

SIEMENS EDA

Calibre® Metrology Interface (CMI) User's Manual

Software Version 2021.2

SIEMENS

Unpublished work. © 2021 Siemens

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

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

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

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

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

Support Center: support.sw.siemens.com

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

Table of Contents

Chapter 1	
Introduction to the Calibre Metrology Interface.....	13
Calibre Metrology Interface Input and Output	14
Calibre Metrology Interface Workflow	16
Calibre Metrology Interface Prerequisites.....	18
Chapter 2	
Using the Calibre Metrology Interface	21
CMi Workflow Overview	22
Saving and Reloading Sessions	25
Loading and Saving Jobspec Files.....	27
Chip Assembly.....	28
Assembling a Chip Over a Job Deck	28
Layer Polygon Tone	29
Marker Operations	32
Graphical Markers In the Chip Layout.....	32
Drawing Hierarchical Markers	33
Adjusting Marker Parameters	34
Reloading Markers	35
Importing Markers	35
Exporting Markers	37
Deleting Markers	38
Adjusting Marker Position (Drag Method).....	39
Marker Magnification.....	40
Failed Marker Detection	41
Browsing Failed Markers	44
Adding SEM Alignment Locations	46
Designate Multiple Measurements Per Marker.....	48
CSV Table Construction	49
Measurements in CMi	52
Multiple Measurements Per Capture	53
Grouping Measurements in a Group Marker	53
Group Measurements in a Region	54
Debugging and Optimizing Multiple Measurements Using a Region.....	55
Tolerance Specification	56
Target Value Specification	56
Searches in CMi.....	58
Searching for Measurement Locations	58
Magnification Settings for Hitachi Output	59
Defining Boolean Operations on Layers	60
Constructing a Four Corner Shape.....	62
Editing a Wafer Map	63

Overlaying SEM Images and Design Clips	68
Automatic Metrology Algorithm Identification	70
Automatic Metrology Algorithm Identification Flow	70
Pattern Library Creation	73
DFM Pattern Capture Command	73
Using the Pattern Library Manager	75
Library Creation Flow Example	77
Using Automatic Identification in CMi	78
Pattern Library Specification for Automatic Metrology Algorithm Identification	82
Multiple Algorithm Matches for the Same Feature	84
Resolving Individual Algorithm Conflicts	87
Modifying Algorithm Conflict Resolution Behavior	87
Automatically Selecting Alignment Markers Using Pattern Matching	90
Defining Contact Pattern Ratios	91
Managing Chiplets	94
Chapter 3	
CMi GUI Reference	99
The CMi Window	100
Assemble Jobdeck Options	107
Layer Options	109
Alignment Marker Options	110
Site Automatic Adjustment Options	111
Site Size Preferences	115
Output Options	116
Display Preferences	120
Sites Table and Properties	121
Layer Processor Window	125
Chiplets Manager	132
Algorithm Manager	135
Allowed Characters in Cell Names	137
Chapter 4	
CMi Command Reference	139
Executing Tcl Scripts for CMi	139
CMi Tcl Script Examples	140
CMi Tcl Commands	142
CMiSession create	146
\$cmi_session configure	147
openSession	148
readJobspec	149
readWaferMapSession	150
getAlgorithmIdentificationConfig	151
getAmatOutputConfig	152
getHitachiOutputConfig	153
getLayoutClipConfig	154
getSitesFilter	155
getLocateAlignmentConfig	156

Table of Contents

sites adjustLocation	157
sites automaticAlgorithmIdentification	158
\$cmi_session sites generateLayoutClips	159
sites locateAlignmentSites	160
sites readSamplePlan	161
sites writeResultsToAMAT	162
sites writeResultsToCSV	163
sites writeResultsToHitachi	164
layout runBooleans	165
\$cmi_session layout setLayout	166
\$layout_clip_config configure	167
\$sites_filter configure	169
\$amat_config configure	170
\$hitachi_config configure	172
\$algorithm_identification configure	174
\$sites_locator configure	176

Appendix A	
Calibre Metrology Interface (CMi) Error Messages	177
Error Messages	177
Warning Messages	178

Index

Third-Party Information

Table of Contents

List of Figures

Figure 1-1. CMi Job Flow for SEM Site Marking.....	13
Figure 1-2. Marked Locations for Measurements	14
Figure 1-3. Validated Measurement Locations With CD	15
Figure 1-4. Capture Field.....	16
Figure 1-5. GDS Clip Output.....	16
Figure 1-6. Windows Used for CMi	17
Figure 1-7. CMi Workflow	17
Figure 2-1. Calibre Metrology Interface	23
Figure 2-2. CMi Sub-Tool Session Links	26
Figure 2-3. CMi Saved Session Directory Structure	26
Figure 2-4. Assemble Jobdeck Dialog Box	28
Figure 2-5. Layer Tone Setting Example.....	30
Figure 2-6. Primary Marking Flow	33
Figure 2-7. Contact Shape Measurement Example	37
Figure 2-8. Converting Markers to SS and GG Formats.....	38
Figure 2-9. Adjusting Marker Size (Drag Method).....	40
Figure 2-10. Moving a Marker (Drag Method).....	40
Figure 2-11. New Marker Dialog Box.....	41
Figure 2-12. Failed Marker Suggested Fix	42
Figure 2-13. Adjusted Sites Highlighted	42
Figure 2-14. Accept or Decline Autofix Changes	43
Figure 2-15. Wrong Marker Orientation	43
Figure 2-16. Marker Size Larger than Target	44
Figure 2-17. Empty Space	44
Figure 2-18. Using the Find Object Dialog Box	45
Figure 2-19. Using Show Find Toolbar	45
Figure 2-20. SEM Alignment Markers	47
Figure 2-21. Set Marker Window Width and Size	47
Figure 2-22. New Marker With Kind Specified	48
Figure 2-23. Kind Value in the Machine Area	49
Figure 2-24. CSV Table.....	49
Figure 2-25. Example of a CSV for a Hierarchical Cell	50
Figure 2-26. Grouped Markers	54
Figure 2-27. Example Results	54
Figure 2-28. Example CSV Table With Search Highlighted.....	58
Figure 2-29. Layer Processing Options	60
Figure 2-30. Layer Processor Window Example.....	61
Figure 2-31. Input Layer Specification	61
Figure 2-32. Options to Specify Four Corner Construction	62
Figure 2-33. Four Corner Construction	63

Figure 2-34. CMi Role in SEM Recipe Generation.	64
Figure 2-35. Wafer Map Manager	65
Figure 2-36. Adding Runs from the Sites Table	66
Figure 2-37. Highlighting in the Runs Tab	67
Figure 2-38. CMi With SEM Loaded	68
Figure 2-39. Pattern Creation Flow	71
Figure 2-40. CMi Automatic Algorithm Identification Flow	72
Figure 2-41. DFM Pattern Capture Command Flow	74
Figure 2-42. Pattern Library Manager File Menu	76
Figure 2-43. Pattern Library Manager Edit Menu	77
Figure 2-44. Conflict Highlighted	79
Figure 2-45. Markers Not Assigned Algorithms Highlighted	80
Figure 2-46. Metrology Algorithm Automatic Identification Flow	81
Figure 2-47. Example Map File	82
Figure 2-48. Pattern Matching Run With Count Markers	84
Figure 2-49. Enabling Count Markers	85
Figure 2-50. Pattern Matching Markers	86
Figure 2-51. Pattern Matching Result Marker	86
Figure 2-52. Algorithm Conflict Highlighted in CMi	87
Figure 2-53. Sites With Common Algorithms Clustered	88
Figure 2-54. Algorithm Manager Example	89
Figure 2-55. Alignment Marker Selection in CMi	90
Figure 2-56. Contact Pattern and Ratio Calculation	91
Figure 2-57. Contact Ratio Option	92
Figure 2-58. Contact Ratio Editor	92
Figure 2-59. Chiplet Manager Work Flow	95
Figure 2-60. Generating a Recipe With Chiplets	95
Figure 2-61. Chiplets Manager	96
Figure 2-62. Open File	96
Figure 2-63. Add Chiplet Transforms	96
Figure 2-64. Delete Chiplet Transforms	97
Figure 2-65. Save Transforms	97
Figure 2-66. Chiplet Output Flow	97
Figure 3-1. Calibre Metrology Interface Main Window	100
Figure 3-2. Show or Hide Bar	101
Figure 3-3. Assemble Jobdeck Options Dialog Box	107
Figure 3-4. Map File Defaults	113
Figure 3-5. Sites Table and Properties	121
Figure 3-6. Sites Table Operation	122
Figure 3-7. Sorting the Sites Table	122
Figure 3-8. Layer Processor Window	125
Figure 3-9. AND/OR Operations in the Input Layers List Dialog Box	127
Figure 3-10. NOT and XOR Operations in the Input Layers List Dialog Box	127
Figure 3-11. Generated Boolean Clips	128
Figure 3-12. Input Layers List	128

List of Figures

Figure 3-13. Add New Layer Dialog Box	129
Figure 3-14. Chiplets Manager Window	132
Figure 3-15. Algorithm Manager.	135
Figure 4-1. Defaults for Map File	174

List of Tables

Table 1-1. Related Products and Their Manuals	18
Table 2-1. Hitachi HSS Template Assignments Based on Layer Tone	31
Table 3-1. Calibre Metrology Interface Fields	101
Table 3-2. Assembly Options	107
Table 3-3. Layer Options	109
Table 3-4. Alignment Marker Selection Options	110
Table 3-5. Site Automatic Adjustment Options	111
Table 3-6. Site Size Preference Options	115
Table 3-7. Output Options	116
Table 3-8. Display Preferences Options	120
Table 3-9. Sites Table Fields	122
Table 3-10. Controls for the Layer Processor Window	125
Table 3-11. Characters Allowed/Not Allowed in Cell Names	137
Table 4-1. Executing Tcl Scripts in CMi	140
Table 4-2. Summary of CMi Commands	142
Table A-1. CMi Error Messages	177
Table A-2. CMi Warning Messages	178

Chapter 1

Introduction to the Calibre Metrology Interface

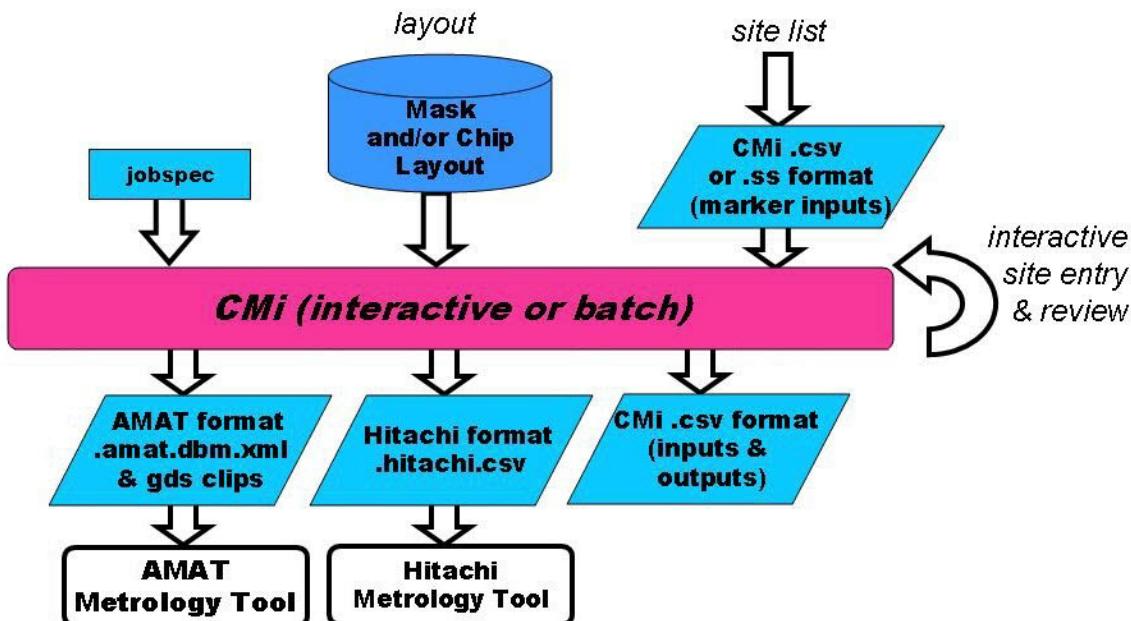
The Calibre® Metrology Interface (CMi) is a Calibre® WORKbench™-based tool created as a design-based off-line marking system for SEM measurements. Use CMi to graphically enter sites and generate a target list in a number of different metrology machine formats.

CMi is a versatile interface to a metrology system for the selection of meaningful measurement locations from the design layout data. The complete system is composed of CMi (an automated method to generate measurement input files), an automated measurement process, and automated analysis and result reporting on the metrology system.

Multiple methods are provided for the user to identify measurement sites from previous analysis, or interactive marking. Once identified, the positions of those features can be used to set up the measurement process. This motivates the development of a direct link from the design data into the measurement setup. Throughput becomes a primary concern as the demand for CD measurements increases. CMi improves efficiency by automating setup and measurement.

[Figure 1-1](#) illustrates the basic job flow for SEM site marking.

Figure 1-1. CMi Job Flow for SEM Site Marking



Calibre Metrology Interface Input and Output	14
Calibre Metrology Interface Workflow.....	16

Calibre Metrology Interface Prerequisites	18
---	----

Calibre Metrology Interface Input and Output

The two major inputs to CMi are measurement locations and the mask layout. The sources of measurement locations are varied. For example, tools like Calibre® OPCverify™ may identify “hot-spot” locations from analysis of the layout, or designers may pass on layout knowledge in a written list of coordinates.

The sources have been reduced to three basic alternatives for input of locations into CMi:

- Interactive drawings
- A written coordinate list
- GDS file shapes

These alternatives provide convenient interfaces to use the results from Calibre OPCverify, and a variety of other Calibre analysis tools. For example, Calibre WORKbench can be used interactively with CMi to visually identify measurement locations using markers.

Figure 1-2 shows a mask pattern (in blue) displayed in Calibre WORKbench where locations have been marked for CD measurements (indicated by red boxes).

