIMUM keyword in the Density operation. MAXIMUM is used with the RDB keyword in Density to limit the number of results to output to an ASCII results database (RDB). The syntax is:

DRC MAXIMUM RESULTS DENSITY {***number*** **| ALL**}

The ***number*** argument specifies a non-negative integer constant for the number of results to output and **ALL** specifies that all results are output. The setting from DRC Maximum Results Density is used whenever MAXIMUM is not specified in a Density operation and limits the size of the density RDB in the same manner as the MAXIMUM keyword. However, DRC Maximum Results Density does not apply in cases for which the MAXIMUM keyword in the Density operation cannot be used, such as with the COMBINE or MAGNITUDE keywords.

Refer to DRC Maximum Results Density in the *Standard Verification Rule Format (SVRF) Manual* for more information.

Support for Layout Cell List

The [Not] Inside Cell, Layout Window Cell, Layout Windel Cell, and Extent Cell operations now support the use of a LIST argument. Instead of specifying individual cell names for these operations, you can use the LIST argument to specify the name of a list of cells as defined by the Layout Cell List statement.

Refer to Inside Cell, Not Inside Cell, Layout Window Cell, Layout Windel Cell, and Extent Cell in the *Standard Verification Rule Format (SVRF) Manual* for more information.

New Secondary Arguments for [Not] Inside Cell STADIUM Argument

The STADIUM argument for the [Not] Inside Cell operation now includes support for two secondary arguments, TOLERANCE_R *value* and TOLERANCE_45 *value*. These arguments adjust the tolerance for selecting polygons that approximate stadium shapes.

[**Not**] **Inside Cell** ***layer1…***
        [**CIRCLE | ANNULUS |** {**STADIUM** [**TOLERANCE_R** *value*] [**TOLERANCE_45** *value*]}]…

The TOLERANCE_R *value* specifies a positive number dimensioned in user units that modifies the requirements for the circular portion of the stadium shape. When specified, the allowed variation between radii must be less than or equal to *value* instead of the default definition of radii variation for an approximate circle.

The TOLERANCE_45 *value* specifies a floating-point numeric between 0 and 1 inclusive that defines the number of degrees that the 45 degree sides may differ by and still be considered a 45 degree rectangle.

Refer to Inside Cell and Not Inside Cell in the *Standard Verification Rule Format (SVRF) Manual* for more information.

2020.3

DFM Circle Analyze Identifies Annular Regions

DFM Circle Analyze (introduced in 2020.1) now identifies annular regions. The DFM property ANNULUS_INRADIUS is now attached to all output shapes. The property value is nonzero only when the shape is identified as an annulus. The value is the radius of the hole in the annular region. See the command reference for exact definitions.

New Vector Access and Sorting Functions for DFM Expressions

There are several new functions that can be used in DFM expressions within the DFM Property family of operations:

VELEMENT — Returns the specified element in a numeric vector.

SVELEMENT — Returns the specified element in a vector of strings.

NETVELEMENT — Returns the specified element in a netID vector.

SORT_INDEX — Returns a one dimensional vector of index values that map the results of the function SORT() on the input vector back to the input vector indices.

UNIQUE_INDEX — Returns a one dimensional vector of index values that map the results of the function UNIQUE() on the input vector back to the input vector indices.

SORT_UNIQUE_INDEX — Returns a one dimensional vector of index values that map the results of the function SORT_UNIQUE() on the input vector back to the input vector indices.

Hyperscaling of Connectivity Operations

The -hyper command line option now supports a connect option in hierarchical DRC. The new abbreviated syntax is this:

```
calibre -drc -hier -turbo -hyper [connect] [remote] …
```

When connect is specified, hyperscaling applies to connectivity-related operations. This option may increase memory use. The connect option can be particularly beneficial for antenna checks. Refer to "Calibre nmDRC and Calibre nmDRC-H Command Line" in the *Calibre Verification User's Manual* for more information.

Net Area Ratio DETAILED Keyword

The Net Area Ratio operation has a new DETAILED keyword, which may only be specified with BY LAYER. If the DETAILED keyword is specified, then certain additional diagnostic properties appear in the NAR RDB results. Additionally, the entire text of any rule check that generates an RDB appears in the check text of the results. See the DETAILED keyword discussion under "Results Database (RDB) Creation" in the *SVRF Manual* for complete information about the properties.

Extent Cell OVER Keyword

The Extent Cell operation now has an OVER keyword, whose function is basically the opposite of the UNDER keyword introduced in 2018.4. OVER expands the output polygons by a specified distance (like the Size operation with a positive value). As with the UNDER keyword, OVER may not be specified with WITH MATCH and is ignored with a warning in flat DRC runs. OVER is primarily intended for certain waiver flows and is otherwise of limited utility.

Layout Window Cell New Keywords

The Layout Window Cell specification statement has three new keywords: HALO, BY LAYER, and PRIMARY.

Recall that the HALO keyword shrinks the computed window extents by a specified value. Now there is an additional OVER keyword that, when specified, expands the extents instead. The modified syntax is this:

HALO [OVER] *value*

There is also a new BY LAYER keyword set:

BY LAYER *layer_name* [PRIMARY]

BY LAYER *layer_name* specifies that shapes on a given layer inside specified cells form window boundaries rather than the extents of the cell placements. Starting from the bottom of the hierarchy of a specified cell, the given layer is flattened to the top-level cell, and the shapes on the layer form the boundaries of the windows. This may be thought of as executing Layout Window Layer on a per-cell basis.

If the layer specified by *layer_name* is empty for a given cell, then a warning is issued, and no window is applied to that cell.

If PRIMARY is additionally specified, then only layer shapes from the primary level of a cell are used to compute windows rather than shapes from throughout the cell's hierarchy.

BY LAYER may not be specified with OCCUPIED.

[Not] Inside Cell New Keywords

The [Not] Inside Cell operations now support three new keywords: ANNULUS, STADIUM, and HIER.

The ANNULUS and STADIUM keywords are similar to the existing CIRCLE keyword. ANNULUS limits selection of polygons to those that approximate annuli, and STADIUM limits selection of polygons to those that approximate stadium, or obround, shapes. All three keywords are mutually exclusive. These keywords can be useful in certain photonics, through-silicon via, and packaging checks.

The HIER keyword modifies the WITH LAYER keyword as follows:

WITH LAYER [HIER] *layer2*

Recall that WITH LAYER limits selection of polygons to those that are on *layer2* and at the primary levels of specified cells. Specifying HIER changes this behavior to include selection throughout the hierarchies of specified cells.

Miscellaneous Updates

- **Net Area Ratio RDBs Contain Check Text Comments** — Net Area Ratio RDBs now contain rule check user comments (those preceded with “@”). These appear in the RDB when it is generated from within a rule check.
- **WITH MATCH Errors in Flat Applications** — It is now a runtime error for Extent Cell or [Not] Inside Cell to be specified with WITH MATCH in a flat Calibre run. Previously, those usages generated warnings.
- **INP5 Compiler Error Message** — The INP5 compiler error has always been given in Calibre when a layer operation has two identical input layers. The name of a duplicate layer and the operation appear in the error message.

  In some cases, the layer name may not be available. In such cases, “<UNKNOWN>” is now printed for the layer name. In other cases, such as an implicitly defined layer (using a layer operation specified in parentheses), the layer name is now printed as “TMP<*number*>”. This may not be immediately meaningful to a user, but should be traceable in the rule file.

2020.2

DFM DV Updates

(2020.2) DFM DV Analyze now has a LOCALIZE keyword. LOCALIZE causes the operation to attempt to promote results from the input layer(s) so that the layer objects are reported in the vicinity of the corresponding DFM DV result. Promotion to the same level as the DFM DV result is not guaranteed.

In addition, you can now specify a DV layer that was generated using WITH DEFAULT VOLTAGES or WITH VOLTAGE PROPERTIES as an input layer to DFM DV Analyze. This caused an error in previous releases because the DV layer was missing the FIRST_NET or SECOND_NET property. See the manual for more information on this usage.

DFM Classify Updates

(2020.2) DFM Classify has new optional keywords KEEPPROPS and DUPCOUNT.

- UNIQUE [KEEPPROPS] — KEEPPROPS can only be specified with the optional UNIQUE keyword. KEEPPROPS causes all DFM properties on the input layer to be included on the output layer. All properties, including "Class", can be accessed by the PROPERTY function in a DFM expression. However, the "dfm_transform" property is not attached when KEEPPROPS is specified.
- DUPCOUNT — Adds the duplicate count property "DupCnt" to the output layer. The "DupCnt" property contains the total flat count of geometries in each class group.

Also, the numbering of the "Class" property now always starts from 1, which was not previously true.

Property Validator Updates

(2020.2) The property validator now support the DFM Pattern Classify and DFM Pattern Capture operations. The property validator is invoked with the -validprop command line argument. See "Property Validator" in the *SVRF Manual*.

New Exception Severities

(2020.2) The Layout Input Exception Severity statement now includes two new severities:

- AREF_DIAGONAL

  This exception identifies diagonal AREF placements. The cell name of the placement, location of the placement, and the containing cell name are placed into the message.

- REPETITION_TYPE_8_9_10_11

  This exception identifies OASIS repetitions of type 8, 9, 10, or 11. The object and location that the repetition applies to, the repetition type, and the containing cell name are placed into the message.

Refer to the *Standard Verification Rule Format (SVRF)* manual for more information.

Miscellaneous Updates

- **Change to Material Enclosure Criterion for Rectangle Enclosure** — Previously, the Material Enclosure Criterion for the Rectangle Enclosure statement applied if polygons on layer X were oriented at 45-degree angles with respect to the x-axis, together with suitable rotation of layer Y polygons.

  With the 2020.2 release, the behavior of the Material Enclosure Criterion for polygons oriented at 45-degree angles has changed such that it does not apply to 45-degree rectangles.

Calibre nmLVS and Calibre nmLVS-H

This section includes changes and enhancements to Calibre® nmLVS™ and Calibre® nmLVS-H™ for the 2021.2 release.

For additional information, refer to the *Standard Verification Rule Format (SVRF) Manual* and the *Calibre Verification User's Manual*. Also see these Release Notes: Calibre Query Server, Calibre PERC, and Calibre Utilities.

For information about Calibre RVE for LVS, see "Calibre RVE" on page 63.

Calibre nmLVS and nmLVS-H Updates

Calibre nmLVS and Calibre nmLVS-H include the following new features in the 2021.2 release.

- New Calibre nmLVS Reconnaissance ERC Mode
- New qs::softchks Options
- Calibre nmLVS Recon Short Isolation Supports -remotedata Option
- Error Given for Improper SVDB in Calibre nmLVS Recon Runs
- Sconnect Optimizations
- Device Filtering Using Initialized Properties
- V2LVS Performance Enhancements
- V2LVS Comment-Coded Warning for Missing Port Connections
- Change in LVS Reduce Split Gates Default

New Calibre nmLVS Reconnaissance ERC Mode

Calibre nmLVS Reconnaissance ERC was introduced in 2021.1. In this release, there is an additional ERC mode supported through this command line:

> **calibre -recon -erc** [-svdb *svdb_dir* [*top_cell*]] [-turbo [*number_of_processors*] [-turbo_all]] {**-exec *tcl_script*** [*args*] }

In this mode, the **-exec** option is required and specifies a script containing Calibre Query Server Tcl shell or Calibre YieldServer commands. The **-exec** option should be the final one specified in this usage. The optional *args* are arguments passed to the ***tcl_script***.

If -svdb is unspecified, then the ***tcl_script*** must open a Mask SVDB Directory database. Otherwise, the ***tcl_script*** commands operate on the *svdb_dir* (unless overridden by the script). As with all Calibre nmLVS Recon runs, the *svdb_dir* must be created using the SI keyword of the Mask SVDB Directory statement. The *top_cell* is an optional primary cell name for a design represented in the *svdb_dir*. This argument is useful if there is more than one such cell in the database.

The -turbo options are as previously documented for other run modes.

See "Calibre nmLVS Reconnaissance Command Line" in the *Calibre Verification User's Manual* for additional details. ER 1436095, 1436841

New qs::softchks Options

The qs::softchks command was introduced in 2021.1 as part of the Calibre nmLVS Recon Softchk tool. This command can appear in the script referenced by the calibre -recon -softchk=*tcl_script* command line (it may not be used in the Query Server Tcl shell, however). In this release, two new options are introduced for this command.

The **-config_file** option specifies a file containing instructions for how to perform soft connection checking. This option can be used instead of **-sconnect_conflict**, which is the default. Either one of these options must be specified.

The -by_contact_count option specifies that the prevailing net in a stamping conflict is chosen based upon the total contact count on a net. Contacts are those defined by the Sconnect BY keyword for layers involved in a conflict. This option can be used instead of -by_vertex_count, which is the default.

Details are in the *Calibre Query Server Manual*. ER 1436752

Calibre nmLVS Recon Short Isolation Supports -remotedata Option

Stand-alone short isolation (calibre -recon -si) now supports the -remotedata option in MTflex mode. Previous release documentation showed this option as being available with -recon -si, but it was not supported until 2021.2. Use of this option is recommended. ER 1441523

Error Given for Improper SVDB in Calibre nmLVS Recon Runs

When calibre -recon is executed with the -svdb option, the Mask SVDB Directory SI keyword must have been in effect when the specified SVDB was created, or this error is now given:

```
Error 302: ERROR: SVDB not produced using Mask SVDB Directory SI.
```

DR 1436661

Sconnect Optimizations

In a circuit extraction transcript, these entries appear when Sconnect is used:

```
FINALIZE NODE NUMBERS ON SCONNECT LAYERS

REMOVING SCONNECT/HIGH SHORT REMOVAL TRIGGERED TRIVIAL PINS/NETS...
```

These steps have been optimized for better runtime. In most designs, the optimizations are not noticeable, but when they are, runtime reductions of up to 90% have been observed for these steps. DR 1433094

Device Filtering Using Initialized Properties

See "Support for Device Filtering Using Initialized Properties" on page 114.

V2LVS Performance Enhancements

V2LVS runtime and memory performance have been improved. Runtime reductions of up to 30% and memory reductions of up to 15% have been observed in large designs. ER 1443330

V2LVS Comment-Coded Warning for Missing Port Connections

V2LVS has been enhanced to write a comment-coded warning when port connections are missing in a Verilog module. When this situation is encountered, the output netlist now has a warning like this in the scope of the subcircuit in which the warning applies:

```
*.CALIBRE v2lvs warning: in <verilog_file> line <N>, Verilog assignment
<instance_pin> is missing the port <net> in the subcircuit <name>.
```

Consider this example. Here is the top-level Verilog netlist:

```
module top (net1, net2, net3, VDD, VSS);
CELL1 inst1( .in1(net1), .in2(net2), .out3(net3), .VDD(VDD), .VSS(VSS));
CELL1 inst2( .in1(net1), .in2(net2), .out3(net3), .VDD(VDD), .VSS(VSS));
endmodule
```

Here is the SPICE library netlist:

```
.global VDD
.global VSS
.SUBCKT CELL1 in1 in2 out3
.ENDS
```

Here are the V2LVS commands:

```
v2lvs::load_verilog -filename design.v
v2lvs::load_spice -filename lib.sp -pin_mode
v2lvs::write_output -filename design.out
exit
```

In past releases, *design.out* would show this by default:

```
.SUBCKT top net1 net2 net3 VDD VSS
Xinst1 CELL1 $PINS in1=net1 in2=net2 out3=net3 VDD=VDD VSS=VSS
Xinst2 CELL1 $PINS in1=net1 in2=net2 out3=net3 VDD=VDD VSS=VSS
.ENDS
```

Now, *design.out* contains warnings:

```
.SUBCKT top net1 net2 net3 VDD VSS
Xinst1 CELL1 $PINS in1=net1 in2=net2 out3=net3 VDD=VDD VSS=VSS
*.CALIBRE v2lvs warning: in design.v line 2, Verilog assignment .VDD(VDD)
is missing the pin VDD in the subcircuit CELL1.
*.CALIBRE v2lvs warning: in design.v line 2, Verilog assignment .VSS(VSS)
is missing the pin VSS in the subcircuit CELL1.
Xinst2 CELL1 $PINS in1=net1 in2=net2 out3=net3 VDD=VDD VSS=VSS
.ENDS
```

These warnings are always shown now when port connections are missing. ER 1416795

Change in LVS Reduce Split Gates Default

In previous releases, the LVS Reduce Split Gates default was YES. Now, it is NO if the PARALLEL or SPLIT keyword is specified for LVS Short Equivalent Nodes. DR 1441923

Calibre nmLVS and nmLVS-H Previous Release Highlights

Calibre nmLVS and Calibre nmLVS-H introduced the following updates in previous releases.

2021.1

Calibre nmLVS Reconnaissance ERC and Softchk

Calibre nmLVS Reconnaissance now supports two additional modes. Previously, stand-alone short isolation was introduced. Now, stand-alone ERC and soft connection checking are also supported.

The new run modes are enabled by this command line:

> **calibre -recon** { { { **-erc=*svrf_script*** } | { **-softchk**[=*tcl_script*] } }
> **-svdb *svdb_dir*** [*top_cell*] [-turbo [*number_of_processors*] [-turbo_all] }

Both run modes take an input ***svdb_dir*** that was generated using the Mask SVDB Directory SI keyword in a previous LVS circuit extraction run. The rule file for the run is one that is typically used for LVS circuit extraction.

If **-erc=*svrf_script*** is specified, then ERC rule checking is performed according to the contents of the ***svrf_script***. The SVRF elements permitted in the ***svrf_script*** are the same ones that can be used for dfm::new_layer in its default mode.

As a basic example, if your circuit extraction rules contain LVS Power Name and LVS Ground Name statements, then you can include this in the ***svrf_script***:

```
float = PATHCHK !POWER && !GROUND
erc.results {
  DFM RDB erc.db float ALL CELLS CELL SPACE COMMENT "no supply path"
}
```

Your ERC results then are output in the *erc.rdb*, which can be loaded into Calibre RVE for analysis.

If **-softchk** is specified, all Sconnect stamped layers (the second layer arguments in Sconnect operations) for which there was a stamping conflict during the circuit extraction run are processed. Soft connection checking is then performed as if LVS Softchk LOWER were specified in a circuit extraction run. The results are output to a *softchk.rdb* file, which is of the same form as the **.softchk* database produced by LVS Softchk.

If **-softchk**=*tcl_script* is specified, then Query Server Tcl shell commands in the *tcl_script* are executed. Two of the commands intended for this flow are discussed in the Query Server's previous release highlights under "2021.1" on page 106.

The multithreading options work as they do for LVS circuit extraction.

See "Calibre nmLVS Reconnaissance Command Line" in the *Calibre Verification User's Manual* for further details. See "Calibre nmLVS Reconnaissance" in the *Calibre Administrator's Guide* for licensing information. ER 1427686, 1427689, 1415115, 1436095, 1432973

Mask SVDB Directory SI Expanded Usage

The Mask SVDB Directory SI keyword historically has applied to interactive short isolation. Now, this keyword also applies to any Calibre nmLVS Reconnaissance run that loads a PHDB. Specifically, if "calibre -recon …-svdb *svdb_dir*" is executed on the command line, then the Mask SVDB Directory SI keyword must have been active in the rule file when the *svdb_dir* PHDB was created. If the SI keyword was not active when producing the PHDB, then an error is given in the transcript. ER 1414756

LVS Softchk Performance Enhancements

LVS Softchk performance has been enhanced. Runtime savings of about 20% have been observed in certain designs, with some showing even greater improvement.
ER 1429874, 1414751

Circuit Comparison Performance Improvement

Circuit comparison has been optimized to run faster in certain designs. Runtime reductions of up to 35% have been observed. Runtime savings will be most prominent in designs containing cells with very large numbers of instances or nets (hundreds of millions). The relevant instance and net counts are those reported for each cell in the NUMBERS OF OBJECTS AFTER TRANSFORMATION sections in the LVS report.

With this optimization, a new "Initialization" step is now reported for each cell in the circuit comparison transcript. This is the step whose performance was optimized. Previously, this step was executed as well, but its run time was not explicitly reported. Here is an example from a new transcript:

```
COMPARING layout and source top levels...

Initialization. CPU TIME = 0 REAL TIME = 0

ANCHOR Points. 0 instances, 0 nets matched. CPU TIME = 0 REAL TIME = 0
```

DR 1430952

New -add Option for erc::setup_device_parameters

The erc::setup_device_parameters command has a new -add option. This option is similar in many respects to the existing -total option, but instead of calculating total counts of parallel instances, the -add option causes summation of an existing SPICE parameter's value for parallel instances and outputs the sum as the value of a new property. For devices not connected in parallel, the sum is equal to the existing property's value. ER 1432426

device::blocked_extent -size_by Option

The device::blocked_extent property computation command introduced in 2020.4 has a new -size_by option. When this option is used, the extents of cells are expanded by a specified value in a similar way to the Size layer operation. ER 1428627, 1432286, 1438771

LVS Recognize Gates WITH SUBSTRATE Keyword

LVS Recognize Gates has a new WITH SUBSTRATE keyword. When this keyword is used, logic gate recognition generates ports that connect to the substrate pins of transistors internal to gates.

A gate is composed of two or more transistors. Inside a gate, a net is called a *substrate net* if it connects to a substrate pin of a transistor. By default, substrate nets are invisible from outside the logic gate and do not contribute to ambiguity resolution. In some circumstances, ambiguity resolution may produce a false INCORRECT comparison because the substrate nets inside gates are mismatched. The WITH SUBSTRATE keyword causes the substrate nets to be connected to ports of a gate and the substrate nets are checked during ambiguity resolution. The false INCORRECT caused by the mismatched substrate nets is eliminated.

Further details are in the *SVRF Manual*. ER 1415185

Trace Property Dimensions Enhancement for User-Defined Devices

The Trace Property statement now supports specification of scaling factors and unit label names for user-defined devices (those that are not built-in for comparison). Together, the factor and label give desired dimensions to reported property values. The new abbreviated syntax is this:

```
TRACE PROPERTY
  <component_type> [ '(' <component_subtype> ')' ]
  <source_property> <layout_property> [<tolerance>]
  [ SCALE <factor> UNITS "<string>" ]
```

The SCALE keyword takes a numeric *factor* as an argument. The *factor* is a positive number that acts as a divisor for a traced property value. The UNITS keyword takes a quoted *string* argument that is the label name for the dimensions of the traced property. SCALE and UNITS must be used together. Failure to provide expected data types for the *factor* or *string* causes a compiler error.

To illustrate these new keywords, assume this Trace Property statement:

```
TRACE PROPERTY mydev z z 0.1
```

This statement may produce a discrepancy as follows:

```
************************************************************************
                       PROPERTY ERRORS

DISC# LAYOUT                                  SOURCE                 ERROR
************************************************************************

  1   X2(72.530,89.950) mydev                 XX0 mydev
      z: 4e-07                                z: 4.5e-07             5e-08
```

The reported property values are dimensionless, and the error is absolute. When the SCALE and UNIT keywords are added:

```
TRACE PROPERTY mydev z z 0.1 SCALE 1.0e-6 UNITS "u"
```

the discrepancy report becomes this:

```
************************************************************************
                       PROPERTY ERRORS

DISC# LAYOUT                                  SOURCE                 ERROR
************************************************************************

  1   X2(72.530,89.950) mydev                 XXTL0 mydev
      z: 0.4 u                                z: 0.45 u              11.1%
```

The properties now have specified dimensions, and the error is relative. ER 1365011

Change to LVS Report Option NOK Behavior

The LVS Report Option NOK keyword historically has caused detailed information to be reported for INCORRECT hcells only. In this release, CORRECT hcells have their object counts reported in addition to the INCORRECT hcell information included previously. Other detailed information, such as warning status, initial correspondence points, and ambiguity resolution details are not reported for CORRECT cells. This change aids in performance analysis of hierarchical LVS runs. The new report information looks like this:

```
                   CORRECT CELL CONTENTS SUMMARY
                   =============================


LAYOUT CELL NAME:         AND2A
SOURCE CELL NAME:         AND2A

----------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                Layout    Source         Component Type
                ------    ------         --------------
 Ports:               5         5
 Nets:                7         7
 Instances:           3         3         MN (4 pins)
                      3         3         MP (4 pins)
                ------    ------
 Total Inst:          6         6


NUMBERS OF OBJECTS AFTER TRANSFORMATION
---------------------------------------

                Layout    Source         Component Type
                ------    ------         --------------
 Ports:               5         5
 Nets:                6         6
 Instances:           1         1         INV (2 pins)
                      1         1         NAND2 (3 pins)
                ------    ------
 Total Inst:          2         2
```

ER 1436574

Calibre nmLVS Reconnaissance Short Isolation Licensing Change

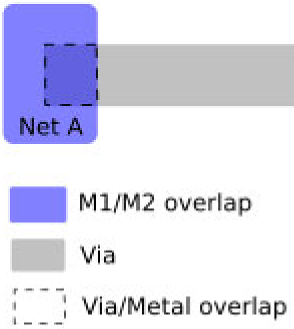
As of this release, a Query Server license is only required for stand-alone short isolation when a Tcl script is specified on the command line, such as in this abbreviated syntax:

```
calibre -recon -si=qs_script.tcl ...
```

ER 1432519

Flat Short Isolation Support for Dogbone-Shaped Vias

Flat short isolation has been enhanced to find short paths in which the contact or via geometries act as primary connectors in the paths. Finding such shorts has been referred to as the dog bone (also H, or dumbbell) via problem because the connection shape resembles such figures.



Assume the rule file has the statement CONNECT M1 M2 BY VIA. The distinguishing feature of the connection in the preceding figure is that the short between nets A and B is made through the VIA layer. Previously a short path that involves this configuration would not have been isolated by the flat algorithm, and the following error would have been written to the transcript:

```
ERROR:  Short isolation failed - no short found.
```

The algorithm now isolates and reports such paths. DR 1432398

New Calibre nmLVS Recon SI SVDB Error

Starting in this release, if a Mask SVDB Directory that is loaded in a calibre -recon -si -svdb run was not created by a previous -recon -si run, or by a circuit extraction run in which the SI keyword was specified in a Mask SVDB Directory statement, then the following error is given:

```
Error 300: Short Isolation on SVDB requires that the SVDB be generated
with MASK SVDB DIRECTORY SI.
```

ER 1429622

Change to LVS Short Equivalent Nodes MATRIX Behavior

The LVS Short Equivalent Nodes MATRIX keyword behavior has been modified. Previously, a criterion for matrix reduction was this:

> No series or parallel MOS reduction, including generic LVS Reduce reduction. (LVS Reduce Parallel MOS is set to YES by default.)

Now, this applies:

> No series or parallel MOS reduction, including generic LVS Reduce reduction for MOS. LVS Reduce Parallel MOS NO must be set or an error results. This reduction restriction does not apply to other built-in device types.

ER 1436592

Wildcard Support in LVS SI Select Connects BY LAYER

The LVS SI Select Connects BY LAYER keyword accepts “*” wildcards in its layer name arguments. The wildcard matches zero or more characters. This feature was present in 2020.4.

V2LVS Performance Improvements

V2LVS performance has been improved. The improvements are most noticeable in cases where there are tens of millions of instantiations of undeclared Verilog modules. Up to 40% reduction in runtime and 75% reduction in memory have been observed. ER 1435342

Reduction Data Logging Optimization

The real-time performance of reduction data logging, which occurs when Mask SVDB Directory XFORMS is specified, has been optimized. ER 1433868

2020.4

LVS Expand On Error Enhancements

LVS Expand On Error was introduced in 2020.3. To review, some hcells lead to LVS errors that can be mitigated by expanding hcells. For example, if an hcell in the source design contains a device that is in its parent hcell in the layout design, there will be LVS discrepancies unless that hcell is expanded. Specifying LVS Expand On Error YES allows such hcells to be iteratively expanded to eliminate LVS errors caused purely by the hcell list.

In this release, a bad-hcell prediction algorithm is introduced. The prediction algorithm assesses whether an hcell will likely contribute to LVS discrepancies. If no hcells are predicted to cause LVS discrepancies, LVS comparison follows immediately.

Bad-hcell prediction occurs after all flattening and transforming has been done, just prior to comparison. If bad hcells are identified, they are removed from the internal hcell list. LVS is immediately restarted before comparison occurs, so no LVS report is generated for the partial run. After restarting, Calibre reads and transforms the netlists again. Then, it reruns bad-hcell prediction, which may find additional bad hcells, leading to their removal from the internal list followed by another restart. When no more bad hcells are found, LVS moves on to comparison with the revised hcell list.

As in the initial release, the comparison results are evaluated for bad hcells when the results are INCORRECT. Bad hcells are marked for removal from the internal list if found. In cases where post-comparison evaluation removes hcells and restarts LVS, bad-hcell prediction runs again in each new pass.

When bad-hcell prediction finds bad hcells, this is reflected in the transcript as follows:

```
Performing PRECOMPARISON PREDICTION work ...

Predicting BAD HCELLS ...

  Predicted bad hcells:
    cell1   cell1
    cell2   cell2

BAD-HCELL prediction completed.  CPU TIME = 0  REAL TIME = 0  LVHEAP = 1/
3/3  MALLOC = 40/40/40  ELAPSED TIME = 0

PRECOMPARISON PREDICTION work completed.  CPU TIME = 0  REAL TIME = 0
LVHEAP = 1/3/3  MALLOC = 40/40/40  ELAPSED TIME = 0

Hcells were predicted bad. Restarting LVS.

Initializing LVS ...
```

If no bad hcells are found, none are listed, and LVS is not restarted. In this case, a transcript entry like the following appears:

```
Performing PRECOMPARISON PREDICTION work ...

  PRECOMPARISON PREDICTION work completed.  CPU TIME = 0  REAL TIME = 0
  LVHEAP = 1/3/3  MALLOC = 40/40/40  ELAPSED TIME = 0
```

Sometimes bad-hcell prediction finds hcells that cannot be expanded because they are listed in cell lists, or because instances of them are contained in other hcells listed in cell lists. The transcript output for such cases looks like this:

```
Predicted bad hcells:
    cell1   cell1   (not expandable due to cell list)
    cell2   cell2
```

If none of the bad hcells can be expanded, LVS does not restart, and they are only listed in the transcript. There is no other side effect.

Bad hcells detected by the prediction algorithm that were expanded are listed in a new subsection of the EXPAND ON ERROR DETAILS portion of an LVS report. Here is an example:

```
**************************************************************************
                           EXPAND ON ERROR DETAILS
**************************************************************************

 The INCORRECT LVS report prior to EXPAND ON ERROR is available in:

         qaout.svdb/qaout.lvs.rep.pass_1

 CELLS EXPANDED DUE TO BAD-HCELL PREDICTION:

    Result         Layout                         Source
    -----------    -----------                    --------------
    EXPANDED       cell1                          cell1
    EXPANDED       cell2                          cell2

CELLS EXPANDED DUE TO ERRORS PRIOR TO EXPANSION:

    Result         Layout                         Source
    -----------    -----------                    --------------
    EXPANDED       cell3                          cell3
```

There is also a new overall status message when bad-hcell prediction expands hcells:

```
Warning:  Some hcells were predicted bad prior to comparison and expanded
(LVS EXPAND ON ERROR).
```

ER 1427616

Database-Only Mode for calibre -recon -si (Calibre nmLVS Reconnaissance Short Isolation)

In 2020.3, stand-alone short isolation was introduced, which is initiated with the calibre -recon -si command line options. (See "Calibre nmLVS Reconnaissance Short Isolation" in the previous release highlights.) In this release, a new DB modal argument is introduced for the -si option. The abbreviated syntax is now this:

calibre -recon -si['='{ ALL | DB | IO | PG | *tcl_script* }] {***rule_file_name*** | **-svdb** ***svdb_dir***}

This new option is intended for use in a two-step flow, like this:

```
calibre -recon -si=DB -turbo rules
calibre -recon -si -svdb svdb
```

The DB modal argument (database-only mode) is specified to create the Mask SVDB Directory with sufficient information to perform short isolation in a later step. This can be beneficial in two cases. First, running connectivity extraction in multithreaded mode without remote servers is the fastest way to produce a PHDB. Second, if it is suspected there will be large numbers of shorts found, it can be faster to generate the SVDB up front and then process it for shorts in later runs using the **-svdb** option.

Whenever the DB argument is used, the transcript will show this message if there is sufficient information in the SVDB to perform short isolation: SVDB READY FOR RECON SI.
ER 1421698

Calibre nmLVS Recon SI Now Supports MTflex

Stand-alone short isolation now supports MTflex distributed mode. (This was mistakenly shown as supported in the *Calibre Verification User's Manual* in 2020.3.) This is the command usage:

```
calibre -recon -si['='{ALL | DB | IO | PG | tcl_script}]
[  -turbo [ number_of_processors ] [ -turbo_all]
[ { -remote host,host,… | -remotefile filename |
    -remotecommand filename count }
 [ -remotedata [ -recoverremote | -recoveroff ] ] ] ]
{ { [ -hyper [ -remote ] ] rule_file_name } | { -svdb svdb_dir
  [ top_cell ] } }
```

If the *tcl_script* modal argument is used, then MTflex only applies to short isolation in the Query Server portion of the run. The license acquisition formula is the same as for other Calibre products that support MTflex.

The primary or local host distributes the shorts to the remote processes and then collects the results from the remotes. By default, each remote process is assumed to consume no more than 100 GB of memory. For example, a remote machine with 1 TB of memory will have at most 10 remote processes for distributed short isolation. This distribution strategy can be adjusted using the new Tcl Query Server command short_db::set_distribution_configuration.

If a remote machine does not have the requested resources, it will be skipped and one of the following messages is written in the transcript:

```
<remote-host-name> has 3 CPUs (16 CPUs are requested) - It will not be
used in distributed SI.
```

or

```
<remote-host-name> has 50000 MB of memory (100000MB minimum is requested)
- It will not be used in distributed SI.
```

If no remote hosts can be found to satisfy the criteria, all processes are executed locally, and the following message is issued:

```
Remote adjustment failed: executing SI on primary.
```

If a remote process fails to complete the analysis of a short, the analysis is restarted on the primary host. The results of locally restarted shorts are in the following RDB files:

./<SVDB>/<TOP_CELL>.phdb/isi/<TOP_CELL>-default-sn<N>.shorts

The time and memory statistics of the overall run are printed to the transcript in the following format:

```
<process ID> <host name> <CPU time> <real time> <lvheap> <malloc> <elapsed
time>
```

Because this mode generates multiple results in parallel, it is possible for the number of results to exceed the limit prescribed in LVS Maximum Short Results. In addition, the short serial numbers found in the RDB files may not match between distributed and non-distributed runs. ER 1399523, 1363683

Calibre nmLVS Recon SI Defaults to BY CELL BY LAYER Results

In 2020.3, when -recon -si was specified on the command line without any modal arguments (ALL, IO, and so forth), stand-alone short isolation was run by default as if LVS Isolate Shorts YES was specified without any additional keywords. If there were additional keywords specified in the rule file, then those also applied. The behavior was the same if a rule file was specified or the -svdb option was used.

In this release, the default is to use the BY CELL BY LAYER keywords of LVS Isolate Shorts YES, which affect results presentation. As before, the behavior is the same if a rule file or the -svdb option is specified. In the latter case, the Query Server short_db::isolate_shorts command options -by_cell -by_layer are used, which are the analogs to BY CELL BY LAYER. Other results presentation behaviors remain as before. ER 1421456

Calibre nmLVS Recon SI Error Message Changes

Error messaging for Calibre nmLVS Recon SI has changed as follows.

For LVS SI Select Connects:

- This message is given when the BY keyword is used, but its layer specification list is inconsistent with the set of Connect and Sconnect statements:

  ERROR: Error EXT16 on line <n> of <rules> - Invalid LVS SI SELECT CONNECTS specification.

- This message is given when the BY LAYER keyword is used, but a layer in its layer list is inconsistent with the set of Connect or Sconnect statements:

  ERROR: Error EXT3 on line <n> of <rules> - cannot establish connectivity of this layer in the MASK connectivity set: <layer_name>.

For short_db::select_connects:

- This message is given when -select_by is specified, but its layer specification list is inconsistent with the set of Connect and Sconnect statements:

  Error EXT17 - Invalid [S]CONNECT layer specification in short_db::select_connects -select_by.

- This message is given when -select_by_layer is used but a layer in its layer list is inconsistent with the set of Connect or Sconnect statements:

  ERROR: - cannot establish connectivity of this layer in the MASK connectivity set: <layer_name>.

- This message is given when -select_by_layer is used, but a layer in its layer list is unavailable:

  Error: invalid layer <layer_name>

ER 1423841

LVS Comparison Reduced Memory Consumption with Additional Disk Space Usage

LVS comparison has been modified to reduce memory consumption when Mask SVDB Directory is specified. Memory savings in the range of 10-20% are typical. One production design showed an improvement of about 35%.