Figure 1-2. Marked Locations for Measurements

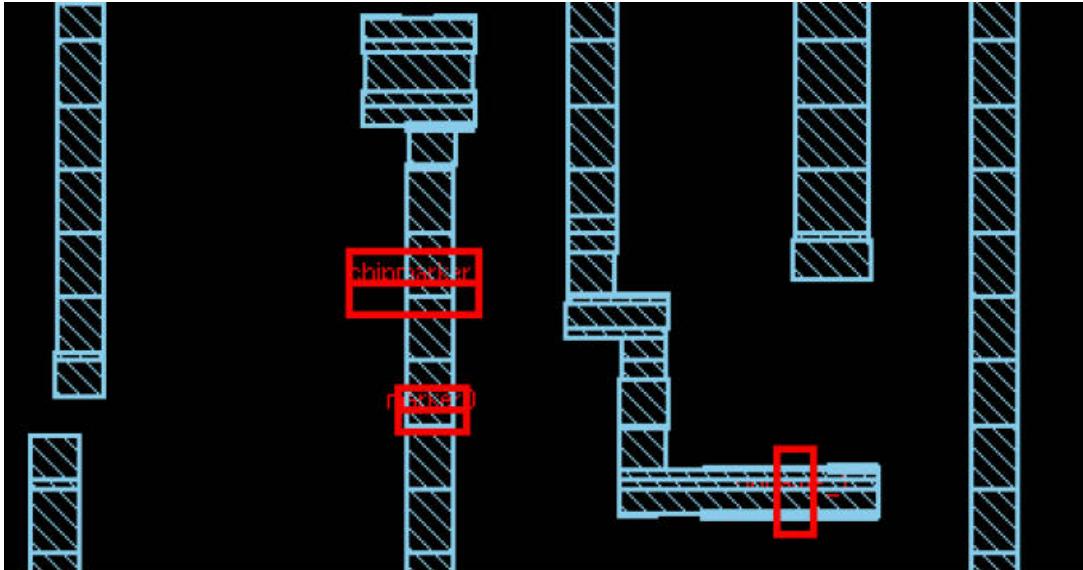
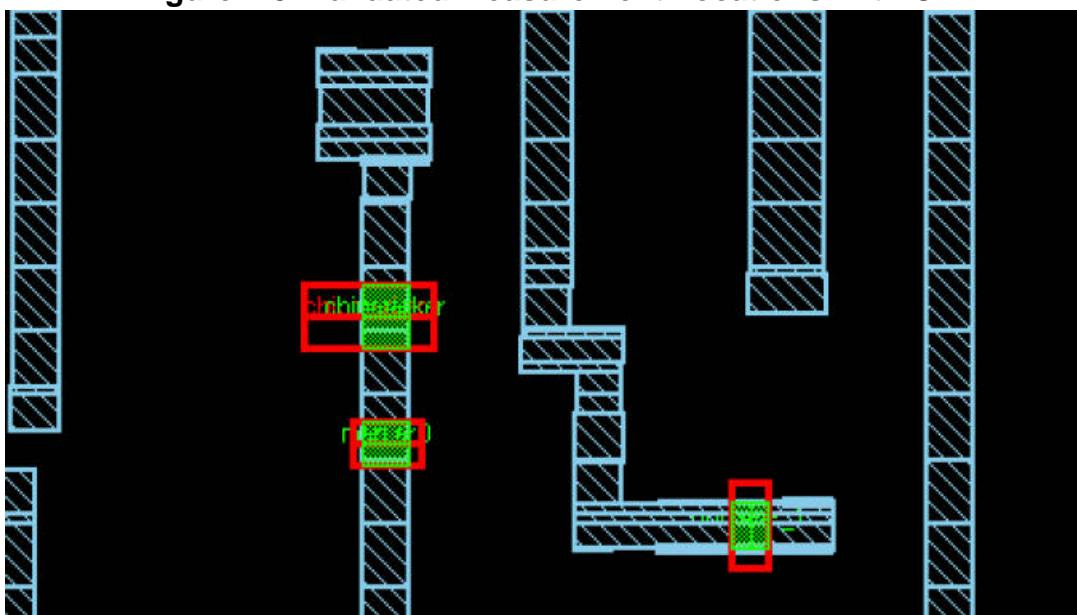


Figure 1-3 shows the validated measurement locations (displayed in green boxes) with CD extracted from the layout.

Figure 1-3. Validated Measurement Locations With CD



The output from CMi can be one or more of the following:

- Applied Materials CD-SEM (AMAT) input
- Carl Zeiss AIMS™ measurement input file written in XML

Note

 AIMS support is deprecated for the Calibre Metrology Interface and is obsoleted as of the 2019.3 release.

- Hitachi CD-SEM input
- GDS or OASIS®¹ clip
- CVS output (this is recommended to preserve hand-drawn markers)

The CMi output includes the specifications of all capture and measurement locations. Group and region designations are attached to the locations, and then transferred to the measurement tool to facilitate further analysis.

The region around each location which will be imaged by the measurement tool is called a Capture Field. If a layer number has been assigned in CMi, these regions can be observed graphically during post-output generation review.

Figure 1-4 shows an example Capture Field rendered for multiple CD measurements (the brown square).

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.

Figure 1-4. Capture Field

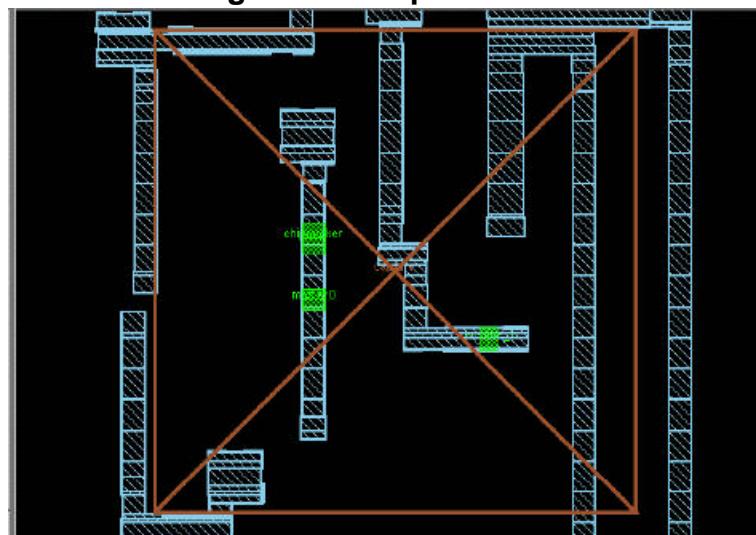
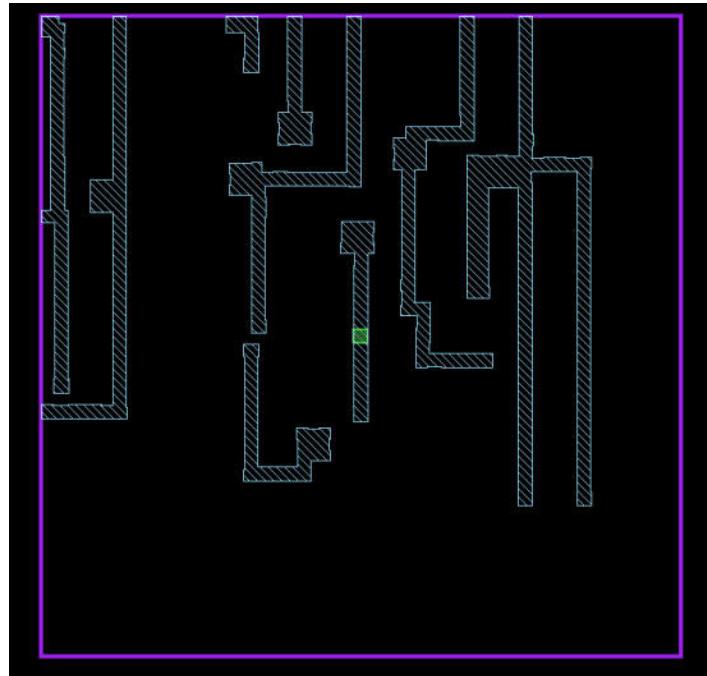


Figure 1-5 shows an example GDS Clip output.

Figure 1-5. GDS Clip Output

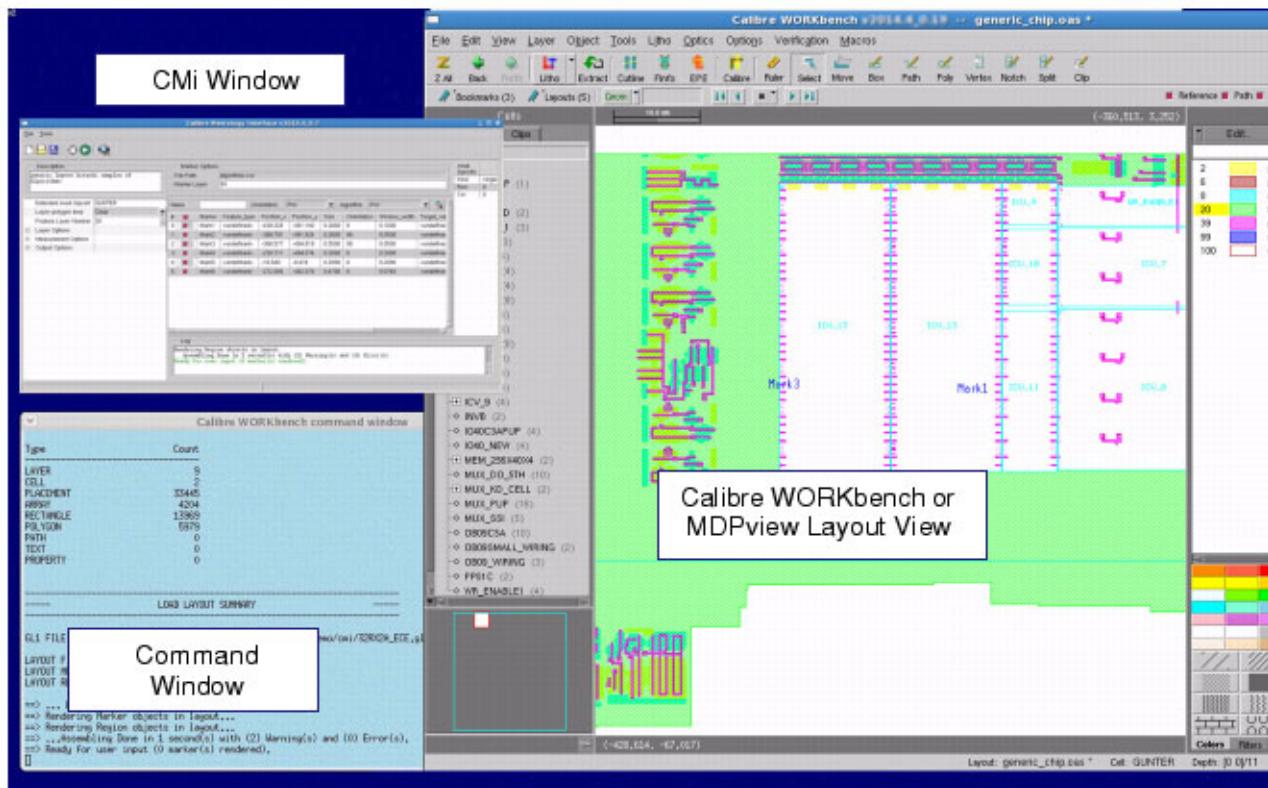


Calibre Metrology Interface Workflow

The Calibre Metrology tool is used interactively with other Calibre tools.

This is illustrated in [Figure 1-6](#).

Figure 1-6. Windows Used for CMi



The basic workflow for the Calibre Metrology Interface can be summarized into the following basic stages:

Figure 1-7. CMi Workflow



1. Invoke Calibre WORKbench with the **-mapi** command line option.

```
<calibre_home>/bin/calibrewb -mapi
```
2. In Calibre WORKbench, select **Tools > Calibre Metrology Interface**.
3. **Load Mask Layout:** Load the mask layout and input job configuration.
4. **Draw or Import Markers:** Manually draw markers or import existing markers or regions.
5. **Generate CMi Output:** Click **Generate Output** to generate all output files. This generates all output files (and is required after all marker entry and algorithm edits).

6. Transfer the files to SEM.

The workflow is further described in “[Using the Calibre Metrology Interface](#)” on page 21.

Calibre Metrology Interface Prerequisites

Prior to using the Calibre Metrology Interface, there are several prerequisites.

- **Platform support** — The Calibre Metrology Interface is available on all supported platforms found in the [Calibre Administrator’s Guide](#). Refer to that document for instructions on how to install Calibre software.
- **Licensing** — Valid licenses for your FRACTURE formats and MDP utilities. The following licenses are required:
 - A Calibre WORKbench or Calibre MDPview license
 - A Calibre Metrology API (MAPI) license

For more information on licensing, refer to the [Calibre Administrator’s Guide](#).

Table 1-1 lists the documents associated with the related Calibre tools.

Table 1-1. Related Products and Their Manuals

Related Products	Documentation
All post-tapeout products	Calibre Post-Tapeout Flow User’s Manual
Calibre® FRACTUREc™	Calibre Mask Data Preparation User’s and Reference Manual
Calibre® FRACTUREh™	Calibre Release Notes
Calibre® FRACTUREi™	
Calibre® FRACTUREj™	
Calibre® FRACTUREm™	
Calibre® FRACTUREn™	
Calibre® FRACTUREp™	
Calibre® FRACTUREt™	
Calibre® FRACTUREv™	
Calibre® MDPmerge™	
Calibre® MDPstat™	
Calibre® MDPverify™	
Calibre® MPCpro™	
Calibre® MASKOPT™	
Calibre® MDP Embedded SVRF	

Table 1-1. Related Products and Their Manuals (cont.)

Related Products	Documentation
Calibre® nmMPC™ Calibre® nmCLMPC Calibre® nmOPC™	<i>Calibre nmMPC and Calibre nmCLMPC User's and Reference Manual</i> <i>Calibre nmOPC User's and Reference Manual</i>
Calibre® MDPview™	<i>Calibre MDPview User's and Reference Manual</i> <i>Calibre DESIGNrev Layout Viewer User's Manual</i> <i>Calibre Release Notes</i>
Calibre® nmDRC™ Calibre® nmDRC-H™	<i>Calibre Release Notes</i> <i>Calibre Verification User's Manual</i> <i>Standard Verification Rule Format (SVRF) Manual</i>
Calibre® WORKbench™	<i>Calibre WORKbench User's and Reference Manual</i>
Tcl/Tk Batch Commands	<i>Calibre DESIGNrev Reference Manual</i>
Calibre® Metrology API (MAPI)	<i>Calibre Metrology API (MAPI) User's and Reference Manual</i>
Calibre® Metrology Interface	<i>Calibre Metrology Interface (CMi) User's Manual</i>
Calibre® Job Deck Editor	<i>Calibre Job Deck Editor User's Manual</i>
Calibre® MDPDefectAvoidance™	<i>Calibre MDPDefectAvoidance User's Manual</i>
Calibre® MPCverify	<i>Calibre MPCverify User's and Reference Manual</i>
Calibre® DefectReview™	<i>Calibre DefectReview User's Manual</i>
Calibre® MDPAutoClassify™	<i>Calibre MDPAutoClassify User's Manual</i>
Calibre® DefectClassify™	<i>Calibre DefectClassify User's Manual</i>

Chapter 2

Using the Calibre Metrology Interface

The Calibre Metrology Interface contains a number of operations to support SEM measurements and design-based off-line marking.