To achieve this reduction, some data formerly stored in memory is now stored in temporary files in the SVDB directory. These files are created any time LVS comparison runs and Mask SVDB Directory is specified. If it is unspecified, then the file-based memory optimizations are not used, and additional memory is required.

Note

Test cases indicate reasonable approximations of the extra disk space required for the SVDB are 25-50% of the combined size of the uncompressed layout and source netlists or 5-10% of the maximum comparison LVHEAP when using a prior version of Calibre. In exceptional cases, disk space requirements were as much as 75% of the combined size of uncompressed netlists and 30% of the previous LVHEAP maximum.

DR 1429404, 1420118, and ER 1427713, 1428959

Improved Transcript Messaging for Hyperscaling Runs

The transcript messaging for hyperscaling runs (-turbo -hyper) has been improved for easier assessment of memory usage and time durations.

The GRAND TOTAL line has been reformatted to include time and memory summary information across all hyperscaling HDBs and the LVS comparison run. This is consistent whether performed in a separate process (-hyper cmp, and so forth), host (-cmp_host/ -cmp_remotefile), or on the main process. The hyperscaling HDB reporting after LVS extraction has been slightly modified to be consistent with the other HDB reporting.

The TOTAL line underneath the CALIBRE::LVS/xRC COMPLETED line has been deleted since is contains the same information as the CALIBRE::LVS/xRC EXECUTIVE MODULE COMPLETED line. The CALIBRE::LVS/xRC EXECUTIVE MODULE COMPLETED line has also been reworded to be more clear.

The following comparison of LVS transcript entries is for MT mode (-turbo -hyper).

Old transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Mon Jul 20
16:57:27 2020
--- TOTAL CPU TIME = 217  REAL TIME = 54  LVHEAP = 55/208/212  SHARED = 0/
64
MALLOC = 256/256/259  ELAPSED TIME = 55
HDB 0: CPU TIME = 0  LVHEAP = 212
HDB 1: CPU TIME = 20  LVHEAP = 68  RX = 22  TX = 7
HDB 2: CPU TIME = 18  LVHEAP = 72  RX = 23  TX = 6
HDB 3: CPU TIME = 20  LVHEAP = 64  RX = 25  TX = 9
HDB 4: CPU TIME = 18  LVHEAP = 74  RX = 23  TX = 7
HDB 0-4 TOTAL LVHEAP = 490

... <Full LVS comparison transcript> ...

--- CALIBRE::LVS/xRC EXECUTIVE MODULE COMPLETED.  CPU TIME = 128  REAL
TIME = 146  LVHEAP = 15/3907/3992  MALLOC = 4018/4018/4064  ELAPSED TIME =
206
--- CALIBRE::LVS/xRC COMPLETED - Mon Jul 20 16:59:58 2020
--- TOTAL CPU TIME = 128  REAL TIME = 146  LVHEAP = 15/3907/3992  MALLOC =
4018/4018/4064  ELAPSED TIME = 206
     .
     .
     .
--- GRAND TOTAL CPU TIME = 345  REAL TIME = 205  LVHEAP = 5/3907/3992
MALLOC = 4018/4018/4064  ELAPSED TIME = 206
```

New transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Mon Aug 24
14:44:50 2020
--- TOTAL CPU TIME = 216  REAL TIME = 52  LVHEAP = 55/208/213  SHARED = 0/
64
MALLOC = 256/256/260  ELAPSED TIME = 53
--- HDB 0     : CPU TIME = 0     LVHEAP = 213   MALLOC = 260
--- HDB 1     : CPU TIME = 20    LVHEAP = 66    RX = 9     TX = 27
--- HDB 2     : CPU TIME = 17    LVHEAP = 66    RX = 8     TX = 24
--- HDB 3     : CPU TIME = 21    LVHEAP = 68    RX = 6     TX = 22
--- HDB 4     : CPU TIME = 18    LVHEAP = 70    RX = 6     TX = 22
--- EXTRACTION TOTAL HDB 0-4   CPU TIME = 76  REAL TIME = 52  LVHEAP = 483

... <Full LVS comparison transcript> ...

--- calibre::lvs comparison module completed. total cpu time = 131  real
time = 151  lvheap = 15/3908/3992  malloc = 4016/4016/4062  elapsed time =
208
```

```
--- CALIBRE::LVS/xRC COMPLETED - Mon Aug 24 14:47:25 2020
     .
     .
     .
--- HDB 0/LVS COMPARE : CPU TIME = 131   LVHEAP = 3992  MALLOC = 4062
--- HDB 1             : CPU TIME = 20    LVHEAP = 66    RX = 9     TX = 27
--- HDB 2             : CPU TIME = 17    LVHEAP = 66    RX = 8     TX = 24
--- HDB 3             : CPU TIME = 21    LVHEAP = 68    RX = 6     TX = 22
--- HDB 4             : CPU TIME = 18    LVHEAP = 70    RX = 6     TX = 22
--- GRAND TOTAL HDB 0-4 TOTAL  CPU TIME = 207  REAL TIME = 208 LVHEAP =
4262
```

The following comparison of transcript entries is for MTflex mode remote LVS comparison (-turbo -hyper with -cmp_host or -cmp_remotefile).

Old transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Tue Jul 14
10:22:36 2020
--- TOTAL CPU TIME = 222  REAL TIME = 71  LVHEAP = 56/208/211  SHARED = 0/
64
MALLOC = 258/258/262  ELAPSED TIME = 74
HDB 0: CPU TIME = 0  LVHEAP = 211
HDB 1: CPU TIME = 19  LVHEAP = 66  RX = 18  TX = 4
HDB 2: CPU TIME = 23  LVHEAP = 64  RX = 21  TX = 8
HDB 3: CPU TIME = 16  LVHEAP = 70  RX = 26  TX = 8
HDB 4: CPU TIME = 17  LVHEAP = 72  RX = 29  TX = 10
HDB 0-4 TOTAL LVHEAP = 483
     .
     .
     .
--- CALIBRE::LVS/xRC EXECUTIVE MODULE COMPLETED.  CPU TIME = 152  REAL
TIME = 200  LVHEAP = 11/3905/3989  MALLOC = 4014/4014/4061  ELAPSED TIME =
245
--- CALIBRE::LVS/xRC COMPLETED - Tue Jul 14 10:25:29 2020
--- TOTAL CPU TIME = 152  REAL TIME = 200  LVHEAP = 11/3905/3989
MALLOC = 4014/4014/4061  ELAPSED TIME = 245
     .
     .
     .
--- GRAND TOTAL CPU TIME = 374  REAL TIME = 245  LVHEAP = 5/207/211  MALLOC
= 247/247/262  ELAPSED TIME = 247
```

New transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Mon Aug 24
14:34:45 2020
--- TOTAL CPU TIME = 216  REAL TIME = 52  LVHEAP = 55/208/212  SHARED = 0/
64
MALLOC = 256/256/260  ELAPSED TIME = 54
--- HDB 0     : CPU TIME = 0     LVHEAP = 212   MALLOC = 260
--- HDB 1     : CPU TIME = 19    LVHEAP = 72    RX = 11    TX = 28
--- HDB 2     : CPU TIME = 17    LVHEAP = 62    RX = 11    TX = 26
--- HDB 3     : CPU TIME = 20    LVHEAP = 72    RX = 5     TX = 23
--- HDB 4     : CPU TIME = 20    LVHEAP = 64    RX = 4     TX = 17
--- EXTRACTION TOTAL HDB 0-4   CPU TIME = 76  REAL TIME = 52  LVHEAP = 482
```

```
     .
     .
     .
--- CALIBRE::LVS REMOTE COMPARISON MODULE COMPLETED. TOTAL CPU TIME = 127
REAL TIME = 165  LVHEAP = 11/3903/3987  MALLOC = 4008/4008/4055  ELAPSED
TIME = 199
--- CALIBRE::LVS/xRC COMPLETED - Mon Aug 24 14:37:12 2020
     .
     .
     .
--- HDB 0      : CPU TIME = 0     LVHEAP = 212   MALLOC = 260
--- HDB 1      : CPU TIME = 19    LVHEAP = 72    RX = 11    TX = 28
--- HDB 2      : CPU TIME = 17    LVHEAP = 62    RX = 11    TX = 26
--- HDB 3      : CPU TIME = 20    LVHEAP = 72    RX = 5     TX = 23
--- HDB 4      : CPU TIME = 20    LVHEAP = 64    RX = 4     TX = 17
--- LVS COMPARE: CPU TIME = 127   LVHEAP = 3987  MALLOC = 4055
--- GRAND TOTAL HDB 0-4 + LVS COMPARE TOTAL  CPU TIME = 203  REAL TIME =
199 LVHEAP = 4469
```

The following comparison of transcript entries is for MT circuit extraction only.

Old transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Mon Jul 20
17:02:36 2020
--- TOTAL CPU TIME = 216  REAL TIME = 52  LVHEAP = 55/208/212  SHARED = 0/
64
MALLOC = 262/262/266  ELAPSED TIME = 53
HDB 0: CPU TIME = 0  LVHEAP = 212
HDB 1: CPU TIME = 19  LVHEAP = 66  RX = 26  TX = 9
HDB 2: CPU TIME = 23  LVHEAP = 62  RX = 25  TX = 10
HDB 3: CPU TIME = 18  LVHEAP = 76  RX = 24  TX = 7
HDB 4: CPU TIME = 15  LVHEAP = 68  RX = 19  TX = 4
HDB 0-4 TOTAL LVHEAP = 484
```

New transcript:

```
--- CALIBRE::HIERARCHICAL CIRCUIT EXTRACTOR COMPLETED - Mon Aug 24
14:48:18 2020
--- TOTAL CPU TIME = 217  REAL TIME = 51  LVHEAP = 55/208/212  SHARED = 0/
64
MALLOC = 256/256/260  ELAPSED TIME = 52
--- HDB 0     : CPU TIME = 0     LVHEAP = 212   MALLOC = 260
--- HDB 1     : CPU TIME = 17    LVHEAP = 74    RX = 9     TX = 27
--- HDB 2     : CPU TIME = 21    LVHEAP = 70    RX = 6     TX = 23
--- HDB 3     : CPU TIME = 18    LVHEAP = 62    RX = 9     TX = 22
--- HDB 4     : CPU TIME = 20    LVHEAP = 66    RX = 6     TX = 22
--- EXTRACTION TOTAL HDB 0-4   CPU TIME = 76  REAL TIME = 51  LVHEAP = 484
     .
     .
     .
--- GRAND TOTAL HDB 0-4 TOTAL  CPU TIME = 76  REAL TIME = 53  LVHEAP = 484
```

DR 1343930

Enhanced Messaging for LVS Softchk Results

Messaging has been enhanced when LVS Softchk Maximum Results is specified with a value other than the default. Warnings are given per LVS Softchk statement showing the maximum results limit. Summary warnings are also given at the end of the transcript and in the LVS report for a comparison run. These warnings are discussed in the *SVRF Manual*.

When no limit is exceeded, the number of SOFTCHK results is now reported per LVS Softchk statement like this:

```
Number of polygons produced by the current SOFTCHK operation = 6
```

and as a summary total at the end of the SOFTCHK module:

```
LVS SOFTCHK TOTAL RESULTS GENERATED = 32
```

ER 1420218, 1420225, 1426207

LVS Softchk Maximum Results Behavior Change

Previously, if the number of LVS Softchk results exceeded the LVS Softchk Maximum Results limit, the processing of soft connections continued, even though the number of results that were output conformed to the maximum setting. In this release, processing of soft connections stops when the maximum is reached, which can result in improved performance. ER 1420217

New device::blocked_extent Function for Generating Shielded Extent Layer

The device::blocked_extent function is introduced for use in compile-time TVF rules. This function is useful for creating layers that are normally generated using the Extent operation but it is desired to shield certain primitive device cells from that Extent derived layer.

Layers like PSUB are frequently defined like this:

```
BULK = EXTENT
PSUB = BULK NOT NWELL
```

In certain parasitic extraction flows, this can create a problem for parameterized device cells that implement devices lacking a PSUB terminal. Since the Extent layer is defined to exist everywhere, it has no way for the user to exclude it from certain cells.

device::blocked_extent provides an extent layer that blocks out specified pcells. This function should only be used in parasitic extraction flows that use pre-characterized parasitic models for cells that require the substrate layer not be present.

For example, to derive a layer like PSUB so it is not present in the cell Dev2T, change the PSUB definition from this:

```
tvf::setlayer PSUB = BULK NOT NWELL
```

to this:

```
package require CalibreDFM_DEVICE

tvf::setlayer EXTENT_BLOCKED = [device::blocked_extent \
              -block_cells { Dev2T } ]
tvf::setlayer PSUB = EXTENT_BLOCKED NOT NWELL
```

Note

Only small parameterized cells should be specified. Using this function for larger cells or blocks results in severe performance degradation.

Use of a layer created by this operation requires a Calibre nmLVS Advanced license. Further details are in the *SVRF Manual*. ER 1428627

Automatic Cell Renaming in the Event of Conflicts With Device Names

Formerly, circuit extraction would create incorrect SPICE netlists when a Device statement for a non-standard SPICE device (a non-standard model or non-standard number of pins) had the same name as a design cell. This resulted in duplicate subcircuit names.

Now, when this condition exists, the cell name is changed in the extracted subcircuit definition to a unique name starting with a sequence of underscore (_) characters. This warning is then issued to the circuit extraction report:

> WARNING: LAYOUT CELL <name> conflicts with a DEVICE with the same name and was renamed to __<name> in the layout netlist.

This change occurred in the 2020.3 release. DR 1025242

Known Problems and Workarounds

DR1413668: When tracing device properties together with LVS Compare Case YES, TYPES, SUBTYPES, or VALUES, then Trace Property validation of those values may not occur correctly without Layout Case YES and Source Case YES also being specified.

Solution: Set Layout Case YES and Source Case YES when case-sensitive comparison is used with device property tracing. ER 1416222

2020.3

-genhcells Option Enhancements

The -genhcells command line option is used to automatically generate an hcell list and has been enhanced in this release.

In default mode (only -genhcells is specified), hcell list generation times have been reduced by up to 32×. The corresponding total runtime for circuit extraction and comparison has been reduced by up to 1.4×.

Note

If you do not have a good hcell selection methodology, then you should consider using the -genhcells command line option. If you get a CORRECT comparison using that option, then use the hcell list that this option generates in subsequent runs by specifying the list with the -hcell option, and omit -genhcells.

The LVS Auto Expand Hcells PRESET, LVS Hcell Report WITH EXPANDED 1, and the new statistics discussed under "Enhanced Memory Reporting" are other features to consider when dealing with hcells.

The option now supports a "select" keyword, and the revised syntax is this: -genhcells[=*qs_tcl_file* | select]. See -genhcells in the *Calibre Verification User's Manual* for a complete discussion.

The -genhcells option does not use the Query Server hcells::select command in the default hcell selection script. Hence, the hcells::select evaluation thresholds are not employed. This can now be changed by specifying the new "select" keyword, which mimics the hcells::select default behavior. (Using "select" is preferred over using the =*qs_tcl_file* suffix.)

In addition to using the default Query Server evaluation thresholds, the select keyword generates an evaluation report like hcells::select. The report is written to one of the following files:

<svdb>/<topcell>.hcells.selectreport

./<lvs_report>.<topcell>.hcells.selectreport

depending on the presence of the Mask SVDB Directory statement. The <svdb> is a Mask SVDB Directory, the <topcell> is the layout primary cell, and the <lvs_report> is the LVS Report filename.

When -genhcells is specified, the following entries appear in the run transcript:

```
Generating HCELLS ...
Genhcells source read completed.
Genhcells source and layout names matching completed.
Genhcells files completed.
Genhcells completed.
```

The hcell list pathnames generated by the -genhcells option are consistent with previous releases. The generated list contains cells specified in -hcell lists, Hcell specification statements, and those selected by the -genhcells option itself, but have not been removed by various internal heuristics that eliminate unuseful cells. The hcells in the generated list are exactly those that appear in the associated LVS Report. ER 1249188

LVS Expand On Error

The LVS Expand On Error specification statement is new. It controls automated removal of certain hcells that may cause false errors during LVS comparison. The default setting is NO. This statement may be specified at most once.

When YES is specified, problematic hcells may be detected after circuit comparison and automatically removed from the internal hcell list. An additional LVS comparison run is triggered, and the reduced hcell list is used. This can occur iteratively. Cross-reference information is only saved from the final comparison run.

The Mask SVDB Directory statement is required when you specify YES. Supplemental LVS reports from each comparison run are stored in the SVDB directory with a .pass_*n* extension. The *n* is an index number starting from one. These reports contain details about the iterative rejection of hcells. See the *SVRF Manual* for more details. DR/ER 1394176, 1414128, 1417346, 1418887, 1421400

Calibre nmLVS Reconnaissance Short Isolation

Calibre nmLVS Reconnaissance Short Isolation (or *Calibre nmLVS Recon SI*) is a new feature that supports stand-alone layout short isolation. This short isolation is performed independently of an LVS run. Calibre nmLVS Recon SI runs have proven to be very fast, and using this flow early in the design verification process is an excellent way to avoid various LVS problems later on.

Using this feature requires a Calibre nmLVS Reconnaissance license, a hierarchical LVS license pair, and a Query Server license.

There are two primary flows: the rule file flow, and the Mask SVDB Directory flow. The rule file flow performs short isolation using LVS circuit extraction rules and a layout as inputs. The Mask SVDB Directory flow loads an SVDB containing a PHDB with connectivity information and performs short isolation on that data. Both flows depend on a Mask SVDB Directory statement in the rules that produces a persistent hierarchical database (PHDB). Some of the common keyword options that do this include CCI, QUERY, or SI.

In either of the aforementioned flows, only connectivity extraction and short isolation are performed (no ERC, no LVS Softchk, and no device extraction or anything dependent upon it, although connectivity statements involving Device pin layers are processed by default). Both of these flows support interactive short isolation immediately upon loading the generated SVDB data into either Calibre RVE or the Calibre Query Server Tcl shell. This interactive capability is provided automatically and requires no additional rule file setup. For more information about interactive short isolation in Calibre RVE, see "Interactive Short Isolation" in the *Calibre RVE User's Manual*. For more information about this topic in the Query Server context, see "Using Short Database Commands" in the *Calibre Query Server Manual*.

The core command line for this flow is this:

```
calibre -recon -si[={ALL | IO | PG | <tcl_script>}] \
  {<rule_file> | {-svdb <svdb_dir> [<top_cell>]}}
```

The -recon and -si options are both required. If *rule_file* is specified, then the rules flow is used (again, these are LVS circuit extraction rules). If -svdb is specified, then the Mask SVDB Directory flow is used. The multithreading options used with the -spice option are also supported, but are not shown here for simplicity.

If -si is specified without the equals sign suffix, then all nets at the primary level of the design are processed for short isolation. If the equals sign is appended to -si, then an additional modal argument must be specified that controls which nets are processed for short isolation. These are either nets at all levels of the hierarchy, I/O signal nets at the primary level, or power and ground supply nets at the primary level. If the *tcl_script* is specified, this is a Query Server Tcl shell script that runs commands from that interface on the PHDB data.

An LVS Isolate Shorts NO statement, specified either explicitly or by default, is ignored in this flow. LVS Isolation Shorts YES secondary keywords are observed if specified in the rules and are applied to the run if they do not conflict with the -si option's modal argument behavior, when specified.

There is also a new LVS SI Select Connects specification statement that supports this feature. This statement may be used to configure which Connect and Sconnect statements are processed prior to short isolation. By default, all such statements that produce connectivity to the level of the device pins are processed. Reducing the number of Connect statements that are processed improves runtime at the potential cost of finding fewer shorts. There is also a new Query Server Tcl shell analogue to LVS SI Select Connects called short_db::isolate_shorts.

For the complete command line usage see the "Calibre nmLVS and Calibre nmLVS-H Command Line" chapter of the *Calibre Verification User's Manual*. The "-recon -si" section and the associated examples are of greatest interest. See also "Short Isolation" in the same manual and "Running Calibre nmLVS Recon SI" in the *Calibre Solutions for Physical Verification* manual. ER 1363685

LVS Short Equivalent Nodes MATRIX Keyword

The LVS Short Equivalent Nodes specification statement has a new MATRIX keyword.

Normally, the SPLIT and PARALLEL options cause LVS to short equivalent nets internally before comparison of the layout and source. This can lead, in certain circumstances, to instance matches that are counter-intuitive and net cross-reference information that makes it difficult to map source nets to layout nets in downstream simulation flows. The MATRIX keyword can, in certain circumstances, give more accurate cross-reference information where the source and layout transistor structures match one another identically. See the *SVRF Manual* for details. DR 1245540

Enhanced Memory Reporting During Cell Flattening

LVS comparison now prints additional instance count information during cell flattening. This helps troubleshoot the root causes of certain performance issues during the transformation phase of comparison.

When cells contain large numbers of flattened instances, memory allocations that occur during the stages “FLATTENING non-corresponding cells” and “EXPANDING unbalanced cells” are reported in the run transcript. (Cells that do not serve as hcells are flattened into any parent hcells. Hcells that are instantiated a different number of times in the layout versus the source are expanded into containing cells.) Performance may be improved by identifying hcells with very large instance counts and finding contained cells that may function as additional hcells. Specifying contained cells identified in this way as hcells helps reduce overall per-cell contained instance counts and can result in improved performance.

To facilitate this identification, the additional memory information provided includes the following:

- Layout or source design.
- Cell name.
- Instance count.
- Memory in use (LVHEAP and MALLOC).
- Elapsed time.

Example output:

```
Resolving DEEP SHORTS ...
DEEP SHORTS resolved.  CPU TIME = 2  REAL TIME = 2  LVHEAP = 1692/1737/1737
MALLOC = 1794/1794/1794  ELAPSED TIME = 32

EXPANDING unbalanced cells ...

Source cell Z1374F now contains 1000071 flattened instances.
LVHEAP = 1880/1882/1882  MALLOC = 1932/1932/1932  ELAPSED TIME = 32

Layout cell Z1374G now contains 1000036 flattened instances.  LVHEAP =
2345/2345/2345  MALLOC = 2486/2486/2486  ELAPSED TIME = 34

   EXPANDING unbalanced cells in LAYOUT cell Z1374G, SOURCE cell Z1374F
      C1802 in Z1374F ( 28 instances, SOURCE )
```

For the preceding report, the cells Z1374G and Z1374F are candidates to search for corresponding child cells to be declared as hcells.

The report threshold for flattened instances is 1E06 and progresses at subsequent intervals of 5E06 instances. DR 1414759

LVS Auto Expand Hcells Enhancement

The LVS Auto Expand Hcells statement has been enhanced to allow for expansion of certain additional high-cost hcells. Those high-cost hcells are listed in the LVS transcript with the text: EXPANDING DENSE OVERLAPS - AREA. For example:

```
RDL (EXPANDING DENSE OVERLAPS - AREA/155: 99.0) - HCELL REJECTED
```

The EXPANDING DENSE OVERLAPS category is not new, but this update does increase the possibility that some hcells will be rejected when a keyword other than NONE is specified. ER 1414759

Calibre Query Server

This section includes changes and enhancements to Calibre® Query Server.

For additional information, refer to the *Calibre Query Server Manual*.

Calibre Query Server Updates

Calibre Query Server has the following new features in the 2021.2 release.

qs::softchks Update

This function is part of Calibre nmLVS Reconnaissance Softchk. See "New qs::softchks Options" on page 82.

Calibre Query Server Previous Release Highlights

Calibre Query Server introduced the following functionality in previous releases.

2021.1

New Soft Connection Checking Commands

The Query Server Tcl shell now has two soft connection checking commands: qs::softchk and qs::softchks. These commands permit LVS Softchk verification to be performed separately from an LVS circuit extraction run, so the commands can effectively decouple soft connection checking from other ERC and LVS tasks. These commands are only executed in a calibre -recon run that uses the -softchk option, which is also new in this release and documented in the LVS section.

The qs::softchk command works essentially the same way as LVS Softchk and has similar configuration options.

The qs::softchks command produces output in the same way LVS Softchk does by default. It performs soft connection checks on PHDB layers for which there was an Sconnect stamping conflict during the circuit extraction run.

Both of these commands produce a *softchk.rdb* results database. This database is of the same format as the **.softchk* results database produced by LVS Softchk. ER1414756

2020.4

short_db::isolate_shorts Performance Improvement

The run time of the short_db::isolate_shorts command has been improved by up to 30% in some cases. This improvement is not seen in “calibre -recon -si=*tcl_script*” runs where short_db::isolate_shorts is executed in the script and MTflex distribution is used. ER 1426406, DR 1427657, and DR 1427384

“ERROR: Short isolation failed - no short found.” Message No Longer Given

Interactive short isolation no longer gives the following error if no short paths are found due to changes made to the database in attempts to fix shorts:

```
ERROR:  Short isolation failed - no short found.
```

DR 1426326

short_db::set_distribution_configuration Command

The short_db::set_distribution_configuration command is new and controls MTflex remote machine configuration for stand-alone short isolation runs that call a Query Server script (calibre -recon -si=*tcl_script*). This command only affects the short isolation portion of the run performed by the Query Server.

For this discussion, a “process” is an instance of the short isolation module. Multiple processes can run concurrently on a host, and each process can support multiple threads simultaneously.

By default, the minimum number of CPUs a remote host may have to run short isolation is 16. This value may be user-specified and also is used to define the process count for a remote. The number of processes and threads per process may be configured by a memory allocation value instead.

By default, the minimum number of shorts to be isolated on a remote host is two (a single short is run on the primary host). This also can be adjusted by this command.

If a remote host specified through the MTflex command line options does not meet the short_db::set_distribution_configuration criteria, then the host is skipped with a corresponding message written to the transcript. If no such remote host meets the criteria, then short isolation is run on the primary host. ER 1413334

OASIS AGF Output Uses CBLOCK Compression

The OASIS AGF output in CCI now uses CBLOCK compression. This can produce substantially smaller file sizes. ER 1421539

2020.3

New Placement Path Parsing Commands

The qs::parse_path command is new in the Query Server Tcl shell. The corresponding standard command is PARSE PATH. Both of these commands decompose a placement path into its constituent cell names and port counts. Device and net instance paths each have outputs specific to those design objects. Here are two examples:

```
> qs::parse_path -source X0/X44/XABC/M0
{{X0 route66 44} {X44 blockA 128} {XABC abc_cell 4} {M0 MN(nfet) 4}}

> qs::parse_path -source X0/X44/XABC/Net[0]
{{X0 route66 44} {X44 blockA 128} {XABC abc_cell 4}} {Net[0]}
```

Each cell instance name is returned in a sub-list together with the corresponding cell name and port count. A primitive device instance is returned with its *type*(*subtype*) designator and a pin count. A net name is returned in a separate list, as is any unresolved portion of a path.

The PARSE PATH command returns its output in the customary response format of the standard Query Server rather than as Tcl lists. Details are in the *Calibre Query Server Manual*. ER 1415635

New short_db::select_connects Command

The short_db::select_connects command is new in the Query Server Tcl shell. This command is used in support of Calibre nmLVS Reconnaissance Short Isolation to select the Connect and Sconnect operations that are executed.

This command is only used in a script referenced by the -recon -si=*tcl_script* command line option. See the Calibre nmLVS release notes for more details about Calibre nmLVS Recon SI.

Calibre Pattern Matching

This section includes changes and enhancements to Calibre® Pattern Matching for the 2021.2 release.

For more information, please refer to the *Calibre Pattern Matching User's Guide*, *Calibre Pattern Matching Reference Manual*, and *Standard Verification Rule Format (SVRF) Manual*.

Calibre Pattern Matching Updates

The following updates were made to Calibre Pattern Matching in the 2021.2 release.

- Change in DMACRO Encryption and Version Dependency
- New and Updated SVRF Operations
- Utility Updates
- Obsoletion and Deprecation Notices

Change in DMACRO Encryption and Version Dependency

Calibre 2021.2 uses a new version of encryption. Due to the new encryption version, DMACRO files created with 2021.2 or later cannot be used by earlier versions of Calibre. DMACRO files created with earlier versions of Calibre can still be used with Calibre 2021.2 and later.

New and Updated SVRF Operations

- **New Operation for Symmetry Analysis**

  The new DFM Pattern Analyze operation analyzes the symmetry of target layer shapes at marked locations. The analysis region is determined by mask or marker shapes. You can analyze groups of shapes for vertical and horizontal symmetry and for 90 and 180 degree rotational symmetry. The operation can output target layer shapes that meet the symmetry requirement, or the difference between the target layers shapes and the same shapes with the symmetry transform applied.

- **New Polygon Classification**

  The new BY_POLYGON keyword in DFM Pattern Classify enables polygon classification by shape, including classification of shapes with angled edges. This classification mode is similar to pattern classification, but classifies individual polygons on the target layer, rather than target layer shapes within the capture region. The SEED or LAYER_MASK keyword is not used to define the capture area and a pattern library is not created.

By default, the output layer is a copy of the input layer, with attached properties. The operation considers orientation by default, meaning that identical shapes with a different orientation have the same class ID. You can specify what orientations are considered duplicates with an optional keyword.

- **Per-Layer Halo From Extent or Edge**

  For both DFM Pattern Capture and DFM Pattern Classify, the FROM_EXTENTS and FROM_EDGES keywords can now be used to specify behavior for a per-layer halo. For example, the pattern halo can be measured from the center of the hotspot, while a per-layer halo for one target layer can be measured from the edges of the hotspot shape. See the *SVRF Manual* for syntax and details.

- **More Flexible Marker Creation in DFM Pattern Classify**

  The LAYER_PATTERN_MARKER keyword is now available in DFM Pattern Classify; previously it was only available in DFM Pattern Capture. You can add multiple markers of various types to the saved pattern library. You can add markers from shapes on a polygon layer, BBOX, BBOX, drawn layer, and matched layer markers.

Utility Updates

The compare utility now reports pattern orientation as an integer from 0 to 7, consistent with the convention for other Calibre Pattern Matching tools. Previously, the orientation was reported as a string; for example, S0 and S90.

In addition, the compare utility was improved for the handling of patterns that have different drawn orientations but are identical to other patterns when allowed transforms are considered.

Obsoletion and Deprecation Notices

The classic Pattern Library Manager GUI is obsolete starting with the 2020.3 release, and it can no longer be invoked. The classic Pattern Library Manager GUI was deprecated starting with the 2019.4 release.

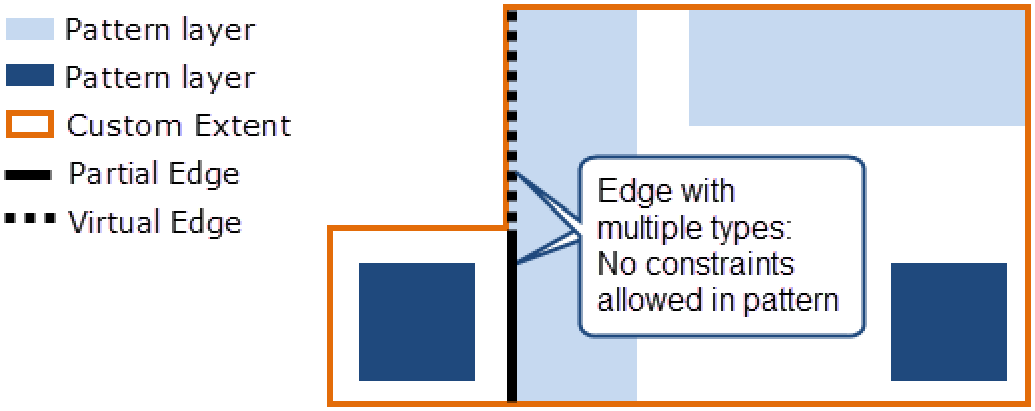
The SVRF command PM_PATTERN_CAPTURE is obsolete with the 2021.1 release. PM_PATTERN_CAPTURE is replaced by DFM Pattern Capture, which has better performance and hierarchical behavior.

Calibre Pattern Matching Previous Highlights

Important changes from the previous release are summarized here.

(2021.1) General Updates

New support for pattern edges with mixed types — In some cases a pattern edge may have more than one type along its length. In particular, this can happen when the pattern extent is not rectangular, as shown next figure. If an edge with multiple types exists in a pattern, the pattern cannot have constraints. Previously, patterns that included edges with mixed types were invalid.



These are two possible scenarios that can result in pattern edges with mixed types:

- Capturing patterns using a hotspot shape that is not rectangular and using the FROM_EDGES keyword in DFM Pattern Capture or DFM Pattern Classify.
- Capturing patterns using mask shape that is not rectangular.

See “Edges in Patterns” in the user’s manual.

(2021.1) Changes to SVRF Commands

DFM Pattern Capture — New keyword to add a pattern attribute:

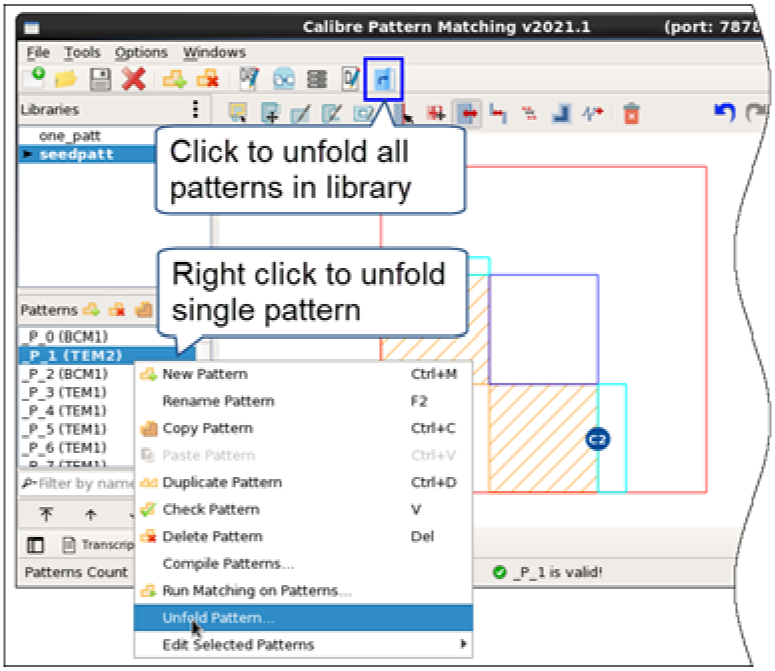
PATTERN_ATTR *attr_name attr_value* [INTEGER | FLOAT | STRING]

The optional keyword PATTERN_ATTR adds a user-defined pattern attribute to each pattern in the saved pattern library. You can specify multiple argument sets. OUTFILE must be specified when using PATTERN_ATTR.

DFM Pattern Capture and DFM Pattern Classify — The minimum width and length is of a BBOX10 output marker is now 1 dbu. This prevents problems when the capture region is very small. (DR 1423799)

(2021.1) GUI Updates

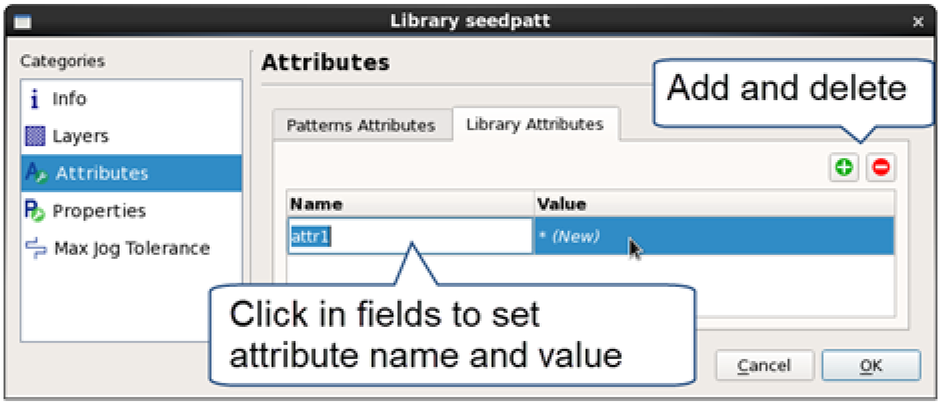
- **Unfold from GUI** — You can now run the unfold utility from the GUI. You can unfold the complete library or a single pattern:



Click to unfold all patterns in library

Right click to unfold single pattern

- **Add and display user-defined library attributes** — User-defined library attributes have one value that is set at the library level, in contrast to pattern attributes that have a value defined for each pattern. To manage library attributes, choose **File > Library Attributes** to open the Library dialog box, then click the new **Library Attributes** tab:



Add and delete

Click in fields to set attribute name and value

(2021.1) Utility Updates

The write_oasis utility has these updates:

- Syntax change for the sort_by argument

The sort_by argument no longer uses the literal string “name” in the argument syntax. The new syntax is this:

> sort_by *prop_or_attr_name1* {descending | ascending}
> [secondary_sorter *prop_or_attr_name2* {descending | ascending}]

If you have scripts that call write_oasis with the sort_by argument, they must be updated.

- Change in default precision

  The default precision is now the lowest common multiple of precision values in your library; this value prevents truncation of any geometries. If oasis_per_pattern is specified, each output file uses the pattern precision by default.

  Previously, if the precision argument was not specified, the precision of the first pattern in the library was used.

- Cannot specify both the output and oasis_per_pattern arguments

  The output argument causes patterns to be written to a single OASIS file. The oasis_per_pattern argument outputs one OASIS file per pattern. You can only specify one of the arguments now.

Calibre PERC

This section includes changes and enhancements to Calibre® PERC™.

For more information about Calibre PERC and Calibre PERC LDL, refer to the *Calibre PERC User's Manual*. For information about Calibre RVE for PERC, see "Calibre RVE" on page 63.

Calibre nmLVS and Calibre nmLVS-H release notes are also of interest for Calibre PERC users.

Calibre PERC Updates

Calibre PERC introduced the following new features with the 2021.2 release.

- Support for Device Filtering Using Initialized Properties
- Distributed LDL CD Test Pools
- Error Given for Unavailable Reducible Properties
- perc_ldl::custom_r0 Results Enhancement

Support for Device Filtering Using Initialized Properties

LVS Filter now supports filtering of devices based upon properties generated with LVS Property Initialize. This can be useful when filtering devices based upon the value of some mathematical expression.

For example, assume you want to filter certain MOS devices based upon the ratio of width to length. You could now generate a property called RWL with a value of W/L using LVS Property Initialize. You can then filter devices having the RWL property using LVS Filter with the *property* argument. ER 1442092

Distributed LDL CD Test Pools

Calibre PERC LDL CD simulates source-sink pin pairs as "tests." In 2021.1 a new feature was added. By default, tests for power supply nets are split into pools of 25 for MTflex distributed processing. This default generally produces good performance. However, the default pool size might require adjustment for optimized performance. To do this, you can set the following environment variable:

```
CALIBRE_PERC_CD_CTRL_FILE_TEST_SPLIT=<n>; # n is the number of tests per
MTflex pool
```

Make any such adjustment in consultation with a Calibre support engineer.

Starting in this release, if you want the aforementioned pool splitting to also apply to signal nets, then set this environment variable:

```
CALIBRE_PERC_LDL_CD_MTFLEX_ENABLE_SIGNAL_SPLIT=1
```

ER 1437277

Error Given for Unavailable Reducible Properties

If the PERC Load XFORM keyword is specified and a property specified by PERC Property is unavailable on a related device instance, this now results in a runtime error:

```
ERROR: Properties missing on instances in design
```

The status of the run is ABORTED. Cells containing such instances are listed in the report with an ABORTED status, regardless of whether they are expanded. DR 1441704

perc_ldl::custom_r0 Results Enhancement

Calibre RVE for PERC results presentation for perc_ldl::custom_r0 output has been enhanced to include P2P flylines for R0 (source to victim) and R1 (victim to sink) layers. ER 1437916

Calibre PERC Previous Release Highlights

Calibre PERC introduced the following functionality in previous releases.

2021.1

UPF Support Enhancement

Recall that UPF power state tables are accessible when PERC UPF Path is specified in the rules. The set_domain_supply_net UPF command is now supported in a PERC UPF Path file. This command defines supply domain names and their corresponding primary supply net names.

The supply domain names defined by set_domain_supply_net commands are accessed in the rule file by the new perc::get_upf_data command. Further details are in the *Calibre PERC User's Manual*. ER 1433266

LDL P2P Results Presentation Now Single Edge By Default

The perc_ldl::execute_p2p_checks -single_edge option is now the default behavior. This option provides results that are easier to understand and can reduce the size of the results database.

For example, if a source short group has four devices and a sink short group has five devices, previously, 20 flyline edges were generated to identify all the connections between the devices. Now, only one representative edge is produced by default. To revert to the historical behavior, an -all_edges option has been added. ER 1429797

Change in Cell Port Handling for LVS Box Cells in LDL Flows

In previous releases, if cell port P was a member of an exported pin pair and P’s cell was declared as an LVS Box cell, then the pin pair export would fail because LVS Box cells are presented as primitive devices in the LVS netlist graph, not cells. Hence, P was presented as a primitive device pin, not a cell port.

In this release, P is maintained as a cell port regardless of the LVS Box status, and the pin pair export succeeds. This change makes it easier to deal with certain IP cells, or cells for which it is advantageous to handle as primitive devices, by “black boxing” them. ER 1432204

LDL CD and P2P Waiver Tolerances

LDL waivers now support tolerance values, which allow for minor changes to the parasitic resistances between sources and sinks without affecting the waiver status. Previously, there were no tolerances, so a minor change in parasitic resistance caused a CD or P2P waiver not to be applied. Now, by default, waivers are annotated with a criteria setting that defines a 10% tolerance when applying a waiver.

For CD waivers, this means that a waived current density value J +/- 10% of J is still waived. This is not configurable.

For P2P waivers, this means that a waived resistance value R +/- 10% of R is still waived. The layout path between a source and a sink may be modified (but not the source and sink locations) and a waiver could still apply if the extracted resistance is within the tolerance. P2P waiver tolerances are configurable.

If you want to change the P2P tolerance, you can add lines like this to a waiver criteria file:

```
P2P:experiment_name P2P [tolerance | {tolerance_above tolerance_below}]
```

The tolerances are floating-point values between 0 and 100, and are interpreted as a percentage. The *experiment_name* is a P2P experiment defined by perc_ldl::design_p2p_experiment. The “*” wildcard is supported in a name and matches zero or more characters. The default is functionally equivalent to this:

```
P2P:experiment_name P2P 10
```

The following entry waives all resistances between 0 and 110% of the waived value for experiment rule_1:

```
P2P:rule_1 P2P 10 100
```

The *tolerance_below* value of 100 means all values below the waived resistance are waived.

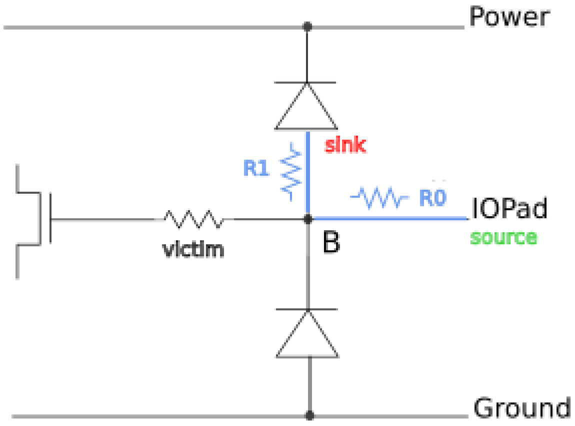
You can then load the waiver criteria file in the **Export Waived Results** dialog box’s **Criteria** > **Input waiver criteria file(s)** field before exporting waivers, and the annotations in your configuration will be available for extraction during a verification run.

See “LDL Geometry Waiver Application” in the *Calibre PERC User’s Manual* for related information. ER 1425764

New perc_ldl::custom_r0 Command

The perc_ldl::custom_r0 command is new in this release. It is used for designs with ESD structures for which it is important to determine something called a *common resistance*.

In the following figure, IOPad could be a source and the pos pin of the upper diode a sink for a P2P experiment. The third pin on the net is the resistor pos pin. A branch location is labeled B, which represents the junction between the source, sink, and a third pin on the simulated net.



Note

It is important to understand B is typically not a point location in the layout. It is often a structure that covers a region.

By default, the resistance between the IOPad and the branch location B is not calculated separately. When perc_ldl::custom_r0 is specified, then the resistance of that segment of the net, called a common resistance, is calculated separately and reported as R0. The resistance between B and the diode pos pin is calculated as R1. The sum R0 + R1 gives the total resistance R between the source and sink. Having this information can assist in understanding whether resistances are within design limits for the simulated net.

Further details are in the *Calibre PERC User’s Manual*.

2020.4

perc::terminate_run -immediately Option

The perc::terminate_run command now has a -immediately option, which serves as a “hard stop” for Calibre PERC. When the command is executed with this option, it aborts the run with an error message. No report file is produced, and any generated Mask SVDB Directory may be incomplete or unusable. This command may be used with -immediately in an initialization, rule check, or -condition proc. Otherwise, the command may only be used in an initialization proc.

perc::get_surviving_net Command

The perc::get_surviving_net command is new. This command determines the status of top-level design nets after netlist transformation.

Input SPICE netlists undergo transformations in which original design nets can survive intact, be merged with another net and reduced, or removed altogether. This command facilitates determination of which of these cases holds for a given top-level net. This information can then be used to determine which nets to assign net types or voltages to.

LDL Results Waivers

Automatic waivers of logic-driven layout (LDL) results are now supported for CD and P2P checks. The flow for these waivers is essentially the same as for DRC-style waivers. The two primary differences versus DRC-style flows are the waivers are generated from the DFM database rather than from an ASCII RDB file (waiving directly from the latter does not apply CD or P2P waivers), and the verification run waiver setup file is specified by the PERC LDL Waiver Path statement in the rules rather than on the command line. This feature requires a Calibre Auto-Waivers license. Further details are discussed under “LDL Geometry Waiver Application” in the *Calibre PERC User’s Manual*. ER 1426573

dfm::get_ldl_data -connection_properties Option

The dfm::get_ldl_data command now has a -connection_properties option that returns a list of lists with data related to a full path connection. This includes the devices and pins involved in the connection along with current and resistance through the connection. Here is an example for a connection exported across a resistor:

```
{
    {{r[r] R2 pos} D}
    {{r[r] R2 neg} R}
    {I 0.50606} {R 94.589}
}
```

ER 1398312

PEX Resistance Open Layer Statement

The PEX Resistance Open Layer specification statement is new. It omits specified layers from parasitic extraction, treating them as if they are not present. Any nets or devices extracted from this layer do not appear in the parasitic network. This can be useful in LDL flows where exclusion of substrate layers is desired. ER 1335805

Message Given for High Resistance Layers

A message is now given in LDL and Calibre xRC applications when resistances of 1E06 ohms or greater are detected during parasitic extraction:

WARNING: Detected High Resistance Value on Cell: "*cell_name*", Layer: "*layer_name*"

This has been added because such high resistances generally are not helpful in results. If you see this warning, check the rules involving the *layer_name*.

MTflex Environment Variables No Longer Required for LDL

In previous releases, LDL applications required the CALIBRE_ENABLE_PERC_SIMULATION_REMOTEFILE_MTFLEX and CALIBRE_ENABLE_PERC_XRC_MTFLEX variables to be set in order for parasitic extraction and resistance simulation to run in MTflex mode. These variables are no longer required. ER 1427063, 1427064

perc::trace_path -minDistanceOnly Option

The perc::trace_path command now supports a -minDistanceOnly option. When this option is used, the number of devices in a minimum-distance path between a source and sink net is returned rather than a Tcl list of instance names. ER 1424227

PERC Load Relaxed Handling of Parentheses in SELECT Syntax

In this release, the handling of parentheses enclosing rule checks in a PERC Load SELECT statement has been relaxed.

Recall that PERC Load can specify rule checks in one of four basic ways indicated here:

```
// mutually exclusive statements
// single-threaded
PERC LOAD tvf_func INIT init SELECT rule1 rule2 rule3 rule4

// multithreaded
PERC LOAD tvf_func INIT init SELECT PARALLEL rule1 rule2 rule3 rule4

// multithreaded with rule2 and rule3 in the same thread
PERC LOAD tvf_func INIT init SELECT PARALLEL rule1 rule4 (rule2 rule3)

// single-threaded then multithreaded, rule2 and rule3 on the same thread
PERC LOAD tvf_func INIT init SELECT rule1 PARALLEL rule4 (rule2 rule3)
```

Previously, the use of parentheses to indicate a sequential parallel check set was limited to when PARALLEL was also specified. Parentheses are now ignored in the context of SELECT without PARALLEL. This can ease structuring the rule file for conditional compilation. This is illustrated by the following code:

```
#IFDEF SELECT_PARALLEL
  PERC LOAD tvf_func INIT init SELECT PARALLEL
#ELSE
  PERC LOAD tvf_func INIT init SELECT
#ENDIF
  rule1 rule4

#IFDEF RUN_ADDITIONAL
  (rule5 rule6)
#ENDIF
  (rule2 rule3)
```

In previous releases, if SELECT_PARALLEL was undefined, (rule2 rule3) would cause a compiler error. Similarly, (rule5 rule6) were problematic when SELECT_PARALLEL was undefined but RUN_ADDITIONAL was defined.

Now, the parentheses do not cause a problem in single-threaded mode. ER 1427815

2020.3

Insertion of Specification Statements in LDL CD and P2P Runs

The perc_ldl::execute_cd_checks and perc_ldl::execute_p2p_checks commands now support a -rules_preamble option that inserts an SVRF specification statement into the rule set. The specification statement applies during the CALIBRE::PERC - EXECUTIVE module only (not during parasitic extraction or resistance simulation). If the statement conflicts with some other statement in the rule file, then the -rules_preamble setting has priority. Incorrect syntax in the specification statement argument causes a rule file compiler error in the LDL portion of the run. This option may be used more than once. Here is an example:

```
perc_ldl::execute_p2p_checks -p2p_experiment_list [list $exp_list] \
  -rules_preamble "INCLUDE supplemental.ldl.rules"
```

ER 1418660

New Rcontrib(%) Statistic for -p2p_debug Run

The perc_ldl::restart command supports a -p2p_debug option that provides additional diagnostic data about P2P results from a previous run. Starting in this release, a new Rcontrib(%) statistic appears in the RDB results of the -p2p_debug run. This statistic is the ratio of the a polygon’s resistance to that of the effective resistance of the entire containing path from source to sink, expressed as a percentage.

The sum of all Rcontrib(%) values for a given path may not be exactly 100%. The value is intended as a relative measure indicating which polygons represent more significant resistance problems relative to others in the same path.

Electromigration Options for dfm::get_ldl_data

The dfm::get_ldl_data command has five new options related to electromigration and current direction through vias.

The -via_direction returns the current direction through a via. Return values can be UP, DOWN, or N/A.

The -em_length_up, -em_length_down, -em_width_up, and -em_width_down options return the EM values consistent with their names. If the via direction is not consistent with any of their names, then the options with the inconsistency return 0.

Calibre 3DSTACK

This section includes changes and enhancements to Calibre® 3DSTACK for the 2021.2 release.

For additional information, refer to the *Calibre 3DSTACK User's Manual*.

Calibre 3DSTACK Updates

The Calibre 3DSTACK tool introduced the following features in the 2021.2 release.

- Enhanced Support for Overlap Checking with Non-Manhattan Geometries
- LEF/DEF Text Enhancement
- Unconnected Ports

Enhanced Support for Overlap Checking with Non-Manhattan Geometries

If an interface layer in your design contains multiple non-manhattan geometries that overlap, there may be slight differences in how the GDS specification and the internal Calibre 3DSTACK engine model the polygons. In some cases, if you are checking for a strict 100% overlap constraint, it may be necessary to merge the overlapping polygons. The tool has been enhanced with a new -merge keyword for the overlap command. Apply the option as follows:

```
overlap -check_name pad_overlap \
   -layer_type1 die_bmp \
   -layer_type2 int_pad \
   -constraint "<100"
   -merge \
   -comment "Pad overlap is less than 100%"
```

Note
Performance is affected when this keyword is applied.

LEF/DEF Text Enhancement

The die -layer_info argument set has been enhanced with a new -texted option for LEF/DEF layouts. When specified, the tool explicitly attaches text objects from that layer to the specified corresponding Calibre 3DSTACK layer. If you do not specify this option, then the text is only attached to layers for which the -text option is applied.

Unconnected Ports

The unconnected_ports command was introduced in 2021.1 but not documented. Use this command to report ports that are both not connected to any other port in the layout and in the source netlist. A source netlist is required. The usage is as follows:

```
unconnected_ports -check_name check_name {-dies die_name_list}
   | {-layer_types layer_types_list}
   [-comment "comment"] [rve_option …]
```

Calibre 3DSTACK Previous Release Highlights

Calibre 3DSTACK introduced the following functionality in previous releases.

- 2021.1 Updates
- 2020.4 Updates
- 2020.3 Updates
- 2020.2 Updates

2021.1 Updates

- **Enhanced Connectivity Options —** As of this release, you can now filter connected results for standalone die checks by specifying the -short_only and -open_only options. These options are mutually exclusive and output only shorts or opens for the connected command, respectively. The usage is as follows:

  ```
  connected -check_name name  \
     [-die1 die_name -standalone [-short_only | -open_only]]
  ```

  Additionally, the connected command has been enhanced to enable you to specify a list of pins to check on a layer type as follows:

  ```
  connected -check_name internal_open_short \
     -stack assembly \
     -layer_type1 bump \
     -pin_list {VDD VSS} \
     -comment "Opens or Shorts detected on dies for VSS/VDD"
  ```

  In this example, the tool only reports shorts or opens on the VDD and VSS pins.

  See “connected” in the *Calibre 3DSTACK User’s Manual* for details.

- **Enhanced Behavior for Warning Severity —** In previous releases, if you changed the behavior of the tool to exit on a warning with “-severity 2”, the tool exited as expected, but no additional messages were output other than the warning that was encountered. As of this release, the tool now generates a message that indicates that the tool exited due to a severity setting.

- **Change to Layer Specification Handling —** In previous releases, if you specified a layer with a decimal point, for example 100.00 as follows:

  ```
  -layer_info {
    -type bump
    -name layer_x
    -layer 100
    -text 100.00
    …
  }
  ```

  This layer could be duplicated and cause issues in the run.

  As of this release, the tool handles a layer with trailing zeros after a decimal point the same as the base number. For example, 100, 100.0, and 100.00 are all considered the same as 100.

- **Change to -svrf_specs Option —** The config -svrf_specs option no longer supports any SVRF statement other than Layout Input Exception Severity. Previous releases included warnings in the documentation to only use this statement to control warning severity. As of this release, the tool now mandates this requirement.

- **Improved Messaging —** As of this release, if a die name used in a rule check does not match the die name specified in the die command, then the tool issues an error message.

2020.4 Updates

- **New Option to Perform Centers Check With One Layer —** As of this release, in order to perform the centers check on a single layer type, you must include the new -same_die option. In other words, you must specify -same_die if you do not specify the -layer_type2 argument set. For example:

  ```
  centers -check_name controller_pads -layer_type1 pad -same_die \
     -constraint "== 80" -alignment orthogonal_only \
     -comment "Pads spaced 80 um in x or y from all pad centers"
  ```

- **Discontinuation of the Check Source Option —** The -cs command-line option and the “check source” checkbox in Calibre Interactive have been removed from the documentation and the Calibre Interactive GUI. The options continue to work in the 2020.4 release if you specify them in a runset file or batch script. The options will be no longer function starting with the 2021.1 release.

- **Enhanced Support for Cell Names —** The Calibre 3DSTACK tool now supports cell names in the assembly that contain the “$” character. In previous releases, cells that contained “$” could result in the following error:

  ```
  3DSTACK_ERROR_608: can't read "<cell>": no such variable.
  ```

2020.3 Updates

- **Verify That Interacting Pads are Exactly the Same Size —** This release includes a new same_size rule check that verifies whether interacting pads are the same size. When verifying a 3D-IC, it can be of interest to ensure that the landing pads of the connected dies are exactly the same size and not just sufficiently overlapping. This can be achieved using other rule check commands, but they are less generalized and require additional logic. The new same_size check is applied as follows:

  ```
  same_size -check_name die_int_s -layer_type1 i_pads -layer_type2
  d_pads \
     -comment "Pads are not sized the same!"
  ```

  See "same_size" in the *Calibre 3DSTACK User's Manual*.

- **Find Unconnected Port Shapes With No Text Labels —** This release includes a new dangling_no_text rule check that reports port shapes that are not connected (dangling) and do not contain text. The new dangling_no_text check is applied as follows:

  ```
  dangling_no_text -check_name orphan_ports
     -dies die_name_list
     -comment "comment"
  ```

- **Specify the Assembly Name —** The -assembly invocation argument is new for this release. Use this argument to specify a custom name for the generated assembly as follows:

  ```
  calibre -3dstack -assembly my_3dic … rules
  ```

  This option is mutually exclusive with the -use_assembly and -create_assembly arguments. If you do not specify this argument, the default assembly name is 3dstack_assembly.

- **Change to the Default Assembly Layout Format —** The default output format for generated layout files is now OASIS with CBLOCK compression enabled. This format reduces the assembly layout file size. The tool no longer generates GDS output, which includes the cross-section layout files.

- **Report All Unconnected Pins —** The Calibre 3DSTACK summary report now includes a list of all unconnected pins.

- **Change to Query Command File —** The LVS Settings Report Write statement in the *.query.cmd* file (generated by the export_layout command) now points to the *.lsrf* file instead of the *.lns* file

- **Support for Pin Mapping in Export Connectivity —** The export_connectivity command now reads specified pin mapping operations and applies them to the generated layout netlist. In previous releases, the pin mapping operations were ignored in export_connectivity operations.

- **Deprecation of Explicit Text Tracing —** The trace_text command has been deprecated. This command is no longer necessary and will generate a warning if it is applied.

2020.2 Updates

- **Enhanced Short Debugging in Connectivity Checks —** The connected rule check now includes an -isolate_path option that significantly improves debugging for shorts in your design. The new enhancement enables you to highlight paths between shorted pins or pads in your design that can help debug the cause of connectivity errors.

  When you enable this option, the shorts in connectivity results are filtered under a <*connect_check_name*>_isolated result to display only the ports that are physically involved in the short. This option has no effect on open circuit results.

  See "Using Path Isolation to Debug Shorts" and "connected" in the *Calibre 3DSTACK User's Manual* for details.

- **SVRF Mapping File Enhancements —** The *.map.svrf* file created by the export_layout command has been enhanced as follows:

  - Placement names are now used instead of the chip names.
  - The file now contains layer grouping.
  - Text and pin text layers are now grouped under their respective placements.
  - The white space formatting has been improved.
  - The file now only supports two dies in two tiers instead of multiple dies in two tiers.

- **Support for Different xCalibrate Versions —** You can now use a different xCalibrate version than your current Calibre version. If you set the XCALIBRATE_HOME environment variable to a custom path, Calibre 3DSTACK uses that version instead of the xCalibrate version specified by MGC_HOME.

- **Detailed Connectivity Information by Default —** The connected -detailed option was disabled by default in previous releases. As of this release, -detailed is always enabled. The option will be deprecated in the 2020.3 release, since it is no longer necessary.

- **Change to Extra and Missing Port Behavior for Mapped Pins —** In previous releases, you could still receive missing or extra port errors for original pins that were mapped to different pin names. Missing or extra port errors due to mapping are no longer reported.

- **Placement-Layer Connectivity Checks Have Been Deprecated —** You can no longer specify placement layers with -placement1 and -placement2 options in connected checks for the standard syntax; you can now only specify die layers. This change does not affect the extended 3DSTACK+ syntax.

- **Debug Tcl Errors —** You can now optionally create additional information about Tcl errors generated from your Calibre 3DSTACK rule files. Set the CALIBRE_ENABLE_3DSTACKPLUS_DECK_DUMP environment variable to a non-null value before running Calibre 3DSTACK to export a file named *<rule_file>_exported.3dstack+*. This file contains information that enables you to identify the source of Tcl errors in your rules.

Calibre Utilities

This section includes changes and enhancements to Calibre® utilities set for the 2021.2 release. Calibre utilities include V2LVS, E2LVS, DBdiff, FastXOR, fdiBA, fdi2gds, and fdi2oasis.

The V2LVS and E2LVS utilities are documented in the *Calibre Verification User's Manual*.

The DBdiff, FastXOR, and FDI utilities are documented in the *Calibre Layout Comparison and Translation Guide*.

Calibre Utilities Updates

This section describes updates to the Calibre utilities in the 2021.2 release.

- New DIEAREA Control for Output DEF in DFM RDB DEF

New DIEAREA Control for Output DEF in DFM RDB DEF

The DFM RDB DEF statement supports a new DIEAREA keyword that enables you to control how the DIEAREA is written to the generated DEF file. The keyword set is described as follows:

```
DFM RDB DEF … DIEAREA {ENH | DEF | {(x1 y1) (x2 y2) [( xn yn ) …]} | NONE }
```

The DIEAREA statement is written to the generated DEF based on one of the following arguments:

- ENH — The DIEAREA is calculated from the extents of the specified enhancement design (see the DEFMODE ENH keyword set).
- DEF — The DIEAREA is used from the specified input DEF. If no DEF file was specified, the tool issues an error message.
- (*x1 y1*) (*x2 y2*) [(*xn yn*) …] — The DIEAREA is specified manually using coordinate pairs. You must specify at least two pairs of coordinates to define a rectangle. If the design is rectilinear, additional coordinate pairs are supported.
- NONE — The DIEAREA is not written to the output DEF.

Calibre Utilities Previous Release Highlights

The following functionality was introduced to DBdiff, FastXOR, fdi2gds, fdi2oasis, and fdiBA in previous releases.

2021.1 Updates

- **Odd-Width Path Handling —** The fdi2gds and fdi2oasis utilities include a new -convertOddWidthPaths argument that enables special handling for paths with odd width dimensions. When you specify this option, all SPECIALNET paths that have odd widths are converted to on-grid polygons that are comprised of the original odd-width path geometry values. This option can prevent grid snapping issues. This argument takes precedence over the FDI_PRESERVE_ODD_WIDTH_PATHS environment variable.

  The tool outputs the following message when it encounters an odd-width path and you specify this option:

  ```
  WARNING_1227: (DEF_LINE): NET (NET): A path on layer (LAYER) at the
  point
  ((coord),(coord) ) has an odd width ((width)) and will be converted
  to
   polygon.
  ```

- **Specify Multiple LEF/DEF Object Types With Subtypes in a Single Mapping Statement —** The fdi2oasis and fd2gds LEF/DEF mapping file format now enables you to output multiple objects types with subtypes to a single layer and datatype on one line. In previous releases, you would need to use multiple lines to perform many-to-one mapping for objects with subtypes. For example, this statement maps LEFPIN and VIAs of a specific size to layer 14, datatype 2:

  ```
  via34 LEFPIN, VIA:SIZE:0.55x0.55 14 2
  ```

- **Mapping Enhancement for DEF Tracks —** The LEF/DEF mapping file now supports DEF track mapping. Use the TRACK object type with the WIDTH:*width* and DIRECTION:*direction* arguments to translate DEF objects of a particular size and direction.

  The WIDTH value is either “DEFAULT” or a floating-point number in microns. The DIRECTION value must be X or Y. For example:

  ```
  M1 TRACK:WIDTH:0.02:DIRECTION:X 10 1
  ```

  This statement translates track objects on the M1 layer that are 0.02 microns wide in the x-direction to layer 10, datatype 1.

- **LEF/DEF FDI Transcript Enhancement —** When reading LEF files, the tool now outputs a summary section to the transcript with the title “LEF TECHNOLOGY

LAYERS” that includes all layers in the technology LEF. In previous releases, the title was “LEF LAYERS READ.”

- **Change to FDI SIZE Mapping Behavior —** In previous releases, the SIZE sub-type for LEF/DEF via cut layer mapping could also apply to routing layers. As of this release, this is no longer the case; the SIZE sub-type now only applies to via layers.
- **Controlling Error Handling in the LEF/DEF Map File —** You can now optionally control how errors are handled in the LEF/DEF map file with the new fdi2gds and fdi2oasis -inputExceptionList option. The usage is as follows:

  ```
  {fdi2gds | fdi2oasis} -system lefdef options
     [-inputExceptionList {parameter severity}…]
  ```

  where the *parameter* argument is one of the following:

  - CELLGDS_DUPLICATE_CELL
  - CELLGDS_PRECISION
  - WAIVER_PRECISION
  - EXCLUDE_CELL_NAME
  - LEFDEF_MAP_WARNINGS
  - FDI1* — (All fdi2gds and fdi2oasis warning messages)

  and the *severity* is either 0, 1, or 2, where:

  - 0 — Silently skip map file issues.
  - 1 — Issue warnings and skip map file issues.
  - 2 — Issue an error and terminate the run on map file issues. This is the default behavior for map file warnings.
- **DEF PIN Net Name Mapping for LEF/DEF —** The -annotatePinNets option for fdi2gds and fdi2oasis is new and optionally maps DEF PIN net names from the fdi2gds and fdi2oasis command-line. When specified, the tool annotates PIN NET objects from LEF/DEF to properties in the output database. The usage is as follows:

  {**fdi2gds** | **fdi2oasis**} ***options*** [-annotatePinNets [TEXT] [PROPERTY {*pnum* | *pname*}]
- **Enhancement to FDI OASIS Output Layer Names for LEF/DEF —** The layer names generated by fdi2gds and fdi2oasis have been enhanced for consistency. When performing certain mapping operations in previous releases, the output OASIS layer name included literal object names such as “_netName_<net>”, “_mask_<mask>”, and “_netType_<type>”. The literal “netName”, “mask”, and “netType” text is no longer

included in the layer name. Layer names and object names are period-separated, for example:

```
M3.VIA
```

Multiple object names are comma-separated, for example:

```
M3.VIA,NET,SPNET
```

Net based mapping or other secondary object modifiers can include a colon, for example:

```
M3.VIA:VDD
V1.LEFOBS:SIZE:0.040x0.040
```

Table 6-1. OASIS Mapping Name Examples

Mapping Option	Map File	Oasis Layer Name
Mapping with multiple object types	METAL1 NET,SPNET	METAL1.NET,METAL1.SPNET
SIZE subtype	VIA12 VIA:SIZE:0.26x0.26	VIA12.VIA:SIZE:0.26x0.26
MASK subtype	METAL2 VIA:MASK:2	METAL2.VIA:2
TYPE subtype	METAL1 SPNET:TYPE:GROUND	METAL1.SPNET:GROUND
Net name subtype	METAL1 SPNET:VDD	METAL1.SPNET:VDD
DIRECTION subtype	METAL1 TRACK:DIRECTION:X	METAL1.TRACK:X

2020.4 Updates

- **New Direct Backannotation LEF/DEF Flow —** This release includes a new direct backannotation flow that enables you to generate enhancement geometries in Calibre (such as redundant vias, line extensions, and fill), and then export the new geometries to a full or incremental DEF file. You control this flow entirely from a Calibre rule file using SVRF statements; no additional executables or utilities are required.

  This flow is controlled by the DFM RDB DEF statement, which is new for this release. Use this statement to specify the Calibre layers and object types to backannotate and set options for the output DEF file.

  Using this new integrated flow, you can read in a LEF/DEF design using the direct read functionality in the Layout Path statement, generate enhancement shapes with DFM or other statements, and then export the enhancement shapes with the original DEF design, all within a single SVRF rule file.

For more information, see "DFM RDB DEF" in the *SVRF Manual* and "Directly Backannotating to DEF Using SVRF" in the *Calibre Layout Comparison and Translation Guide*.

- **Support for LEF VIA Obstruction Object Types in LEF/DEF Mapping —** The fdi2gds and fdi2oasis utilities now support LEF via obstructions. The LEFOBSVIA[:SIZE][:MASK] mapping keyword is new. If you do not specify a mapping file, LEF VIA obstructions are written to the generated mapping file as comments (you must uncomment the lines to output obstructions in LEF vias).

  If you specify a map file with LEFOBSVIA mapping statements, then the LEF obstruction via names include the "_LEFOBSVIA" suffix. Layers with a LEFOBSVIA type are only included in the translated output if a map file is specified that includes LEFOBSVIA mapping statements.

- **Enhanced Banner for FDI Utilities —** The banner written to the transcript at the beginning of an fdi2gds and fdi2oasis run has been enhanced to include the following information:

  - Host name
  - Host OS
  - Memory Information
  - CPU Info
  - CALIBRE*, MGC*, FDI*, DBDIFF* environment variables

  The following is an example of the new information:

  ```
  // Running on Linux machine 2.6.32-754.el6.x86_64 #1 SMP Tue Jun 19
  // 21:26:04 UTC 2018 x86_64
  // OS: CentOS release 6.10 (Final)[2.6.32-754.el6.x86_64]
  //
  // Entries in /proc/meminfo:
  //
  // MemTotal: 74365348 kB
  // MemFree: 46289108 kB
  // Cached: 19081708 kB
  // SwapCached: 129288 kB
  // Dirty: 11404 kB
  // Writeback: 0 kB
  //
  // Processor : Intel(R) Xeon(R) CPU X5550 @ 2.67GHz
  // CPU Info : Cores = 4.
  // Process ID: 15537
  // Starting time: Thu Jun 18 12:54:31 2020
  // Command: fdi2oasis -system lefdef -lef file.lef -def file.def -
  outfile
  // out.oas
  ```

- **Support For Box Record Exceptions —** The FastXOR utility now supports the Layout Input Exception Severity BOX_RECORD option. In previous releases, this statement was ignored by FastXOR runs.
- **Support for Additional LEF/DEF Properties —** The fdi2gds and fdi2oasis utilities now support the LEF58_WIDTH and LEF58_MINWIDTH properties for LEF/DEF systems.
- **Minor Enhancement to LEF/DEF Translation Performance —** Internal enhancements were made in fdi2gds and fdi2oasis to reduce run times for LEF/DEF designs with large amounts of warning messages.
- **fdi2spice and cvt_dfmdb2icc.tcl Utilities Discontinued —** Support for the fdi2spice utility was discontinued in the 2011 release. The utility has now been removed from the *$MGC_HOME/bin* directory. The *$MGC_HOME/pkgs/fdi/tcl/cvt_dfmdb2icc.tcl* script has also been removed.

2020.3 Updates

- **Only Generate an Automatic Layer Map —** Only Generate an Automatic Layer Map

  You can now specify to generate a layer map file with fdi2gds or fdi2oasis without performing an immediate translation. To use this feature, specify the new -mapOnly option for fdi2gds or fdi2oasis.