CMi Workflow Overview	22
Saving and Reloading Sessions	25
Loading and Saving Jobspec Files	27
Chip Assembly	28
Assembling a Chip Over a Job Deck	28
Layer Polygon Tone	29
Marker Operations	32
Graphical Markers In the Chip Layout	32
Drawing Hierarchical Markers	33
Adjusting Marker Parameters	34
Reloading Markers	35
Importing Markers	35
Exporting Markers	37
Deleting Markers	38
Adjusting Marker Position (Drag Method)	39
Marker Magnification	40
Failed Marker Detection	41
Browsing Failed Markers	44
Adding SEM Alignment Locations	46
Designate Multiple Measurements Per Marker	48
CSV Table Construction	49
Measurements in CMi	52
Multiple Measurements Per Capture	53
Tolerance Specification	56
Target Value Specification	56
Searches in CMi	58
Searching for Measurement Locations	58
Magnification Settings for Hitachi Output	59
Defining Boolean Operations on Layers	60
Constructing a Four Corner Shape	62
Editing a Wafer Map	63
Overlaying SEM Images and Design Clips	68

Automatic Metrology Algorithm Identification	70
Automatic Metrology Algorithm Identification Flow	70
Pattern Library Creation	73
Using Automatic Identification in CMi	78
Pattern Library Specification for Automatic Metrology Algorithm Identification	82
Multiple Algorithm Matches for the Same Feature	84
Resolving Individual Algorithm Conflicts	87
Modifying Algorithm Conflict Resolution Behavior	87
Automatically Selecting Alignment Markers Using Pattern Matching	90
Defining Contact Pattern Ratios	91
Managing Chiplets.....	94

CMi Workflow Overview

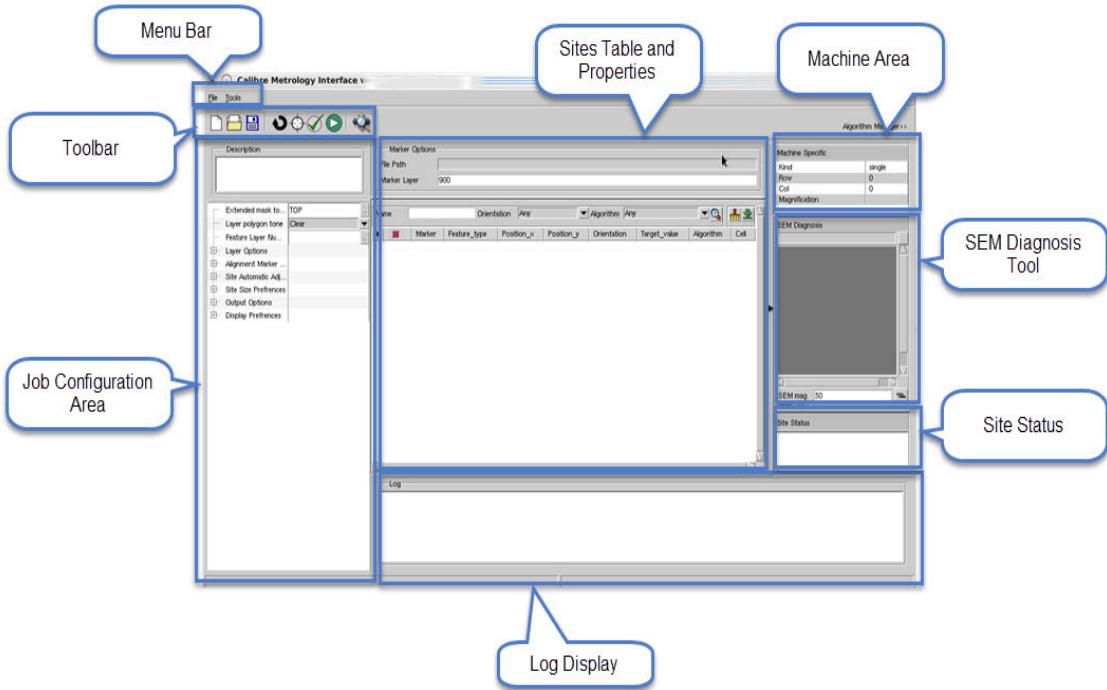
The basic workflow for the Calibre Metrology Interface uses CMi in conjunction with Calibre WORKbench (or Calibre MDPview) to graphically enter sites and generate a target list in different metrology machine formats.

Prerequisites

This workflow presumes that a number of steps have been followed.

- Invoke Calibre WORKbench with the -mapi command line option, in addition to any command line arguments you might also require.
`<calibre_home>/bin/calibrewb -mapi`
- See the [Calibre WORKbench User's and Reference Manual](#) for complete information on command line arguments.
- In Calibre WORKbench, select **Tools > Calibre Metrology Interface** to invoke CMi.

Figure 2-1. Calibre Metrology Interface



- The mask layout should be loaded into Calibre WORKbench (or Calibre MDPview).

Procedure

1. In the Calibre Metrology Interface window, perform one of the following:
 - Select **File > New Session** to start a new CMi session. See “[Saving and Reloading Sessions](#)” on page 25 for information on CMi sessions.
 - Select **File > Open Session** to import a previously-saved CMi session file.
 - Select **File > Load Jobspec** to load a previously-saved jobspec file (XML or Tcl). See “[Loading and Saving Jobspec Files](#)” on page 27 for further information.
2. Add markers to your layout in Calibre WORKbench. If you already have markers from a previous CMi session, you can skip this step. There are different methods available to add markers to your layer.
 - **Import Markers** — Select **File > Import Markers** to import markers from a marker layer, CSV file, super spreadsheet, gauge, or RDB file. The markers are drawn in the layout from text files and populate the Sites table (see “[Sites Table and Properties](#)” on page 121). If the source of the marker is a marker layer, the layout is scanned to get existing markers to populate the table.

You can optionally click **Scan Markers Algorithm and Type** to scan all markers in the layout to check the feature type (either width or space) and the algorithms assigned to the markers in this job.

- **Draw Markers on the Layout** — You can add markers directly to the layout loaded into Calibre WORKbench on the marker layer. It is recommended that you enable the Select tool in the toolbar of the Layout view before using the New Marker or Edit Marker tools. Otherwise, some unintended operations may occur.
 - i. In Calibre WORKbench or Calibre MDPview, place the cursor at a measurement center point, then press Ctrl and double-left-click to invoke the New Marker dialog.
 - ii. In the New Marker dialog box, enter a Marker Name, and adjust the values for Center X, Center Y, Orientation, Window Width, Size, and Algorithm.
 - iii. Click **OK**. Marker information can be viewed in the Sites table (see “[Sites Table and Properties](#)” on page 121).

To edit an existing Marker, select the marker entry in the Sites table and click on any of the columns to edit values. Marker operations are covered in more detail in “[Marker Operations](#)” on page 32.

3. In the Job Configuration Area, specify the job parameters (including layout, measurement, and output options). You must minimally specify parameters for the Extended mask topcell and Feature Layer Number.
4. You can optionally add a chip to the job deck for assembly by performing the following:
 - a. Select **Tools > Assemble Jobdeck**.
 - b. In the Assemble Jobdeck dialog box, enter chip parameters such as the Chip File Name, Chip Topcell, Mask Chip Cell Name, as well as fracture parameters.
 - c. Click **OK**. The chip is added to the mask layout.

Further description of the assembly process are described in “[Chip Assembly](#)” on page 28.

5. Click **Generate Output** to generate all output files. This generates all output files (and is required after all marker entry and algorithm edits).

Error, Warning, and Info messages appear in the log window and the command window (the terminal where Calibre WORKbench was invoked). Some messages also appear in the status line at the bottom of the Calibre Metrology Interface GUI. More severe errors appear as popup windows. The command window should be monitored for correct operation. Refer to “[Calibre Metrology Interface \(CMi\) Error Messages](#)” on page 177 for a complete listing of possible messages.

The results are put into a directory specified by the user’s choice of Output String:

- *results/output_string*

Any previous results for that directory are moved to a date-coded backup directory:

- results/output_string_yymmdd_hhmmss

You can use **Generate Output** as an optional point-by-point for review (extra results can be discarded).

This operation validates sites against the layout geometries and reduces user effort for precise locations. You can activate or deactivate this function in the CMi GUI.

- Confirms exactly 2 parallel edges in the marker (for Touching Edge method of validation).
- Adjusts the center coordinate.
- Extracts the layout feature width.
- Draws the result on the Measurement layer for success, or on the Failed Markers layer.
- Users can **Generate Output** repeatedly, after drawing each marker.

6. **Transfer the files to SEM.**

7. Save the CMi session by selecting **File > Save Session** or **Save Session As**. The CMi session is saved as a directory. The job parameters file (jobspec) is saved as an XML file.

Saving and Reloading Sessions

You can save the current state of your Calibre Metrology Interface session and reload it at any time. This enables you to restore the exact state of your work at the point that you save.

The following information can be saved into an XML-format file:

- Job configuration file path. If you used a Jobspec file from previous CMi versions, the session file will save the location.
- Job parameters.
- Layout path for the layout loaded in CWB window.
- Sites source information.
- Source type.
- Path to the file or layout marker layer number.
- Hitachi Template file path. This is a path to a file generated by the Algorithm Manager.
- Boolean operations file path. This is a path to a file generated from the [Layer Processor Window](#).

- Links to CMi sub-tool session files. Several sub-tools used by CMi have their own states saved in a session file. The main CMi session file saves the links to these files, even if they are not in the same CMi session directory. The CMi sub-tools are:
 - Boolean Layer Processing
 - Contact Ratio
 - Chiplets Manager
 - Algorithm Manager
 - Wafer Map

The following figure shows an example of the links generated in the main CMi session file.

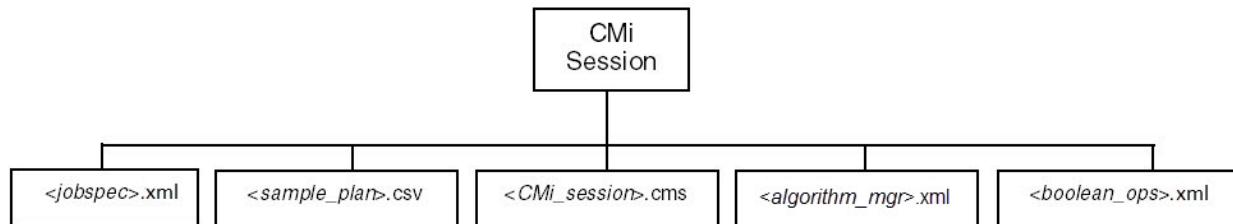
Figure 2-2. CMi Sub-Tool Session Links

```
mljf.json
{
    "Input_job_parameters": "cmi_multi_match/jobspec.xml",
    "Layout_path": "/d2s/d2s_calibre_eng_data1/semi_mfg_eng/MDP/CMi/GF_2017/Algorithm_conflict_resolution/design/test_multi_match.oas",
    "site_source_type": "csv",
    "site_source": "/d2s/d2s_calibre_eng_data1/semi_mfg_eng/MDP/CMi/GF_2017/Algorithm_conflict_resolution/cmi_multi_match/sample_plan.csv",
    "wafer_map_file": "",
    "chiplets_file": "",
    "algorithm_mgr": "/d2s/d2s_calibre_eng_data1/semi_mfg_eng/MDP/CMi/GF_2017/Algorithm_conflict_resolution/alg_session_tst3.xml",
    "contact_ratio": "/d2s/d2s_calibre_eng_data1/semi_mfg_eng/MDP/CMi/GF_2017/Algorithm_conflict_resolution/cmi_multi_match_tst_3/contact_pattern_ratio.xml",
    "layer_processing": ""
}
```

Procedure

1. You can start a blank CMi session by selecting **File > New Session**.
2. To save your current CMi session, select **File > Save Session** or **Save Session As** (to rename an existing session). When you save a session, a directory structure is created as shown in the following figure.

Figure 2-3. CMi Saved Session Directory Structure



The CMi session directory can contain the following files:

- **<CMi_session>.cms** — A CMi session file that defines the paths of the inputs to CMi Design layout, jobspec template file, and Sites source.
- **<jobspec>.xml** — Job configuration XML file as defined in the latest state in the GUI.
- **<sample_plan>.csv** — A list of all the sites in the current session.

- *<algorithm_mgr>.xml* — An Algorithm Manager file containing metrology names for each default algorithm.
 - *<boolean_ops>.xml* — If any boolean operations have been defined with the Layer Processor Window, they are saved automatically to this file.
3. To reload a CMi session, select **File > Reload Session** and select the existing session file.

Loading and Saving Jobspec Files

Job specification (jobspec) files are Tcl files that contain parameters for Calibre Metrology Interface runs. These were utilized by previous versions of CMi and are now maintained for backwards compatibility. They are no longer required as the same information is now saved in a CMi session file instead.

Procedure

1. To load a previously-existing jobspec file, select **File > Load Jobspec** file.
2. When a dialog box appears, select the jobspec file to load and click **OK**. If the loaded jobspec is saved by the current version of the Calibre Metrology Interface, then it includes the layout path and marker source information. Loading this jobspec automatically loads the layout to Calibre WORKbench, add markers to the layout, and populates the table.
3. To save updates to a jobspec file, select **File > Save Jobspec Template**. However, layout or site information is not saved, as this is a template instead of a session file.

Chip Assembly

An optional step in the Calibre Metrology Interface is to add a chip to your mask layout for assembly.

Assembling a Chip Over a Job Deck	28
Layer Polygon Tone.....	29

Assembling a Chip Over a Job Deck

You can add a chip to your mask layout in the Calibre Metrology Interface using the **Tools > Assemble Jobdeck** option.

Prerequisites

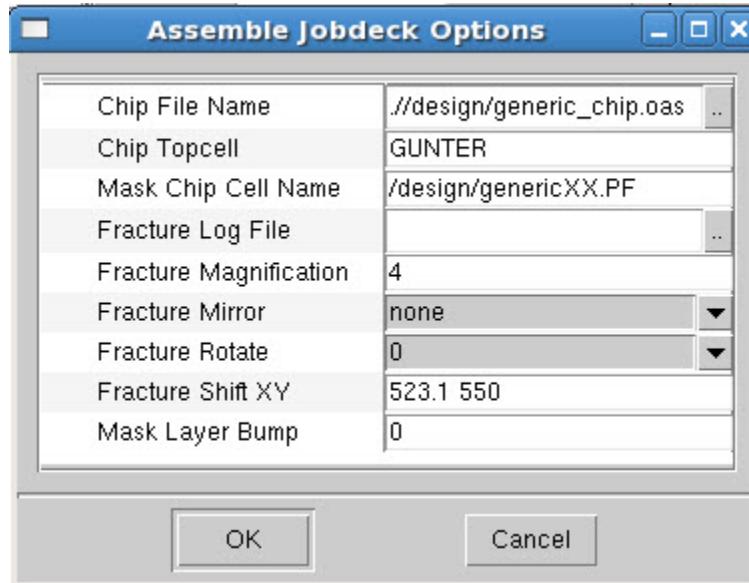
Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. In the CMi window, in the Job Parameters Area, enter the Extended Mask top cell.
2. For Hitachi HSS CSV formats, select a Layer Polygon Tone, either Clear or Dark. See “[Layer Polygon Tone](#)” on page 29 for information.
3. Select **Tools > Assemble Jobdeck**. The Assemble Jobdeck Options dialog box appears.

Figure 2-4. Assemble Jobdeck Dialog Box



4. In the Assemble Jobdeck Options dialog box, manually enter or browse to the chip file for the Chip File Name entry (OASIS® or GDSII format).
5. In the Assembly Jobdeck Options dialog box, set the following parameters:
 - Chip Top cell name: Enter the top cell name of the chip.
For version 2.0 and previous, the chip top cell name can be determined by opening the chip layout file.
 - Mask Chip cell name: The cell names can be observed from the job deck when loaded into Calibre WORKbench.
 - Fracture options and transformations: Though some fields can be determined through trial-and-error, it is much easier to examine the parameters that were used during fracture.

The fracture parameters can be loaded from a Calibre fracture log file. If the fracture log file is not available, then these values must be manually entered

The transformations specified in the job deck are automatically handled. The transformations from fracture are not explicitly available in the job deck, pattern files, or chip layout. Some assistance is provided for magnification and shift after assembly. The suggested values are not always accurate, since they are determined from the extent of the data, rather than the actual fracture window. Mirror and rotation can be more difficult; they should be set correctly before the XY shift.

For assembly of a single layout input, the “Mask file name” is sufficient. If the mask file is a MEBES job deck, the design layout can be overlaid on the mask to give access to the design hierarchy and additional layers.

Note that the shift value is an XY pair equal to the negative of the lower-left corner of the fracture extent (only in the case of mirror “none” and rotate 0). This is in design scale (before magnification). The mirror (0 or X) and rotate (0, 90, 180, 270) are entered in the conventions of the layout viewer as the conventions of fracture are different.

6. Enter a Mask Layer bump value (such as 1000) which avoids collision of the mask layer (usually 1) with the chip file layers, which may also include layers 1 or 0.
7. It is important to confirm the alignment of the chip layout over the job deck graphically to avoid errors.

Layer Polygon Tone

The Calibre Metrology Interface enables you to set the tonality of the measured layer formats. Layer tonality specifies if the polygon defines a feature (width) or a space. This is needed to correctly assign an MP Template name in the Hitachi HSS CSV format.

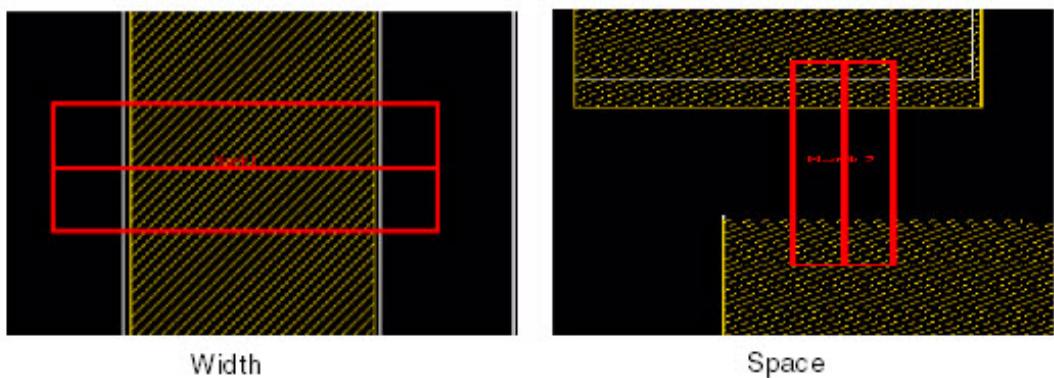
Use the **Layer Polygon Tone (Line On Wafer)** control on the Calibre Metrology Interface to set the layer tone of the layer to be measured, either Clear or Dark. Once selected, CMi will then

set the Template Name for each measurement point in the HSS spreadsheet by using the algorithm value, layer tonality and the feature type that will be measured at that point.

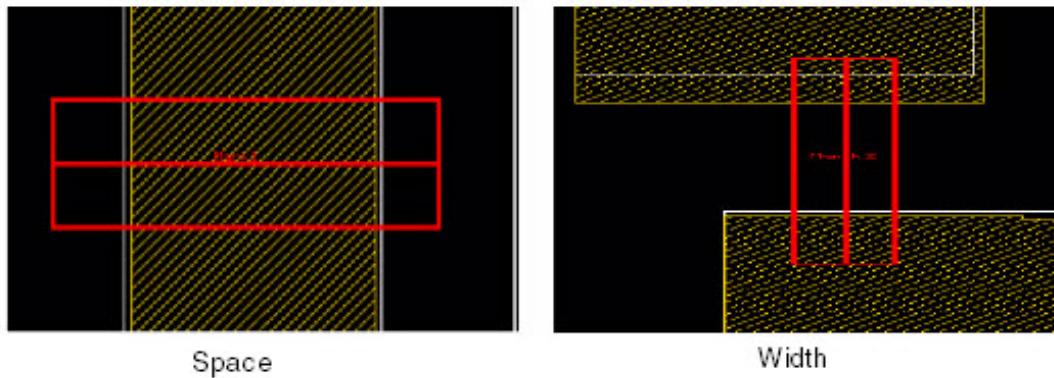
If the Layer tone is set to Clear, all targets measured on a polygon are considered to be features (width) and all other between polygons are a space. If set to Dark, then all targets measured on the polygon are treated as a space and all other between polygons are considered features (width). This is summarized in [Figure 2-5](#).

Figure 2-5. Layer Tone Setting Example

If the Layer Tone is set to Clear, Width and Space are defined as follows:



If the Layer Tone is set to Dark, Width and Space are defined as follows:



[Table 2-1](#) summarizes the specific MP Templates that are assigned based on the relevant algorithm and layer tone settings.

Table 2-1. Hitachi HSS Template Assignments Based on Layer Tone

Algorithm	Layer Tone	Feature Type	MP Template
Ave	Dark	Space	MP_SPACE_AVE
		Width	MP_WIDTH_AVE
	Clear	Space	MP_SPACE_AVE
		Width	MP_WIDTH_AVE
Min	Dark	Space	MP_SPACE_MIN
		Width	MP_WIDTH_MIN
	Clear	Space	MP_SPACE_MIN
		Width	MP_WIDTH_MIN
Max	Dark	Space	MP_SPACE_MAX
		Width	MP_WIDTH_MAX
	Clear	Space	MP_SPACE_MAX
		Width	MP_WIDTH_MAX
Image	Dark	NA	NA
	Clear		
Contour	Dark	NA	NA
	Clear		
Contact	Dark	Space	MP_SPACE_RADIAL
		Width	MP_WIDTH_RADIAL
	Clear	Space	MP_SPACE_RADIAL
		Width	MP_WIDTH_RADIAL
Ellipse	Dark	Space	MP_SPACE_ELLIPSE
		Width	MP_WIDTH_ELLIPSE
	Clear	Space	MP_SPACE_ELLIPSE
		Width	MP_WIDTH_ELLIPSE

Marker Operations

This section describes marker-based operations in the Calibre Metrology Interface.

Graphical Markers In the Chip Layout	32
Drawing Hierarchical Markers	33
Adjusting Marker Parameters	34
Reloading Markers	35
Importing Markers	35
Exporting Markers	37
Deleting Markers	38
Adjusting Marker Position (Drag Method).....	39
Marker Magnification.....	40
Failed Marker Detection.....	41
Browsing Failed Markers	44
Adding SEM Alignment Locations	46

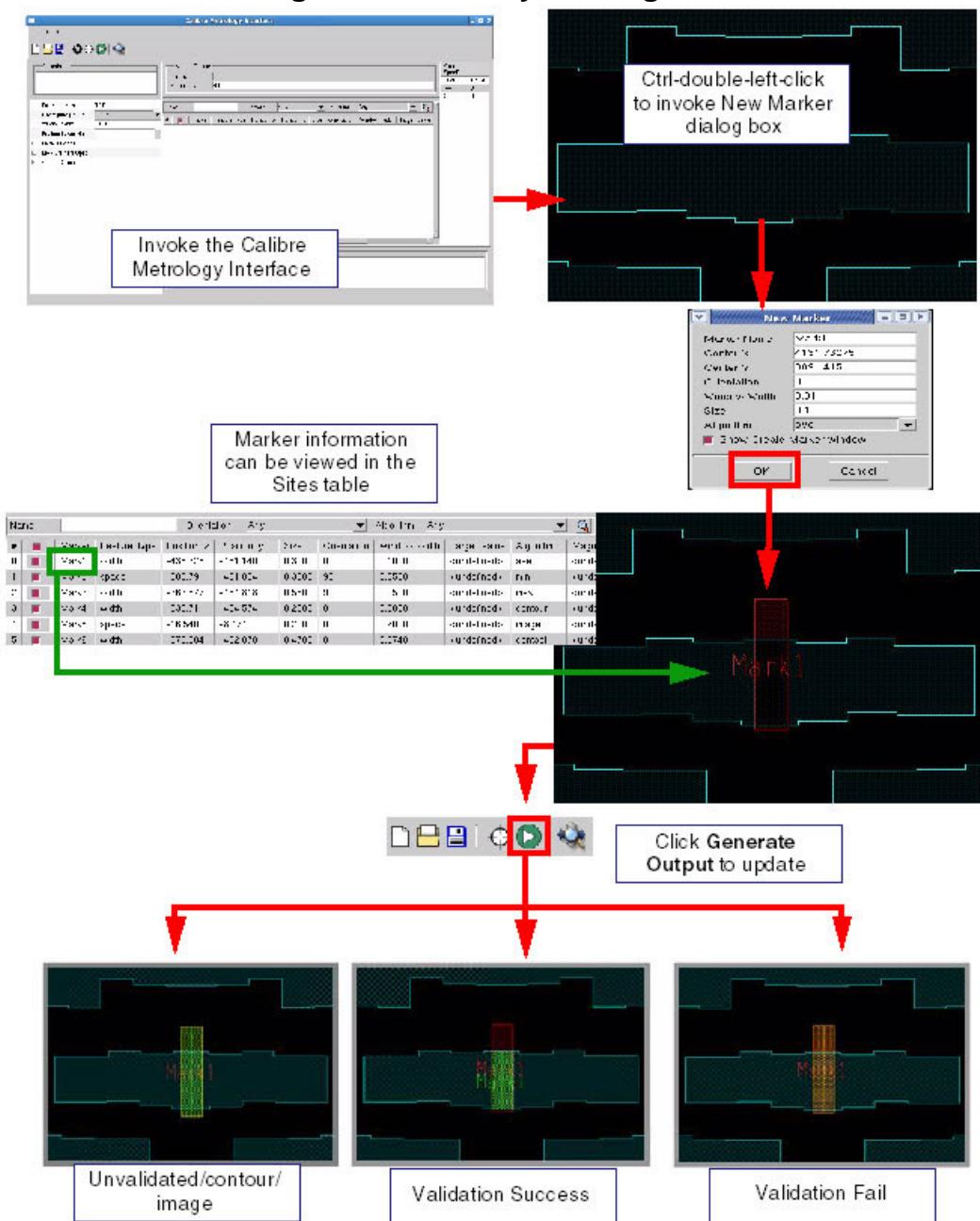
Graphical Markers In the Chip Layout

Graphical markers can be drawn in any of the layout files which are assembled by CMi. They may be drawn in any hierarchical cell.

Names may be applied by placing text on or within the marker boundary. Unnamed markers are given arbitrary names by CMi of the form “noname_#”. Marker names may be edited graphically after assembly, and before pressing the **Generate Output** button. Multiple names within the same marker are resolved where possible by use of the closest to the region boundary; a warning is issued for multiple names.

The basic marker flow process is illustrated in [Figure 2-6](#).

Figure 2-6. Primary Marking Flow



Drawing Hierarchical Markers

For hierarchical markers, when you add a marker in a cell that is not the TOP cell, the Calibre Metrology Interface automatically identifies its position in the hierarchy.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. In Calibre WORKbench, enable “Reference” selections located in the toolbar above the Layers palette.
2. Set the **View Depth** to 0 see the chip cell outline, or greater to see sub cells.
3. Select the desired instance with a left-click to get the highlight of the instance outline.
4. Push into the sub cell instance:
 - a. Right-click in the view area to invoke the pull-down menu, then select **Set as Context**.
 - b. Observe the sub cell name in the bottom status line. It will have the following syntax:

```
Cell: TOP (markers_patternname)
```
 - c. To see the shapes on the feature layer, zoom to the desired position and increase to the full view depth.
 - d. Add a marker on the marker layer (9 or 99). To add a marker, point the cursor the location on the layout where you want to insert the marker, then Ctrl-double-left-click to invoke the New Marker dialog box. Enter the marker parameters and click **OK**. The marker appears in the layout and an entry appears in the Sites table in CMi. See “[Sites Table and Properties](#)” on page 121.
 - e. Optionally set context back to the top using **Edit > Clear Context** (or Alt + C).
 - f. Observe the marker as it appears in each instance of that cell on the mask.
5. In CMi, click **Generate Output**.

Adjusting Marker Parameters

You can use the Calibre Metrology Interface to adjust the parameters for your markers.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface

- A mask layout
- A marker list loaded into CMi

Procedure

1. Select the desired marker row in the CMi Sites table (see “[Sites Table and Properties](#)” on page 121). The selection is indicated by lighter color background in the row.
2. Click on any of the columns to adjust the value. The changes will be made in the layout and table.

Reloading Markers

You can reload preexisting markers in CMi as part of an error recovery function. For example, you accidentally imported markers to an incorrect layout file and you need to import the same marker list to another layout file instead.

Prerequisites

Before starting this procedure, you should have:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout
- A marker list that was previously loaded into CMi

Procedure

In CMi, select **Tools > Reload Markers**. The markers listed in the Sites table (see “[Sites Table and Properties](#)” on page 121) are reloaded from your original marker source.

Examples

For example, you have two layouts (layout1 and layout 2) loaded into Calibre WORKbench, and you are importing markers. You intended to import markers to layout1, but you accidentally imported them to layout2 instead, as it was the one currently displayed in Calibre WORKbench.

In this case, you can select layout to display in Calibre WORKbench from the **Layouts** drop-down list, then select **Tools > Reload Markers** in CMi to reload the imported marker list to layout1.

Importing Markers

You can import markers from a number of different sources.

Procedure

1. In Calibre CMi, select **File > Import Markers** and select one of the following types:
 - **Marker Layer** — Directly from a marker layer.
 - **CS** — The most common marker input format for general application. Markers, Regions, and Searches may be imported from a CSV format table. This table format is like the CSV output format. It is easiest to create an input file by editing a previous output file.
 - **SS** — Spreadsheet format for interfacing from other test pattern metrology tools.
 - **Gauges** — A gauge is a drawn line along which a CD or a space measurement is performed. A gauge data file (.gd) is similar to a spreadsheet (.ss) file in that both contain CD or space measurements.
 - **Rdb** — ASCII format written from a Calibre job, commonly used for the Calibre OPCverify hotspot flow.
 - **XY List** — Enables the import of a list of marker XY coordinates that represents the measurement location. The tool automatically identifies the best measurement point around the specified coordinates and calculates the measurement direction and the feature type either width or space. The XY list is in CSV format, as in the following example:

1487.23, 95867.7 201, 95667.7 301, 4567.932

You specify the maximum CD acceptable for the technology by entering a value in the **Site Automatic Adjustment > Location > Maximum CD** entry in the Job Configuration area.

2. When you import markers, a dialog box appears that allows you to select the source. If you are importing from a marker layer, you can optionally select **Find Hierarchical Markers** from this dialog box. When this option is enabled, marker cells are created over the pattern files to allow drawing of hierarchical markers.