  ```
  {fdi2gds | fdi2oasis} -system LEFDEF -lef "tec.lef std.lef" -def
  top.def …
      -mapOnly
  ```

  Note

  The -map, -layerMap, and -outFile options are mutually exclusive with this option.

  The MGC_CALIBRE_LAYERMAP_FILE environment variable does not support the -mapOnly option.

- **Automatic Layer Mapping for LEF/DEF —** The algorithm for automatically mapping LEF/DEF objects to layers and datatypes has been enhanced for this release. The enhancement only applies when you do not specify the -map option (no user-defined layer mapping file). The objects are now assigned layers and datatypes based on the following priority:
- **LEF/DEF Row and Site Mapping —** When mapping DEF ROW objects, the site ID is no longer required. If you include the site, then it must be specified after the orientation.

  When no mapping is specified, the order of the mapping statement now includes the orientation before the site. The default layer and datatype mapping has not been changed.

- **LEF/DEF Translation Performance Improvement —** Internal enhancements were made to improve the run time when translating LEF/DEF designs with the fdi2oasis or fdi2gds utilities. This improvement also addresses duplicate layer counting that may have been entered into the log, mapping information, and layer names files.

- **New LEF/DEF Warning Message Control —** You can now control the number of LEF/DEF warning messages generated when creating a DFM database (DFMDB) with the new DFM LEFDEF Warning Limit statement. Use this statement to specify the maximum number of warning messages generated for a single message type. Use only with LEF/DEF input flows that generate a DFM database (DFMDB).
- **Ignore Incomplete Cadence Installation —** When translating OpenAccess databases, the fdi2gds and fdi2oasis utilities optionally use the Cadence *cdsLibDebug* tool if it exists in your path. This could cause problems in previous releases if the Cadence tool was not correctly configured. The FDI utilities now detect whether the Cadence tool is operational. If this test fails, then the FDI tools use an internal tool instead.
- **Support for OpenAccess 22.60 —** The fdi2gds and fdi2oasis utilities now support OpenAccess 22.60 (oa22.60p025). The default version, OA 22.43, is unchanged. To enable 22.60 support, do one of the following:
    - Set the following environment variable:

      ```
      MGC_FDI_OA_VERSION="22.60"
      ```

    - Specify the “-system OA 22.60” argument set for fdi2gds or fdi2oasis on the command line.
    - For direct read, specify the following in your SVRF rule file:

      ```
      LAYOUT SYSTEM OA 22.60
      ```

2020.2 Updates

- **Enhancement For OASIS Output —** The fdi2oasis utility has been enhanced to provide improved compression in OASIS output files. This can reduce the generated file size on disk. To enable this feature, set the FDI_FDI2OASIS_USE_XYRELATIVE environment variable to a non-null value.
- **Enhancement For OA Mapping File —** You can now use an OpenAccess mapping file for LEF files. The fdi2gds and fdi2oasis utilities have been enhanced to support the -oamap option for LEF using the following syntax:

  ```
  {fdi2gds | fdi2oasis} -oamap file
  ```

  Where the file uses the following format:

  ```
  layer_name { pin | label | other } layer_number data_type \
     [material [mask]] [qualifier] [photomask] [color_state]
  ```

- **Translate Obstruction Spacing —** The fdi2gds and fdi2oasis utilities include the following new annotation option for LEF/DEF databases that enables you to output obstruction spacing values:

  ```
  -annotateObsSpacing: [TEXT] [PROPERTY {pnum | pname}]
  ```

- **Output Row and Site Information Using LEF/DEF Mapping File —** As of this release, the automatically generated map file (created when translating LEF/DEF systems without the -map option) contains separate mapping statements for each row object in the DEF. The mapping statements include unique <*SiteName*>:<*Orientation*> pairs that are properties of the row in the DEF.
- **Enhanced Messaging —** The fdi2gds and fdi2oasis run summary in the transcript now includes FILL objects for LEF/DEF layouts.
- **Update to OpenAccess Version —** The fdi2gds and fdi2oasis utilities now support OA 22.50p064.

Chapter 7
Design for Manufacturability (DFM)

This chapter describes the 2021.2 changes and enhancements for the Calibre® Design for Manufacturability applications.

See "Corrected Defects and Known Issues" on page 12 for instructions on accessing the complete list of corrected defects and known issues for the release.

See "Calibre Utilities" on page 127 for updates to LEF/DEF and OpenAccess database handling.

Calibre YieldAnalyzer

This section includes changes and enhancements to Calibre® YieldAnalyzer for the 2021.2 release.

For additional information, refer to the *Standard Verification Rule Format (SVRF) Manual* or the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

Try It!	**Calibre Tutorials and Example Kits**
	Tutorials and Example Kits (eKits) contain design data, rule files, and instructions for running and learning more about the tool. There are several eKits for DFM applications. See "Calibre Tutorials and Example Kits (eKits)" on page 22.

Calibre YieldAnalyzer Updates

There were no updates introduced for Calibre® YieldAnalyzer for the 2021.2 release.

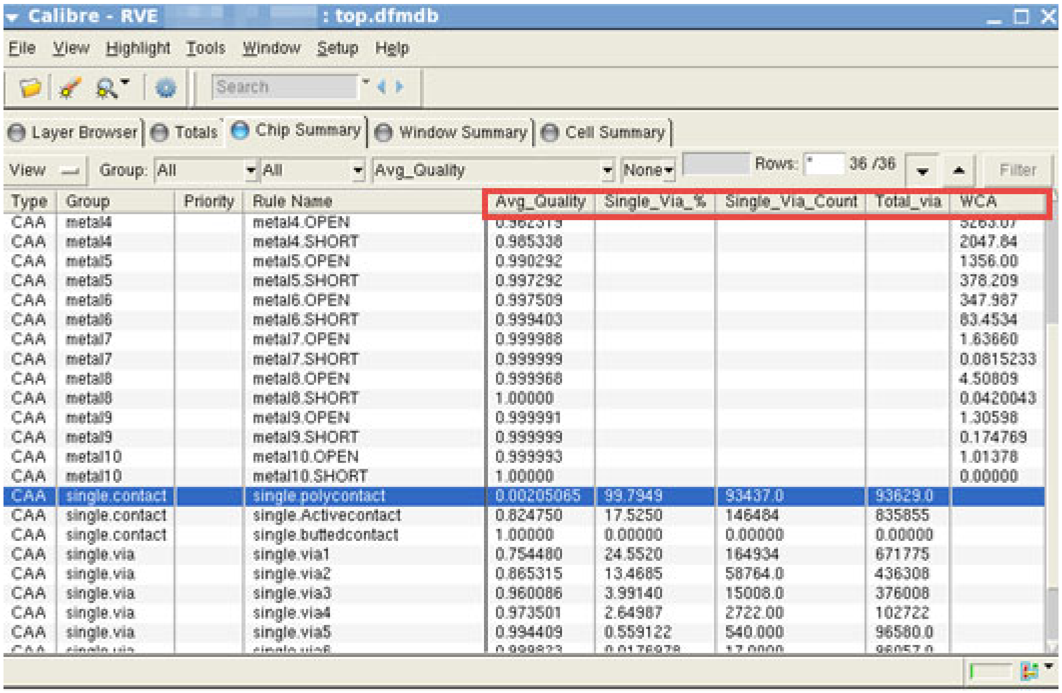
Calibre YieldAnalyzer Previous Release Highlights

Important changes from previous releases are summarized here.

Metric Column Update

(2020.4) The weighted critical area (WCA) metric column now displays in Calibre RVE for DFM for Calibre YieldAnalyzer critical area analysis (CAA) using the "no defect density" (NDD) flow. You can use the standard CAA tabs and options in Calibre RVE for DFM to review the results of the CAA NDD metric scores. See "CAA with No Defect Density" in the

Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual for information on the metrics used in this flow.



View — Group: All All Avg_Quality None Rows: * 36 /36

Type	Group	Priority	Rule Name	Avg_Quality	Single_Via_%	Single_Via_Count	Total_via	WCA
CAA	metal4		metal4.OPEN	0.982319				3263.07
CAA	metal4		metal4.SHORT	0.985338				2047.84
CAA	metal5		metal5.OPEN	0.990292				1356.00
CAA	metal5		metal5.SHORT	0.997292				378.209
CAA	metal6		metal6.OPEN	0.997509				347.987
CAA	metal6		metal6.SHORT	0.999403				83.4534
CAA	metal7		metal7.OPEN	0.999988				1.63660
CAA	metal7		metal7.SHORT	0.999999				0.0815233
CAA	metal8		metal8.OPEN	0.999968				4.50809
CAA	metal8		metal8.SHORT	1.00000				0.0420043
CAA	metal9		metal9.OPEN	0.999991				1.30598
CAA	metal9		metal9.SHORT	0.999999				0.174769
CAA	metal10		metal10.OPEN	0.999993				1.01378
CAA	metal10		metal10.SHORT	1.00000				0.00000
CAA	single.contact		single.polycontact	0.00205065	99.7949	93437.0	93629.0	
CAA	single.contact		single.Activecontact	0.824750	17.5250	146484	835855	
CAA	single.contact		single.buttedcontact	1.00000	0.00000	0.00000	0.00000	
CAA	single.via		single.via1	0.754480	24.5520	164934	671775	
CAA	single.via		single.via2	0.865315	13.4685	58764.0	436308	
CAA	single.via		single.via3	0.960086	3.99140	15008.0	376008	
CAA	single.via		single.via4	0.973501	2.64987	2722.00	102722	
CAA	single.via		single.via5	0.994409	0.559122	540.000	96580.0	
CAA	single.via		single.via6	0.999823	0.0176978	17.0000	96057.0	

Calibre YieldEnhancer

This section includes changes and enhancements to Calibre® YieldEnhancer for the 2021.2 release.

For additional information, refer to the *Standard Verification Rule Format (SVRF) Manual* or the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

Try It!	Calibre Tutorials and Example Kits
	Tutorials and Example Kits (eKits) contain design data, rule files, and instructions for running and learning more about the tool. There are several eKits for DFM applications. See "Calibre Tutorials and Example Kits (eKits)" on page 22.

Calibre YieldEnhancer Updates

The following updates were introduced for Calibre® YieldEnhancer in the 2021.2 release.

- STRICT Mode Supported for DFM RDB OASIS and Compressed Fill Layers
- Post-Fill Filtering of Fill Shapes
- DFM Optimize Updates
- Calibre YieldEnhancer PowerVia Updates
- DFM Spec Via Shift and DFM Via Shift Updates
- DFM Reshape Update

STRICT Mode Supported for DFM RDB OASIS and Compressed Fill Layers

When generating an OASIS database with DFM RDB OASIS, STRICT mode is now the default. In addition, STRICT mode is now supported when writing out compressed fill layers using DFM RDB OASIS. (Compressed fill layers are created with the COMPRESS keyword in DFM Fill.) (ER 1440765)

To disable STRICT mode for *all* OASIS results databases, include the DRC Results Database statement with the NOSTRICT keyword in your rule file.

Post-Fill Filtering of Fill Shapes

The new BLOCKOUT keyword in DFM Fill specifies keep out regions for the generated fill layer. Shapes on the fill layer that interact with shapes on the *blockout_layer* are removed. Interaction is determined between fill shapes and *blockout_layer* shapes, rather than their extents, as is the case for most spacing measurements in fill operations. BLOCKOUT filtering is done after fill is generated, in contrast to the SPACE set of keywords in DFM Spec Fill, which affect the calculated fill region. BLOCKOUT is supported for RECTFILL and POLYFILL. (ER 1442089)

DFM Optimize Updates

The DFM Optimize capability has these updates:

- The DFM Spec Optimize statement has a new RDB keyword that causes the operation to generate a results database that contains statistics about the optimization criteria. The output is similar to that produced by the RDB keyword in DFM Spec Fill. (ER 1441917)
- DFM Spec Optimize Data now supports a range constraint for the optimization limit. Previously, only a constraint with an upper bound was supported. This syntax is now supported:

  '[' *optimization_expression* ']' > *lower_bound* < *upper_bound*

  The *lower_bound* has a higher priority than the *upper_bound*. In order to meet the *lower_bound* requirement of the local optimizer, some windows may exceed the *upper_bound* for the local or global optimizer.
- The optimization expression in DFM Spec Optimize Data now supports multiplication of the generated fill area by a scaling factor. The numerator of the optimization expression must be of this form:

  ```
  c + x*AREA(_FILL_)
  ```

  Where c may be an expression and the "c +" portion is optional. The scaling factor x is optional and x must be a positive scalar when used; x cannot be an expression. (ER 1441009)

Calibre YieldEnhancer PowerVia Updates

The Calibre YieldEnhancer PowerVia flow is updated with the following new *techlib.tcl* file variables:

- DEBUG_SPACE — Produces an output OASIS file with debugging information on fillable regions and spacing layers that represent empty areas for the list of specified vias.
- DEBUG_SPACE_FILE — Specifies the output filename for DEBUG_SPACE.
- INSERTED_VIA_FILE_PREFIX — Specifies a relative path to the output database filename with the filled vias. The default is *via_punch_fill*.(*oas*/*gds*). The tool automatically appends the file extension .(*oas*/*gds*).

See "powervia Variables" in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

DFM Spec Via Shift and DFM Via Shift Updates

The DFM Spec Via Shift specification has a new optional keyword that supports center-to-center measurements in DFM Via Shift.

[CENTER_SPACE *center_space_value*]

This keyword can be specified for all types of via interactions for DFM Spec Via Shift usage.

See "DFM Via Shift Commands" in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

DFM Reshape Update

The DFM Reshape operation can now generate CORNER _CONVEX or CORNER_CONCAVE results together with LINE_END_SPEC on the same layer. For example:

- CORNER_CONVEX with LINE_END_SPEC

  ```
  DFM RESHAPE CORNER_CONVEX(CONCAVE_SPEC corner_fill_spec CONVEX_SPEC
  corner_chop_spec LINE_END_SPEC line_end_spec)…
  ```

  The CORNER_CONVEX transformation plus LINE_END_SPEC produces corner chop and line-end pullback shapes in the same output layer.

- CORNER_CONCAVE with LINE_END_SPEC

  ```
  DFM RESHAPE CORNER_CONCAVE(CONCAVE_SPEC corner_fill_spec
  CONVEX_SPEC corner_chop_spec LINE_END_SPEC line_end_spec)…
  ```

  The CORNER_CONCAVE transformation plus LINE_END_SPEC produces corner fill and line-end extension shapes in the same output layer.

See "DFM Reshape Commands" in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

Calibre YieldEnhancer Previous Release Highlights

Important changes from previous releases are summarized here.

- New TOP Keyword in DFM RDB ASCII
- DFM Optimize Updates
- New NOTCH Keyword for DFM Expand Enclosure
- Enhancements to PowerVia techlib.tcl Procedures
- New Options for DFM Via Shift

New TOP Keyword in DFM RDB ASCII

(2021.1) DFM RDB ASCII now supports the TOP keyword, which causes coordinates to be reported at the top level. (ER 1377797)

DFM Optimize Updates

(2021.1) The DFM Optimize operation now supports hyperscaling mode (the -hyper command line argument), which enables the concurrent, parallel execution of rule file operations. (ER 1434949)

The shape removal process of DFM Optimize is improved so that removal of adjacent shapes is avoided if possible. (ER 1434110)

New NOTCH Keyword for DFM Expand Enclosure

(2021.1) The new optional NOTCH keyword in DFM Expand Enclosure affects how the spacing rule is applied. When NOTCH is specified, the operation distinguishes between edges associated with the same or different connectivity layer shapes. NOTCH enables you to disable or specify a different spacing constraint for edges associated with the same connectivity layer shape. When used, the keyword must be specified after the SPACE keyword set:

SPACE *min_space* [NOTCH *min_notch*]

See the command reference for details and examples.

Enhancements to PowerVia techlib.tcl Procedures

(2021.1) The Calibre YieldEnhancer PowerVia flow has the following enhancements for this release:

- proc assign_via_insertion_order {via_name} {} — New Tcl procedure that enables you to specify the order of insertion for different via types. For example, if the area of a BAR via is greater than the area of a LRG via, you can specify to insert the larger via type first for optimal insertion.

  ```
  "viaBAR viaLRG"
  ```

- proc configure_rdb_results {via_layer} {} — Updated Tcl procedure with new definitions per via layer (instead of per via size) for output to the results database (RDB). The procedure returns a list of pair values.

  ```
  {layer_specifier  net_option}
  ```

See "Calibre YieldEnhancer PowerVia techlib.tcl File Specifications" in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual.*

New Options for DFM Via Shift

(2021.1) The DFM Via Shift operation has the following enhancements for this release:

- TARGET_ENCXY — New option enabling specification of the targeted via enclosure in the x- and y-dimension. The syntax is specified as follows:

  ```
  TARGET_ENClosureXY enc_value[::enc_value_y]
  ```

- NO_CORNER_CHOP_OVERLAP — New option that uses the corner-chop via shape instead of the actual via shape to determine the overlap of the exclusion region with the via. The syntax is specified as follows:

  ```
  exclusion_layer‘(’EXCLUSION [NO_CORNER_CHOP_OVERLAP]‘)’
  ```

See “DFM Via Shift” in the *Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual*.

New FIXED Option for DFM Reshape

(2021.1) New input layer attribute that applies to the UNIFORM transformation. This attribute identifies the orientation (horizontal or vertical) of Manhattan rectangles from the top cell view by checking the BY RECTANGLE values. For example:

```
//To identify vertical rectangles
rect_viaV = DFM RESHAPE UNIFORM via_in(BY RECTANGLE 0.03 0.10 FIXED)
//To identify horizontal rectangles
rect_viaH = DFM RESHAPE UNIFORM via_in(BY RECTANGLE 0.10 0.03 FIXED)
```

See “DFM Reshape” in the *Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual*.

DFM Rectangles: New Operation

(2020.4) DFM Rectangles generates an output layer consisting of rectangles with the specified dimensions and spacing, where these parameters can be provided by explicit numeric values or by DFM properties attached to the input layer. The rectangles are generated within the polygons on an input layer. (ER 1415474)

DFM Optimize Update

(2020.4) When using DFM Optimize, the optimization expression specified in DFM Spec Optimize Data now supports expressions that are not in the form of a density calculation. Specifically, it is not required that the denominator of the optimization expression be AREA(), the area of the current capture window. See the description with DFM Spec Optimize Data and the new Example 3 in DFM Optimize. (ER 1419764)

Update to Compressed Cell Naming for DFM Fill

(2020.4) When COMPRESS is specified in DFM Fill without specifying a *basename* in the corresponding DFM Spec Fill statement, the compressed cell name is formed using the *spec_name* from DFM Spec Fill. Now, characters in *spec_name* that are not allowed in the cell name, such as underscores (_) and spaces, are replaced with a dollar sign ($). Previously such characters were replaced with a hyphen (-). (DR 1429392)

DFM Via Shift Updates

(2020.4) The DFM Via Shift operation has the following enhancements for this release:

- **Marker Layer SRAM ARRAY Flow**

  This flow option preserves symmetry across an SRAM array by taking a marker region as input and using an external operation to generate a shifted result. The result is expanded and replicated across the entire SRAM array without introducing MRC violations.

- **MAX_SHIFTXY**

  The MAX_SHIFTXY optional keyword is added to support different shift values in the x- and y-direction. The following limitations apply:

  - Specifying different maximum shift x/y and short/long shift values together for same-layer input vias is not supported.
  - Specifying different maximum x/y shift values triggers via promotion and can potentially alter hierarchy structure.

- **CHOP BY**

  The CHOP BY SHRINK value can now be specified for a single-side shrink on a rectangular via. For example:

  ```
  CHOP BY SHRINK 0.0::0.005
  ```

  And

  ```
  CHOP BY SHRINK 0.005::0.0
  ```

For more information, see “DFM Via Shift Commands” in the *Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual*.

Calibre YieldEnhancer PowerVia Updates

(2020.4) The Calibre YieldEnhancer PowerVia flow has the following enhancements for this release:

- **DEF Backannotation Flow** — Adds the ability to output via enhancements and special-net types (“ANALOG”, “CLK”, “GROUND”, “POWER”, …) in incremental Design Exchange Format (DEF) for import to place and route tools. See “Backannotation Flow Specifications”.
- **DRC Command-Line Options** — Supports expandable Calibre nmDRC/nmDRC-H command-line options for the powervia utility. For example, if all inputs are located in the working directory:

  ```
  setenv POWER_VIA_TECH /usr/project/work
  powervia -cal_options -turbo -hyper remote -remote host1,host2,host3
  ```

- **Extended Vias Extraction** — Supports Tcl procedures to extract original and extended (BAR1, BAR2, ... LRG1, LRG2, ...) via layers for technology flows using the generated via variables flow.
- New additions to the powervia utility variables and the *techlib.tcl* file procedures.

    - Variables to enable the backannotation flow and DEF specifications:
        - RUN_DEF_BA
        - DEF_FILE
        - DEF_VERSION
    - Tcl procedures to support the backannotation flow and DEF specifications:
        - proc get_def_layer_name {layername} {}
        - proc set_def_net_type {netname} {}
        - proc via_name_in_DEF {via} {}
    - Tcl procedure to support extraction of extended vias (BAR1, BAR2, ... LRG1, LRG2, ...):
        - proc get_extended_type_count {via_name} {}
- Updates to file specifications:

  See your Siemens representative for assistance with the files and specifications used for your foundry process flow.

For more information, see "Calibre YieldEnhancer PowerVia Flow", in the *Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual*.

New DFM Stripe Command

(2020.3) The new DFM Stripe operation fills polygons on an input layer with horizontal or vertical stripes. The stripe width is fixed at the specified width. The stripe spacing can be fixed or adjustable, depending on the setting of the DISTRIBUTE keyword, which controls the stripe placement method. See the command reference for details.

New METACELL Keyword for DFM RDB {GDS | OASIS}

(2020.3) When writing out a GDS or OASIS file with DFM RDB, you can now create a subcell below the topcell and place the layer data in that subcell. You can use multiple DFM RDB operations with the new "METACELL *cell_suffix*" keyword set to create multiple subcells below the topcell. For example:

```
DFM RDB OASIS "output.oas" layer1 1 METACELL _cellA
DFM RDB OASIS "output.oas" layer2 2 METACELL _cellA
DFM RDB oasis "output.oas" layer3 3 layer4 4 METACELL _cellB
```

The preceding code results in this cell hierarchy:

```
TOPCELL
--TOPCELL_cellA  // all layer1 and layer2 data
--TOPCELL_cellB  // all layer3 and layer4 data
```

When METACELL is used, all DFM RDB operations writing to the same file must specify METACELL. Within a set of DFM RDB operations writing to the same file, each layer can be written out only once; that is, the same layer cannot be written to multiple subcells. METACELL can only be used with the GDS or OASIS keyword.

DFM Spec Fill Updates for SHAPESPACE and SPACEXY

(2020.3) You can now specify separate x and y spacing for the SHAPESPACE keyword in DFM Spec Fill and DFM Spec Fill Shape. See the command references for the new syntax.

When using SPACEXY and SPACEXY EDGE the default behavior is to convert all angled edges to a set of small stair-step edges and apply the specified x and y spacing to the stair-step edges. You can now specify whether to keep 45 degree edges and only convert skew edges to stair-steps, or to not convert any angled edges. This new optional keyword set is available:

> [DEANGLED {ALL | KEEP45 | NONE}]
>
> The default behavior (DEANGLED ALL) is the same as the previous behavior. The DEANGLED KEEP45 option provides better performance when there are very long 45 degree edges in the space layer. See the command references for details.

DFM RDB ASCII New Keyword HIGHPREC

(2020.3) The new HIGHPREC keyword in DFM RDB ASCII specifies to print numeric properties with twelve digits of precision, rather than the default of six digits. (ER 1415617)

```
DFM RDB ASCII OUTPUT FILE filename [HIGHPREC]
   FORMAT SPECS
   output_spec ` ;'
   [{output_spec ` ;'} …]
   END FORMAT SPECS
```

New DFM Remove Edge Operation

(2020.3) The DFM Remove Edge operation removes edges shorter than a specified value from the input polygons and outputs the resulting polygons.

DFM REMOVE EDGE *layer* {***length*** | ***x_length y_length***} [NONOTCH] [NOHOLE]

Separate x and y length constraints can be specified. Notches and holes in the input polygons can be ignored.

DFM Segment New Syntax

(2020.3) DFM Segment was introduced in the 2019.4 release and divides an edge layer into alternating segments and outputs the alternate segments. Segmentation is done by length or by count of segments. The new syntax is the following:

Syntax 1: Segment by Count

```
DFM SEGMENT {edge_layer | ‘(’edge_layer‘)’} count
```

Syntax 2: Segment by Length

```
DFM SEGMENT {edge_layer | ‘(’edge_layer‘)’} primary_length gap_length
            [DISTRIBUTE mode]
```

The ***edge_layer*** and **(*edge_layer*)** notations output alternate segments, and replace the previous ODD and EVEN keywords. Additional segment placement options are provided with the DISTRIBUTE keyword.

The following examples show usage in the previous and new syntax.

Segment by Count

```
// Old syntax
M_Odd_N6 = DFM SEGMENT M_edge  ODD  N 6
M_Even_N6 = DFM SEGMENT M_edge EVEN N 6

// New syntax
M_A_N6 = DFM SEGMENT  M_edge  6  // First and alternating segments
M_B_N6 = DFM SEGMENT (M_edge) 6  // Second and alternating segments
```

Segment by Length

In the prior implementation of segmentation by length, if a segment could not be placed in the final position (top or rightmost), it was dropped. In the new implementation, leftover edge length is added to a gap segment.

```
// Old Syntax
M_Odd_L1 = DFM SEGMENT M_edge ODD L 3.0 // length 3 for both odd and even
M_Even_L1 = DFM SEGMENT M_edge EVEN L 3.0

// New Syntax  - Uses default DISTRIBUTE 0
M_A_L3 = DFM SEGMENT  M_edge 3.0 3.0
M_B_L3 = DFM SEGMENT (M_edge) 3.0 3.0
```

Uniform Distribution of Segments With Minimum Gap Specification

The prior DISTRIBUTE syntax is comparable to segment by length with DISTRIBUTE 6 in the new syntax.

```
// Old Syntax
M1_seg_1 = DFM SEGMENT M1_edge_all DISTRIBUTE 1.0 0.1

// New Syntax, output only the primary segment
M1_seg_new = DFM SEGMENT M1_edge_all 1.0 0.1 DISTRIBUTE 6
```

DFM Density: Output Gradient Property Values

(2020.3) You can now attach gradient property values to the output layer generated by DFM Density. The optional GRADIENT keyword causes gradient properties for each data capture window to be attached to the output layer. Eight gradient values are calculated for each data capture window, one for each neighboring window of the window being considered. The maximum gradient value is also calculated. Specify MAX ONLY to only attach the maximum gradient value. See the command reference for details.

New DFM Reshape Commands

(2020.3) The DFM Reshape operation is introduced for this release. This operation is used to correct corner and line-end shapes and generate uniform via layers. These shape transformations imitate the actual metal shapes and improve via enclosures in a subsequent DFM Via Shift operation.

The following commands are used in the DFM Reshape operation:

- **DFM Spec Reshape** — This command specifies the set rules and values used for the DFM Reshape geometry transformation.
- **DFM Reshape** — This command performs the specified type of geometry transformation on the input layer.

For more information, see “DFM Reshape Commands” in the *Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual*.

DFM Via Shift Keyword Updates

(2020.3) The DFM Via Shift operation has the following enhancements for this release:

- DFM Spec Via Shift
    - **SHORT_LONG** — Added keyword for enabling an asymmetric MRC specification that uses layer order to distinguish LONG_SHORT and SHORT_LONG constraints for rectangular via layers.
    - **PROJecting** — Added keyword option for OPPOSITE to use with or without SPACE to enable multiple minimum spacing values along an edge projection range.
- DFM Via Shift

- **SHRINK** — Added keyword option for CHOP BY to shrink the via size from both ends (side-chop) during the calculation of the coverage area. The size of the output via is unchanged. For example:
    - Only side-chop (shrink) is used.

      CHOP BY SHRINK 0.001
    - Both corner-chop and side-chop (shrink) are used.

      CHOP BY 0.005 SHRINK 0.001

  The CHOP BY value (corner-chop) can be specified with or without SHRINK (side-chop). If both are specified, the CHOP BY value must be specified before the SHRINK value.

Calibre YieldEnhancer PowerVia Enhancements

(2020.3) The PowerVia flow has the following enhancements for this release:

- **Connectivity-Based Spacing** — This functionality has updated variable specifications to perform connectivity-based spacing from upper-via layers to lower-via layers (Vx to Vx-1).
- **powervia** — This utility has added keywords for specifying local and remote hosts for multithreaded operations.
    - **-remote *host***[,*host*…]
    - **-remotefile *config_file_name***
- Additions to *techlib.tcl* file variables and procedures:
    - **MERGED_OUTPUT** — Merges the output of the original input and the *via_punch_fill*.(*oas*/*gds*) layout database files.
    - **PERC_FLOW** — Enables the *techlib.tcl* flow to use Calibre PERC flow layers defined in the specified file instead of generating these layers.
    - **RDB_FILE** — Specifies the name of an output file that is a results database (RDB) (default is *powervia.rdb*).
    - **RESOLUTION** — Specifies the step size and causes a square grid step (equal in x- and y-directions).
    - **SVRF_VERSION** — Enables Calibre version checking in the SVRF rule file.
    - **DISABLE_ SPECIAL_FILTERS** — Specifies to disable special via filters for empty areas.
    - **proc exclude_ip_cells {via_name} {}** — Excludes via insertion from certain IP cells. The specified via_name applies to all types for that via.

- Updates to file specifications:
    - **Net Names Mapping** — Ensures proper handling of user-specified nets with matching names, differing by case (for example, “VDD” and “vdd”). The nets are mapped and renamed (internally only) during the flow.

      Note

      If your process design kit contains “VIRTUAL CONNECT NAME *name* [*name*…]” statements, you must add the “Layout Preserve Case Yes” statement if you want to use case sensitivity for net names.

    - **Enclosure Template File (Generated Flow)** — Generates the enclosure template file with via parameters for the specified layer stack. This file is used as input for certain foundry process flows only. An input enclosure template file is required with via type specifications. An example enclosure template file specification is shown.

      Given input enclosure template file line with “Vx” via type:

      ```
      Vx M_LOWER width 0.8 0.9 enc 0.18 0.03
      ```

      Corresponding generated enclosure template file lines with via parameters:

      ```
      VIA1i M1i width 0.8 0.9 enc 0.18 0.03
      VIA2i M2i width 0.8 0.9 enc 0.18 0.03
      ```

  See your Siemens representative for assistance with the files and specifications used for your foundry process flow.

For more information, see “Calibre YieldEnhancer PowerVia Flow”, in the *Calibre YieldAnalyzer and YieldEnhancer User’s and Reference Manual*.

DFM Optimize: INSIDE OF LAYER Capability

(2020.2) You can now specify that DFM Optimize analyze and optimize fill within the extent of a specified layer. Specify the layer with the INSIDE OF LAYER keyword in DFM Spec Optimize.

DFM Segment Updates

(2020.2) DFM Segment now has the DISTRIBUTE keyword, which divides the input edge into fixed length segments that are evenly distributed and have a minimum gap between the segments.

DFM SEGMENT *edge_layer* DISTRIBUTE *length min_gap*

Calibre YieldEnhancer PowerVia Spacing Updates

(2020.2) The PowerVia flow has the following spacing enhancements for this release:

- Same-Layer Spacing Rule — This functionality controls spacing between vias (original and fill vias) located on the same layers. Spacing constraints between bar, large, and square vias are supported.
- Connectivity-Based Spacing for Same-Layer Vias — This functionality uses certain variable specifications to enable connectivity-based spacing for same-layer vias on the same nets and different nets.
- Spacing by PRL for Same-Layer Vias — This functionality uses PRL-based spacing for same-layer vias on the same nets and different nets.

DFM Via Shift Enhancements

(2020.2) The DFM Via Shift operation has the following enhancements for this release:

- DFM Spec Via Shift — Updated usage specifications that support both rectangular and square via interactions between movable vias and unmovable vias with unknown size.
- DFM Via Shift — Updated usage specifications and keywords that support options for filtering by via size and shape, via cluster selection, improved via centering, handling of invalid via layers, and same-layer via interactions.
    - map_option — An optional argument set that outputs additional layers containing vias with invalid size and shape from the input layer or separators between via geometries.
    - *via_layer_options* — An optional argument set that includes new keywords for specifying via size (SIZE), required for movable via layers and optional for unmovable via layers, and improved via centering (CHOP BY).

SPACE WRAP Deprecation

(2019.2) The SPACE WRAP keyword in DFM Spec Fill is deprecated as of the 2018.4 release. The functionality may continue to operate but it is no longer supported. Contact your Siemens EDA representative for information on transitioning to a replacement.

Calibre YieldServer

This section includes changes and enhancements to Calibre® YieldServer.

For more information, see the *Calibre YieldServer Reference Manual*.

Calibre YieldServer Updates

The following updates were made to Calibre YieldServer in the 2021.2 release.

LVS Expand On Error YES Supported by dfm::run_compare

The dfm::run_compare command now observes the LVS Expand On Error YES statement if specified in the rule file that generated the DFM database. This setting follows the same behavior as in hierarchical LVS. ER 1437131

Calibre CMPAnalyzer

This section includes the changes and enhancements to Calibre® CMPAnalyzer for the 2021.2 release.

For additional information, refer to the *Calibre CMPAnalyzer User’s Manual*.

Try It!	**Calibre CMPAnalyzer Tutorial and Example Kit** Run a batch CMP analysis using the Calibre CMPAnalyzer simulator flow. Also learn how to calculate a user-defined hotspot, run a CMP batch bucketing flow analysis, and export CMP thickness data for parasitic extraction tools. 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.

See “Calibre Tutorials and Example Kits (eKits)” on page 22 for a listing of all eKits.

Calibre CMPAnalyzer Updates

The following updates were introduced for Calibre® CMPAnalyzer in the 2021.2 release.

New Calibre CMPAnalyzer Rerun Flow

The Calibre CMPAnalyzer rerun flow enables you to selectively re-extract and perform simulation on designs with newly added layers, layout changes, and process recipe changes. It uses a modified runset and Calibre YieldServer commands to rerun the CMP analysis.

During the analysis, the rerun flow only re-extracts and simulates the selected layers in the modified runset and reuses the data for any unchanged layers when regenerating the DFM database. This reduces the time spent regenerating the data for the entire DFM database and enables you to quickly assess intermediate design and process changes.

You specify a required environment variable to enable the flow and use Calibre Interactive for DFM and Calibre YieldServer commands to run the flow in batch or GUI mode.