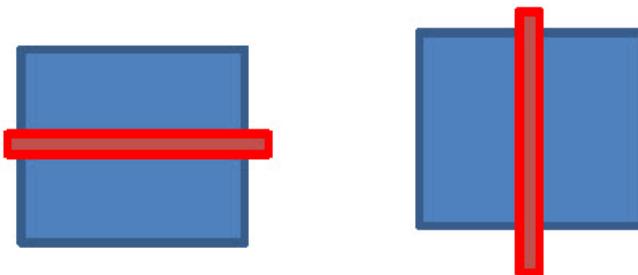
Results

The appear on both the layout in Calibre WORKbench and as individual entries in the CMi window.

When an X Y list is imported, there are special cases that can occur:

- **Contact shape measurement** — If the direction of measurement could be vertical or horizontal, CMi highlights sites with possible conflicts of measurement direction as shown in the following figure.

Figure 2-7. Contact Shape Measurement Example



CMi attempts to center sites on contacts (sites assigned the contacts algorithm) in both X and Y directions. Once the true center is found, the marker is perpendicular to the longest edge.

- **Point with no surrounded polygons** — If the calculation could not find two parallel edges for a certain point in the given search area, CMi assumes it to be a site with the Image algorithm and creates a marker in the layout of 1.5um x 1.5um, then highlights it in the Sites table.

CMi highlights marker points that can be measured in different directions as well as points with no nearby target CD to measure.

Exporting Markers

You can export marker information in CMi in various formats.

Procedure

To export markers, select **File > Export Markers**. You can select one of the following formats to export marker information to:

- CSV: The most common marker input format for general application. Markers, Regions, and Searches may be imported from a CSV format table. This table format is like the CSV output format.
- SS: Spreadsheet format for interfacing from other test pattern metrology tools.
- GG (Gauges): A gauge is a drawn line along which a CD or a space measurement is performed. A gauge data file (.gd) is similar to a spreadsheet (.ss) file in that both contain CD or space measurements.

Exporting SS and GG format converts CMi marker information using the following site properties:

- Enable and Disable
- Name
- Center_x (for SS)

- Center_y (for SS)
- Size (for GG)
- Target_value
- Feature_type

The following figure illustrates how CMi marker information is converted into SS and GG formats.

Figure 2-8. Converting Markers to SS and GG Formats

#	Marker	Feature_type	Position_X	Position_y	Size	Orientation	Window_width	Target_value	Algorithm	Magnification	Cell
0	pitch_55	width	39.9995	50.0	0.02750	0	0.010	0.0275			

55

Pitch_55

79544,100000 80454,100000

SS format (dbu)

Name	Row	Col	X	Y	Loc	Target	other	Meas	weight
Pitch_55	1	-1	80054	100000	0	55	-1	-1	-1

gg format (dbu)

flag	structure	Row	Col	X1	Y1	X2	Y2	Loc	drawn	...
1	Pitch_55	1	-1	79544	100000	80454	100000	0	55	-1

Deleting Markers

You can delete existing markers using CMi.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout
- A marker list loaded into CMi

Procedure

1. Select the desired marker row in the table. The selection is indicated by darker color background in the row. Multiple rows can be selected together. To select a range of rows, click on the beginning row, and Shift-click the end row. Add or remove selections with a Ctrl-click on individual rows.
2. Right-click on the highlighted row(s) and select **Delete Markers**. A pop-up dialog box will appear for confirmation. The selected markers disappear from the table and layout views.

Adjusting Marker Position (Drag Method)

The marker position and size can be adjusted by the drag method in the layout window. This is an alternative to directly editing the marker properties in the Sites table. More care is required here to maintain the text object within the marker shape. Layout changes will not reflect in the Sites table until after generating CMi output. Generally, the other methods are preferred.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout
- A marker list loaded into CMi

Procedure

1. Select the desired marker by either area selection of both the shape and the text, or individual selection of the object or edge. Some useful options include:
 - Selecting Individual object or edge selection with a left-mouse-button click.
 - Adding or removing incrementally from the selection using Ctrl-click.
 - Selecting an area with a left-drag, followed by pressing Alt-S.
 - Unselect using the **u** key.
 - Enabling the desired objects for editing with the Layout View buttons: **Polygon**, **Path**, **Text**, or **Edge**.
 - Restricting the visibility of layers can help to select the correct object. Double-click a layer in the Layers palette to toggle the visibility. Layer visibility can be stored in the layerprops file, which is re-applied automatically during CMi output generation. For example, it is suggested to make the Measurement layer invisible to facilitate editing the marker.
2. Move or stretch the marker shape, as required, with the **m** key. Objects can be deleted with the Delete key.
3. The marker name text must be maintained within the marker shape, and on the marker layer. The name can be adjusted by editing the text properties. If the name is not kept inside the marker, it will receive a new name (“noname_x”) during output generation.
4. Click **Generate Output** to update the table, validate, and create the output files. Before generating CMi output, the table will not be consistent with the drag method edits.
5. This method is illustrated in [Figure 2-9](#) and [Figure 2-10](#).

Figure 2-9. Adjusting Marker Size (Drag Method)

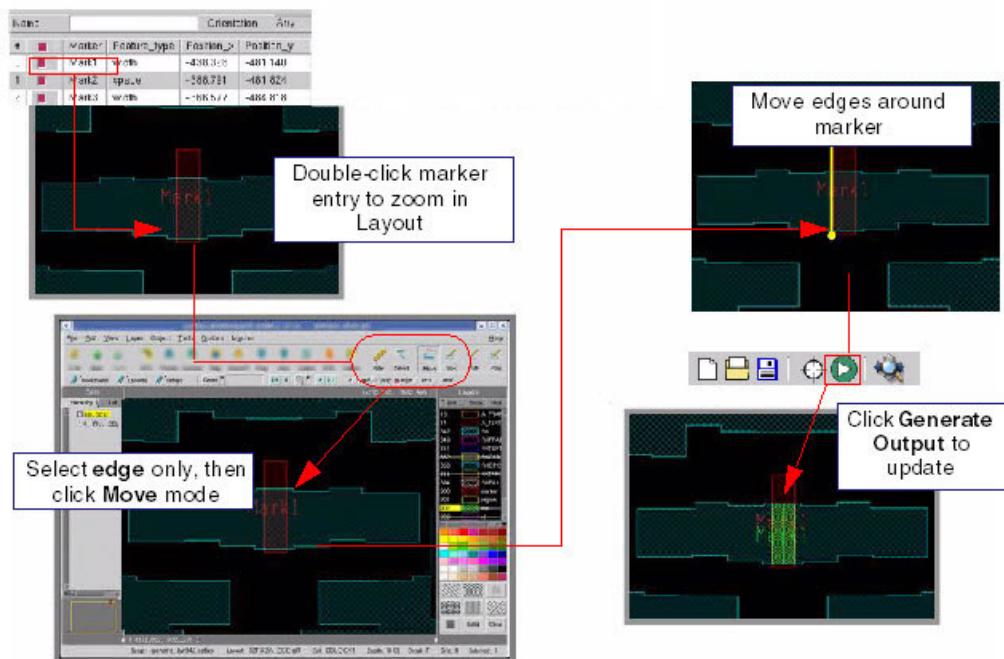
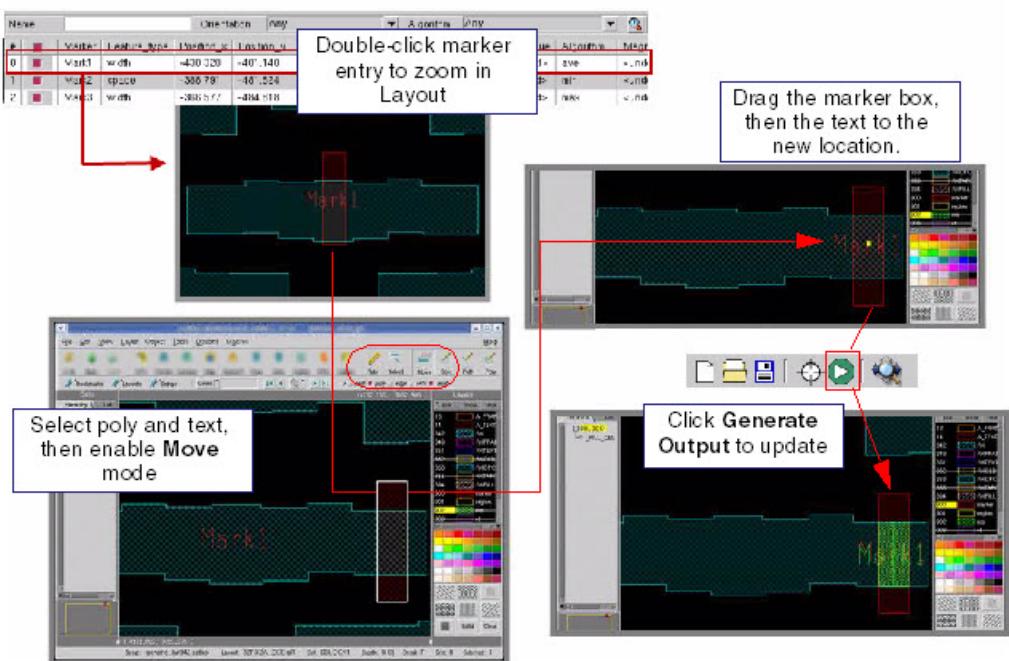


Figure 2-10. Moving a Marker (Drag Method)

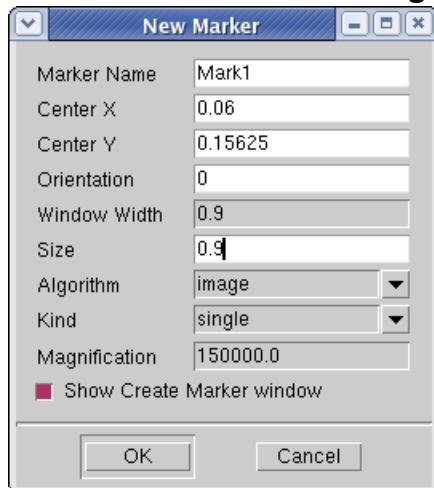


Marker Magnification

The SEM magnification is controlled individually for Markers with algorithm values of “Image”, “Contour”, or “Group”. The size of these markers indicates the SEM field-of-view.

The marker dialogs require the Window Width value to be equal to the Size value, such that the field is square. The resulting magnification value is indicated in the dialog window (see [Figure 2-11](#)).

Figure 2-11. New Marker Dialog Box



Any field size is allowed. A warning is presented for illegal magnification < 100,000X. A warning is also presented for uncommon values that are not 150,000X (0.9 um field) or 200,000X (0.675 um field).

Note

-  For markers in sub cells, note that magnification and orientation may be different in the final top-level context.
-

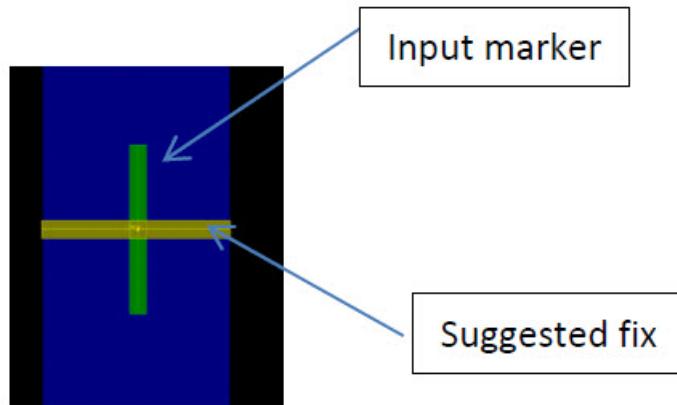
Failed Marker Detection

When you run CMi to confirm the validity of markers for measurements, the tool automatically detects and display the failed markers.

When you click the **Scan Markers Algorithm and Type** button (and the **Location** radio button is enabled), CMi detects cases where the marker either are not located or specified in a way that could lead to incorrect measurements in the design. CMi displays the reason for the failure of the markers, and attempts to automatically correct the problem. The reason for the marker failure is displayed in the Site Status area in the CMi window.

An example suggested fix of a failed marker is shown in the following figure.

Figure 2-12. Failed Marker Suggested Fix



The markers with suggested fixes are also highlighted in the markers table for review. CMi highlights parameters that are adjusted for site validation, which includes Position_x, Position_y, Orientation, and Target_value.

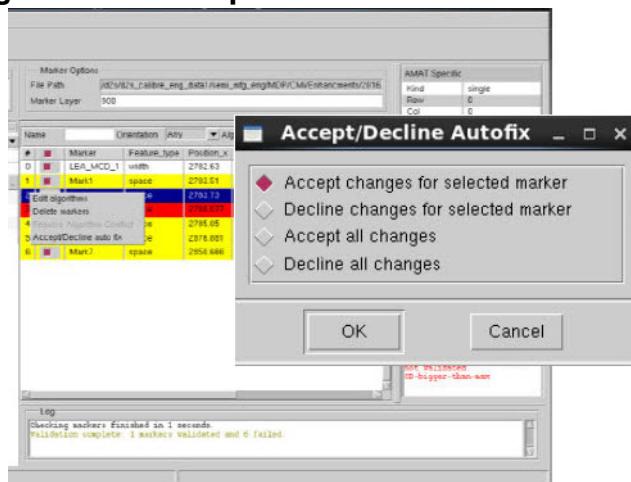
Figure 2-13. Adjusted Sites Highlighted

Name	Marker	Feature_type	Position_x	Position_y	Orientation	Target_value
0	Mark8	width	0.655	22.818	90	2.292
3	Mark12	width	1.229	-11.784	90	2.292
5	Mark14	width	3.194	-40.392	90	2.292
7	Mark17	width	32.258	18.056	90	2.292
9	Mark19	width	32.659	-10.761	90	2.292
1	Mark10	width	0.573	15.033	90	2.292
2	Mark11	width	1.802	1.596	90	2.292
6	Mark16	width	32.168	23.841	90	2.292
10	Mark20	width	34.623	-27.318	90	2.292
4	Mark13	width	3.681	-28.341	90	2.292
8	Mark18	width	32.652	2.619	90	2.292
11	Mark21	width	34.53	-39.369	90	2.292
12	Mark30	space	1.909	9.737	0	0.453
13	Mark31	space	15.985	1.87	0	0.453

You can accept or decline the suggested fixes by right-clicking on one of the highlighted markers and to either:

- Accept or decline the change for selected markers
- Accept or decline changes for all markers

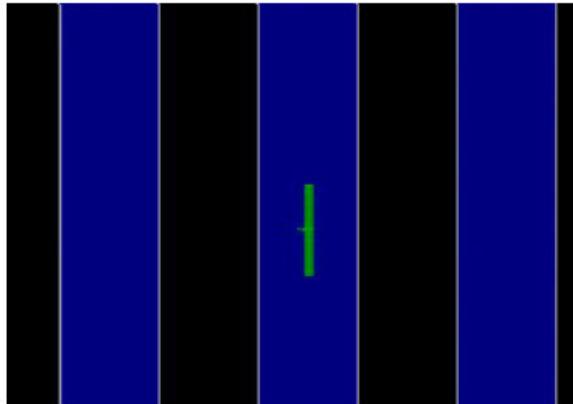
Figure 2-14. Accept or Decline Autofix Changes



The following are the most common failed marker cases:

- Marker specified with incorrect measuring direction.

Figure 2-15. Wrong Marker Orientation

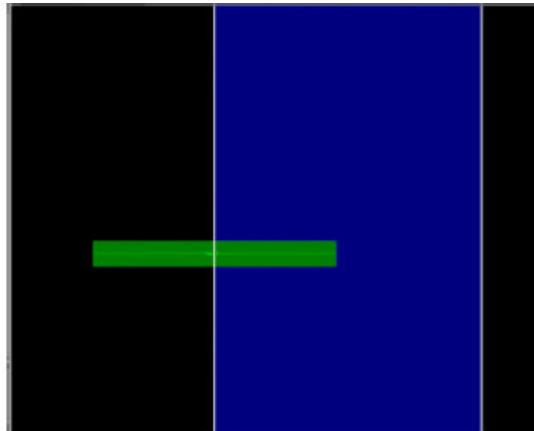


Failure message: Wrong Marker orientation.

Suggested Fix: Rotate marker and assign it 0 orientation.

- Search area identified by the marker size does not cover a valid feature.

Figure 2-16. Marker Size Larger than Target

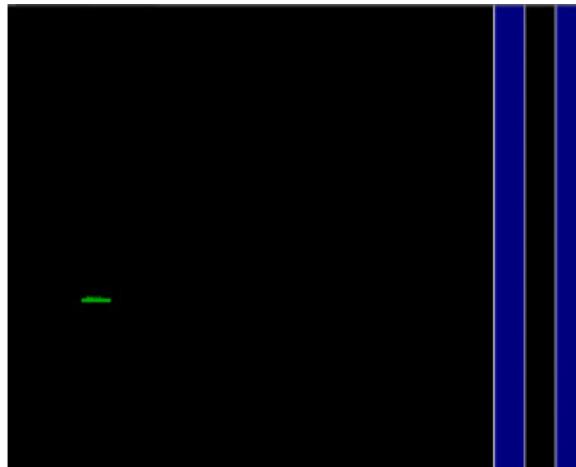


Failure message: Marker size is smaller than expected Target value.

Suggested Fix: Shift the center of the marker to the accurate center And increase the size of the marker for better visualization.

- A marker in an empty space.

Figure 2-17. Empty Space



Failure message: No feature found for this marker.Suggested Fix: Highlight for review.

Browsing Failed Markers

You can use the CMi interface to browse for failed markers.

Prerequisites

Before starting this procedure, you should have the following:

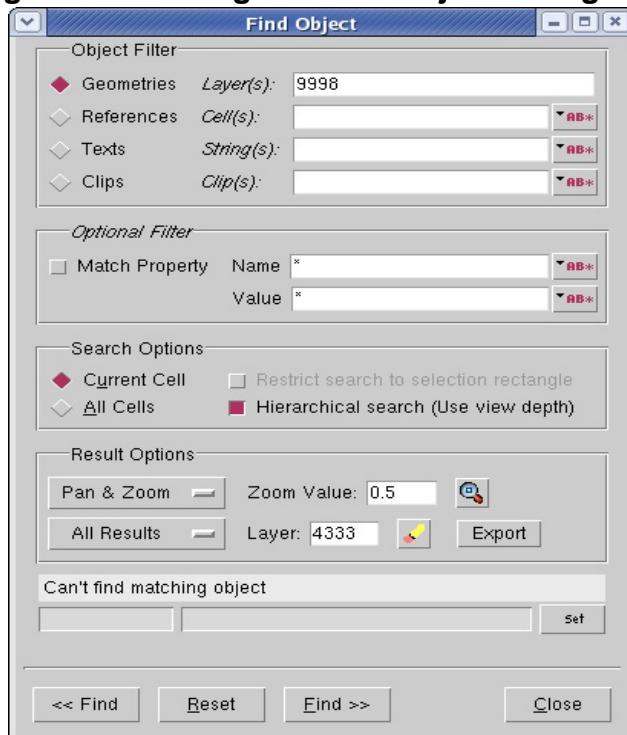
- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

- Existing markers (drawing markers are described in “[Marker Operations](#)” on page 32).

Procedure

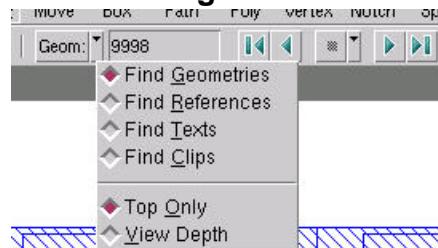
1. In the Layer Options section of CMi, enter a layer number **Failed Markers Layer Number** text field. Click **Generate Output** after this value has been set.
2. In the layout viewer (such as Calibre WORKbench), select **Object > Find Object** to invoke the Find Object dialog box (see [Figure 2-18](#)). Use the controls in the Find Object dialog box to browse for markers.

Figure 2-18. Using the Find Object Dialog Box



Alternatively, you can use the object browser toolbar to browse the markers, as shown in [Figure 2-18](#) (select **Object > Show Find Toolbar** to show the toolbar).

Figure 2-19. Using Show Find Toolbar



- Note**
- 
- While browsing, you may encounter small triangles drawn by CMi at the origin. These are required to maintain the layers when there are no other shapes. In general, they can be ignored.
-

Adding SEM Alignment Locations

You can specify the marker extent for the SEM measurement method. The markers are used to align the SEM pattern recognition region.

Prerequisites

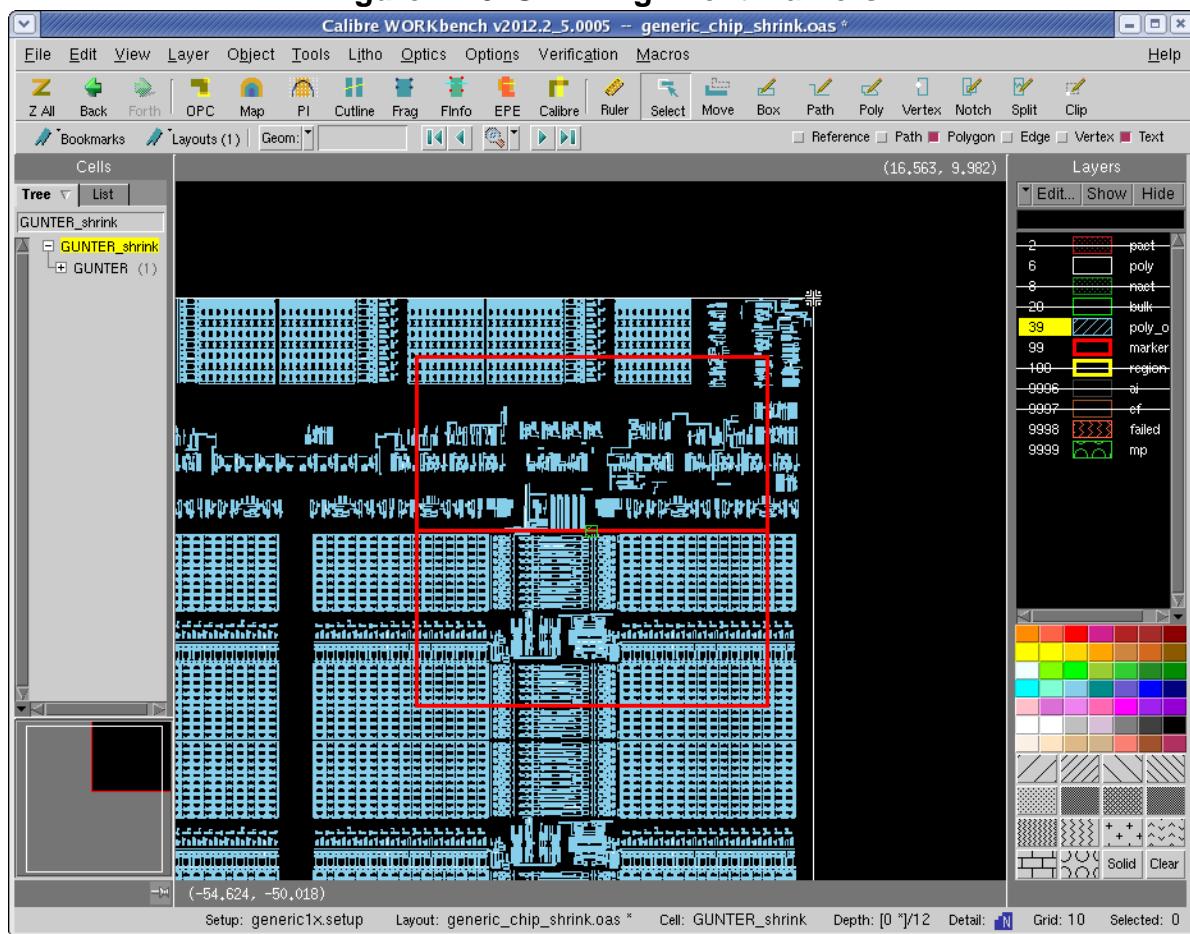
Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. In the Calibre WORKbench viewer, add markers at the desired locations for SEM alignment (see [Figure 2-20](#)).

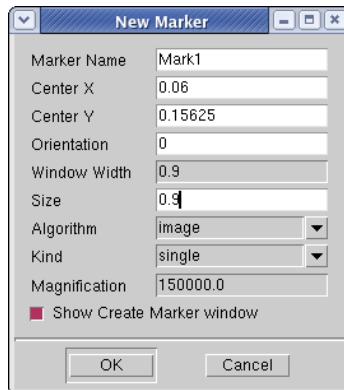
Figure 2-20. SEM Alignment Markers



2. Set the parameters for SEM Alignment in the New Marker or Edit Marker window (see [Figure 2-21](#)):

 - Set the algorithm to “SEM_align.”
 - Set the Window Width and Size to the field of view.

Figure 2-21. Set Marker Window Width and Size



3. Click **OK** to input the changes. The new SEM_align marker is added to the end of the list in the Sites table. After generating CMi output, the alignment locations are included in the AMAT or Hitachi output file.

Designate Multiple Measurements Per Marker

Multiple measurements can be made from a single marker using the Kind property. This feature is only available for the AMAT SEM format.

Three values are available:

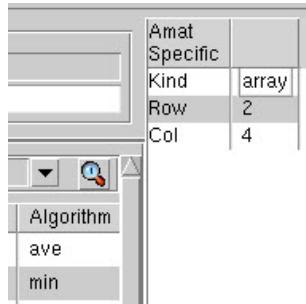
- single: This specifies a single marker and is the default setting.
- array: This is for multiple sites and specifies a number of rows and columns from a root site. The root site specifies the upper-left site and is handled similar to a single site.
- window: This is for multiple sites and specifies a window width and height, though not necessarily centered on a measurement.

The Kind value can be set in the New Marker dialog box (see [Figure 2-22](#)) or in the Kind column of the Machine Area (see [Figure 2-23](#)) as you set parameters for SEM alignment locations (see “[Adding SEM Alignment Locations](#)” on page 46). These values are displayed in the Machine Area of CMi.

Figure 2-22. New Marker With Kind Specified



Figure 2-23. Kind Value in the Machine Area



In this figure, an array is the selected Kind, which also means you must specify the number of Rows and Columns (Cols) for the array.

After you click **OK** to input the changes, the new SEM_align marker with the Kind assignment is added to the end of the list in the Site Tab. After generating CMi output, the alignment locations are included in the AMAT .dbm output file.

CSV Table Construction

The CSV format is a general-purpose interface for measurement sites. It is available for both input and output and can be easily created and modified with a text editor, spreadsheet program, or user software.

It is recommended to use a CSV output file as a template to modify. Output CSV files can be directly used as inputs to subsequent runs, and unnecessary objects are ignored. CSV files may be viewed or edited as text files, or more conveniently using a spreadsheet editor (such as Microsoft Excel). [Figure 2-24](#) shows an example CSV table.

Figure 2-24. CSV Table

A	B	C	D	E	F	G
Marker	position_x	position_y	size	orientation	window_width	description
table1	17.540(cell=ICV_16)	4.220(cell=ICV_16)	0.240(cell=ICV_16)	90(cell=ICV_16)	0.200(cell=ICV_16)	
Region	position_x_min	position_y_min	position_x_max	position_y_max	description	
regionXYZ	-436.240(cell=GUNTER)	-484.600(cell=GUNTER)	-433.740(cell=GUNTER)	-481.260(cell=GUNTER)		
searchregion	74143.78	74141.16	74187.3	74182.22		
Search	FeatureSpec	Region	division_x	division_y	count_limit	pick_method
search1	horiz	searchregion	1	1	200	random
FeatureSpec	size_min	size_max	measure_type	feature_type	orientation	field
horiz	0.8	0.8		width	0	clear
vert	0.8	0.8		width	90	clear

All parameters on a line are separated by commas. All strings are case sensitive. Text values may not have spaces within the string. Spaces adjacent to the commas are ignored. The context of values is top-level (mask or wafer), unless indicated to be in a “cell”. The cell context is indicated by the “cell” field. Alternatively, the context can be indicated by the suffix “(cell = cellname)”, where cellname must exactly match an existing cell name in the assembled mask layout. All numeric values in each object must be in the exact same context. [Figure 2-25](#) illustrates an example CSV for a hierarchical cell.

Figure 2-25. Example of a CSV for a Hierarchical Cell

A	B	C	D	E	F	G	H
1	Marker	cell	position_x	position_y	size	orientation	window_width
2	EP_marker	GUNTER	-267.634	-485.405	4.5	0	4.5 group
3	MP4	GUNTER	-266.681	-483.879	0.7	90	0.1 ave
4	MP5	GUNTER	-269.211	-484.824	0.7	90	0.1 ave
5	MP6	GUNTER	-267.357	-486.373	0.6	0	0.2 ave
6	MP7	GUNTER	-267.934	-485.876	0.6	0	0.1 ave
7	MP8	GUNTER	-268.211	-486.608	0.5	0	0.2 ave
8	MP3	GUNTER	-265.841	-485.569	0.6	0	0.2 ave
9	#MP5b	GUNTER	-271.391	-484.824	0.7	90	0.1 ave
10	MP5_alone	GUNTER	-280.111	-484.824	0.7	90	0.1 ave
11							
12	Region	position_x_min	position_y_min	position_x_max	position_y_max	description	
13	#EP_region	-269.884	-487.655	-265.384	-483.155		

Each collection of objects has a header line followed by object lines. A collection begins with a line containing an allowed object type in the first field (header line). The minimum file would contain only the “Marker” collection (a Marker header and a Marker object). Other collections are not required. Markers are not required when Search is used. A Search requires a FeatureSpec and a Region. The order of the collections of the objects is not required. A header line begins with the object type name (for example, “Marker”). The values following on the line are the allowed property names for the object type. The names must all be spelled correctly with the correct capitalization. The order of the fields is not required, except that the order of the object values must match with the order in the header.

The lines immediately following a header line are objects of that type. The name of the object is in the first field. Object names must be unique within each collection. Objects lines beginning with “#” are ignored.

Tip

 CSV objects can be deactivated by adding a # prefix before each entry.