```
setenv CMP_ENABLE_RERUN_SIMULATION 1
```

This re-extracts and simulates again the layers in the modified runset for the selected layer indexes, for example, [list 2 4]:

```
> cmp::rerun_extraction -layers [list 2 4] -runset ./modified_runset
-resimulate
```

See “Calibre CMPAnalyzer Rerun Flow” in the *Calibre CMPAnalyzer User’s Manual* for the complete command syntax and steps for running this flow.

Calibre CMPAnalyzer Previous Release Highlights

Important changes from previous releases are summarized here.

- Calibre CMPAnalyzer Enhancements
- Extent Layer
- Backup and Wrap Flow
- Hotspots Clustering
- Results Database Generation
- Calibre YieldServer CMP Command Updates
- Vector Property Format
- Calibre Interactive GUI (Input Pane) — CMPAnalyzer Flow
- Thickness Data Readings

Calibre CMPAnalyzer Enhancements

(2020.4) Calibre CMPAnalyzer has the following updates:

- **Predefined Layers** — The file format now supports encrypted predefined layer files for SVRF and TVF rules.
- **Layer Definition** — The CMP GUI in the Calibre Interactive Inputs pane now accepts multiple layer definitions separated by a semicolon (;) to support the multiple-etch flow for predefined layers.
- **Hotspots Clustering** — The property data for the hotspot clusters displays as text labels after the clusters are highlighted in Calibre DESIGNrev or other layout viewer.
- **Export Colormap** — The colormap in Calibre RVE for DFM is now updated with a **Coords** submenu selection that enables the exporting of coordinate information from a colormap tile.

Extent Layer

(2020.3) The updated **Extent layer** option changes the definition of the DB_EXTENT marker layer, which defines the extent of the layers needed for the Calibre CMPAnalyzer analysis. When **Extent layer** is enabled in the Calibre Interactive GUI, it behaves as follows:

- If the layer field is empty (default), all layers are used to determine the extent, whether or not they are part of the simulation.

DB_EXTENT = MERGE(EXTENT DRAWN ORIGINAL)

- If the layer field has a specified layer number, this layer is used to determine the extent.

  DB_EXTENT = MERGE(EXTENT *layer*)

Backup and Wrap Flow

(2020.3) Calibre CMPAnalyzer runs are enhanced with the backup and wrap flow option, which is enabled in the **Area DFM** tab in the Calibre Interactive GUI. This functionality extends the design on all four sides symmetrically, ensuring that tiles (windows) at the design edges are completely filled. It derives a new design extent and fills the additional area with shapes copied from the existing data in the design. When the **Backup and wrap flow** setting is enabled, there are two modes to choose from:

- **Backup** — Selects shapes starting at the original design edges and shifts them outward to fill the empty areas of the new extent layer (default).
- **Wrap** — Emulates the die in wafer-context by copying shapes from edges and corners to fill empty spaces in the new extent layer. The “Gap size” field specifies the space between existing and copied shapes.

The backup and wrap flow is launched in Calibre CMPAnalyzer. It redefines the design content before the analysis. After the run completes, Calibre RVE for DFM is used for generating reports and viewing results.

Hotspots Clustering

(2020.3) Dynamic hotspot clustering combines nearby single-pixel hotspots into larger clusters of rectangular shapes. The cluster shapes can then be passed to other tools for redesign and optimization. Reporting clusters of hotspots by level, instead of single-pixel hotspots, improves reporting by significantly reducing the number of reported hotspots.

Hotspot clustering can be run from the Calibre CMPAnalyzer GUI in Calibre RVE for DFM from the **Tools** > **CMP** > **HotSpots Clustering** menu item or from a Calibre YieldServer command line.

From the Calibre YieldServer command line, the cmp::run_clusterization command generates the hotspot clusters. For example:

```
set Erosion_clusters [cmp::run_clusterization -layer M1_data_detail
-property Hotspot_Rule_Erosion -distance 80 -level 10]
```

Results Database Generation

(2020.3) Two new options make it easier to generate results database (RDB) files:

- You can specify the cmp::export_grid command with the -rdb argument to output results in RDB format for a given grid.

- You can use **File** > **Save Hotspots Report** and choose either *.txt* or *.rdb* format in the Calibre CMPAnalyzer GUI in Calibre RVE for DFM.

Calibre YieldServer CMP Command Updates

(2020.3) The Calibre YieldServer commands for CMP are updated to support new functionality.

The following new commands support hotspot clustering functionality:

- cmp::run_clusterization
- cmp::save_clusters
- cmp::export_clusters
- cmp::cleanup_clusters

The cmp::export_grid command is updated with arguments to support RDB file generation and property data enhancements.

For complete command and argument syntax, see "CMP YieldServer Command Reference" in the *Calibre CMPAnalyzer User's Manual*.

Vector Property Format

(2020.3) Calibre CMPAnalyzer runs have a **Save Vector Properties** option in the Calibre Interactive GUI **Simulator Options** tab. If selected, the data for each property is saved as a vector on a single-geometry CMP layer. The new vector format collects all data for each property, while requiring only one access to the DFM results database.

The vector property format is compatible with user-defined hotspot (UDHS) Tcl scripts implemented with Calibre YieldServer grid commands. Iterator-based UDHS Tcl scripts are not compatible with the new vector property format.

Calibre Interactive GUI (Input Pane) — CMPAnalyzer Flow

(2020.2) The Calibre CMPAnalyzer flow in Calibre Interactive has the following updates for this release:

- **Simulation Options Tab** — Control the number of CPU threads for CMP simulation in the Simulator CPUs field. The default number is 4 CPUs for simulation.
- **CMP Tab GUI Columns** — Show columns in the data table for these parameters:
    - **Window Size** — Specifies the window size for the simulator. The default is 20, which means an approximate 20 um by 20 um window (tile). A single number or two space-separated numbers for the x- and y-dimensions can be entered.
    - **Array Recognition Region (ARR)** — Specifies an optional parameter that enables array recognition and the extraction of geometric grid information for the search

region. The specified value is the radius in um of the search region for array recognition.

Use the right-click menu in the data table to select the columns to display and to adjust layer parameters.

Thickness Data Readings

(2020.2) The Calibre CMPAnalyzer simulator flow reports results on certain thickness parameters. All thickness data is now reported in whole angstroms to simplify simulation analysis.

Calibre CMP Model Builder

This section includes the following changes and enhancements to Calibre® CMP Model Builder for the 2021.2 release.

For additional information, refer to the *Calibre CMP Model Builder User's and Reference Manual*.

Calibre CMP Model Builder Updates

The following updates were made to Calibre® CMP Model Builder for the 2021.2 release.

New Enhanced Perimeter Model (EPM)

The enhanced perimeter model (EPM) is used to modify the removal rate of the material depending on the contact area of the different materials along the pattern perimeter. See "CMP Model" in the *Calibre CMP Model Builder User's and Reference Manual* for the GUI parameter fields and values used with the perimeter model.

Process Recipe Command and GUI Updates

The following process recipe commands are enhanced with new arguments that provide the following functionality:

- define_model
    - rfb=value — This argument specifies a rate-factor bias value used for the enhanced perimeter model (EPM) that decreases (value < 1) or increases (value > 1) the polishing rate. It is specified together with the wb argument and affects the removal rate of the material.
    - wb=value — This argument specifies a width bias value used for the enhanced perimeter model (EPM) that defines the transition region between trench and non-trench patterns. It is specified together with the rfb argument and affects the removal rate of the material.

  Available in the GUI from the CmpModel dialog box (RFB and WB).

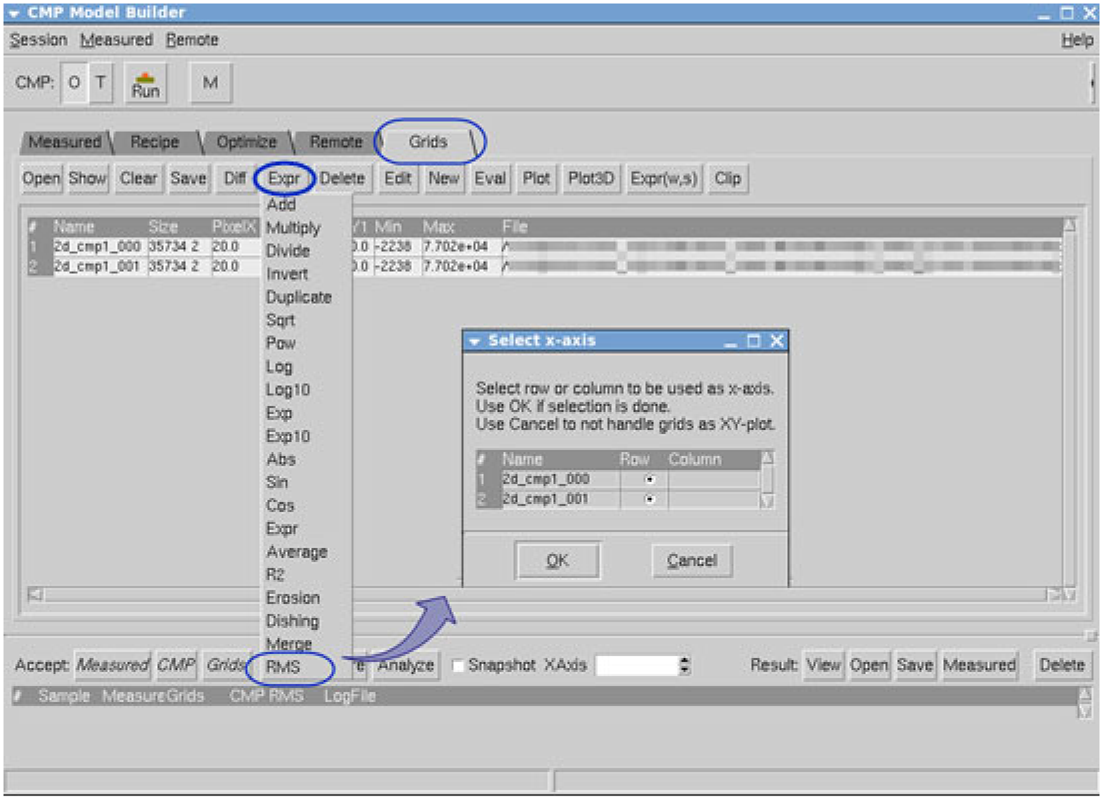
- save
    - topology — This keyword specifies thickness topology data related to the Thickness_Topology property used for analysis in Calibre CMPAnalyzer.

      Available in the GUI from the Save dialog box.

- RMS — This expression specifies an RMS error calculation for comparing two grids of the same size.

  Available in the GUI from the **Grids** tab under **Expr** > **RMS**.

Figure 7-1. RMS Expression



#	Name	Size	PixelX	/1	Min	Max	File
1	2d_cmp1_000	35734 2	20.0	).0	-2238	7.702e+04	/
2	2d_cmp1_001	35734 2	20.0	).0	-2238	7.702e+04	/

Select row or column to be used as x-axis.
Use OK if selection is done.
Use Cancel to not handle grids as XY-plot.

#	Name	Row	Column
1	2d_cmp1_000	●	
2	2d_cmp1_001	●	

□ Snapshot

#	Sample	MeasureGrids	CMP RMS	LogFile

CMP Modeling Previous Release Highlights

Important changes from previous releases are summarized here.

- Calibre CMP Model Builder Enhancements
- SidewallsOnly
- Thickness Data in Whole Angstroms

Calibre CMP Model Builder Enhancements

(2021.1) The Calibre YieldEnhancer PowerVia flow has the following enhancements:

- The etch command now supports etch type empty_trenches with Thickness values to control the corresponding material thickness for each window.
- The pattern density is now kept for all pixels, instead of being reset to zero, after deposit fill with TargetThicknessNT=0 setting.

- The **Grids** tab now opens by default files with the *.grid* extension (such as those exported from Calibre CMPAnalyzer).

SidewallsOnly

(2020.4) The Deposit dialog box in the **Recipe** tab has a new SidewallsOnly parameter selection. You can use this option to support material deposition only on the trench sidewalls. If enabled, the thickness value of the deposition in angstroms (Thickness, A) is used to update the geometry data.

Thickness Data in Whole Angstroms

(2020.2) All values for thickness data grids Z1, Z2, ThicknessT, and ThicknessNT are now reported in whole angstroms.

Calibre Litho-Friendly Design (LFD)

This section includes changes and enhancements to Calibre® LFD™ for the 2021.2 release.

For a detailed description, refer to the *Calibre Litho-Friendly Design User's Manual*.

Calibre Litho-Friendly Design (LFD) Updates

There were no updates introduced for Calibre LFD in the 2021.2 release.

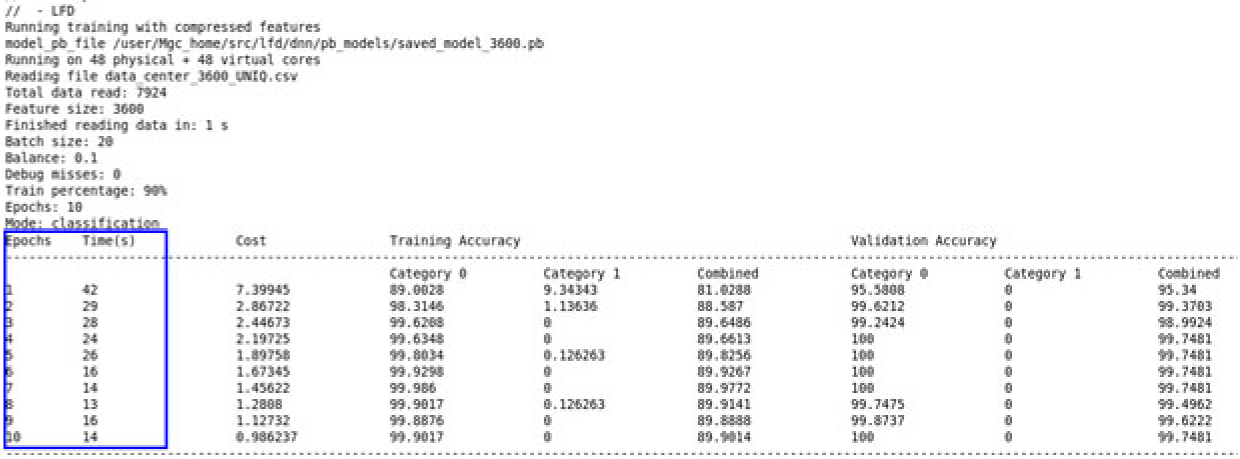
Calibre Litho-Friendly Design (LFD) Previous Release Highlights

The following updates were made in previous releases of Calibre LFD.

- LFD Machine Learning Transcript Update — Epoch Times
- LFD DNN Training Transcript Enhancements — Metrics

LFD Machine Learning Transcript Update — Epoch Times

(2020.4) The lfd_dnn_train flow now displays updated transcript epoch times in seconds instead of milliseconds to facilitate evaluating results.



```
//  - LFD
Running training with compressed features
model_pb_file /user/Mgc_home/src/lfd/dnn/pb_models/saved_model_3600.pb
Running on 48 physical + 48 virtual cores
Reading file data_center_3600_UNIQ.csv
Total data read: 7924
Feature size: 3600
Finished reading data in: 1 s
Batch size: 20
Balance: 0.1
Debug misses: 0
Train percentage: 90%
Epochs: 10
Mode: classification
Epochs    Time(s)         Cost          Training Accuracy                                 Validation Accuracy
---------------------------------------------------------------------------------------------------------------------------------
                                        Category 0    Category 1    Combined      Category 0    Category 1    Combined
1         42              7.39945       89.0028       9.34343       81.0288       95.5808       0             95.34
2         29              2.86722       98.3146       1.13636       88.587        99.6212       0             99.3703
3         28              2.44673       99.6208       0             89.6486       99.2424       0             98.9924
4         24              2.19725       99.6348       0             89.6613       100           0             99.7481
5         26              1.89758       99.8034       0.126263      89.8256       100           0             99.7481
6         16              1.67345       99.9298       0             89.9267       100           0             99.7481
7         14              1.45622       99.986        0             89.9772       100           0             99.7481
8         13              1.2808        99.9017       0.126263      89.9141       99.7475       0             99.4962
9         16              1.12732       99.8876       0             89.8888       99.8737       0             99.6222
10        14              0.986237      99.9017       0             89.9014       100           0             99.7481
---------------------------------------------------------------------------------------------------------------------------------
```

LFD DNN Training Transcript Enhancements — Metrics

(2020.2) The lfd_dnn_train flow now has additional information included in the transcript output.

- Updated logging information for the number and type of CPU cores utilized during the run:
    - Non-turbo case — Running on *X* physical + *Y* virtual cores
    - -turbo *X* case — Running on *Y* cores where,
        - *Y* = *X* (if *X* <= *physical_cores_count*)
        - *Y* = *physical_cores_count* (otherwise)
- Improved accuracy reporting for machine learning classifiers for hotspot (HS) and non-hotspot (NHS) statistics:
    - Recall (HS Accuracy)
    - Precision
    - Extras Percentage
    - F1 Score

See "Training Transcript Information" in the *Calibre Litho-Friendly Design User's Manual* for a description of terms and an example training transcript.

New Product: Calibre LSG Synthetic Layout Generator

Calibre LSG Synthetic Layout Generator (Calibre LSG) is a new software tool used in conjunction with Calibre WORKbench to create user-defined pattern definitions for randomly-generated layout clips. The layout clips are suitable for pattern analysis and other early design phase applications for specific layers and technologies.

For further information on Calibre LSG, refer to the new *Calibre LSG for Synthetic Layout Generation User's Manual*.

Chapter 8
Parasitic Extraction (PEX)

This chapter describes the 2021.2 changes and enhancements for the Parasitic Extraction applications.

See "Corrected Defects and Known Issues" on page 12 for instructions on accessing the complete list of corrected defects and known issues for the release.

See "Calibre Utilities" on page 127 for updates to LEF/DEF and OpenAccess database handling.

Calibre xACT Updates

This section includes information on the following changes and enhancements to the Calibre® xACT™, Calibre® xACT 3D, Calibre® xACT 3D Reference, and Calibre® xACTView tools for the 2021.2 release.

- Enhanced Xcell List Format for Calibre xACT
- Changes to SVRF Statements Supported by Calibre xACT

For more information, refer to the *Standard Verification Rule Format (SVRF) Manual* or the *Calibre xACT User's Manual*.

Enhanced Xcell List Format for Calibre xACT

As of the 2021.2 release, two new flags have been added to the xcell file format:

- -CONLY— Indicates that the SPICE model for the cell only contains parasitic capacitance. This means only parasitic resistance is extracted for the cell. These resistors are identified in the extracted netlist with the property called $pcell_res=1. This flag can only be specified together with -PCDEF.
- -NOBLOCK — Indicates the cell is not a pcell. The contents are still extracted and written to the netlist. Use this flag only in cases where the cell name is part of a wildcard list and should be treated differently.

See “Hierarchy Control with xCells” in the *Calibre xACT User’s Manual* for more information.

Changes to SVRF Statements Supported by Calibre xACT

The 2021.2 release adds the following new Calibre xACT SVRF statements:

- PEX Bulk Model — Use this statement to control whether or not the PEX Extract Bulk...Ideal statement is applied during extraction. Use only with Calibre xACT and the Calibre xACT 3D direct netlisting flow.

The 2021.2 release changes the behavior of the following existing Calibre SVRF statement:

- PEX Xcell — Syntax 1 for this statement has been enhanced as follows:
    - A new optional keyword CONLY has been added to this release for use with Calibre xRC, Calibre xACT, Calibre xACT 3D. CONLY is an optional keyword specified with the PCDEF keyword to ignore capacitance and only extract parasitic resistance for the cell. These resistors are identified in the extracted netlist with the property called $pcell_res=1. Use this keyword when the SPICE model for the cell includes parasitic capacitance but not parasitic resistance.
    - A new optional keyword NOBLOCK has been added to this release for use with Calibre xRC, Calibre xACT, and Calibre xACT 3D. NOBLOCK is an optional keyword that indicates the cell is not a pcell. The contents of the cell are still extracted and written to the netlist. Use this keyword in when you want to explicitly specify cells to be extracted.
- PEX BA Mapfile — The optional keyword set, PINEXCEPT *pinids*, found in the optional section for adding LPE parameters to the device instance statements for specific intentional device models in the parasitic netlist, has been enhanced. For the Calibre xACT direct netlisting flow, *pinids* can be specified numerically or by pin name.

    As of the 2021.2 release, MOS devices numeric *pinids* follow the order shown Table 8-1. For all other devices including built-in or user defined devices, the numeric *pinid* uses the device pin order as defined by the Device statement declaration.

    Table 8-1. Pinids

MOS PIN	ID
DRN	0
GATE	1
SRC	2
BULK	3

- PEX Netlist — This statement has been enhanced as follows:
    - The PINDELIM *string* keyword set can now be used in the Calibre xACT and Calibre xACT 3D direct netlist flows for the CALIBREVIEW netlist format. Use

this keyword set to specify the delimiter character between device names and pin names in the CALIBREVIEW netlist.

- The TOTEM keyword now appends the bounding box information to the intentional devices found in the Instance Section of the DSPF netlist as lower left ($llx $lly) and upper right ($urx $ury) coordinates.
- The LAYER_MAP keyword now appends the bounding box information to the intentional devices found in the Instance Section of the DSPF netlist as lower left ($llx $lly) and upper right ($urx $ury) coordinates.

Calibre xRC Updates

This section includes information on changes and enhancements to the Calibre® xRC™ tool for the 2021.2 release.

- Enhanced Calibre xRC Formatter Log File Output
- Changes to SVRF Statements

For more information, refer to the *Standard Verification Rule Format (SVRF) Manual* or the *Calibre xRC User's Manual*.

Enhanced Calibre xRC Formatter Log File Output

The log file information generated by the Calibre xRC formatter now includes information about the SVRF settings used during your formatter run. The information is found under the heading "PEX OPTIONS USED DURING FORMATTER". For example:

```
//  PEX OPTIONS USED DURING FORMATTER:
//  ----------------------------------
//  LAYOUT CASE NO
//  SOURCE CASE NO
//  LVS COMPARE CASE NO
//  PEX BA MAPFILE svrf/BA_mapping_pin_2
//  LPE setting in BA MAPFILE svrf/BA_mapping_pin_2
//  PARM setting in BA MAPFILE svrf/BA_mapping_pin_2
//  flag setting in BA MAPFILE svrf/BA_mapping_pin_2
//  PEX REDUCE STUB YES
//  PEX NETLIST "netlist.dspf" DSPF LAYOUTNAMES GROUND 0 LOCATION RCNAMED
//  HIERARCHICAL SEPARATOR "/"
//  PEX NETLIST LINEWRAP NO
//  PEX NETLIST LPE USING EXTMODE YES
```

Changes to SVRF Statements

The 2021.2 release changes the behavior of the following existing SVRF statements:

- PEX Reduce Mincap — This statement can now be used when the netlist includes parasitic self or mutual inductance as extracted by Calibre xL or the Calibre xACT inductance flow. Use this statement to combine or remove parasitic capacitors below a threshold value to reduce netlist size.

- PEX Reduce Minres — This statement can now be used when the netlist includes parasitic self or mutual inductance as extracted by Calibre xL or the Calibre xACT inductance flow. Use this statement to combine or short parasitic resistors below a threshold value to reduce netlist size.

Calibre xL Updates

This section includes information on changes and enhancements to the Calibre® xL tool for the 2021.2 release.

- Changes to Inductance SVRF Statements

For more information, refer to the *Standard Verification Rule Format (SVRF) Manual* or the *Calibre xL User's Manual*

Changes to Inductance SVRF Statements

The 2021.2 release changes the behavior of the following existing SVRF statements:

- PEX Reduce Mincap — This statement can now be used when the netlist includes parasitic self or mutual inductance as extracted by Calibre xL or the Calibre xACT inductance flow. Use this statement to combine or remove parasitic capacitors below a threshold value to reduce netlist size.
- PEX Reduce Minres — This statement can now be used when the netlist includes parasitic self or mutual inductance as extracted by Calibre xL or the Calibre xACT inductance flow. Use this statement to combine or short parasitic resistors below a threshold value to reduce netlist size.

xCalibrate Updates

This section includes information on changes and enhancements to the xCalibrate™ rule file generator for the 2021.2 release.

- Enhanced ITF2MIPT2 Translation
- Enhanced MTflex Support for xCalibrate

For more information, refer to the *xCalibrate Batch User's Manual*.

Enhanced ITF2MIPT2 Translation

The optional argument set, -itfmap *map_file_name* [*map_file_name*...], is used by the itf2mipt2 translator to convert an ITF file with one or more ITF mapping files to an MIPT file (*out.mipt*) with embedded SVRF mapping and ignore statements.

As of the 2021.2 release, a new optional argument, -separate, has been added to the -itfmap argument set.

```
[-separate] -itfmap map_file_name [map_file_name ... ]
```

Specify the -separate argument just before the -itfmap argument in the command line to write the SVRF statements that would otherwise be found in the svrf_verbatim section of the *out.mipt* file into a separate mapfile called *out.map*.

Enhanced MTflex Support for xCalibrate

As of the 2021.1 release, the VER and DIR configuration file header statements are no longer required or supported. Also, the MGC_HOME *mgcPath* argument set, used in both the REMOTE COMMAND and REMOTE HOST configuration file statements, is also no longer required by xCalibrate.

xCalibrate now uses the same configuration file syntax and setup as Calibre with the exception of the executable file called by the REMOTE COMMAND statement. For xCalibrate jobs, the executable file called by the REMOTE COMMAND statement must specify the *$CALIBRE_HOME/bin/rxcalibrate* utility to initiate an xCalibrate MTflex run on remote hosts.

For details on how to use MTflex with xCalibrate, see “Executing xCalibrate With Calibre MTflex Using IBM Spectrum LSF” in the *Calibre Administrator’s Guide.*

Calibre View Updates

No major updates were made for Calibre View in the 2021.2 release.

For more information, refer to “Creating a Calibre View” in the *Calibre Interactive User’s Manual*.

Chapter 9
Resolution Enhancement Technology (RET)

This chapter describes the 2021.2 changes and enhancements for the Calibre® RET applications.

See "Corrected Defects and Known Issues" on page 12 for instructions on accessing the complete list of corrected defects and known issues for the release.

RET Release Information

This section includes general release announcements for Calibre RET for the 2021.2 release.

Output Compatibility

It is our objective to keep the release output consistent from release to release. However, some releases will have differences due to changes in the default settings and algorithms that improve the functionality of the tools. These differences are documented here. It is the responsibility of each customer to use this information as applicable to his or her design methodology.

Note

OPC Results Differ Between AOI and AOJ Calibre Trees

Differences may be observed in the results from OPC jobs that are run using an AOI and an AOJ Calibre tree. The differences in results are due to the different hardware processors and operating system libraries that are supported by the different Calibre trees.

Table 9-1 describes any known output geometry differences between 2021.2 and the previous release.

Table 9-1. Output Compatibility 2021.2

Application	Differences That Can Exist
Calibre® nmOPC™	None.
Calibre® OPCverify™	None.
Calibre® OPCpro™	None.
Calibre® TDopc™	None.
Calibre® OPCsbar™	None.
Calibre® nmSRAF™	None.
Calibre® LFD™	None.
Calibre® LPE	None.
Calibre® pxOPC™	None.
Calibre® Multi-Patterning	None.

Deprecation and Obsoletion

This section describes any plans for future deprecation or obsoletion of existing functionality.

Planned Deprecation

Deprecated functionality should be removed from setup files before the functionality becomes obsolete. Obsolete functions can prevent your run from completing.

Table 9-2 lists all functionality for which support will be removed in a future release.

- Deprecated functionality may continue to operate, however it will no longer be supported by the Calibre team.
- All deprecated functionality that is currently documented may remain in the documentation, with a deprecation notice reminding customers that it is no longer recommended.
- Deprecated functionality may continue to exist for a subsequent release after first notification through the Calibre release notes or documentation. After that time it may become obsolete.

Table 9-2. Planned Deprecated Functionality

Keyword or Variable	Notification	Deprecation Release	Targeted Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
sraf_apa command (Calibre OPCverify)	2021.2	TBD	TBD	apa_check
applyProperty(*tag*) arguments (various commands)	2021.2	2021.2	TBD	property(*tag*)[:default(*value*)]
maximum_error_number (Calibre OPCverify)	2021.1	2021.1	2021.3	None
SITES_SHIFT (nmOPC)	2020.3	2020.3	TBD	SITES_CREATE -shift | -offset
FRAGMENT_MAX_DISPLACEMENT	2020.3	TBD	TBD	DISPLACEMENT -limit
FRAGMENT_SET_DISPLACEMENT	2020.3	TBD	TBD	DISPLACEMENT

Table 9-2. Planned Deprecated Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Targeted Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
pw_anchor, pw_select group_target, and pw_select group_anchor	2019.3	2019.3	TBD	metric_impact
job decoratedarksraf	2019.1	2019.2	TBD	job decorate with scatter_offset and metric_impact
collecttype (nmSRAF)	2018.4	2018.4	TBD	None
cleanupversion 0 (nmSRAF)	2018.4	2018.4	TBD	cleanupversion 1
mdf smo source {gensocs | delsocs} (in the RET Selection tool)	2017.3	2017.3	TBD	None. No longer required.
keep_bin (in Calibre OPCverify setlayer for classification blocks)	2017.1	2017.1	TBD	None. No longer required.
scatter_weight (Calibre pxOPC)	2016.4	2016.4	TBD	None. Not needed for tuning recipes.
Litho PSMgate	2016.3	2016.3	TBD	New development should use the Calibre Multi-Patterning family of products.
pwopc_mode	2016.2	2016.2	TBD	None.
imagegrid *max_grid_microns*	2016.2	2016.2	TBD	imagegrid aerial.
feedback_balance	2016.2	2016.2	TBD	None (algorithm 1 is deprecated for nmOPC only).
algorithm 0 and algorithm 1 for nmOPC only	2016.2	2016.2	TBD	algorithm 2 (default)
Calibre Multi-Patterning commands RET TP and RET QP	2016.1	2016.1	TBD	Use DFM MP instead.

Table 9-2. Planned Deprecated Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Targeted Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
OPC_MIN_EXTERNAL OPC_MIN_INTERNAL opcMinCheckTag OPC_DO_MORPH	2015.4	2015.4	TBD	MRC_RULE
preclean	2015.3	2015.3	TBD	Pre-cleaning is no longer required for dense OPC.
JOG_SMOOTH	2015.2	2015.2	TBD	Older method of tag-based scripting (<script> ...<endscript>) is obsolete.
SET_COST_TOLERANC E	2013.2	2013.2	TBD	cost_tolerance keyword was deprecated in 2013.1.
le_pb and se_pb (pwopc pw_condition option)	2013.2	2013.2	TBD	Tolerance-based PWOPC no longer supported.
corner_ignore (pwopc pw_condition option)	2013.2	2013.2	TBD	Tolerance-based PWOPC no longer supported.
keep_middle (pwopc pw_condition option)	2013.2	2013.2	TBD	Tolerance-based PWOPC no longer supported.
sraf_adjust	2013.1	2013.2	TBD	Keyword will be replaced by future functionality.
tol (pwopc pw_condition option)	2013.1	2013.1	TBD	Tolerance-based PWOPC no longer supported.
weight (pwopc pw_condition option)	2013.1	2013.1	TBD	Weight-based PWOPC no longer supported.

Table 9-2. Planned Deprecated Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Targeted Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
weight (image command option)	2013.1	2013.1	TBD	Weight-based PWOPC no longer supported.
cost_tolerance	2013.1	2013.1	TBD	Command no longer required.
allow_delete argument in sraf_print_avoidance	2012.2	2012.2	TBD	Option no longer required.
map_blocked_epes	2011.3	2011.3	TBD	Option no longer required.
max_epe_variance	2011.3	2011.3	TBD	Option no longer required.
modifiedTarget argument in opcIter and pwOpcIter	2011.3	2011.3	TBD	targetLayer argument in both.
opc_grid_multiplier 10 opc_grid_multiplier 100	2011.1	2011.1	TBD	The command will be "on" by default. No other options will be required (aside from "off").
script_variable	2010.3	2010.3	TBD	Not required.
script_output_layer	2010.3	2010.3	TBD	Not required.
exec keyword not required anymore in jogsmooth, pwopcIter, or sitemodify	2010.2	2010.2	TBD	Not required.
exec keyword not required anymore in opcIter	2008.3	2008.3	TBD	Not required.
VT5 Model Support in Calibre nmOPC	2007.4	2007.4	TBD	CM1 Model

Obsolete Keywords

Obsolete keywords and variables may cause a run to halt with an error. The functionality once referred to by the keyword or variable has been removed from Calibre, which means even if the run completes the simulation and optimization may not have occurred as designed.

Table 9-3 lists all previous functionality that has been disabled or removed from the release.

- Obsoletion will occur no sooner than six months after first deprecation of the functionality.
- The functionality description will be removed from documentation and will no longer be supported by the Calibre team.

Table 9-3. Obsolete Functionality

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
SVRF Optimizer	2020.4	2020.4	2021.2	None
mode train (SONR)	2021.1	2021.1	2021.1	sonr --tree or sonr --cluster
-outCSV and -outmod in SONR_COLLECT	2021.1	2021.1	2021.1	None
fragalign, fragcoinc (nmOPC)	2020.3	2020.3	2021.1	Not needed
-layer argument in target_curve	2020.4	2020.4	2020.4	None
debugReducedTarget (Multi-patterning)	2020.4	2020.4	2020.4	effectiveTarget
DFM Spatial Sample	2020.1	2020.1	2020.3	None
mdf smo anchor (in the RET Selection tool)	2020.2	2020.2	2020.2	None
min_sampling_distance (SONR)	2020.2	2020.2	2020.2	None. This is now set by Calibre Sonr and cannot be changed manually.
rect45corner orient options (nmSRAF)	2020.2	2020.2	2020.2	None
optimizeCutCandidateLineEnds (Multi-Patterning)	2020.2	2020.2	2020.2	optimizeLineEnds
min_sampling_distance (Vector Capture)	2020.2	2020.2	2020.2	Not needed; now automatic
criticalNets (Multi-Patterning)	2019.3	2019.3	2020.1	criticalTimingMarkers
DBCLASSIFY	2015.4	2016.1	2019.3	classify_options in Calibre OPCverify

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
Calibre SVRF Analyze	2018.2	2018.2	2019.2	None.
controller	2010.4	2010.4	2019.2	None.
Double-Dipole Lithography OPC (ddlOpcIter, ddlfragalign, ddlGrowRule, ddlParams, ddlTag commands)	2015.2	2015.2	2019.2	None.
-flat in Calibre invocation (Distributed Flat platform)	2018.1	2018.1	2019.1	Calibre FullScale platform
hha_sectorization auto in Optical Parameters File	2018.1	2018.1	2018.4	hha_sectorization energy
rxopc	2015.4	2015.4	2018.4	Calibre LPE
-lft in calibrewb invocation	2014.3	2015.2	2018.4	Use RET Flow Tool (on by default) instead.
approximate_context (in Calibre OPCverify setlayer for classification blocks)	2017.1	2017.1	2017.2	None. No longer required
approximate_stopping (in Calibre OPCverify setlayer for classification blocks)	2017.1	2017.1	2017.2	None. No longer required
peaktype 1peak, 2peak and 3peak	2017.2	2017.2	2017.4	None.
target_curve curveOnly 0	2017.4	2017.4	2017.4	target_curve curveOnly 2 (the new default)
fix_limited_preopc_mrc_violations	2016.2	2016.2	2017.2	None.

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
Calibre Multi-Patterning keywords: COMPLEXITY_CHECK_ COMBINED COMPLEXITY_CHECK_ POLYGONS COMPLEXITY_CHECK_ SEPARATORS	2016.3	2016.3	2017.1	None.
Calibre nmSRAF templates using ordered values instead of keywords	2015.4	2015.4	2017.1	Use the standard keyword naming conventions.
Calibre Multi-Patterning effort keyword	2016.2	2016.2	2016.4	None.
plane argument in Calibre nmOPC image	2017.4	2016.4	2016.4	pw_condition or setlayer image
Calibre OPCverify anchored contouring	2016.1	2016.1	2016.3	None.
Calibre WORKbench CM1 modelform 12	2016.1	2016.3	2016.3	None.
Calibre nmSRAF dynamicsizing BY keyword	2015.4	2015.4	2016.2	None.
OUTPUT_SHAPE box	2017.1	N/A	2016.2	None.
Calibre WORKbench VEB models with negative exponents in BTERMs	2016.1	2016.1	2016.1	None.
RET PIXBAR	2015.3	2015.3	2016.1	Calibre pxOPC
RET SMO	2015.2	2015.2	2015.4	Calibre pxSMO
dynamicsize_version	2015.2	2015.2	2015.4	No longer needed.
mbsraf modes: 45only, all, horizontal, no45 and vertical	2015.2	2015.2	2015.4	Use remaining mbsraf modes
Flare layers for long range EUV flare	2014.4	2014.4	2015.3	PNG files

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
PV Band tool in Calibre WORKbench	2015.2	2015.2	2015.2	PV band simulation can be performed in the RET Flow Tool in Calibre WORKbench.
Parallel linear models (PLM)	2014.3	2014.4	2015.2	CM1 or VT5 models as appropriate.
<script> ... <tagscript>	2015.2	2015.2	2015.2	Use current Calibre nmOPC Tcl scripting commands.
move_allangle_edges	2011.3	2011.3	2015.2	Option no longer required.
dpStitchMode violationPaths	2014.3	2014.3	2015.1	Use targetNodePairs or cycleAware modes instead.
conflictMode output layer	2014.3	2014.3	2015.1	None. Used only with dpStitchMode violationPaths.
cleanmode postconflict	2014.2	2014.2	2014.4	Use cleanmode postsizing instead.
Adaptive OPC (adaptive_opc, FRAGMENT fix, FRAGMENT unfix)	2014.2	2014.2	2014.3	Use rule-based fragmentation instead.
nmDPC output all conflicts keyword	2014.2	2014.2	2014.3	All conflicts are output by default.
nmDPC progress indicator keyword	2014.2	2014.2	2014.3	None required.

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
LITHO commands in Embedded SVRF blocks (<SVRFSTART> and <SVRFEND>)	2014.2	2014.2	2014.2	LITHO commands should be defined outside of the <SVRFSTART> and <SVRFEND> block.
num_flare_layers {layer_name...} keywords for flare_model command	2014.2	2014.2	2014.2	Long-range flare models are associated by the longrange keyword of the flare_model_load command and flare_longrange for Litho models.
nmDPC verbose keyword	2014.1	2014.1	2014.2	None.
RET SRAF_FILL and cnsraf fill keywords	2013.4	2013.4	2014.2	RET MBSRAF
RET NMDP	2013.2	2013.4	2014.1	Use action :stitch with RET NMDPC instead.
Calibre pxOPC lmrc_negative_cuts_enabled	2016.4	N/A	2014.1	lmrc_inside_assist_cuts_enabled
pitch (nmDPC keyword)	2013.2	2013.4	2014.1	Use action instead.
MAP violated MAP violated2 (RET NMDPC)	2013.2	2013.4	2014.1	Use SVRF instead. See Calibre Multi-Patterning manual.
LITHO_NEW_TRANSCRIPT	2013.2	2013.3	2014.1	No replacement.
nmSRAF ring	2013.1	2013.1	2013.4	Template no longer supported.
nmSRAF ringedge	2013.1	2013.1	2013.4	Template no longer supported.

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
reopc	2013.2	2013.3	2013.3	Keyword has been replaced by the rxopc command.
symmetry_type	2013.1	2013.1	2013.3	Use clip_layer instead.
costFunction	2013.1	2013.1	2013.2	Keyword no longer required.
faceCosts	2013.1	2013.1	2013.2	Keyword no longer required.
zeroCostFaces	2013.1	2013.1	2013.2	Keyword no longer required.
zeroCostRegion	2013.1	2013.1	2013.2	Keyword no longer required.
maxStitchesPerPolygon in dpStitchCandidates	2012.3	2012.3	2013.1	Option no longer required.
depthMarkers in dpStitchCandidates	2012.3	2012.3	2013.1	Option no longer required.
concurrent_simulation_measurement	2013.1	2013.1	2013.1	Command no longer required due to obsoletion of Calibre nmOPC CPA.
Calibre nmOPC CPA (Co-Processor Acceleration)	2012.4	2012.4	2013.1	Functionality no longer supported.
dpCost	2012.2	2012.2	2012.4	Use dpStitchCandidates instead.
dpStitch	2012.2	2012.2	2012.4	Use dpStitchCandidates instead.
VIALAYER	2012.2	2012.2	2012.4	Use other clean up control keywords.
OPCsbar V1 processing mode	2012.2	2012.2	2012.4	Default processing mode is V2.

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
map_jog_epes	2011.4	2011.4	2012.2	Option no longer required.
mrc_mode 0	2010.4	2010.4	2011.3	Option no longer required. “mrc_mode 1” is already default.
consistency_postprocess	2011.2	2011.2	2011.3	This command is no longer helpful for improved consistency. Follow all consistency best practices instead.
USE_OLD_NEG_CLEANUP	2010.4	2010.4	2011.1	Option no longer required. A new negative cleanup is the default.
OPCSBAR_V2_CAREFUL_CLEANUP	2010.1	2010.3	2010.3	CAREFUL_CLEANUP
FRAGMENT_GET_PRESET_DISPLACEMENT	2009.4	2009.4	2010.1	Not required.
SITE_ADD	2009.3	2009.3	2009.4	All site functionality in the nmOPC Tcl custom scripting capability will be replaced by new SITES commands to be released in 2009.3 Update Release 1.
SITE_PREPARE	2009.3	2009.3	2009.4	
SITE_ITERATOR	2009.3	2009.3	2009.4	
SITE_GET	2009.3	2009.3	2009.4	
mrc_violation_free_input	2009.3	2009.4	2009.4	Not required.
LITHO_NEW_OPC = 0	2008.2	2008.2	2008.4	
Verification Center™ (in Calibre OPCverify)	2008.4	2009.1	none	None

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
LITHO_AERIAL_CONTOUR			2008	Migrate to OPCverify.
FRAGMENT_FINALIZE (nmOPC)	2007.3	2007.4	2007.4	n/a
LITHO_DENSE_PRINT	2007.2	2007.2	2007.3	Migrate to OPCverify. Use the setlayer image command to generate dense contours.
maxcp (in setlayer curve)	2007.2	2007.2	2007.2	none (handled internally)
-simpixsz (in modelflow_v2)	2007.2	2007.2	2007.2	none (handled internally)
-keepbest (in modelflow_v2)	2007.2	2007.2	2007.2	none (default behavior)
-genmode (in modelflow_v2)	2007.2	2007.2	2007.2	none (handled internally)
passfilter | excludefilter (in modelflow_v2)	2007.2	2007.2	2007.2	none (intended for internal use only)
timer (OPCverify setup variable)	2007.1	2007.1	2007.2	none (on by default)
SPLAT	2005.1	2005.1	2005.3	Migrate to TCCCalc. SPLAT will be read, but will not support generation of SPLAT models
vtre model	2005.1	2005.1	2005.3	Migrate to VT5. Vtre models will always be read, but will not support generation of VTRE

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
Inline setup file -inline, as part of the LITHO operation in the SVRF file without using LITHO FILE	2005.1	2005.1	2005.2	n/a
OPC_MIN_EXTERNAL_LE2LE	2004.4	2004.4	2005.2	setMoveLimitForTag
OPC_MIN_EXTERNAL_CX2LE	2004.4	2004.4	2005.2	setMoveLimitForTag
OPC_MIN_EXTERNAL_CX2CX	2004.4	2004.4	2005.2	setMoveLimitForTag
DYNAMIC_RULE_TABLE_SIZE	2004.4	2004.4	2005.2	n/a
non-integer kernel grids for optical model	2004.4	2004.4	2005.2	Use integer multiples of database unit in optical grids
aspect	2004.4	2004.4	2005.2	Use tagging for constraints
VT5 version 2	2004.4	2004.4	2005.2	Migrate to VT5 version 3
minedgelength enforce	2004.4	2004.4	2005.2	Use OPC_MIN_INTERNAL, opcMinCheckTag
FRAG_AT_EXTERNAL_LAYERS	2004.4	2004.4	2005.2	Use island layers
sample points in “samfile” format (for option -s) -s option in: samlinev2 samlinev3 cdsample cdruler	2004.4	2004.4	2005.2	Use spreadsheet format (option -ss)

Table 9-3. Obsolete Functionality (cont.)

Keyword or Variable	Notification	Deprecation Release	Obsoletion Release	Recommended Setting / Replacement Keyword or Variable
vt5crosscheck	2004.4	2004.4	2005.2	Use -xcheck option to vt5gen command instead

Default Changes

Calibre RET tracks changes to defaults for the prior 10 years. Although changes to defaults are only made after extensive testing proves the change is an improvement, you can usually retain prior behavior by explicitly setting the keyword or variable to the value shown in the Previous Default column.

Table 9-4. Default Changes for Calibre RET

Keyword or Variable	Previous Default	New Default	Version
metric in mrc_rule and MRC_RULE (Calibre nmOPC)	No default	Euclidean	2021.2
Reason for Change: "Euclidean" has been the recommended value for many years, and not having to specify the metric improves ease of use.			
-epochs for mdf ml optimize_n2e	8000	6000	2020.4
Reason for Change: Best practice update for N2E only. N2R default remains the same.			
Optical Model version	12	13	2020.4
Reason for Change: Adds internal symmetry detection that recognizes sourcemaps that are only symmetric in the X or Y directions.			
Stochastic Model version	1	2	2020.4
Reason for Change: Supports stochastic failure rate calculations in Calibre nmModelflow and random contour generation in Calibre OPCverify.			
gauge_reduce -method	diverse	coverage	2019.4
Reason for Change: Best representation of input gauge set.			
mdf gauges create	No lower limit for gauge length	minimum of 3*simulation pixel size	2019.4
Reason for Change: Add a default.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
mdf optimize set_resist	maxaidiff 10 stable	maxaidiff 10 stable anchor	2019.4
Reason for Change: Defaults will work for all modelforms.			
CLASSIFY_FRAGMENTS -compresskey	0	1	2019.3
Reason for Change: Reduce memory.			
Global Lithomodel Creation Wizard, default slits provided	7	13	2019.2
Reason for Change: Updates for current best practices.			
CM1 COEFMAX default starting value	none	0.00012. (1.2e^{-4})	2019.2
Reason for Change: Updated to reflect recommended values. Models calibrated in this release are automatically updated to this value.			
CM1 resist model curvature diffusion length coefficients (CURVATURE_D*n* and CURVATURE_S*n*)	• search interval lower limit: 3*pixel size • recommended upper bound: 50nm • nominal value: 30nm	• DUV lower limit: 6*pixel size if the number of curvature cross terms is not equal to zero (EUV and DUV with zero curvature cross terms is 3*pixel size) • upper bound: 100nm • nominal value: 45nm	2019.2
Reason for Change: Updates reflect current calibration best practices.			
pvband defaults gauges add_properties ldof operation	rectangle and model_poly8	ellipse and model_poly13	2019.2
Reason for Change: Update to defaults.			
DDM Model Format version	4	5	2019.1
Reason for Change: Improves corner signal generation time for EUV libraries and improves accuracy for mask simulation.			
CM1 Model version	9	10	2018.4
Reason for Change: Improves the normalization of symmetrization type -1 dissolution kernels.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
Optical Model version	11	12	2018.4
Reason for Change: Improves optical model accuracy over version 11 for EUV.			
DDM Model Format version	3	4	2018.4
Reason for Change: Improves EUV modeling accuracy and is independent of object plane.			
hha_sectorization -tight	None	1	2018.4
Reason for Change: This new keyword produces HHA sectors with the tightest possible bounds around the source map for each sector. To restore prior behavior, specify -tight 0.			
hha_sectorization -strict_center	None	1	2018.4
Reason for Change: This new keyword creates a central HHA sector only if there is a dip in the radial intensity. This is now the default behavior for version 12 optical models but not earlier versions. To restore prior behavior, specify -strict_center 0.			
minoffset long2lineend	projecting	lineend	2018.4
Reason for Change: The minoffset long2lineend value defaulted to projecting values and not lineend values as expected.			
minfeaturespace long2lineend	projecting	lineend	2018.4
Reason for Change: The minfeaturespace long2lineend value defaulted to projecting values and not lineend values as expected.			
cleanupversion	0	1	2018.4
Reason for Change: Siemens EDA recommends always setting cleanupversion 1 to achieve optimum SRAF generation. The cleanupversion 0 setting has been deprecated and is to be obsoleted in release 2019.2.			
pixel_alignment_v2	off	on	2018.4
Reason for Change: Improved consistency between full-chip runs and layout clips for setlayer operations run in litho flat mode.			
algorithm (RET MBSRAF)	0	None	2018.1
Reason for Change: Siemens EDA recommends always setting algorithm 1 to achieve optimum SRAF generation. Setting this keyword to 0 restores the previous behavior.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
cleanupversion	0	None	2018.1
Reason for Change: Siemens EDA recommends always setting cleanupversion 1 to achieve optimum SRAF generation. Setting this keyword to 0 restores the previous behavior.			
hha_sectorization	auto	energy	2018.1
Reason for Change: The default auto is deprecated as of the 2018.1 release.			
maxmove	0	Using algorithm 0, the default is width/2. Using algorithm 1, the default is 1nm.	2018.1
Reason for Change: Provides users direct control over the degree of movement for SRAFs undergoing cleanup.			
Optical Model version	10	11	2017.4
Reason for Change: Adds additional information to the optical model directory to improve accuracy and consistency of some DDM models with no performance penalty if the DDM library version is greater than or equal to 3.			
CM1 Model Format version	8	9	2017.3
Reason for Change: Improvements in contour consistency for shrink and sidewall bias terms by changing the intensity and threshold characteristic functions slightly.			
enable_default_checks (pxOPC)	no (off)	yes (on)	2017.3
Reason for Change: Saves time in developing Calibre pxOPC recipes. It can be turned off for production.			
flare_longrange	No default. Command had to be specified if flare models existed in a litho model.	**flare_longrange *litho_model* none**	2017.2
Reason for Change: Improved usability.			
CM1 Model Format version	7	8	2017.2
Reason for Change: Improvements for contour consistency in shrink and sidewall operations for NTD-based CM1 models. PTD-based model users do not need to regenerate.			
iterations (pxOPC Open job)	4	10	2017.2
Reason for Change: In most cases, four iterations was not enough for the Open job to converge, especially affection SRAF insertion.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
job 1 start (SMO)	iterations 20	iterations 50	2017.1
Reason for change: Extensive testing shows significant quality improvement.			
modelflow_v2 biasing_type	1	0	2016.4
Reason for Change: Initial calibration speed improvements.			
(RET Selection and pxSMO) CM1 vs Aerial Image	CM1 Adjust	Image Model: Aerial	2016.4
Reason for Change: Formerly, CM1 Adjust and Use Aerial Image were separate controls and not prevented from being both selected. In 2016.1 and 2016.2, CM1 Adjust was on by default. Now the two possibilities are grouped under Image Model so you can only select one, and by default it is Aerial. Sessions imported from previous databases do not have the new default.			
sraf_print_avoidance -max_pair_dist	2*Nyquist	2.5*Nyquist	2016.4
Reason for Change: The former default did not allow pairing for EUV SRAFs.			
modelflow_v2 fasttopo	1	2	2016.3
Reason for Change: Faster model calibration. Note that several topo calibration scripts may require modification, because fasttopo 0 and 1 use different calibration parameters than fasttopo 2.			
mbSRAF shielding	no	seedpixel	2016.3
Reason for Change: Results in better quality SRAFs.			
OUTPUT_RETARGET	write to layers of type opc	write to layers of type target if they exist; opc if not.	2016.3
Reason for Change: Version 2016.3 introduces the layer type "target" for Calibre nmOPC. It is intended specifically for use in retargeting. Rule files without target layers will work as before.			
OPC_ITERATION min_space_pw min_space_interpattern priority change	min_space_pw	min_space_interpattern	2016.2
Reason for Change: Spacing between masks should be controlled by min_space_interpattern when it is specified. If min_space_interpattern is not specified, it defaults to min_space. For the same results in 2016.2 and forward as 2015.2 through 2016.1, set min_space_interpattern to the same value as min_space_pw.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
refine_visible_sim	no	yes	2016.2
Reason for Change: Better results for common use case of overlapped repair regions.			
algorithm	0	2	2016.2
Reason for Change: Both algorithm 0 and 1 are deprecated as of the 2016.2 release.			
cnsrafpromotion	0	1	2016.2
Reason for Change: Improved SRAF consistency.			
controller	on	off	2016.1
Reason for Change: Improved OPC consistency, replaced by using Matrix OPC instead.			
shielding	no	seedpixel	2016.1
Reason for Change: Improved SRAF consistency.			
fragment_inter -distancePriority	0	1	2015.4
Reason for Change: Improved output consistency.			
LITHO_CONTEXT_RELAX_MRC	0	Number of iterations - 2	2015.3
Reason for Change: The best practice has been to set this variable to two less than the number of iterations. This change prevents overlooking the best practice.			
LITHO_PRIORITY_BASED_MOVEMENT	OFF	ON	2015.3
Reason for Change: The best practice has been to set this variable to ON. This change prevents overlooking the practice.			
freeze_skew_edges	is_error	on and no_error	2015.2
Reason for Change: This prevents cases where Calibre exited an OPC run with an error and no results were produced.			
setlayer mbsraf slopeweight	If you use DDM/HHA models, mbsraf does not use narrowweight, and slopeweight and verticalweight must be explicitly specified.	See the section entitled *Calibre nmSRAF Previous Release Highlights* in the 2015.4 Release Notes.	2015.2
Reason for Change: Enables slopeweight for DDM/HHA models.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
refine_distance0	1 micron	optical radius	2015.1
refine_distance1	10*min_frag_len	optical radius	
refine_distance2	1 micron	optical radius+0.2 microns	
Reason for Change: Basing the region sizes on optical radius allows more sensitivity to process conditions and produces better results.			
NEWTAG external/ internal	NA	-shielding_check simple	2014.3
Reason for Change: Provides a default tag-level shielding check when performing NEWTAG internal/external/enclose/enclosing operations. The new default preserves the current behavior of the NEWTAG external and NEWTAG internal commands. Changes in output are not expected in all single layer Calibre nmOPC applications (non-MPOPC). Tagging is expected to be correct in the multi-layer cases (Calibre mpOPC and Calibre nmBIAS).			
iterations in pxOPC when invoked by LPE	4, 6, 8, 10, 10	10, 6, 16, 10, 24	2014.3
mrc_iterations in pxOPC when invoked by LPE	30	0	2014.3
Reason for Change: Because Calibre LPE is typically used on layouts that have already been through OPC, more iterations are necessary to fix hotspots without disrupting optimized regions.			
sraf_print_avoidance	-rectangle	-nonrectangle	2014.1
Reason for Change: To improve sraf_print_avoidance results at hierarchical boundaries, -nonrectangle is enabled by default. To restore the previous behavior, use -rectangle.			
sraf_print_avoidance -allow_cut	false	true	2014.1
Reason for Change: To improve sraf_print_avoidance results, -allow_cut is enabled by default, which means that SRAFs are cut if they are at the MRC limits. To restore the previous behavior, use the -disallow_cut keyword.			
CM1 model "version"	5	7	2014.1
Reason for Change: The new version (7) improves simulation quality. CM1 models produced in version 2014.1 and later produce version 7 models by default.			
freeze_skew_edges	on	is_error	2014.1
Reason for Change: Setting freeze_skew_edges to is_error instructs Calibre nmOPC to specify skew edges as an error condition and exit the program. This is to prevent bad OPC results due to skew edges.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
minoffsetlayerspace offsetlayer	In releases prior to 2013.4, all SRAFs were kept at a spacing equal to minfeaturespace from the offsetlayer.	The new default allows SRAFs to abut the offsetlayer. To model previous SRAF-to-offsetlayer spacing behavior, use minoffsetlayerspace to enforce the required space.	2013.4
Reason for Change: This new default allows greater flexibility for SRAF treatment, extending contour fidelity.			
assist_inside_dose_multiplier assist_outside_dose_multiplier	assist_inside_threshold_multiplier assist_outside_threshold_multiplier	Keyword now uses aerial image.	2013.4
Reason for Change: The new behavior (using aerial image instead of print image for scaling) provides better results.			
-enforce parameter in fragment_island	0	1	2013.3
Reason for Change: The new default setting provides more stable results for fragment_island operations.			
jog_cleanup_dist parameter in setlayer curve if left unset	NA	0.5 microns	2013.3
Reason for Change: The new default setting provides more stable results when running setlayer curve. The 0.5 setting (the maximum possible) yields results that are closest to the previous infinite setting.			
job refine iteration	4	6	2013.2
Reason for Change: Improved consistency for main features.			
setlayer mbsraf peaktype	4Dir	3Dir	2012.3
Reason for Change: The new default setting, 3Dir, is more stable along lines than 4Dir, and produces a more consistent collection of seeds in many environments than 4Dir.			
mrc_mode	1	2	2012.3
Reason for Change: This option enables a post-movement approach that improves the output of the existing movement code. It does not change the functionality of the existing movement code.			

Table 9-4. Default Changes for Calibre RET (cont.)

Keyword or Variable	Previous Default	New Default	Version
setlayer mbsraf slopeweight	None	slopeweight 1	2012.2
Reason for Change: Improved SRAF consistency.			
OPTION FAST_MRC	NO	YES	2011.4
Reason for Change: Enabling this option provides better performance.			
LITHO_SIMULATE_CONTEXT	0	1	2011.3
LITHO_SIMULATE_CONTEXT_DISTANCE	0	0.25	2011.3
Reason for Change: Gives better results and improved hierarchical consistency for most customer designs.			

Global Changes Affecting Post-Tapeout Products

This section covers changes that affect multiple post-tapeout products.

Utility for Identifying Base Layers

The Calibre tree now includes a script that analyzes SVRF or TVF rule files that contain at least one LITHO operation to suggest an optimal set of base layers. The syntax is