The fields following the object name have property values for the object, corresponding to the property name order in the header. It is not required to have values for some of the fields (such as description, target_value, and tolerance), but most are required. The order of the objects in the collection is not used.

Required fields for Marker are: Marker, position_x, position_y, size, and orientation. “Marker” must be first and the others can be in any order.

Other pre-defined fields are optional:

- window_width: The Window Width job-based default value is used.
- target_value: Specifies a target value.
- tolerance: Specifies a tolerance value.

- algorithm: The default Metrology Algorithm job-based default is used (in **Output Options**). The possible algorithms are:
 - ave: Take the average of measurements within the marker, across the window width.
 - min: Measure the minimum width. This is useful for finding pinches in hot spots.
 - max: Measure the maximum width. This is useful for finding bulges in hot spots.
 - contacts: Measure a two-dimensional hole (for contacts). Sites using the contacts algorithm in CMi are automatically centered in both the X and Y directions. Once the center is found, the correct site orientation is perpendicular to the longest edge of the feature.
 - image: Get the image only.
 - ellipse: Measure a two-dimensional elliptical area (similar to contacts).
 - contour: Get the image fit contours to the field.
 - group: Specify the field-of-view and designate the measurements inside to be extracted from that image.
 - undefined: Apply the Default Metrology Algorithm value.
 - SEM_align: Specify the marker extent for the AMAT SEM measurement method. The marker extent indicates the pattern recognition region. See “[Adding SEM Alignment Locations](#)” on page 46.
 - OM_align: Specifies a Global Alignment (GA) Optical Microscope (OM) marker. OM_align markers are treated as SEM_align markers.
 - algorithm is treated as SEM_align markers.
- kind: Allows you to designate multiple measurements per marker for AMAT SEM (see “[Designate Multiple Measurements Per Marker](#)” on page 48).
 - rows: Specify the number of rows in an array (if specified as kind = array).
 - cols: Specify the number of columns in an array (if specified as kind = array).
- cell: Specifies that the context is the top-cell of the job.
- description: A text description of the entry.

Measurements in CMi

The Calibre Metrology Interface supports a number of measurement-based operations.

Multiple Measurements Per Capture	53
Tolerance Specification	56
Target Value Specification	56

Multiple Measurements Per Capture

The CMi Capture Field represents the SEM field-of-view. Multiple measurements can be grouped into a single field to reduce the run time on the tool and prevent over scanning. The user designates the grouping by one of two methods: a Group Marker or a Region.

Grouping Measurements in a Group Marker.....	53
Group Measurements in a Region	54
Debugging and Optimizing Multiple Measurements Using a Region.....	55

Grouping Measurements in a Group Marker

Measurements can be designated to be in groups that are defined by rectangular regions of the layout. these regions can be specified in the input layout or CSV file, in which case they will be rendered in the layout after assembly.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. Add a marker at the center of the field.
2. Set the Size corresponding to the desired tool magnification. Common values are 150,000X (0.9 um filed) or 200,000X (0.675 um field).
3. Set the Orientation to the desired scan direction.
4. Mark the desired measurements inside the bounds of the group marker.
5. After generating CMi output, there will be a single Capture Field at the location of the group marker, indicating a successful grouping.
6. [Figure 2-26](#) illustrates an example of grouped markers. [Figure 2-27](#) shows an example of the results.

Figure 2-26. Grouped Markers

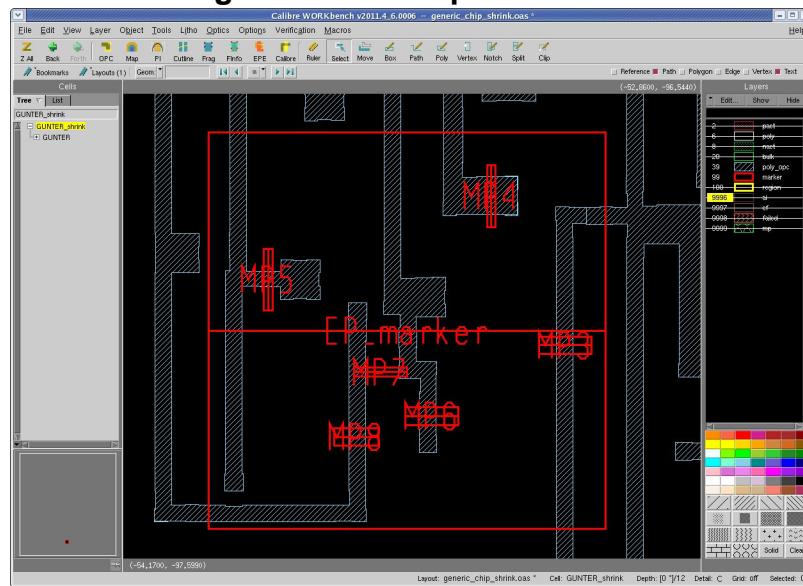
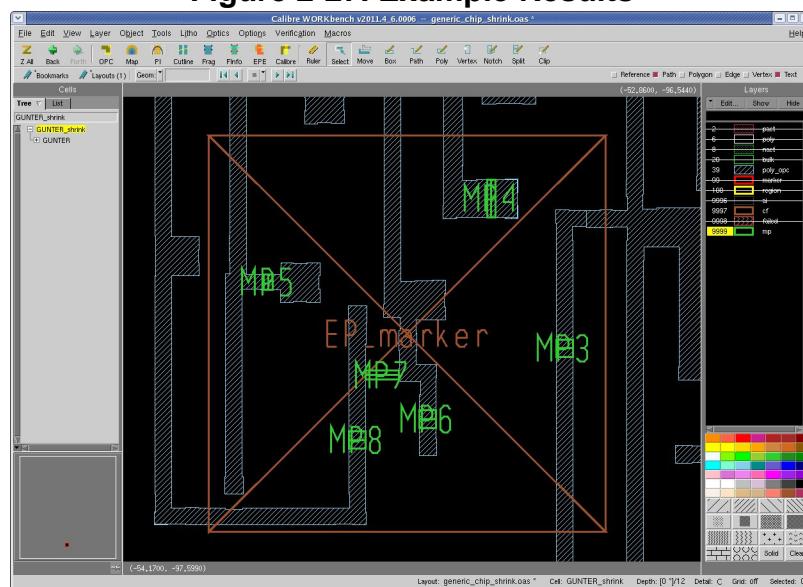


Figure 2-27. Example Results



Group Measurements in a Region

Regions can be drawn into the layout after assembly. The specified Region Layer number is used. Region names may be edited as text within the rectangle on the same layer. Hierarchical regions are allowed.

Multi-measurements per capture are controlled by the user drawing a region. There are three requirements:

- Regions must be drawn. The center of each Region is the proposed center of the Capture Field. The region must be no larger than the Capture Width parameter to be used for multi-measurements. The recommended size is the capture width reduced by capture tolerance on each side, which corresponds to the capture field graphics after generating CMi output.
- The first (“dictionary order”) marker name within the Region must be close enough to the center of the proposed capture field, such that the Alignment Image would be fully contained. For example, the marker naming convention “align#” may be used to distinguish alignment positions. The distance requirement involves the values for Capture Width, Capture Tolerance, and Image Width parameters. Otherwise, this marker will generate a separate single-measurement Capture Field centered on the Measurement location.
- Each measurement point must be fully contained in the proposed capture field. Otherwise, the measurement will generate a separate single-measurement Capture Field centered on the Measurement.

If your tools do not support multi-measurements, avoid creation of multi-measurements by not using regions altogether. Failure to meet requirements results in warning messages, and reverting back to a single measurement per capture field. The results may be less effective, but are otherwise correct. The efficacy of multi-measurements is indicated by the run statistics in the status line and log window. The number of captures is less than the number of measurements. Multi-measurement efficacy can be tuned by viewing the layers for capture field and alignment image. When layer numbers are specified, outlines are rendered. Examine the log messages and graphics to meet the three requirements.

Note

 The Capture Field boundary is adjusted smaller on all sides by the Capture Tolerance value.

To get the desired result, it may be necessary to re-run after adjustment of region boundary, marker position, or marker names. The presence of multi-measurements are also indicated in the XML file by formatVersion >= 0.31.

Debugging and Optimizing Multiple Measurements Using a Region

An iterative method may be used to configure multiple measurements.

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. Draw a region box exactly the size of an undersized capture field (capture width minus capture tolerance on all sides).
2. Choose the marker names considering that the first alphabetic name will be used for the alignment image. The alignment image centered on the marker must be fully contained in the region. Other measurement points must be fully contained in the region, including the oversize and clearance values.
3. Assign layers for Measurement and Capture Field so that the first pass results can be interpreted more easily.
4. Run the job.
5. Examine the results in the graphic window. Make the “cf” and “ai” layers visible. There should be a single “cf” box coincident with the region if the requirements have been met. If not, then move the region box to a position where the desired “ai” box does no extend outside. Check the measurement object names to be sure that the desired one is first. Check all measurement objects for clearance inside the region.
6. Run the job again.

Tolerance Specification

If tolerance is given in a CSV table for a Marker, then it is used for Measurements from that Marker. If the Marker has no tolerance (including drawn markers), then the default value for the job is used.

Note that the tolerance value is not used within CMi. It will instead be later used for analysis of measurement results.

Target Value Specification

Commonly, the target_value is taken from the feature size on the specified “Measure Layer”.

There are two exceptions:

- If a target_value is given in a CSV table for a Marker, then it is used for Measurements from that Marker. Note that the Measurement is still re-centered to the layout in this case, but the target CD from the layout is discarded.

- If validation is deactivated for the job, then the global default value for target_value (in the GUI) is used for all measurements.

Searches in CMi

Searches are defined in the CSV input table. An output table can be used as a template for the proper object header and property names. One search requires an object for each of the following: search, feature spec, and region.

The search parameters are detailed in the *Calibre Metrology API (MAPI) User's Guide and Reference*.

Searching for Measurement Locations 58

Searching for Measurement Locations

Searches are specified only from a CSV file.

Figure 2-28 illustrates an example

Figure 2-28. Example CSV Table With Search Highlighted

A	B	C	D	E	F	G
Marker	position_x	position_y	size	orientation	window_width	description
1 table1	17.540(cell=ICV_16)	4.220(cell=ICV_16)	0.240(cell=ICV_16)	90(cell=ICV_16)	0.200(cell=ICV_16)	
Region	position_x_min	position_y_min	position_x_max	position_y_max	description	
5 regionXYZ	-436.240(cell=GUNTER)	-484.600(cell=GUNTER)	-433.740(cell=GUNTER)	-481.260(cell=GUNTER)		
6 searchregion	74143.78	74141.16	74187.3	74182.22		
Search	FeatureSpec	Region	division_x	division_y	count_limit	pick_method
9 search1	horiz	searchregion	1	1	200	random
FeatureSpec	size_min	size_max	measure_type	feature_type	orientation	field
12 horiz	0.8	0.8		width	0	clear
13 vert	0.8	0.8		width	90	clear

Prerequisites

Before starting this procedure, you should have the following:

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout

Procedure

1. To set up the search:
2. In the CSV file, create a FeatureSpec object.
3. Create a Region object.
4. Create a row under the Search title for your search object.
5. Fill in the properties to link the FeatureSpec and Region entries.
6. In CMi, select **Read Text Objects from File** and specify your CSV table.

Magnification Settings for Hitachi Output

You can use CMi to manually set magnification settings for different parameters for Hitachi output (more specifically, the EP_MAG_SCAN_X and GP_Data values in the HSS CSV file).

The following can be manually assigned when you select **Hitachi Output** in CMi:

For all algorithm types except “Image” the EP_MAG_SCAN_X value will be assigned as follows:

- Maximum magnification — Specifies the maximum magnification value when the feature size is smaller than a ratio of the FOV (specified using the “FOV ratio” field).
- Minimum magnification — Specifies a minimum magnification value when the feature size is greater than or equal to ratio of the FOV (specified using the “FOV ratio” field).
- Field of View (FOV) ratio — Specifies the ratio of the (Field of View) FOV that will be compared with the feature size. This ratio will be used to determine whether minimum or maximum magnification settings are used. The default value is $2/3 = 0.66$
- Image capture magnification — Specifies the image capture magnification value in cases where an Image algorithm is used.
- Multiple Region of Interest (MROI) magnification — Specifies the magnification value of the Multiple Region of Interest (MROI) EP marker. This is required if MROI markers are used (they are used with the Group algorithm).

If the magnification parameters are not set, the EP_MAG_SCAN_X value in the HSS will be left blank, or else it will be set with the calculated value.

The following magnification options set the value for the GP_Data section in the HSS file:

- SEM Align magnification — Specifies the magnification value in cases where the SEM_align algorithm is used.
- OM Align magnification — Specifies the magnification value if an OM_align algorithm is used (for Global Alignment Optical Microscopes).

The Magnification value can be reviewed through the Sites table (under the Magnification column) and Create New Marker dialog boxes.

Note the following:

- If a loaded Jobspec or previous session has the magnification parameters already specified, the magnification value is calculated during output generation and is sent to the HSS file.
- If you modify the values of the magnification parameters in the Hitachi Output options, then the magnification is re-calculated during the run and the Sites table is updated accordingly.

- If the magnification parameters are set, you can also automatically re-calculate magnification by clicking the **Scan Markers Algorithm and Type** button.
- If the values of the Hitachi magnification parameters have been deleted, the previously calculated magnification value for each marker is kept unchanged, and these values are output to the HSS file. You can still manually edit the value of the magnification entry for each marker.

Defining Boolean Operations on Layers

In CMi, you can define boolean operations more than any two layers from the original layers in the layout or from derived intermediate layers from previous boolean operations.

Prerequisites

- You should have completed the “Assemble” step as described in the “[CMi Workflow Overview](#)” on page 22.

Procedure

1. In the CMi Layer Options section, click the **Enable Layer Processing** radio button to enable the layer processing.

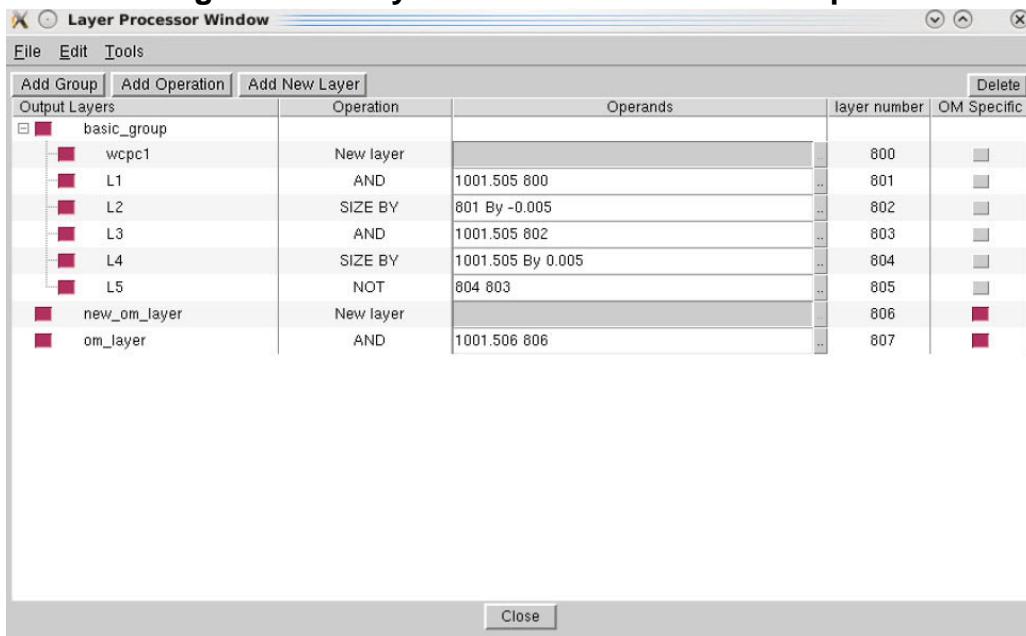
Figure 2-29. Layer Processing Options

Layer Options	
Enable Layer Processing	<input checked="" type="checkbox"/>
Region Outline Layer Number	100
Measurement (Output) Layer...	9999
Capture Field (Output) Laye...	9997
Alignment Image (Output) La...	9996
Failed Markers Layer Number	9998
Layer Property File Name	generic_chip.oa ..