```
$CALIBRE_HOME/bin/calflowoptimize input_file [-o output_file]
```

The input file has some restrictions:

- It must not contain undefined variables, such as might normally be set at run time by a shell script.
- It must not contain any obsolete keywords or pre-production features.

The default output file is *dummy.svrf*. In addition to the LAYOUT BASE LAYER statement, it contains the equivalent SVRF of the input file.

The script is included in the discussion “Optimal Performance Using LAYOUT BASE LAYER” in the *Calibre Post-Tapeout Flow User’s Manual*.

Calibre Directed Self-Assembly (DSA)

There were no changes or enhancements to Calibre® DSA for the 2021.2 release.

Detailed product information can be found in the *Calibre Directed Self-Assembly User's and Reference Manual*.

Calibre Local Printability Enhancement

There were no changes or enhancements to Calibre® LPE for the 2021.2 release.

For complete product information, see the *Calibre Local Printability Enhancement User's and Reference Manual*.

Best Practice Updates for Calibre LPE

The table summarizes changes and updates to the Best Practices for Calibre LPE.

Table 9-5. Calibre LPE Best Practice Updates

Best Practice	New or Update?
2019.2	
Removed references to the Calibre nmOPC controller keyword. **Reason**: The controller keyword is obsolete.	Update
2017.1	
The denseopc_context_parameters setting recommendation for the zone nearest the boundary between the visible and context region should be "displacement limit off feedback off". **Reason:** This setting results in fewer XOR differences across the boundary than the former recommendation.	Update
For designs that require significant amounts of memory on the CPU, the run can use hyperscaling provided the remote configuration file includes "HYPER SMTFACTOR 0" to disable multithreading on remote CPUs. **Reason:** Additional experiments after the 2016.4 release showed that hyperscaling without multithreading used acceptable levels of local memory.	Update
2016.4	
For designs that require significant amounts of memory on the CPU, such as triple patterning or EUV runs, make use of SOCS shared memory (set LITHO_SOCS_KERNELS_SHARED to TRUE) and avoid hyperscaling and SMT (set CALIBRE_MT_SMT_FACTOR to 0). **Reason:** These settings provide the most memory per CPU on remote hosts. When distributed processes do not have enough local memory, runs can hang or even fail.	New

Table 9-5. Calibre LPE Best Practice Updates (cont.)

Best Practice	New or Update?
2016.1	
Use the script generator (calibre -lpe) to create setup files for refine_exec … denseopc operations. **Reason**: Calibre LPE requires many modifications to Calibre nmOPC setup files for best operation. The script generator reduces both errors and manual effort.	New
2015.4	
For Calibre OPCverify setup files that generate marker layers with commands that include a classification block, the classification block should include “keep_no_context duplicates”. **Reason:** By default, classification suppresses duplicates. If the classification block does not include “keep_no_context” the marker layer does not indicate all hotspots.	New
2015.2	
For runs using Calibre nmOPC (dense OPC), use the following command settings: • denseopc_copy_fragments yes • denseopc_insert_correction yes • denseopc_limit_excess yes • denseopc_smooth_corrections yes • denseopc_context_parameters displacement_limit off feedback auto displacement_limit -0.012 0.012 feedback -keep -0.015 displacement_limit -0.007 0.007 feedback -keep -0.07 displacement_limit -0.003 0.003 feedback -keep -0.03 displacement_limit -0.001 0.001 feedback -keep -0.01 • (Calibre nmOPC setup file) controller off **Reason:** Provides smoother correction between the repair region and visible region for re-OPC with Calibre nmOPC.	New
2015.1	
Removed recommendations to size the regions by optical radius. **Reason:** The recommended sizes are now the default.	Update
2014.4	
The size of the core region should be approximately the size of the optical diameter. **Reason:** Better quality of results.	New

Table 9-5. Calibre LPE Best Practice Updates (cont.)

Best Practice	New or Update?
2014.3	
Formerly the best practices advised to use "pxopc_init_mode target". This advice is now regarded as not always a best practice. **Reason:** Based on new test cases, we are re-evaluating the general advice. If you have a working recipe with pxopc_init_mode target, you do not need to change it.	Update
Formerly the best practices advised against 45-degree edges. This advice is now obsolete. **Reason:** The software has improved and now handles 45-degree edges.	Update
The size of the context region (also known as the transition region) should be approximately the size of the optical radius. **Reason:** Better merging of repairs into the existing layout.	New
The size of the visible region should be 0.2 microns larger than the optical radius. **Reason:** The increased interaction distance improves simulation accuracy.	New
2014.2	
The value of mrc_small_feature_size should be smaller than mrc_min_square. **Reason:** When this is violated, square SRAFs that are acceptable to the foundry are removed.	New
Use "pxopc_init_mode target". **Reason:** Calibre pxOPC produces better results when starting from the designer's intended shapes. The default is for backwards compatibility.	New

Calibre Multi-Patterning

This section includes information regarding changes and enhancements to Calibre® Multi-Patterning for the 2021.2 release.

For more information, refer to the *Calibre Multi-Patterning User's and Reference Manual*.

Calibre Multi-Patterning Updates

This section covers changes affecting Calibre Multi-Patterning products for this release.

New or Changed Commands and Keywords

There are no new or changed commands for this release.

Deprecated and Obsoleted Commands

- **debugReducedTarget —** This MAP keyword is obsoleted as of the 2020.4 Calibre release. Use the MAP keyword effectiveTarget instead.
- **optimizeCutCandidateLineEnds —** This parameter is obsoleted as of the 2020.2 Calibre release. Use the optimizeLineEnds parameter instead.
- **trackSpacing —** This parameter is obsoleted as of the 2020.4 Calibre release. Use the sidewallWidth parameter instead.
- **criticalNets —** This layer and keyword are obsolete as of the 2020.1 Calibre release. Use the criticalTimingMarkers layer and keyword instead.
- **RET TP and RET QP —** These commands are deprecated as of the 2016.1 Calibre release and are to be obsoleted in a future Calibre release. Siemens EDA recommends migrating these commands to DFM MP.

Best Practice Updates for Calibre Multi-Patterning

Calibre Multi-Patterning has several documented best practices.

Table 9-6 summarizes the changes and updates to the Best Practices for Calibre Multi-Patterning.

Table 9-6. Calibre Multi-Patterning Best Practice Updates

Best Practice	New or Update?
2020.1	
Use sidewallWidth for SADP and SAQP processes instead of trackSpacing going forward. **Reason:** trackSpacing is deprecated and is to be obsoleted in release 2020.4.	New
2018.4	
Use DFM DP for all double-patterning applications going forward. **Reason:** RET NMDPC is superseded and is to be deprecated in a future release.	New
2016.4	
Siemens EDA recommends users develop phase assignment decomposition flows using the Calibre Multi-Patterning family of products. **Reason:** Extended processing efficiency.	New
2016.1	
Use DFM MP for all triple- and quadruple-patterning applications going forward. **Reason:** RET TP and RET QP are deprecated.	New
2012.2	
A method of polygon to edge conversion for separators is demonstrated for users wishing to use separators generated by dpStitchCandidates. **Reason:** Facilitates derivation of separator layers for decomposition.	New
A table of technology node, hardware requirements and the respective tilemicrons settings were added. **Reason:** Optimizes implementation.	Update

Calibre Multi-Patterning Previous Release Highlights

Significant changes from previous releases are summarized here.

Commands and Keywords

- **RET DPSTITCH —** The RET DPSTITCH syntax is now simplified as of the 2021.1 Calibre release.

- **debugReducedTarget —** This MAP keyword is obsoleted as of the 2020.4 Calibre release. Use the MAP keyword effectiveTarget instead.
- **optimizeCutCandidateLineEnds —** This parameter is obsoleted as of the 2020.2 Calibre release. Use the optimizeLineEnds parameter instead.
- **trackSpacing —** This parameter is obsolete as of the 2020.4 Calibre release. Use the sidewallWidth parameter instead.
- **criticalNets —** This layer and keyword is obsolete as of the 2020.1 Calibre release. Use the criticalTimingMarkers layer and keyword instead.
- **RET TP and RET QP —** These commands are deprecated as of the 2016.1 Calibre release and are to be obsoleted in a future Calibre release. Siemens EDA recommends migrating these commands to DFM MP.
- **effectiveTarget —** The MAP keyword, debugReducedTarget, is renamed to effectiveTarget. The MAP keyword debugReducedTarget is obsoleted. Renamed for release 2020.4.
- **condition(numNeighbors value [,optional]) —** The existing option to condition, numNeighbors, was simplified: its operator was removed and the value now represents the minimum number of neighboring shapes. Changed for release 2020.3.
- **maskGridSpacing —** The maskGridSpacing parameter was simplified, together with line-end extension enhancements. The previous maskGridSpacing implementation referenced line-ends associated with each cut for their grid alignment. This led to inconsistent results if the original target was not aligned to grid. A number of improvements were made for release 2020.3:
    - Target line-ends are now extended as required to put line-ends on grid.
    - Line-ends extended to match grid are now included in a new debug layer, PitchAlignTargetExtensions, used to debug pitch align operations.
    - The debugReducedTarget layer now includes target line-end extensions.
    - The illegalTargetGap layer now shows locations where the grid cannot be matched between two line-ends.
- **PitchAlignTargetExtensions —** A new debug keyword, PitchAlignTargetExtensions, outputs polygons that extend target line-ends to meet requirements of maskGridSpacing or PitchAlign restrictions. Released in 2020.3.
- **optimizeLineEnds —** A new SADP LITHO FILE parameter, optimizeLineEnds, was added to better administer the RET SIDCUTFILL line-end optimization algorithm. This parameter has replaced optimizeCutCandidateLineEnds, which is obsoleted. Released in 2020.2.

Calibre CellArray RET

This section includes information regarding changes and enhancements to Calibre® CellArray RET for the 2021.2 release.

For more information, refer to the *Calibre CellArray RET User's and Reference Manual*.

Calibre CellArray RET Updates

This section covers changes affecting Calibre® CellArray RET products for this release.

New Commands and Functionality

There are no new commands for this release.

Calibre CellArray RET Previous Release Highlights

Significant changes from the previous four releases are summarized here.

Commands and Keywords

There are no previous command updates.

Calibre nmOPC

This section includes information on changes and enhancements to Calibre® nmOPC™ for the 2021.2 release.

For more information, refer to the *Calibre nmOPC User’s and Reference Manual*.

Calibre nmOPC Updates

The following is a list of Calibre nmOPC new and updated features this release.

Support for Curvilinear Layouts

The 2021.2 release includes a new product, Calibre® nmCLOPC, to perform OPC on curved layouts such as are typical in photonic designs.

New Commands

The following commands have been added to the documentation:

- **LITHO CL DENSEOPC**

  Performs OPC on curvilinear input.

- **setlayer layer_simplify**

  Derives a layer that can be corrected from curvilinear input.

- **curvilinear_opc**

  Activates algorithms specific to curvilinear OPC and specifies the basic OPC constraints to use on curved shapes.

Updated Commands

The following commands were updated:

- **fragment_nearly_coincident** … [-cornerDist *val*]

  Use the -cornerDist option to prevent fragmentation points from being copied within a certain distance to a corner.

- **OUTPUT_OPC -curvilinear**

Use -curvilinear with OUTPUT_OPC to produce curvilinear shapes based on the current OPC displacements of the fragments.

- **setlayer curve_target** …[minStepTanDist *val*]… and **target_curve** …[minStepTanDist *val*] …

  Use minStepTanDist to prevent overly aggressive minimum curvature on steps.

Calibre nmOPC Previous Release Highlights

Important changes from the previous four releases are summarized here.

Changes to Behavior

As of version 2020.3, NEWTAG external, internal, enclose, and enclosing require all constraints to have an upper bound (< *value*). The run exits with an error.

New Commands

The 2021.1 release added the following commands:

- **allow_single_site_correction** {**off** | **on**}

  Allows jog fragments to be corrected.

- **fragment_consistency_mode** {**0** | **1**}

  Controls how fragments generated by fragment_max are promoted around tile boundaries.

- **TAG_FRAG_SRC** ***layer source_number*** **-out** ***tag***

  Finds all fragments tagged with a specific fragment_corner source number and places them in a tag set.

- **TAG_MRC_SRC** ***tag_in source_number*** {**-out** | **-aout**} ***tag***

  Identifies which MRC rule blocked a fragment when used in conjunction with MRC_RULE … source.

There were no new commands for the 2020.4 or 2020.2 releases.

The 2020.3 release added the following command:

- **adjust_moved_target_sites** {**on** | **off**}

  Adjusts dense sites after retargeting with target_curve.

Updated Commands

The 2021.1 release modified the following commands:

- **fragment_corner** … [source *N*]

  The fragment_corner command adds an option, source *N*, for marking fragments the command generates. Use this in conjunction with TAG_FRAG_SRC to analyze unexpected fragmentation results.

  Additionally, fragment_corner corner type now accepts "any" in addition to concave and convex. Use "any" when the rule requires length2 but the second corner's type is not important.

- **target_curve** …[{emulate | spline} [convex_adjacent *val*] [concave_adjacent *val*]]

  The target_curve command adds controls for retargeting fragments that are in the neighborhood of a corner. Emulate and spline control how freely sites are shifted. The convex_adjacent and concave_adjacent options specify how far from the corner is considered "adjacent."

- **FEEDBACK *tag* property(*annotated_tag*)**[:default(*value*)]…

  The FEEDBACK command can now read feedback values from annotated tag sets. Feedback values outside the range of -2 to 2 are treated as 0. Fragments in ***tag*** that are not in the annotated tag set receive a value of -0.4 unless :default(*value*) is specified.

- **ML_VECTOR_CAPTURE_END -tag *set*** …[-gauges *filename*]

  The ML_VECTOR_CAPTURE function can now restrict the collection of feature vectors to only those that intersect gauges. The -gauges argument cannot be combined with -old, which is used to combine data sets.

- **MRC_RULE** {**external** | **internal** | **enclosure**} ***tag1*** '**{**' {***rule*** [source *N*]}… '**}**'

  The tag-based MRC_RULE scripting command adds the option "source *N*," which can be specified after each set of spacing constraints. Fragments that are blocked by that particular set of spacing constraints are marked with the set's *N*. Use this in conjunction with TAG_MRC_SRC to analyze results.

- **NEWTAG expression**

  The NEWTAG expression command adds two built-in properties, :feedback and :rl_control. The :feedback property returns the feedback set on a fragment for the current iteration. The :rl_control property returns how much a control site on the fragment is shifted after retargeting.

- **TARGET_CONTROL *tag* -constant property(*annotated_tag*)**[:default(*value*)]

  The TARGET_CONTROL command adds the ability to read the shift value from an annotated tag set. Fragments in ***tag*** that are not in the annotated tag set do not change their shift value unless :default(*value*) is specified.

The 2020.4 release modified the following commands:

- **fragment_corner *edge_spec rule*** … [allowzero]

  The fragment_corner command adds the option “allowzero.” In the rare situation where a fragmentation scheme can exactly fragment an edge with no remainder, by default the last pair of fragments are not created because a zero-length remainder is less than the minimum remainder (either rem or the minimum fragment length). If allowzero is specified, the last pair of fragments are created when the remainder is either greater than the specified minimum remainder or exactly 0.

- **DISPLACEMENT *tag*** [-applyLimitIn | -applyLimitOut | -applyProperty]
  [{-fixedOffset | -fixedOffsetIncr | -hintOffset | -hintOffsetIncr} *val*]
  [-freeze | -unfreeze] [-limit {*inner outer*} | clear]
  [-moveLimitsFromTarget] [-unlock]

  OPC_ITERATION -set_epe [*pvband_pwopc_options*] [*matrix_retargeting_options*]
  [*gradient_optimizer_options*] [-coarse *num_of_simulations*]
  [-simulate_sites | -simulate_sites_epe] [-simulate_no_opc_pw]

  OPC_XMEEF_ITERATION [opcLayer *layer*] [[iterations] *count*]
  [coarse *num_simulations*] [iterativeXmeef {true | false}]
  [recycleXmeef {true | false}] [xmeefRadius *dist*]
  [*pvband_pwopc_options*] [*matrix_retargeting_options*] [*gradient_optimizer_options*]

  SITES_CREATE *tag* … **-numx *x* -spacing *s*** [-epe_spacing *es*] [-min_corner *dist*]
  [-offset {*offset* | applyProperty}] [-rotate *angle*] [-shift [ratio] *value*]

  SITES_SHIFT *tag* {*value* | applyProperty} [-min_cornerdist *dist*] [-type *type*]

  For the arguments indicated with an underline, values can now be expressed either as a number or read from an annotated tag set using the following syntax:

  ```
  property(annotated_tag_set)[:default(value)]
  ```

  For example, any of these are acceptable:

  ```
  DISPLACEMENT tag1 -fixedOffset 0.01
  DISPLACEMENT tag1 -fixedOffset property(tag2)
  DISPLACEMENT tag1 -fixedOffset property(tag2):default(0.01)
  ```

  The first line represents the syntax before version 2020.4. The second and third show reading the value from an annotated tag set, tag2. In the case of the syntax of the second line, if a fragment from tag1 is not represented in tag2, it receives the fixed-offset default value. In the syntax of the third line, if a fragment in tag1 is not represented in tag2, it receives the user-specified default value of 0.01.

- **SITES_CREATE *tag* -bylayer *layer* -numx *x* -spacing** s …
  [-frag {pt1 | pt2 | midpt} [-shift [ratio] *shift_value*]]

The “by layer” form of the SITES_CREATE command can now shift sites when creating them, removing the need for SITES_SHIFT. The -shift argument requires identifying where to initially place the site on the fragment (-frag).

- **target_curve** [-moved]…

  The “-layer *name*” argument for target_curve is obsolete. Formerly target_curve did not verify if the specified layer was suitable for use. The target_curve command now only supports layers of type “opc” and “correction” to be used, which are handled by other arguments.

The 2020.3 release modified the following commands:

- **DENSE_SIMULATE** … [-dose3 *dose_list*]…

  To support hybrid topological resist models, which are also new in version 2020.3, the DENSE_SIMULATE Tcl scripting command can now accept a third dose list.

- **fragment_layer** … {**not_inside** | **not_outside**} …

  The denseopc_options command fragment_layer can now fragment edges that are coincident with the marker layer.

- **layer *name* underlying active2**

  To support hybrid topological resist models, which are also new in version 2020.3, the denseopc_options layer keyword adds “active2”.

- **mask_chip_corner -table {*list_of_conditions*}**

  The mask_chip_corner command now supports specifying variable corner chipping for VEB. Generally, variable corner chipping is specified in a litho model, but the VEB commands require non-litho models.

- **NEWTAG** {**external** | **internal** | **enclose** | **enclosing**} …-moved [-tag_only] …
  -out *tag* -range *constraint* [-outRest *tag*]

  The measurement group of NEWTAG Tcl script commands has two changes:

  - The option “-moved,” which causes the measurement to use the current position of the fragments instead of the original position, can also specify -tag_only. The -tag_only option causes the measurements to make use only of those fragments that were fragments originally; jogs that grow into fragments are ignored.
  - All ***constraint*** ranges must include an upper bound. Formerly, a constraint could be “> a”. This must now be “> a < b” or “>a <= b”. Use -outRest for creating tag sets with an undetermined upper bound.

- **NEWTAG property *tag_in layer property*** [-inside] **-out *tag_out***

The NEWTAG property Tcl script command adds “-inside” to transfer ***property*** to all ***tag_in*** fragments that interact with the ***layer*** shapes. Without -inside, only ***tag_in*** members within a certain distance of the property markers on ***layer*** received ***property***.

- **NEWTAG topological** … [-moved]…

  The NEWTAG topological Tcl script command adds the option “-moved.” This option causes the topological check to use the current fragment positions rather than the original positions.

- **NMBIAS_MEASURE_TAG** … [-c2c [<u>!=45</u> | ==45 | >=0]] …

  The NMBIAS_MEASURE_TAG Tcl script command has two changes:

  - The existing option “-resolution {min | max}” can now be specified with measurements of space or width.
  - The command adds the -c2c option set. When this is specified, NMBIAS_MEASURE_TAG also measures corner-to-corner pairs of fragments (that is, fragments that are parallel and not projecting). By default, any pair of corners that are on the same axes and do not project are measured.

- **setlayer *output* = curve_target** …[jogRampConcave *value* jogRampConvex *value*]

  The litho setup file command “setlayer curve_target” adds the ability to control the curve’s adjustment specifically for the corners on either side of a jog. This gives three possible levels of adjustment for corners on a polygon: cornerRadius (with overrides for concave or convex) for any corner, stepRadiusConcave/Convex for a concave and convex corner separated by a single fragment, and jogRampConcave/Convex for a concave and convex corner separated by a jog.

- **simulation_consistency** 3

  The litho setup file command simulation_consistency adds another setting, “3”. Compared to “2”, “3” tunes settings for improved consistency but slightly slower run time. The recommended setting is still “2” for DUV processes and “euv” for EUV processes.

- **SITES_CREATE *tag* -bymarker *layer* -numx *x* -spacing *s*** [-min_cornerdist *dist*] [-offset *offset* | applyProperty] [-dist *md*] [-replace]

  The SITES_CREATE Tcl scripting command adds a new mode, “-bymarker,” that creates sites by projecting the shapes on ***layer*** to the nearest ***tag*** fragment.

- **SITES_CREATE** … [-offset *offset* | applyProperty]

  All four SITES_CREATE modes now allow setting the offset value by reading a property on the fragment. The fragment tag set must be annotated prior to using applyProperty.

- **target_curve** … [-moved | -layer *name*] [stepRadiusConcave *value* stepRadiusConvex *value*]

[jogRampConcave *value* jogRampConvex *value*] [colinearAngleTolerance *degrees*]
[lineEndWidth *value*] [spaceEndWidth *value*]
[preferMid {true | false} [midSpanExt *value*] [maxMidSpan *value*]]
[roundCorner {true | false} [roundedCornerRadius *value*]]…

The target_curve Tcl script command adds most setlayer curve_target options.

The 2020.2 release modified the following commands:

- **OUTPUT_SHAPE fragment** … [-point | -point1 | -point2] **-property**…

  The annotated OUTPUT_SHAPE command now allows the -point, -point1, and -point2 options. These place boxes at either the first, second, or both endpoints of the fragment rather than the center of the fragment. The properties are attached to the boxes.

- **setlayer *output* = curve_target** … [stepRadiusConcave *val* stepRadiusConvex *val*]

  The setlayer curve_target command adds the stepRadiusConcave and Convex options. This pair of arguments can be used to specify radii smaller than the cornerRadius settings for use around jogs.

Best Practice Updates for Calibre nmOPC

Changes and updates to the Best Practices for Calibre nmOPC are summarized here.

Table 9-7. Calibre nmOPC Best Practice Updates

Best Practice	New or Update?
2020.2	
Removed “algorithm” from “Recommended Settings for Best Performance.” **Reason**: The information was incorrect. Algorithm 2 has the same performance as algorithm 1. Algorithm 2 (the default) is still recommended because of improved consistency and convergence.	Update
2019.4	
Use fragment_inter -ripplestyle 2. **Reason:** This provides better consistency than the default, -ripplestyle 1.	Update
2019.1	
Removed recommendation that setlayer curve only be used for simple cases in “Optimize Results From setlayer curve.” **Reason**: The command can also be used in more complicated cases.	Update
2018.4	

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
The references to maxcp and width were removed from “Optimize Results From setlayer curve.” **Reason:** The maxcp option has been disabled since v2007.2.	Update
The reference to epe_spacing was removed from “Recommended Settings for Best Runtime.” **Reason:** The recommendation did not apply with algorithm 2, which is the only algorithm setting that is not deprecated.	Update
2018.2	
For EUV processes, use simulation_consistency euv. **Reason**: The euv keyword is new in v2018.2. It encapsulates recommended values for imagegrid and final_upsample to improve consistency.	Update
2018.1	
The reference to controller was removed from “Recommended Settings for Best OPC Results.” **Reason:** The controller has been deprecated since v2010.4.	Update
2017.3	
The best practice for mrc_rule is now ≤ fragment_min rather than strictly less than. **Reason:** Correcting the manual.	Update
2015.2	
The recommended setting for algorithm is now 2 (the default) instead of one. This deprecates two existing best practices: • Setting feedback_balance is no longer recommended for the setup file as it is specific to algorithm 1. • The recommendation to use algorithm 1 in the “OPC Output Consistency” section is no longer valid. **Reason:** Setting algorithm to 2 produces more accurate OPC results.	Update

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
2015.1	
A new section, "Simplifying a Calibre mpOPC Setup File" has been added that describes how to create a simpler Calibre mpOPC setup file for double and multi-patterning mask correction. **Reason:** Double and triple patterning significantly increases the complexity of the setup file as, in many cases, you must define fragmentation and MRC constraints for every pattern. Often, the fragmentation and constraints are same for every pattern, but they are still explicitly defined for each combination of patterns.	New
A new section, "Simplifying a Calibre mpOPC Setup File" has been added that describes how to create a simpler Calibre mpOPC setup file for double and multi-patterning mask correction. **Reason:** Double and triple patterning significantly increases the complexity of the setup file as, in many cases, you must define fragmentation and MRC constraints for every pattern. Often, the fragmentation and constraints are same for every pattern, but they are still explicitly defined for each combination of patterns.	New
2014.4	
A new section, "Fragmentation Best Practices" has been added. This includes examples on how to deal with asymmetric fragmentation that occurs for intra- and interfeature fragmentation on long and short edges. This section also includes the "Long Edge Fragmentation" best practice that was previously documented. **Reason:** Fragmentation affects the consistency of results produced during an OPC run.	New
The "SRAF Print Avoidance Best Practices" section has been updated to include asraf layers and optimizing negative SRAFs for print avoidance. **Reason:** Negative SRAF layers are now supported in Calibre nmOPC.	Update
The recommended range for fragment_max is now 0.1 to 0.4 um. **Reason:** Updates to technology.	Update
The "max_iterations nopw" and pwopc_mode best practices have been deprecated.	Update
2013.4	
The Hardware Requirements and tilemicrons settings recommendations has been updated to include Hyperthreading and non-Hyperthreading core recommendations as well as a 10 nm node. **Reason:** Updates to technology and node size.	Update

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
If you are using mask_chip_corner with OPC and SRAF layers, to perform a Calibre OPCverify or Print Image contour simulation, you must use the Calibre OPCverify cornerchop command on both layers. **Reason:** To ensure processing for OPC and SRAF layers during Calibre OPCverify and Print Image contour simulation when corner-chopping polygons.	New
2013.3	
The entire Calibre sraf_print_avoidance best practices have been updated due to the redesign of the entire sraf_print_avoidance process. This includes: • Simplified method to create sites on sraf layers using the new sraf_sites_create. Simplified fragmentation rules using the new “mrc_rule area” check instead of prior rules. • Obsoletion of the fragalign rules (sraffragalign now automatically aligns SRAF fragmentation). • Process windows for SRAFs are now specified using the pw_condition “spa” keyword. Refer to the section “SRAF Print Avoidance Best Practices” in the *Calibre nmOPC User’s and Reference Manual* for complete details. **Reason:** The introduction of sraffragalign, sraf_create_sites, and a number of other enhancements have made a number of the former best practices obsolete.	Update
2013.2	
The entire Calibre sraf_print_avoidance best practices have been updated and reorganized. Refer to the section “SRAF Print Avoidance Best Practices” in the *Calibre nmOPC User’s and Reference Manual* for complete details. **Reason:** The deprecation of weight and tolerance-based PWOPC and replacement by PVband-based PWOPC affected how pw_conditions were defined for sraf_print_avoidance operations.	Update
All cost_tolerance-based best practices have been removed. **Reason:** The cost_tolerance command was deprecated in 2013.1.	Update
2013.1	
For PVband PWOPC, sites (including default sites) must to be deleted and then manually re-created for involved tag sets before OPC correction and site-related tagging. **Reason:** In PVband PWOPC, default sites are created before execution of tag scripts and thus will not see target-based controls such as TARGET_CONTROL and TARGET_FUNCTION (both are tag script commands). This practice allows for these target controls to function.	New

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
When implementing sraf_print_avoidance, use a fragalign statement for the SRAF layer (particularly the copy and allowDelete options) to ensure that there are no residual MRC violations at the tile boundaries due to the variation of the fragments during promotion at tile boundaries. **Reason:** MRC violations after sraf_print_avoidance at tile boundaries can occur because the fragmentation on the SRAF layer between child versus parent for the fragment promoted was different. This causes fragment pairing to fail and subsequently prevent deletion during the sraf_print_avoidance process.	New
2012.4	
Process Window OPC Updates: • Only two extreme process conditions are recommended for PWOPC (the nominal condition is no longer required). • PVband is the only recommended mode for PWOPC. All weight-based PWOPC is no longer recommended. • The initial recommendation for PW iterations is to use a 50/50 ratio between non-process window and process window iterations settings, ranging down to a ratio of 3/12 no-PW/PW for good quality of results • You are no longer required to create sites on the contact/via layer. **Reason:** Streamlining of process and improvement of PWOPC results.	Update
SRAF Print Avoidance Best Practice Updates: • fragment_min should be larger than both the minimum internal and external MRC constraint. • All layers containing SRAFs should be defined as an “sraf” layer, otherwise the sraf layer will not be moved. • Sites should be created and simulated before the use of sraf_print_avoidance, since sraf_print_avoidance only checks the simulation; it will not perform the simulation itself. • OPC_ITERATION is recommended to be used after sraf_print_avoidance since sraf_print_avoidance may impact OPC convergence. • Fragmentation should be large enough that the mrc_rule internal for the SRAF times the minimum fragment length is a legal shape for the mask shop. • Use the -multi keyword for the SITES_CREATE command as it will give more consistently cleaner results. • New recommendations for debugging printed SRAFs include checking sites at the printing location, checking the MRC limit, fragment alignment, and threshold. **Reason:** Overall improvement of SRAF detection.	Update

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
2012.3	
In ddlOpcIter, the upper limit recommendation of ddlPrimaryFeedback changes from -0.25 to -0.4. **Reason:** This improves the ddlOpcIter results.	Update
In ddlOpcIter, the upper limit recommendation of ddlPrimaryFeedback changes from -0.25 to -0.4. **Reason:** This improves the ddlOpcIter results.	Update
If you use pwOpcIter in conjunction with ddlOpcIter, the pwOpcIter parameters maxPerpGaugeLength and minGap should be set to the same value as ddlDecompCD. **Reason:** This prevents certain conflicts between ddlOpcIter and pwOpcIter.	New
2012.2	
Set simulation_consistency to 2. **Reason:** Setting simulation_consistency 2 uses faster simulator settings that still provide high consistency of results.	Update
Set mrc_mode to 2. If you have mrc_mode explicitly set in your setup file, then you should change it to reflect this new setting. **Reason:** This is an advanced MRC Solver that provides lithographically better results in highly-constrained layout regions. OPC results will be unchanged in unconstrained regions, and improved in constrained regions.	Update
In DDL OPC, the minedgelen parameter (which defines the smallest allowable edge length) in fragalign should either be equal to fragment_min or omitted so it can be set to fragment_min by default. **Reason:** This prevents fragment misalignment during DDL OPC.	New
For ddlOpcIter, use partial intensity-based decomposition with the following factors: • Set ddlUsePI to "true." • Optimize the ddlDoubleExposureContrastTolerance, ddlDSCutOffContrast, ddlDNCutOffContrast parameters to provide robust initial decomposition. • The value of these parameters are greatly dependent on the optical model and layout and can't be generalized.	New

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
For ddlOpcIter, the starting point for growing secondary fragments is recommended to be around half way from the starting iterations in order for OPC to perform the best correction before adding additional double exposure areas. Recommended practices include the following: • Set ddlGrowOn to 0.5 * (number of total iteration)+/- 10% (total number of iterations). For example, when nIterations is 10, set ddlGrowOn to 5 or between 4-6. • End the growing process before the last OPC iteration to allow correction for newly-added secondary features. • Set ddlGrowOff is 0.7 * (total number of iterations). For example, when nIterations is 10, set ddlGrowOff to 7.	New
For ddlOpcIter, the grow threshold values should be as large as possible while maintaining good results. The smaller the value, the more the double exposure area generated. Recommended practices include the following: • Set ddlGrowThreshold to minCD/4. • Set ddlGrowImageControlledThreshold to minCD/8. • Choose a smaller ddlPrimaryFeedback value (between -0.1 and -0.25) to refine the primary fragment placement. Choose a larger values for ddlSecondaryFeedback (between -0.2 to -0.4) since the image is not as sensitive as it is to the primary fragments.	Update
Use CornerRadius in the range between 0.060 um to 0.080um. A small corner radius can cause more ripples in the image. **Reason:** A large corner radius can cause large EPE on the nearby fragments.	New
The recommendations for Calibre nmOPC and Calibre OPCverify hardware requirements and tilemicrons settings have been updated to the following: 45-40 nm/ 2 GB per core /44-55 um 32-28 nm/ 3 GB per core/35-40 um (30 um for M1) 22-20 nm/ 4 GB per core/25-30 um (20 um for M1) 14 nm/ 4 GB per core/20 um	Update

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
2012.1	

In DDL, do not use fragmentation with unbounded definitions such as "length > 0.250". For example:
`fragment_corner concave not_end_adjacent length > 0.250 fragment 0.060 0.050 0.050 breakinhalf`
The following statement uses the correct format:
`fragment_corner concave 0.060 0.050 0.050 0.050 0.040 0.040 0.040 0.040 0.040 0.040 breakinhalf`
Reason: This will cause incorrect fragmentation.
	New

2011.4	
For jog_ignore_metric, the recommended setting is now *tech_node*/4, where *tech_node* is the technology node size. **Reason:** This should provide better jog handling based on your technology node.	Update
2011.3	
For retarget_layer, "emulate" is now the recommended option (instead of "spline"). **Reason:** The "emulate" option is similar to the "spline" option, except that it allows retarget edges to differ from the original OPC.	New
For mask_sample_grid, "rsm" is now the default setting. **Reason:** As "rsm" is the default, it will no longer need to be explicitly set in the setup file.	Update
In ddlOpcIter, the values of ddldecompCD and maxpolywidth should be equal. **Reason:** This prevents certain feature misalignments in the ddlOpcIter process. The fragments that are decomposing are typically the ones to be aligned.	New
2011.2	
The notchfill operation should not be done after setlayer veb_retarget. **Reason:** Using notchfill inside the VEB setup file after the VEB retargeting process can introduce potential MRC violations after OPC.	New

Table 9-7. Calibre nmOPC Best Practice Updates (cont.)

Best Practice	New or Update?
When specifying MRC constraints for nub or jog edges, be sure the “length” criteria in the MRC rules matches the length definition of the nubs or jogs. For example, if the jog length definition is 5nm, the “length” criteria should also be 5 nm so that the sub-check can be applied properly at the nub or jog edges. **Reason:** To make sure the rule is applied at small “length” definitions.	New
When using the pw_condition command to implement process window-based OPC (or PWOPC), use PVband PWOPC. **Reason:** The documented practice has been simplified to highlight the best solution, PVband PWOPC.	Update
Explicitly set fragment_max and fragment_min outside fragment_layer blocks. **Reason:** Global settings are not passed to fragment_layer blocks.	New
With CPA, use SMOOTH_NEUTRALIZATION 1 and SMOOTH_GRADIENT 0 inside CM1 model. Without CPA, use SMOOTH_NEUTRALIZATION 2 and SMOOTH_GRADIENT 1. **Reason:** Improves simulation consistency.	New
In DDL, use “clone_transformed_cells decide” (do not turn off cloning). **Reason:** Cloning should *not* be turned off for DDL.The ROTATED placements must be cloned for DDL to work properly. Cloning of reflected placements is not important for DDL since the process has symmetry with respect to reflections.	New
In DDL, enable the “copy allowDelete” options in the fragalign command. **Reason:** When using the fragalign keyword to align all fragments on a target layer in DDL, there are certain conditions where fragments cannot be aligned without deleting a fragment. This allows a fragment to be deleted if the final result will be aligned.	New
When using negative SRAFs in DDL, use fragmaxedge, which applies the OPCpro minedgelength operation to the tag set (the negative SRAFs). **Reason:** In DDL, as fragmentation is disabled, maxedgelength fragmentation is not run on the correction layer. In most cases, this is not an issue as fragmentation is copied from the opc layer to the OPC. However, in the case of negative SRAFs, they are not included in the opc layer, so there is no fragmentation to copy.	New

Calibre nmSRAF

This section includes information regarding changes and enhancements to Calibre® nmSRAF™ for the 2021.2 release.

For more information, refer to the *Calibre nmSRAF User's and Reference Manual*.

Calibre nmSRAF Updates

This section covers changes affecting Calibre® nmSRAF™ products for this release.

New Commands and Functionality

The following new commands and capability have been added to Calibre nmSRAF. For more information, see the *Calibre nmSRAF User's and Reference Manual*.

- **scalefactor method —** The rectcorner template now includes a new option set to manage placement of SRAFs around corners with variable offsets is available through the rectcorner scalefactor method.

Best Practice Updates for Calibre nmSRAF

Best practices for Calibre nmSRAF are continually updated.

Table 9-8. Calibre nmSRAF Best Practice Updates

Best Practice	New or Update?
2019.4	
When passing tags in to the newtag command, Siemens EDA recommends always applying an upper bound to the internal constraint. **Reason:** For predictable output of SRAFs using tags.	New
2019.2	

Table 9-8. Calibre nmSRAF Best Practice Updates (cont.)

Best Practice	New or Update?
Siemens EDA recommends always specifying cleanupversion 1. The rectedge center and exclusion45, centermerge, intrapriorityalign, interpriorityalign, lineendmerge no, and intersection LTP keywords are all incompatible with cleanupversion 0, and no new features will be compatible with cleanupversion 0 going forward. **Reason:** For predictable, high quality SRAF generation.	New
2019.2	
Siemens EDA recommends never using the output of debug layers for SVRF-related post-processing. The only purpose for using the cnsraf debug capability is for debugging SRAF-generated intermediate layers within small regions. A high possibility exists that generated debug layers may be missing or not MRC clean near tile boundaries. Therefore, Siemens EDA does not guarantee clean generation of debug-specific SRAF layers for large designs or hierarchical runs. **Reason:** Prevents perpetuation of unnecessary errors from improperly generated debugging layers.	New
2018.4	
Siemens EDA recommends only specifying lineendmerge by priority. **Reason:** When lineendmerge is specified globally, line-ends of different priorities may be merged.	New
2018.4	
Conditional MRC parameters such as lineend2long, long2lineend, square2long, etc., must be specified by priority only for minfeaturespace. Do not define these conditional parameters globally. **Reason:** Although parsed as valid, global specification of these parameters degrades performance.	New
2018.2	
Siemens EDA recommends using minoffset, minfeaturespace, projecting and lineend keywords for global MRCs. **Reason:** Provides for predictable SRAF generation.	New
2018.2	
Siemens EDA recommends limiting conditional MRC keywords (lineend2long, square2long, squarecorner2corner, etc.) by priority only or where required. **Reason:** To avoid increased runtime due to additional measurement and cleanup with use of these extra MRC keywords.	New
2018.2	

Table 9-8. Calibre nmSRAF Best Practice Updates (cont.)

Best Practice	New or Update?
Siemens EDA recommends setting lineendmerge no and lineendmerge 0 by priority. **Reason:** Enables use of interpriorityalign and intrapriorityalign globally.	New
2018.2	
Siemens EDA recommends specifying strictcenter for SRAF generation using rectedge-center templates also specifying spacelimit. **Reason:** Two tagged target edges provide predictable SRAF generation.	New
2018.1	
The previous default setting of maxmove 0 has changed, and can now be explicitly specified. However, recipes previously using algorithm 1 must now explicitly specify maxmove 0. **Reason:** To obtain identical SRAF results prior to 2018.1.	New
2018.1	
Siemens EDA recommends always setting cleanupversion 1 for all new recipes. Additionally, set cleanupversion 1 when using the lineend2sqedge, sqedge2sqedge, sqcorner2sqcorner, corner2sqcorner, long2sqedge, sqedge2long, sqcorner2corner or sqedge2lineend keywords. **Reason:** For predictable, high quality SRAF generation.	New
2018.1	
Siemens EDA recommends always setting algorithm 1. **Reason:** For predictable, high quality SRAF generation.	New
2017.4	
Siemens EDA recommends using reducedbarsplit 0 to improve line layer SRAFs. **Reason:** For predictable, high quality long line SRAF generation.	New
2017.4	
Siemens EDA recommends always using the bbox option of the cnSRAF cleanup keyword conflictmode. **Reason:** For predictable, high quality SRAF generation.	New
2017.3	
Siemens EDA recommends always explicitly specifying minlength, minwidth and minarea in Calibre nmSRAF recipes. **Reason:** For predictable, high quality SRAF generation.	New
2017.3	

Table 9-8. Calibre nmSRAF Best Practice Updates (cont.)

Best Practice	New or Update?
Specify dynamicsizing values to be greater than or equal to minlength. If they are not specified as such, a warning message is issued. **Reason:** To maintain syntactically dependable recipes.	New
2017.3	
Specify sbwidth values to be greater than or equal to minwidth. If they are not specified as such, a warning message is issued. **Reason:** To maintain syntactically dependable recipes.	New
2017.3	
Specify sraf values to be greater than or equal to minsquarelength or the square root of minsquarearea. If they are not specified as such, a warning message is issued. **Reason:** To maintain syntactically dependable recipes.	New
2017.2	
For EUV-specific nodes, Siemens EDA recommends using maxoffset instead of multiple localmaxgradient templates to control the number of SRAF rings. **Reason:** For high quality SRAF generation at EUV nodes.	New
2016.4	
It is imperative that the Advanced Vector Extensions (AVX) environment is identical for recipe development and production runs; AVX environmental differences may cause slight variations in gradient maps which could result in final SRAF differences. **Reason:** For consistent SRAF generation between recipe development and production treatment.	New
2016.4	
The interpriorityalign and intrapriorityalign keywords require you to set cleanupversion 1. **Reason:** For improved SRAF cleanup.	New
2016.3	
Siemens EDA recommends using the algorithm 1 keyword as a part of all Calibre mbSRAF recipes. **Reason:** For improved SRAF consistency of generated SRAFs.	New
2016.1	

Table 9-8. Calibre nmSRAF Best Practice Updates (cont.)

Best Practice	New or Update?
All layers input to Calibre nmSRAF recipes must be unique. When attempting to reuse any input layer as originally named in subsequent operations within a Calibre nmSRAF recipe, the layer name must be changed through a second derivation. This best practice affects input layers for both cnsraf and mbsraf: target layer, offsetlayer, sblayer and correctionlayer. **Reason:** Failure to re-derive the input layer may potentially result in erratic results.	New
2015.4	
A comprehensive table was added to the Best Practices chapter of the Calibre nmSRAF User's and Reference Manual outlining defaults, recommendation and user notes for important keywords. **Reason:** Provides users with better control over Calibre nmSRAF for generating optimal SRAFs.	Update
2014.4	
Specify cleanupversion 1 for all new recipes. **Reason:** Provides for more precise MRC enforcement and better support for line layers, and 2D structures.	Update
2014.3	
Specify lineendmerge keyword with a value greater than minoffset when using in conjunction with intersection J and intersectionnegative J. **Reason:** Provides for more continuous SRAF.	New
Specify minwidth. **Reason:** Insures no SRAFs below MRC values are inadvertently placed.	New
2014.2	
Use cleanmode postsizing. **Reason:** Provides consistent SRAF cleanup.	New
Use the Lithofile model format instead of individual calls to background, optical_model_load and resist_model_load. **Reason:** Provides better model organization, and reduced specification set in the LITHO FILE statement and sraf_options block.	New
2014.1	
Use mbSRAF to fill SRAF areas instead of the family of fillMaxOffset keywords. **Reason:** fillMaxOffset keywords have been deprecated.	New

Table 9-8. Calibre nmSRAF Best Practice Updates (cont.)

Best Practice	New or Update?
2013.1	
Use correctionLayer for mbSRAF **Reason:** Provides better performing scattering bars.	New
rectEdge space-based option **Reason:** Provides proficient spaced-based scattering bars.	New
2012.3	
Template indexing capability added. **Reason:** Facilitates new template additions.	New
Match cnsraf specifications with OPC MRC specification. **Reason:** Ensures MRC-clean SRAF output after SA/SPA.	New
Siemens EDA recommends use of SrafRingHV template. **Reason:** Template simplifies the recipe.	New
Siemens EDA recommends use of GUI-based Litho File Tool to generate debug layer information. **Reason:** Simplifies and speeds SRAF debugging.	New
General tips for mbSRAF recipe development and control. **Reason:** Simplifies and speeds recipe development and improves results of SRAF output.	New
2012.2	
As of release 2012.2, Siemens EDA recommends the use of cnsraf_fill instead of RET SRAF_FILL. **Reason:** Provides optimal SRAF.	New
A table of technology node, hardware requirements, and the respective tilemicrons settings was added. **Reason:** Optimizes implementation.	Update

Calibre nmSRAF Default Changes and Deprecations

Calibre® nmSRAF™ includes the following default changes, or deprecated or obsoleted commands in this release.

Default Changes

- **cleanupversion 1 —** As of release 2018.4, Siemens EDA recommends using cleanupversion 1 for all new recipes.
- **cleanupversion 0 —** For release 2019.2, the rectedge center and exclusion45, centermerge, intrapriorityalign, interpriorityalign, lineendmerge no, and intersection LTP keywords are all incompatible with cleanupversion 0. For release 2019.2 and later releases, all new nmSRAF features will be incompatible with cleanupversion 0.
- **minfeaturespace long2lineend —** For 2018.4 and later, values for this keyword pair now default to lineend instead of projecting.
- **minoffset long2lineend —** For 2018.4 and later, values for this keyword pair now default to lineend instead of projecting.
- **strictcenter —** For 2018.2 and later, rectedge-center recipes generate center SRAFs between one tagged edge and its opposing untagged edge by default. If center SRAFs must be generated between two tagged edges only, strictcenter must be specified. Prior to 2018.2, center SRAFs were generated between two tagged edges by default. Use strictcenter to achieve pre-2018.2 results for rectedge-center recipes.
- **interpriorityalign and intrapriorityalign —** Prior to 2018.2, the default search distance for intrapriorityalign and interpriorityalign was incorrect set at two times the specified value. For 2018.2, this was fixed. The default search distance was changed to the specified value.
- **maxmove 0 —** As of 2018.1, the previous default setting of maxmove 0 has changed, and can now be explicitly specified. However, recipes previously using algorithm 1 must now explicitly specify maxmove 0 to obtain identical SRAF results prior to 2018.1.
- **cleanupversion 0 —** As of 2018.1, cleanupversion is now a required keyword. All recipes previously using the default must now have the keyword explicitly declared: cleanupversion 0.
- **Symmetric dynamicsizing —** As of Calibre release 2017.3, a default change to dynamicsizing was made so that square SRAFs, when trimmed, remain square when minsquarelength or minsquarearea keywords are specified.
- **Asymmetric dynamicsizing —** As of Calibre release 2017.3, a default change to dynamicsizing was made, that where trimmed SRAFs of the same priority are in conflict, their probability of contribution is significantly increased.
- **shielding —** Calibre mbSRAF shielding default was changed as of release 2016.1 from no to seedpixel.
- **cnsrafpromotion —** The 2016.1 manual incorrectly states that cnsrafpromotion default mode is 1 (enabled). Default enabling of cnsrafpromotion was not introduced until release 2016.2.

Deprecated or Obsoleted Commands

- **SRAF Optimizer Tool —** The SRAF Optimizer tool was deprecated and is obsoleted in Calibre release 2021.2.
- **SRAF Analyze Tool —** The SRAF Analyze tool was deprecated and is to be obsoleted in Calibre release 2019.2.
- **cleanupversion 0 —** The cleanupversion 0 setting was deprecated and is to be obsoleted in a future Calibre release.
- **collecttype —** The collecttype keyword was deprecated and is to be obsoleted in Calibre release 2019.2.

Calibre nmSRAF Previous Release Highlights

Significant changes from the previous four releases are summarized here.

Commands and Keywords

- **corner —** The rectcorner center template method now includes a new keyword, corner. Arguments to corner include pair, both, and center. Released in 2021.1.
- **centermove —** A new option, centermove, is now available for the intra- and interpriorityalign keywords. Released in 2021.1.
- **center multicorner Method —** A new method, center multicorner, is now available for the rectcorner template. Released in 2021.1.
- **Conditional Cleanup —** A new conditional cleanup capability is now available and accessible through the priority keyword. Released in 2021.1.
- **protectionradius —** The scalefactor method of the rectcorner template now includes a new keyword, protectionradius, to enable scalefactor-based SRAF placement without conflict. Released in 2021.1.
- **orient 45 —** The rectcorner scalefactor template and method now includes a new keyword, orient 45. Released in 2020.4.
- **centermove —** A new option, centermove, was added to the intra- and interpriorityalign keywords. Released in 2020.3.
- **intersection 45, C —** Two new options, C and 45, were added to the intersection keyword, extending methods of SRAF intersection merging. Released in 2020.3.
- **widthc —** A new option, widthc, was added to the intersection C keyword to control 45-degree SRAF segment width. Released in 2020.3.

- **rect45corner angledalso —** A new option, angledalso, was added to the rect45corner template, facilitating generation of axis-aligned SRAFs at corners of 45-degree oriented target shapes. Released in 2020.2.
- **rectcorner strictcenter —** A new option, strictcenter, was added to the center keyword of the rectcorner template. Released in 2020.2.

Gradient Calculation Dependencies Between mbsraf and pxopc for 3D Mask Models

Since Calibre release 2013.3, mbsraf has supported 3D mask models (DDM and HHA optical models). To support 3D models, mbsraf automatically uses a *gradientType* called *pxopcSlope* to accommodate the pxopc gradient computer, and the required gradient is then generated by the pxopc gradient computer using the weight values from mbsraf for its calculations. The mbsraf *gradientType pxopcSlope* supports the necessary weights for pxopc of *slopeWeight* and *verticalWeight*, where:

slopeWeight + verticalWeight = 1

Since initial support of 3D models, Siemens EDA has discovered that these weight values as interpreted by the pxopc gradient computer were wrong when using 3D models:

- **No weights specified —** If no weights were specified in mbsraf, the pxopc gradient computer used *slopeWeight* of 0.8 and *verticalWeight* of 0.2.
- **Explicit weights specified —** If explicitly specified weights were used in mbsraf, they were ignored and instead the pxopc gradient computer used a *slopeWeight* of 0 and a *verticalWeight* of 1.

In both cases, weight values specified by the user for mbsraf were not used by pxopc. This was documented in the Corrected Defects and Known Issues document for 2015.3_25 UR1 (Update Release) and fixed in that release. From 2015.3 UR1 and later, if results normally obtained from use of releases 2013.3-2015.2 are desired, use this corrective workaround:

- **If no weights were specified in the original recipe —** In your 2015.3 mbsraf recipe specify *slopeWeight* of 0.8 and *verticalWeight* of 0.2.
- **If any weights of any value were specified in the original recipe —** In your 2015.3 mbsraf recipe specify *slopeWeight* of 0 and a *verticalWeight* of 1.

Obsolete Commands

These keywords and commands have been obsoleted:

- **rect45corner orient —** All options of the orient keyword for the rect45corner template are obsoleted. No alternative is provided for this capability. Obsoleted in Calibre release 2020.2.

- **Calibre SRAF Analyze —** No alternative is provided for this capability. Obsoleted in Calibre release 2019.2.
- **cleanmode postconflict —** Use cleanmode postsizing instead. Obsoleted in Calibre release 2014.4.
- **dynamicsize_version —** Obsoleted in Calibre release 2015.4.

  A outdated syntax of cnSRAF, where corresponding ordered values were specified after the template name without keywords was obsoleted.

Five arguments of the mode keyword were obsoleted in Calibre release 2015.4:

- 45only
- all
- no45
- horizontal
- vertical

Calibre nmModelflow

This section includes information regarding changes and enhancements to Calibre® nmModelflow for the 2021.2 release.

Information on Calibre nmModelflow can be found in the *Calibre nmModelflow User's and Reference Manual*.

Calibre nmModelflow Updates

The following is a list of Calibre nmModelflow new and updated features this release.

For more information, see the *Calibre nmModelflow User's and Reference Manual*.

GUI-Related Changes

- The Stages Wizard initial page now has its **Center Gauges** tab set to 'off' by default.
- The Contour Analysis Tab now has additional controls to hide or enable filtered sites, as well as a control to release the loaded contour library information file to renumber the sites in order.
- The Flow Stage Wizard, Resist Calibration page now supports the multi-objective genetic algorithm (moga) search objective. It allows you to select two objectives instead of one for resist model tuning.
- Gridshift consistency settings can now be accessed from the Flow Stage Wizard, Resist Calibration page in the optimizer settings when the gridshiftconsistency objective is selected.
- When creating a verification job via the Calibre nmModelfow GUI, there is now a radio button available to turn off gauge simulation statistics generation for the job to save time.
- The Split Gauges dialog box now has a new operating mode (Feature Split) that allows it to work on sites instead of gauges.
- The Calibration Job Record Dialog Box now displays entries for the Iterstats Pareto results from running a moga search.

Functionality Changes

- The CLI now supports the mdf param sensitivity command, which checks a user-specified parameter for sensitivity to small changes in a specified gauge file and litho model.

```
mdf param sensitivity  -qlist "quantity_list" -pardata parameter low
high [-num steps] [-degree order] [-out file]
```

- The CLI now supports he mdf gauges etch_risk_table, which scans a VEB model inside a litho model and returns a report on which density kernels and BTERMs might be a high risk of canceling out another term.
- The CLI command mdf gauges computeils now has been modified to return NILS and EPE data in addition to the ILS information.
- The CLI command mdf optimize set_resist gridshiftconsistency now has an -obj_type keyword, which sets a choice of objective metric and may use the new -prec1D and -prec2D precision keywords.
- The CLI command mdf optimize set_etch has had the keyword max_linear_coeff added, which sets an upper limit on the absolute value of BTERMS.
- The CLI command mdf optimize set_resist mem_report has been added. When active, it returns the estimated memory usage for contour calibration.
- The CLI now supports additional mask loading resist model parameters (MASK_LOADING_SPAN2 and MASK_LOADING_SPAN3).
- The CLI supports changing the FRM output using the mdf setupfile param -resist frm_output_type to either energy or height. Energy is the default.

Calibre OPCpro

This section includes information on changes and enhancements to Calibre® OPCpro™ for the 2021.2 release.

For complete product information, see the *Calibre OPCpro User's and Reference Manual*.

Calibre OPCpro Updates

The following new keyword has been added to Calibre OPCpro.

- optical_model ***index model***

  This keyword is similar to opticalmodel, but allows for up to 12 optical models. The index is a number from 0 to 11. The model argument can either be the filename of an optical model located in modelpath, or an in-line optical model enclosed by braces ({ }).

Best Practice Updates for Calibre OPCpro

The table summarizes changes and updates to the Best Practices for Calibre OPCpro.

Table 9-9. Calibre OPCpro Best Practices Updates

Best Practice	New or Update?
2015.3	
Leave LITHO_CONTEXT_RELAX_MRC and LITHO_PRIORITY_BASED_MOVEMENT at default or set to previous best practice values. **Reason:** The defaults have changed to follow the best practice values. For new rule files, you do not need to set them now.	Update
2015.1	
Use the SVRF DEANGLE command rather than the litho variable LITHO_DEANGLE. **Reason:** More consistent layout preparation between Calibre OPCpro and Calibre nmOPC.	Update

Table 9-9. Calibre OPCpro Best Practices Updates (cont.)

Best Practice	New or Update?
2014.3	
Do not set LITHO_ASYM_SOURCE_CELL_CLONE. **Reason:** Since 2012, all the Calibre RET tools automatically determine whether cell placements require cloning based not just on the optical source but also based on how computations are shared across processors.	Update
2013.3	
When setting the LITHO_SIMULATE_CONTEXT variable, also set LITHO_SIMULATE_CONTEXT_DISTANCE. **Reason:** More consistent OPC results. When you set LITHO_SIMULATE_CONTEXT explicitly, it changes the default of LITHO_SIMULATE_CONTEXT_DISTANCE to unlimited.	Update
2012.3	
During length-based tagging, set an upper bound on length checks to keep the tags out of invalid regions. The upper bound should be less than or equal to the interaction distance. **Reason:** Better OPC results. Allowing tags to apply to fragments longer than the interaction distance can cause false tags and bad OPC results.	New
2011.3	
When setting the LITHO_SIMULATE_CONTEXT_DISTANCE variable, it is now recommended to start at 0.25 instead of 0.1 microns. **Reason:** LITHO_SIMULATE_CONTEXT_DISTANCE is now on by default with a value of 0.25.	Update
2011.2	
The LITHO_DEANGLE variable if set greater than 0 should not be set lower than 0.001. Good values are generally between 0.001 and 0.01. **Reason:** Better performance.	New

Calibre OPCsbar

This section includes information regarding changes and enhancements to Calibre® OPCsbar™ for the 2021.2 release.

For more information, see the *Calibre OPCsbar User's and Reference Manual*.

Calibre OPCsbar Updates

This section covers changes affecting Calibre® OPCsbar™ commands for this release.

New Commands and Capabilities

There are no new commands for this release.

Best Practice Updates for Calibre OPCsbar

This section summarizes the changes and updates to the Best Practices for Calibre OPCsbar.

Table 9-10. Calibre OPCsbar Best Practice Updates

Best Practice	New or Update?
2018.4	
Siemens EDA recommends setting OPTION CLEAN_RESEAT to YES. **Reason:** To allow for better scattering bar output.	New
2016.4	
If OPTION CAREFUL_CLEANUP is set to 0, OPTION FAST_MRC is disabled by default and a warning message is issued to the transcript. **Reason:** To allow for management of the interaction between these two options.	New
2016.2	
Always enable LAYOUT CLONE TRANSFORMED PLACEMENTS when running OPCSBAR. **Reason:** To allow for correct simulation of rotated structures.	New
2012.2	

Table 9-10. Calibre OPCsbar Best Practice Updates (cont.)

Best Practice	New or Update?
V1 processing mode has been obsoleted. The only mode available is V2, which is default behavior. **Reason:** V1 processing mode is obsoleted.	Update
OPTION TILEMICRONS provides improved processing speed through design hierarchy management. Provide a value to the command that approximately matches the process node of the design. See the *Calibre OPCsbar User's and Reference Manual* for detailed information. **Reason:** New command OPTION TILEMICRONS provides processing improvement.	New

Calibre OPCsbar Previous Release Highlights

Significant changes from the previous four releases are summarized here.

Commands and Keywords

None.

Obsolete Commands

There are no keywords or commands that have been obsoleted.

Calibre OPCverify

This section includes information on changes and enhancements to Calibre® OPCverify™ for the 2021.2 release.

For additional information, refer to the *Calibre OPCverify User's Guide and Reference*.

Try It!	**Calibre OPCverify Tutorial and Example Kit** Contains known working SVRF rule files, optical models, and a sample OASIS design database. 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.

See "Calibre Tutorials and Example Kits (eKits)" on page 22 for a listing of all eKits.

Calibre OPCverify Updates

The following is a list of Calibre OPCverify new and updated features this release.

Information on these commands and features can be found in the *Calibre OPCverify User's and Reference Manual*.

Command Updates

The following commands are modified for this release:

- Adds the apa_check command, which calculates the average printability analysis (APA) function value in relation to specified regions. This command replaces the sraf_apa command (which is currently maintained in this release for backwards compatibility).

  ```
  apa_check image_layer constraint
  [[not] {inside | outside} filter_layer]
  {{include_layer inc_layer [size_by val_um]} |
  {exclude_layer exc_layer  include_condition constraint
     [include_condition_over_under value]}}
  stochastic_model  modelName
  # error-centric section
  [property '{'
   apa [apa_actual] [apa_area] [apa_pixel_count]
  '}']
  [classify, limit, or histogram block]
  ```

- The curvature_check command adds the extent argument , which connects smaller error markers into a single marker up to a specified extent size.

- The maskgen command now has the rounded argument, which converts chopped corners to rounded corners with the same area.
- The sraf_apa adds two new properties keywords (sraf_area and sraf_pixel_count), but the command is also deprecated (and replaced by the apa_check command).

Best Practice Updates for Calibre OPCverify

This section summarizes the changes and updates to the Best Practices for Calibre OPCverify.

Table 9-11. Calibre OPCverify Best Practice Updates

Best Practice	New or Update?
2018.4	
Updated “Choose Grid Settings Based On Your Configuration”: Specific recommendations for imagegrid, contour_options, and optical_transform_size for all DUV and EUV nodes.	Update.
2018.2	
Updated “Choose Grid Settings Based On Your Configuration”: Specific recommendations for imagegrid, contour_options, and optical_transform_size for DUV sub-14nm and EUV.	Update
2015.4	
Update to “Choose Appropriate tilemicrons Settings”: Recommendations for the 14/10 nm process node have been added, and a recommendation for triple patterning layers needing at least 4G memory.	Update
Update to “Use an Error-centric Flow”: The anchor recommendation has been obsoleted and a new recommendation for settings for pm_classify parameters has been added (max_search = 1.0*halo, max_length = 3-10*halo).	Update
Update to “Add a Classification and Limiting Block to Process Window Contours Used With output_window”: If a setlayer check outputs duplicate errors and the setlayer check is used in the output_window statement, then the classify block for this check should contain the keep_no_context keyword. **Reason:** Failure to do use the keep_no_context keyword in this circumstance can cause a large impact in run time and performance.	Update
Update to “Use Filter Layer Markers to Check Square Contacts”: An example using the Expand Edge SVRF command has been added to show generation of a cross-layer filter.	Update

Table 9-11. Calibre OPCverify Best Practice Updates (cont.)

Best Practice	New or Update?
2014.3	
Avoid the use of max_error_number in classification blocks, except as a safety measure against extremely long classification times on unexpectedly large number of errors. The guideline from 2013.3 still applies; if you do use max_error_number, it should be at least 10*x* relative to worst_count.	Update
2013.3	
Specify a max_error_number that is at least 10x relative to worst_count for classification blocks. **Reason:** Avoids turning off classification too quickly.	New
2012.3	
Use the critical_dimensioncommand to quickly set default values consistently across Calibre OPCverify commands. **Reason:** Ease of use and accuracy.	New
2012.2	
Use the jog_options argument to contour_diff when operating around jog corners, especially when using the command on contours. **Reason:** Improves accuracy.	New
2012.1	
Added the section “Soft Bridge and Soft Pinch Checks Have Strange Error Markers Near Line Ends”. **Reason:** Improperly set values of the output_expand arguments to bridge and pinch can create unexpected results, such as multiple markers for a single contour, or overly large markers.	New
2011.3	
Use area_ratio and contour_diff to check contact/via layers instead of bridge and pinch checks. **Reason:** Results of improved best practices.	Update
Do not use overly large length values for measure_distance. **Reason:** Overly large interaction distance can affect runtime.	New
2011.2	
Added the section “Choose Appropriate tilemicrons Settings.” Reason: Improper settings for tilemicrons can affect runtime and memory usage	New

Calibre pxOPC

This section includes information on the changes and enhancements to Calibre® pxOPC™ for the 2021.2 release.

Detailed information is available in the *Calibre pxOPC User's and Reference Manual*.

Global Grid Alignment in Calibre pxOPC

Calibre pxOPC now supports default global grid alignment of tiles in the 2021.2 release.

By default, Calibre pxOPC now sets the origin (that is, lower left corners) of tile grids to a multiple of the grid step size. Global grid alignment improves the following:

- Consistency between images generated by Calibre pxOPC and Calibre OPCverify.
- Shift-based consistency of Calibre pxOPC output.

Calibre pxOPC GUI Changes

The 2021.2 release supports the following changes to the pxOPC interface:

None.

New Keywords in Calibre pxOPC

This release adds the following new keywords to Calibre pxOPC.

- None.

Changes to Commands in Calibre pxOPC

This release changes the following commands:

- None.

Best Practice Updates for Calibre pxOPC

The table summarizes changes and updates to the Best Practices for Calibre pxOPC.

Table 9-12. Calibre pxOPC Best Practice Updates

Best Practice	New or Update?
2019.2	
Information for EL and CDOF jobs were added.	New
2018.4	
Best practices for Decoratedarksraf jobs were added, in particular advice on process window conditions that differ from those for Decorate jobs. **Reason:** Experiments have provided more information on best practices for dark features.	New
2017.2	
When using three process window conditions in the last job, set tilemicrons to 5 for EUV; 19 for single-pattern DUV; 10 for double-pattern DUV; and 6 or 7 for triple-pattern DUV. **Reason**: Further testing has shown that smaller tilemicrons are necessary as the number of patterns or the frequency increases.	Update
2016.3	
To improve Calibre pxOPC runtime when using pw_bridging and pw_pinching, keep the constraints to less than 60nm. **Reason:** Formerly this was shown as 90nm but additional testing has shown that 60nm provides nearly the same quality and delivers better runtime.	Update
If the Detail job does not satisfy process window constraints, make sure the output from the LMRC job does not show hard bridging or very bad pinching. If also using pw_enclose, the LMRC should keep the enclosed layer fully covered by the target. **Reason:** Improving the input allows the Detail job to converge faster.	New
2016.2	
Process window constraints should be *slightly* more aggressive than verification constraints. **Reason:** Calibre pxOPC tries to keep contours in the vicinity of the target. Constraints that are too aggressive not only take longer but can result in larger EPE.	New

Table 9-12. Calibre pxOPC Best Practice Updates (cont.)

Best Practice	New or Update?
2015.1	
To fix oscillating contours, try one of these techniques: • Tune setlayer curve. • Use weight layers. • Add straight_contour 1 to the Correct, Refine, or Detail job. **Reason:** Released straight_contour for use with rippling contours.	Update
To reconcile Calibre OPCverify and Calibre pxOPC results, make sure that all process window conditions in Calibre OPCverify are represented in the Calibre pxOPC recipe. **Reason:** Calibre OPCverify syntax is not identical to Calibre pxOPC syntax for process window conditions. See "Reconciling Results with Calibre OPCverify" for specifics.	New
2014.3	
Shapes on a weight_layer layer should only cover the part of the edge that is approximately straight (that is, would not be rounded by target smoothing). **Reason:** When weight_layer shapes extend to the corners of polygons, Calibre pxOPC runs much longer for little improvement in final output.	New
2014.2	
The value of mrc_small_feature_size should be smaller than mrc_min_square. **Reason:** When this is violated, square SRAFs that are acceptable to the foundry are removed.	New

Calibre pxOPC Previous Release Highlights

Important changes from the previous four releases are summarized here.

Previous GUI Changes in Calibre pxOPC

The following capabilities have been recently added:

- None.

Previous Changes to Features in Calibre pxOPC

The following capabilities have been recently enhanced:

- The following commands added support for curvilinear MRC constraints (2021.1):
  - **mrc_min_external**
  - **mrc_min_external_sraf**
  - **mrc_min_internal**
  - **mrc_min_internal_sraf**

Previous New Commands in Calibre pxOPC

The following new keywords have been recently added to Calibre pxOPC:

- **job *number type*** (2020.3)

  The job command adds two new job types — detailhybrid and detailrectsraf. The detailhybrid job produces curvilinear main features and rectilinear SRAFs. The detailrectsraf job produces rectilinear main features and rectilinear SRAFs.

- **mask_periodicity** yes | <u>no</u> (2020.2)

  Forces mask output to repeat patterns periodically.

- **mrc_enabled** (2021.1)

  Activates MRC constraints for curvilinear mask output.

- **mrc_min_area** (2021.1)

  Constrains the minimum SRAF area during MRC.

- **mrc_min_area** (2021.1)

  Constrains the minimum SRAF area during MRC.

- **objective_region** (2020.4)

  Optimizes a region using specified objectives.

- **pw_vertical_constraint** (2020.4)

  Optimizes contours to be inside or outside of the shapes of a specified layer for a certain process window condition or group of conditions.

Calibre pxSMO

This section includes information on changes and enhancements to Calibre® pxSMO™ for the 2021.2 release.

Detailed product information can be found in the *Calibre SMO RET Selection User's and Reference Manual*.

Updated Keywords in Calibre pxSMO

Calibre pxSMO uses keywords to provide advanced functionality that is not always available in the template setup files.

Calibre pxSMO adds keywords or modifies functionality to the following commands:

- **mdf smo source** (keywords added)

  The MDF command "mdf smo source" adds the "maskgen" keyword to execute pxSMO in the mask (that is, OPC) command block.

RET Selection deprecates the following MDF commands in this release:

- None.

RET Selection adds the following new MDF commands for this release:

- None.

Calibre pxSMO adds the following new keywords for this release:

- **mrc_max_internal_sraf**

  This pxSMO setup file command constrains maximum internal spacing during MRC for rectilinear SRAFs only.

- **output_unblurred_source_file**

  This pxSMO setup file command outputs an illuminator-level source map file at the end of a session run.

- **save_job_raw_source**

  This pxSMO setup file command outputs an illuminator-level source map file at the end of a job for debug purposes.

RET Selection GUI Changes

The 2021.2 release includes the following changes to the RET Selection window:

- The OPC setup pane adds mask-only pxSMO as an OPC engine option.
- The Anchor/Threshold/Dose Settings dialog box adds Pre-SMO and Post-SMO tabs for pre-SMO and post-SMO anchoring.

Best Practices Updates for Calibre pxSMO

The table summarizes changes and updates to the Best Practices for Calibre pxSMO.

Table 9-13. Calibre pxSMO Best Practice Updates

Best Practice	New or Update?
2018.4	
"How to Use Gauge Selection" training videos are available on Support Center and the Calibre InfoHub.	New
2018.2	
"Getting Started with Parametric Explorer" training videos are available on Support Center and the Calibre InfoHub.	New
2018.1	
Add Smooth Target as an SMO layer type when setting up layout and gauge pair.	New
In many cases, an optimized source can be created using fewer than 20 gauges.	New
Changes to process window best practices. The changes include recommendations for focus and dose ranges, and weights used with point tools.	Update
Change to recommended mask delta setting for MEEF calculations. **Reason**: Change over to dense image grid.	Update
2017.4	
Training videos are available on Support Center in a KB article.	New
2016.1	
Changes to dark background inner and outer process window conditions for mask size. **Reason**: Better accounts for MEEF.	Update
New section of recommendations on tuning.	New

Calibre pxSMO Previous Release Highlights

Important changes from the previous four releases are summarized here.

Previous Changes to Commands

The 2021.1 release modified the following commands:

- **job** (keywords added)

  The pxSMO setup file command “job” now accepts the “start_mo”, “decorate_mo”, “improve_mo”, “polish_mo”, “cdof_mo”, and “el_mo” keywords to support mask-only optimization flows.

- **scatter_offset** and **scatter_offset_negative**

  The pxSMO setup file commands “scatter_offset” and “scatter_offset_negative” default to a new value for EUV optical models.

The 2020.4 release modified the following commands:

- **job** (keywords added)

  The pxSMO setup file command “job” now accepts the “start_so”, “improve_so”, “polish_so”, “cdof_so”, and “el_so” keywords to support source-only optimization flows.

- **mdf litho measure** (keyword added)

  The MDF command adds the -compute_pw keyword to calculate NILS, MEEF, ILS, EILS, and EPE at custom process window conditions.

The 2020.2 release modified the following commands:

- **fixed_source_optimization_region** (keywords changed)

  The pxSMO setup file command “fixed_source_optimization_region” now accepts the following keywords to match optical model syntax:

  - “illumangle” replaces “illum_angle”.
  - “rotangle” replaces “rot_angle”.

- **objective_region** (keyword added)

  The pxSMO setup file command “objective_region” now accepts the “contour_diff” keyword to minimize contour differences between process window conditions.

Previous New Keywords and Commands

The following new commands and keywords were added to Calibre pxSMO in the 2021.1 release:

- **mrc_enabled**

  This pxSMO setup file command activates MRC constraints for curvilinear mask output.

- **mrc_min_area**

  Constrains the minimum SRAF area during MRC.

- **mrc_min_edge**

  Sets a target length for MRC geometries.

- **mrc_min_external**

  Constrains external spacing during MRC; supports curvilinear mask output.

- **mrc_min_external_main2sraf**

  Constrains external spacing during MRC between a main feature and an SRAF; supports curvilinear mask output.

- **mrc_min_external_sraf**

  Constrains external spacing during MRC for SRAFs only; supports curvilinear mask output.

- **mrc_min_internal**

  Constrains the minimum distance during MRC between two internal edges; supports curvilinear mask output.

- **mrc_min_internal_sraf**

  Constrains the internal spacing during MRC between two internal SRAF edges; supports curvilinear mask output.

- **mrc_min_length**

  Constrains the minimum distance between two internal edges during MRC.

- **mrc_min_rect_length**

  Constrains the minimum distance between two internal edges lengthwise during MRC.

- **mrc_min_rect_width**

  Constrains the minimum distance between two internal edges widthwise during MRC.

- **mrc_min_small_feature_area_excess**

Constrains the minimum area for mask features during MRC.

- **mrc_min_square**

  Constrains the minimum distance between two internal edges in a square feature during MRC.

- **mrc_small_feature_size**

  Constrains the minimum distance between two internal edges during MRC.

The following new commands and keywords were added to Calibre pxSMO in the 2020.4 release:

- **parameterization**

  This pxSMO setup file command defines the mask shapes that represent features. The "compliant" keyword represents all features using Manhattan edges. The "compliantrectsraf" keyword represents main features using Manhattan edges and assist features with rectangles.

The following new commands and keywords were added to Calibre pxSMO in the 2020.2 release:

- **anchor_gid**

  This pxSMO setup file command specifies an anchor gauge.

- **anchoring_enabled**

  This pxSMO setup file command enables optimization with anchoring to a gauge.

- **enable_periodicity**

  This pxSMO setup file command reads periodicity information and applies it to custom clips.

- **optimize_dose**

  This pxSMO setup file command optimizes the dose threshold when anchoring to a gauge.

The following mdf commands were added to RET Selection in the 2021.1 release:

- **mdf smo configopt**

  Configures the in-session optical models before loading the model.

The following mdf commands were added to RET Selection in the 2020.4 release:

- **mdf gauges computeils**

  Calculates image log slope (ILS) at polygon edges in a layer for all gauges.

- **mdf smo flarelongrange**

  Configures “flare_longrange” statements in the pre_so_model and post_so_model *setup.in* files.

The following mdf commands were added to RET Selection in the 2020.3 release:

- **mdf litho clip_bias**

  Applies a CD bias to specified clips.

- **mdf litho inter_copy**

  Copies specified clips from one layer to another. Commonly used in the presmo block to copy an anchor gauge from the target layer to the correction layer.

- **mdf litho merge_layouts_gauges**

  Combines multiple layout and gauge pairs into a single pair of files.

- **mdf smo copyclip**

  Copies a specified gauge to a gauge id in the *retsel_cliparray.oas* layout file and *retsel_cliparray.gg* gauge file.

The following mdf commands were added to RET Selection in the 2020.2 release:

- **mdf smo setclips**

  Copies a custom clip file to the session directory and database.

- **mdf smo setgauges**

  Copies a custom gauge file to the session directory and database.

Previous Changes to the RET Selection GUI

The following parts of the RET Selection window changed in the 2021.1 release:

- The Clips and Gauges editor adds Weight to control clip weights.
- The Clips and Gauges editor adds flip with reflection symmetry options “ST” and “SL”.
- The PW Table pane adds the Spec option to open the Optical Model Spec Manager to create, edit, and manage optical model specifications (specs) with custom Zernike coefficients. The PW Table pane also adds PW Options to manage global process window condition settings, set image plane interpolation modes, and ignore specs from input optical models.
- The pxSMO Setup pane adds pxSMO Mode for selecting SMO, source-only, and mask-only optimization modes.

- The Process Window plot adds a uDOF plot to Gauge Plots to visualize usable depth of focus.

The following parts of the RET Selection window changed in the 2020.4 release:

- The pxSMO Setup pane adds Constant Flare to support adding constant flare across all process window conditions.
- The Verify Setup tree adds the Property PW item to simulate custom process window conditions.
- The Clips and Gauges editor adds reflection symmetry options “|”, “—”, and “+”.
- The pxSMO Setup, Advanced tab adds After Read options to Delete SOCS.
- Session Plots adds plots for edge-based image log slope (edge-based ILS). Session Plots also adds Property PW Plots to display metrics across focus and dose.
- The Inter-Plots window adds a Save Table to CSV button to write all contents of the session summary table to a CSV file.
- The Process Window Plot adds the Gauge Plots tab to plot metrics by gauge id. The Process Window Plot table adds Focus and Dose columns to display best focus and dose for individual gauges.
- The Score Configuration Dialog Box adds CD [%] as a property selection for scoring.

The following parts of the RET Selection window changed in the 2020.3 release:

- The Layout and Gauge Pair loader started supporting adding multiple layout/gauge pairs.
- The Anchor/Thres/Dose option was added to pxSMO Setup pane to support dynamic anchoring. Dynamic anchoring runs during each pxSMO iteration and improves optimization robustness and consistency.
- The Anchor/Threshold/Dose Settings Dialog Box added the following options:
    - The Optimize Dose option was added to the threshold settings.
    - The Copy Target Clip option copies the anchor clip from the target layer to the correction layer.
    - The OPC option uses the anchor clip in the OPC and Verify stages.
- The 2nd Stage Run option was added to the pxSMO Setup pane to support mask-only optimization (MO) of SMO-disabled clips.
- The Objective Region tab was added to each pxSMO and pxOPC job to apply custom objectives to marker layers at unique process window conditions.

The following parts of the RET Selection window changed in the 2020.2 release:

- The pxSMO setup pane added Anchor options to replace the Anchoring Setup Dialog Box and corresponding items in the Setup Items tree.
- The Clips and Gauges Editor added to support creating and managing custom clips and gauges. The Gauge Analysis Pane added an icon to open the Clips and Gauges Editor.
- The Session Browser added a Tags button to support creating, coloring, and managing tags for sessions. The Run Control/Layout Session View table added a column to display session tags.

Calibre Rule-Based OPC

This section includes information regarding changes and enhancements to Calibre® Rule-Based OPC for the 2021.2 release.

For more information, refer to the *Calibre Rule-Based OPC Reference and User's Manual*.

Calibre Rule-Based OPC Updates

This section covers changes affecting Calibre Rule-Based OPC for this release.

New Commands and Algorithms

The following new commands have been added to Rule-Based OPC:

- **-shift —** A new keyword set for VIA_SHIFTING, -shift, is now available for shifting property-based layers.
- **CL —** A new option, CL, is added for the LITHO NMBIAS command to enable curvilinear biasing.
- **CURVILINEAR_SMOOTHING —** A new command, CURVILINEAR_SMOOTHING, is available for nmBIAS for skewed angle and curved shape biasing.
- **curvilinear_bias —** A new keyword, curvilinear_bias, is available to enable curvilinear biasing for nmBIAS.

Default Changes

There are no default changes for this release.

Deprecated And Obsoleted Commands

There are no deprecated or obsoleted commands for this release.

Calibre RBOPC Previous Release Highlights

Significant changes from the previous four releases are summarized here.

Commands and Keywords

- **angle —** The Biasrule table mode now supports an angle filter for space and width classifications to support curvilinear geometries. Released in 2020.4.

Obsolete Commands

No commands were obsoleted.

Calibre SONR

This section includes information on changes and enhancements to Calibre® SONR™ design analysis and fab solutions for the 2021.2 release.

For more information, see the *Calibre SONR User's and Reference Manual*.

New Commands in Calibre SONR

This release adds the following new applications and keywords to Calibre SONR.

Command-Line Applications

- sonr --model_creator model_2

  Creates fully supervised machine learning models from feature vectors that include labeled data.

- sonr --concatenate_cv

  Combines two sets of calibration vectors.

Changes to Keywords in Calibre Sonr

This release changes the following keywords:

- Formerly, the command "cluster_method" was required in the setup files for SONR Layout Analyze. It is now optional.
- The feature vector file format renames the multilayer terms (MN and MT) for clarity. The new term names incorporate references to the measurement being made.
- **SONR_AUX_LAYER** … [-densityLayer *layer*]

  Use -densityLayer to identify a layer that should be used for calculating density.

- **SONR_COLLECT** … [-quantize *number*]

  Use -quantize to reduce memory requirements on large runs by treating similar features as identical.

  Additionally, SONR_COLLECT now outputs a database and *phvo.mod* by default.

- **SONR_PREDICT** … [-prop_thresh *value*] [-dfm_property Probability *layer* float max]

Use the new DFM property "Probability" and -prop_thresh options with fully supervised machine learning models.

- **sonr --cluster** …[--target_fa *float* [--hsl *col*] [--hsv *int*…]]

  Use --target_fa to calculate the resolution automatically based on a false alarm rate. (There is no guarantee the target rate is reached.)

  Additionally, sonr --cluster and sonr --tree no longer output a database by default. To write out results, you must specify the -o option.

Calibre SONR Previous Release Highlights

Important changes from the previous four releases are summarized here.

2021.1

New Tool: sonr Command-Line Utility

The 2021.1 release introduces the sonr command-line interface, with three applications:

- sonr --cluster groups feature vectors by similarity. This application can generate an initial sonr model or be used for selecting representative feature vectors.
- sonr --read exports feature vectors from the database into a text file.
- sonr --tree creates a prediction tree from an initial sonr model. It can be grown with training data from multiple layouts collected at different times.

New Keywords

The 2021.1 release added these keywords:

- **density_image_load**

  (Litho Setup File) Loads a PNG file with precomputed density convolution.

- **dist** ***distance***

  (sonr_options Block) Specifies how far sonr_vector_capture searches for DFM properties.

- **ml_model_load** ***name filepath***

  (Litho Setup File) Loads the sonr model for use by SONR_PREDICT.

Changed Keywords

- **mode**

  The setting "train" is obsolete. Use sonr --cluster and sonr --tree instead.

- **SONR_AUX_LAYER**

The SONR_AUX_LAYER command adds -dist *search_distance* to specify how far to search from the fragment for a DFM property marker. The default is 60 dbu.

- **SONR_COLLECT**

  The SONR_COLLECT command has undergone substantial revision.

  - There is no longer any default output. You must specify -outlayer or -outdb to see the results of the run.
  - The 2020.4 options -hsid, -outmod and -outCSV are no longer available.
  - The following optional arguments were added:
    - -append [*database*] adds data from the current run to the database. If one is not specified with -append, it uses the database in -outdb.
    - -density *png_file* allows the SONR_COLLECT run to read density information from a PNG file and include it in the feature vectors.
    - -densityLayer *layername* specifies a layer to use for density calculations. By default, the layer being analyzed is used. In multi-patterning layouts, this may not reflect the expected manufactured density.
    - -dist *search_distance* specifies how far from the fragment to search for a DFM property marker. The default is 60 dbu.

- **SONR_PREDICT**

  The SONR_PREDICT command has undergone substantial revision.

  - There is no longer any default output. You must specify -outlayer or -outdb to see the results of the run.
  - The required -snr_mod argument now takes the name of a model loaded in ml_model_load and not a directory path.
  - The 2020.4 options -hsid, -outmod and -outCSV are no longer available.
  - The following optional arguments were added:
    - -density *png_file* allows the SONR_PREDICT run to read density information from a PNG file and include it in predictions.
    - -dfm_property {Class | Gen} *outlayer type* max allows filtering the output based on DFM property markers.
    - -dist *search_distance* specifies how far from the fragment to search for a DFM property marker. The default is 60 dbu.

2020.4

New Tool: Hotspot Predict

The 2020.4 release introduced Calibre SONR for hotspot prediction. It can help predict unknown hotspots in full chip layouts using machine learning models. The SONR Hotspot Predict operation can be used along with the previously released SONR Vector Capture to extract the features of a design from known hotspot locations. These features are then used to generate and train a machine learning model and predict new unknown hotspots on full chip layouts.

New Keywords

The following SVRF operation and setup file keywords were added for hotspot prediction:

- **SONR** [EUV] **HOTSPOT PREDICT**

  (SVRF) Analyzes a layout with a trained machine learning model and predicts hotspots.

- **SONR_AUX_LAYER**

  (Litho Setup File) Prepares auxiliary layers for use with SONR_COLLECT and SONR_PREDICT.

- **SONR_COLLECT**

  (Litho Setup File) Collects feature vectors.

- **SONR_PREDICT**

  (Litho Setup File) Predicts layout hotspots based on one or more layout layers, a feature vector model, and the trained hotspot model.

2020.3

The following sonr_options block keywords were added in the 2020.3 release:

- **append** [*existing_database*]

  Adds new feature vectors to an existing database.

- **label *property_layer property_name*** …

- **prop *property_layer property_name*** …

  The label and prop keywords read numeric DFM properties from ***property_layer*** and write them to the database. Use “label” to add the property for information only. Use “prop” for the property to be considered part of the feature vector’s value for clusters.

2020.2

New Keywords

The following keywords can be added in the sonr_options block:

- **euv_slit_x_center**

  When using Calibre Sonr for EUV processes, add euv_slit_x_center for best results. This is a standard keyword for Calibre EUV solutions, and is described in "euv_slit_x_center" in the *Calibre OPCverify User's and Reference Manual*.

- **normalize**

  This optional keyword causes setlayer sonr_analyze to normalize the input data in the feature vector database when training a model.

- **poi**

  Use "poi" instead of "marker" to indicate the seed layer for SONR Layout Analyze. This keyword indicates the layer contains points of interest.

- **representative**

  This optional keyword causes setlayer sonr_analyze to flag one feature vector per cluster as "representative" of the cluster.

Changed Keywords

The SVRF operation SONR Vector Capture adds the option "EUV." Use EUV to have the run use the appropriate defaults for EUV manufacturing processes.

The following sonr_options keywords were modified:

- In Layout Analyze setups, use "**poi**" instead of "**marker**". The two layers are inherently different and it created confusion using the same keyword in both Layout Analyze and Vector Capture. The Vector Capture sonr_options block should continue using "marker".
- **min_sampling_distance** is now set automatically; the keyword is obsolete.

Calibre WORKbench

This section includes information on the following changes and enhancements to Calibre® WORKbench™ for the 2021.2 release.

For additional information, refer to the *Calibre WORKbench User's and Reference Manual*.

Calibre WORKbench Modeling and Command Changes

Calibre WORKbench contains multiple modeling structures and batch commands that are run when a GUI control is used.

- New Calibre WORKbench Invocation Options
- Changes to Calibre WORKbench Invocation Options
- New Calibre WORKbench Commands
- Modified Calibre WORKbench Commands
- Changes to the Optical Model Format
- Changes to the Masksim and DDM Formats
- Changes to the Test Pattern Format

New Calibre WORKbench Invocation Options

None.

Changes to Calibre WORKbench Invocation Options

None.

New Calibre WORKbench Commands

- cwb_sample — Selects a set of representative gauges, sites, or clips from an input group.

Modified Calibre WORKbench Commands

- None.

Changes to the Optical Model Format

The Optical Parameters File Format now supports the following updates:

- pupilObscuration — Now requires the image_normalization method set to either source_area or integ_source_area for a dark field system.

Changes to the Masksim and DDM Formats

None.

Changes to the Test Pattern Format

None.

Calibre WORKbench GUI Changes

In the 2021.2 release, the main window in the Calibre WORKbench interface supports the following changes:

- None.

Updates to the Optical Model Tool

None.

Updates to the VEB Model Tool

None.

Updates to the Calibre RET Flow Tool

The Calibre RET Flow Tool (RFT) v2.0 has had the following changes:

Adds the contourClearance parameter to specify a method to handle sites near clip edges.

Updates to Calibre ContourCal

The Contour Layer Info (CLI) File Format now supports the following added or modified options:

None.

Updates to Calibre IPSE

None.

Updates to Calibre Test Pattern Tools

The Calibre Pattern Generator now supports the following added or modified options:

- None.

Updates to SEMSuite Tools

Adds the SEMSuite toolset that includes:

- The Raw Data Filtering (RDF) tool for processing CD SEM image and data files to produce a gauge file.
- The Contour Data Flow (CDF) tool for extracting contours from CD SEM images to produce a Sites CSV file.

Dense Modeling Changes

This section includes information on the following changes and enhancements to Calibre® Dense Modeling handling for the 2021.2 release:

Changes to CM1 Center

None.

Changes to the CM1 File Format

For more information, see the section “CM1 Model File Format.”

The MASK_LOADING parameter now supports up to four mask-loading terms.

The CROSSTERM parameter now supports up to four mask-loading terms plus horizontal bias terms as cross terms.

Changes to the EUV-Related Model Formats

None.

Changes to the Topo Model Format

For more information, see the *Calibre WORKbench Topography Modeling User’s and Reference Manual*.

- None.

Changes to modelflow_v2

The modelflow_v2 syntax now supports the following modifications:

None.

Changes to the VEB Model and Commands

The Variable Etch Bias (VEB) Model File Format now supports the following modifications:

The BTERMs, xBTERMs, and yBTERMs parameters add support for three-term and four-term cross terms for physical reaction contributions to reactive ion etch (RIE) processes.

The average_bias parameter defaults to on, which provides additional contour regularization. The average_bias consistency keyword now supports a value of 3 for the most aggressive cases where the final consistency is expected to be less than 0.5 nm.

New Dense Modeling Commands

None.

Changed Dense Modeling Commands

None.

Chapter 10
Mask Data Preparation (MDP)

This chapter describes the 2021.2 changes and enhancements for the Calibre® MDP applications.

See "Corrected Defects and Known Issues" on page 12 for instructions on accessing the complete list of corrected defects and known issues for the release.

Calibre MDP Utility Updates

This section describes new features added to Calibre® utility commands for this release.

Refer to the *Calibre Mask Data Preparation User's and Reference Manual* for details.

Updates

- This release adds a new utility command, MDP DATAPREP, that generates input for data-to-database detection tools. MDP DATAPREP processes extended MEBES job decks and layer input according to mask type to produce an extended MEBES job deck with OASIS placements.

Calibre MDPview Updates

This section describes new features added to Calibre® MDPview™ for this release.

Refer to the *Calibre MDPview User's and Reference Manual* for details.

Global Viewer Updates

A number of global updates apply to all viewers (Calibre® DESIGNrev™, WORKbench™, LITHOview™ and MDPview™). Refer to the release notes for "Calibre DESIGNrev" on page 47 for complete details.

Calibre nmMPC, Calibre nmCLMPC, and Calibre MPCverify Updates

This section describes the 2021.2 changes and enhancements that apply to all Calibre® nmMPC™, Calibre® nmCLMPC, and Calibre® MPCverify applications.

See the *Calibre nmMPC and Calibre nmCLMPC User's and Reference Manual* and *Calibre MPCverify User's and Reference Manual* for details.

Calibre nmCLMPC Updates

This release adds two new commands:

- **mpc_poi_measurement** — Optimizes EPE measurement time for CLMPC by reducing the number of EPE measurements.
- **mpc_preserve_angles** — Preserves edge angles in CLMPC movement.

Calibre nmMPC Updates

This release adds a new command, mpc_skip_fragments, that omits specified fragments from EPE measurement computation.

Calibre DefectReview Updates

This section describes new features added to Calibre® DefectReview™ for this release.

See the *Calibre DefectReview User's Manual* for details.

Updates

- Defect tracking in the Defect List now supports two new options from the right mouse menu: Track Defect in Normal Mode and Track Defect in Fast Mode. The modes are determined from the input inspection report. Track Defect in Normal Mode selects Normal Mode inspection reports and Track Defect in Fast Mode selects Fast Mode inspection reports.
- The Defect List now allows certain pairs of columns to be sorted simultaneously. The Primary Classification and Defect Progress Classification and the Auto Defect Type

Classification and Auto Defect Progress Classification column pairs support two column sorting.

- The Image Measurement Unit toolbar now supports the following new buttons:
    - The **DI-SIGN/DI-ABS** button switches the display of Defect Information entries between signed or absolute values. This button is also supported in the Defect Progress Tracking window.
    - The **Create Defect Bounding Box** button displays a bounding box rectangle on Aligned Sem, Sem, Defect Contours, and Defect Bounding Box images.
    - The **Save Defect Info** button saves any manually-created defects.
- Defect Progress Tracking supports several new enhancements:
    - Defect Progress Tracking now supports tracking defects by Reflectance Residue, Transmitted Flux (T Flux), Reflected Flux (R Flux), and X-axis.
    - You can now modify the range of the Y-axis for Upper and Lower Control Limits by clicking the new **Scale Axis** button.
- The Wafer Defect Management utility now supports wafer map coloring which uses colors to indicate the status of dies on the wafer map. The status and colors used are set using the waferMapInfo node in the *dat-ini-wafer.xml* file.

Calibre DefectClassify Updates

This section describes new features added to Calibre® DefectClassify™ for this release.

See the *Calibre DefectClassify User's Manual* for details.

Updates

- The Image-to-Layout Alignment tool now supports two different types of offsets that are used for defect locations that do not have corresponding SEM images: global and local. A global offset is a common shift value applied to all defect locations without SEM images. A local offset calculates the shift for each defect location individually.

Calibre MDPAutoClassify Updates

This section describes new features added to Calibre® MDPAutoClassify™ for this release.

See the *Calibre MDPAutoClassify User's Manual* for details.

Updates

- In this release, you can now define defect dispositions. Defect dispositions are customized combinations of classification codes from other tiers, including defect and source type.

Calibre MDPDefectAvoidance Updates

This section describes new features added to Calibre® MDPDefectAvoidance™ for this release.

See the *Calibre MDPDefectAvoidance User's Manual* for details.

Updates

- You can now generate a defect avoidance parallel run report, an HTML-based report that provides information from a distributed MTflex run.
- You can now control what timers are output in the defect avoidance log file by configuring the LogLevel setting in the defect avoidance run configuration file.

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.