2. Select **Tools > Layer Processing (Booleans)** to invoke the Layer Processor window. In the Layer Processor window, perform the following steps:
 - a. Click **Add Group** to add a group of operations or **Add Operation** to add entries for individual boolean operations.

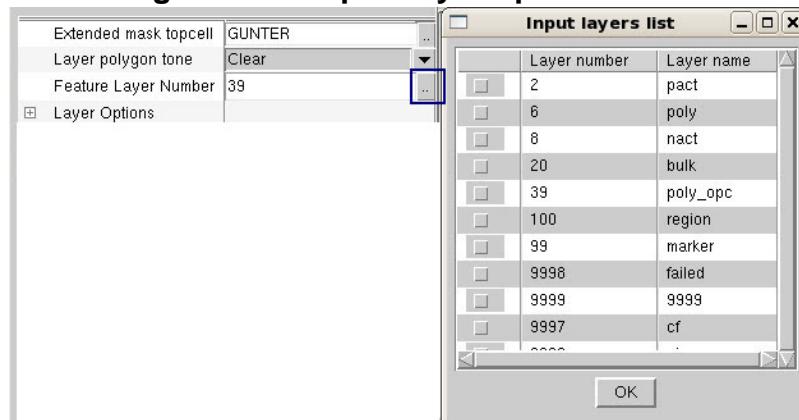
Clicking **Add Group** generates a tree with a parent node (Group Name) and a child node (Operation). The group is a method to collect operations in a tree structure and does not affect the processing of the operations.
 - b. You can specify operation, operands, layer numbers by clicking on each column for the entry. You can also specify that the boolean operations be processed on OM_align markers only by clicking the **OM Specific** radio button.
 - c. To delete an entry from the table list, select the entry and click the **Delete** button.

Figure 2-30. Layer Processor Window Example



3. Click **Close** to close the window.
4. You can select any of the original layers or newly-defined layers by entering the Feature Layer number (or click the “..” button to invoke the Input Layers List dialog box to select multiple layers).

Figure 2-31. Input Layer Specification



5. Click **Run** in the CMi window. If the **Layout Clip Output** option is enabled, the defined operations are executed and the newly-defined layers are added to the output clip.
6. If the defined boolean operations reference input layers that do not exist in the clip, the operations are not executed and an error similar to the following is added to the output transcript:

```
Can't execute operation clip_temp AND: Input layer, 1001.505, does
not exist or is empty
```

Constructing a Four Corner Shape

You can use the Calibre Metrology Interface to construct a complete square shape constructed from clips of the four corners of your layout, starting from the SEM/OM marker position. OM/SEM markers are typically positioned at the origin X1, Y1 of the layout (the lower left corner). The upper-right quarter of the clip is placed inside the design while the other three portions are placed outside and are empty.

Refer to [Figure 2-33](#) for an illustration.

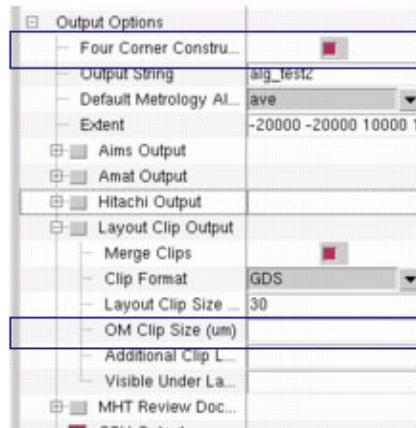
Prerequisites

- You should have your layout loaded into Calibre WORKbench (see “[CMi Workflow Overview](#)” on page 22 for information on the overall process flow).

Procedure

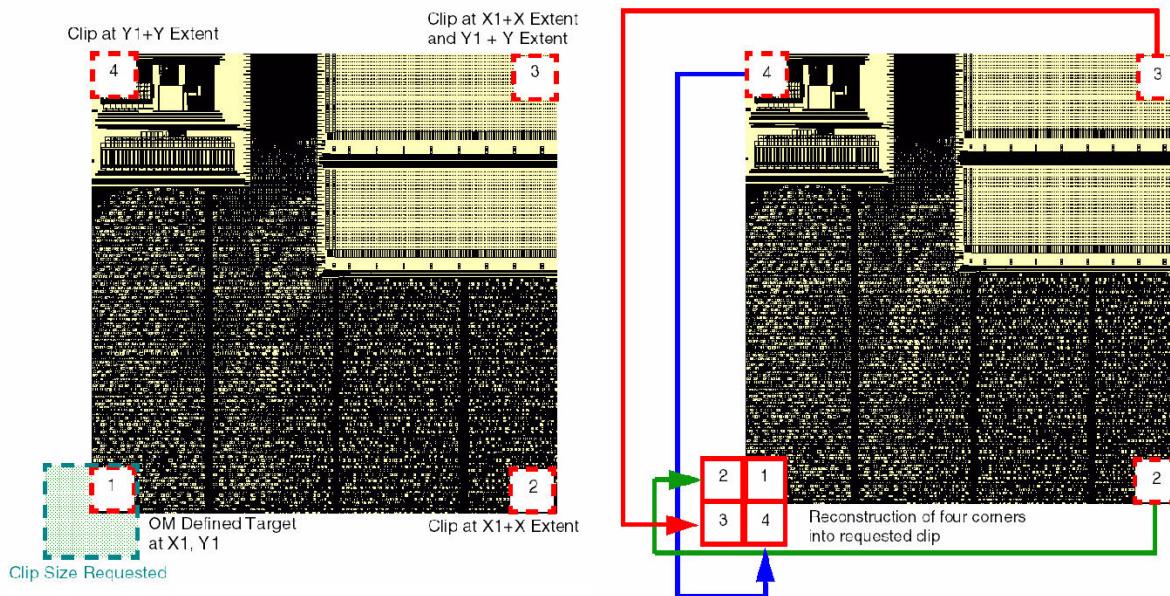
1. In Calibre WORKbench, add a new marker by performing a Ctrl + double-left-click in the loaded layout to invoke the New Marker dialog box (see “[Marker Operations](#)” on page 32). The markers can also be loaded from sample plan (.csv or .ss) files during assembly.
2. For each of these markers, specify that they use either the SEM_align or OM_align algorithms. This can be specified in the New Marker dialog box, be previously set in the sample plan, or it can be edited in the Sites table.
3. In the Calibre Metrology Interface, click **Four Corner Construction**.
4. Define the extent field in the Extent text entry field and the clip size value in the OM Clip Size entry field. The design extent is defined as two coordinates: the first coordinate value defines the lower left corner of the layout and the second coordinate defines the upper right corner.

Figure 2-32. Options to Specify Four Corner Construction



5. Click **Generate Output**. [Figure 2-33](#) illustrates the process that follows.

Figure 2-33. Four Corner Construction



The corner clipping begins from the OM/SEM marker position X1, Y1. The size of the upper right quarter of the desired square is determined is calculated from the center of the marker.

CMi calculates the position of the other three clips as follows:

- The lower-left corner of the clip calculations include:
 - Clip 2: X1 + X extent , Y1
 - Clip 3: X1 + X extent, Y1 + Y extent
 - Clip 4: X1, Y1 + Y extent
- CMi calculates the clip extent by using the lower left corner position and the clip size.
- CMi places clips 2, 3 and 4 around the SEM/OM clip to construct the four-corner box.

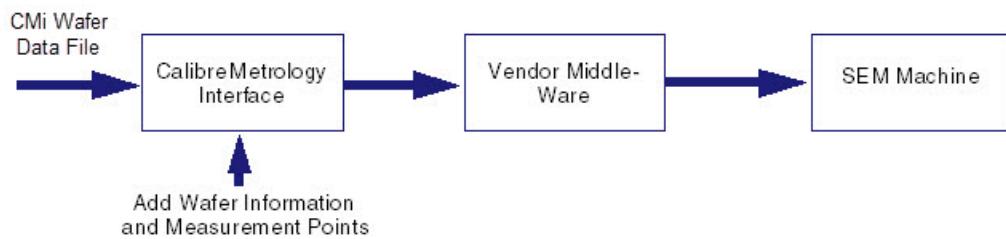
Editing a Wafer Map

Wafer data is often used to complete SEM recipes for machine vendor middle-ware to speed up and enhance the metrology process cycle. You use CMi to generate or edit an existing wafer data XML file that includes wafer parameters needed to generate an AMAT DWM and Hitachi IDW file.

CMi's main role in the SEM recipe generation flow is to generate, complete, and correct measurement point recipe files and layout clips. The output of CMi runs are used as an input to

a vendor's middle-ware. Typically, a metrology engineer uses the middle-ware to complete the SEM recipe by adding wafer data before sending them to the SEM machine. However, using CMi, you can reduce the time it takes to include wafer data in this process by adding it as input to CMi instead.

Figure 2-34. CMi Role in SEM Recipe Generation



You view and edit the wafer data parameters loaded from the XML file through the Wafer Map Manager utility.

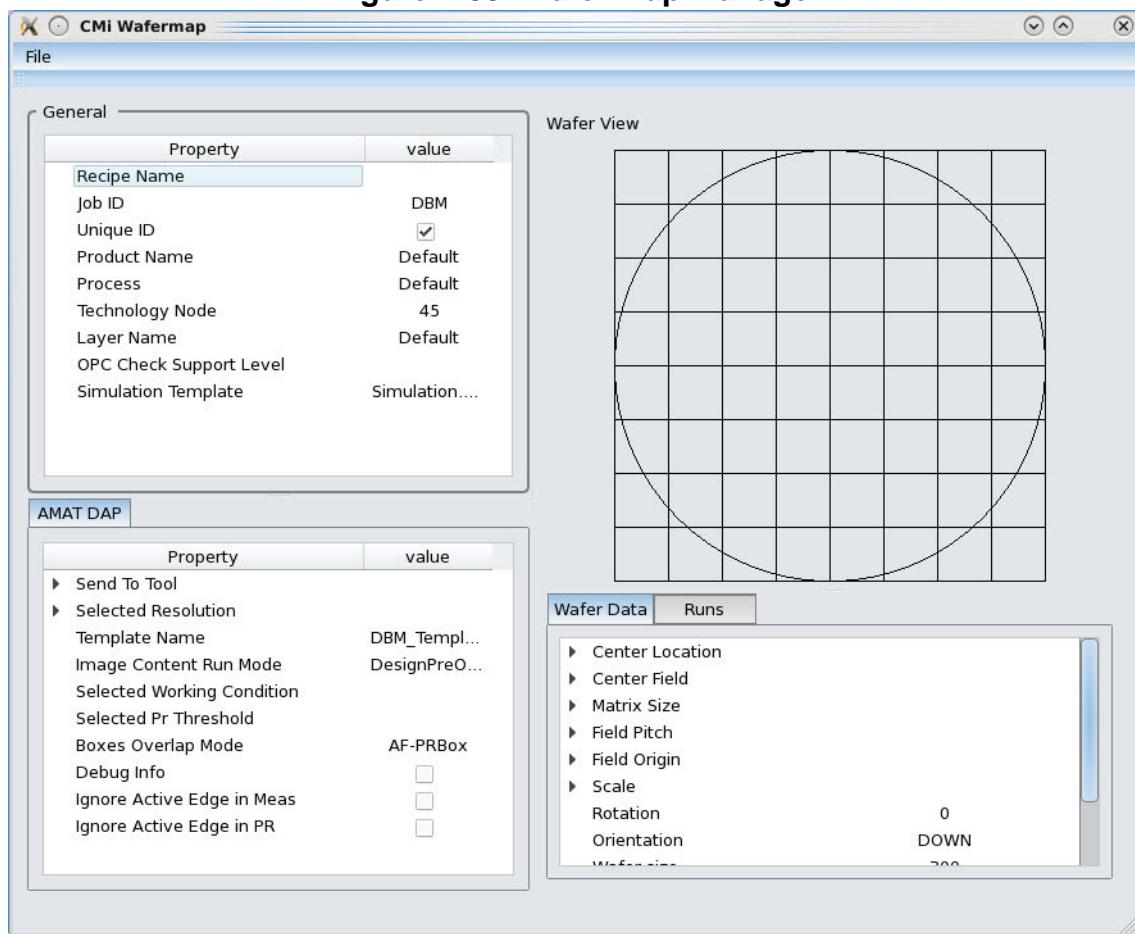
Prerequisites

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A wafer session file in XML format.

Procedure

1. In the CMi window, select **Tools > Wafer Map Manager**.
2. Select **Tools > Wafer Map Manager**. The Wafer Map Manager appears.

Figure 2-35. Wafer Map Manager



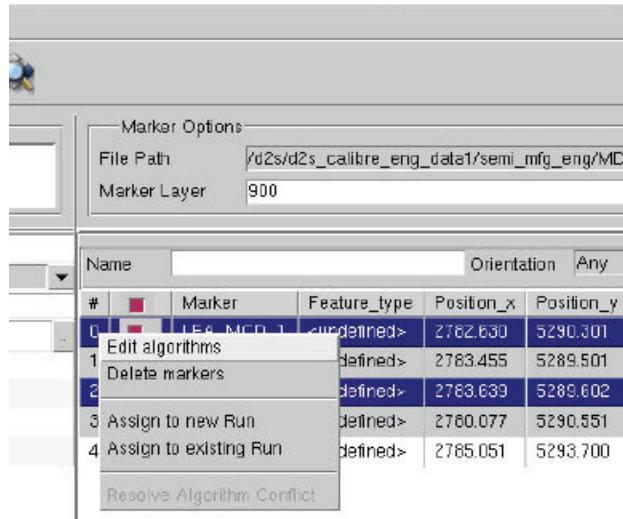
- In the Wafer Map Manager, edit any of the parameter values as needed.

Note

 By default, wafer maps are displayed and edited in mm units. However, for AMAT output, the wafer map units are converted to ums.

- Define logical groupings of markers called Runs to generate an output wafer map using one of the following methods:
 - Select markers from the Sites Table, then right-click and choose **Assign to New Run** (to create a new marker group) or **Assign to Existing Run** (to add to an existing group).

Figure 2-36. Adding Runs from the Sites Table

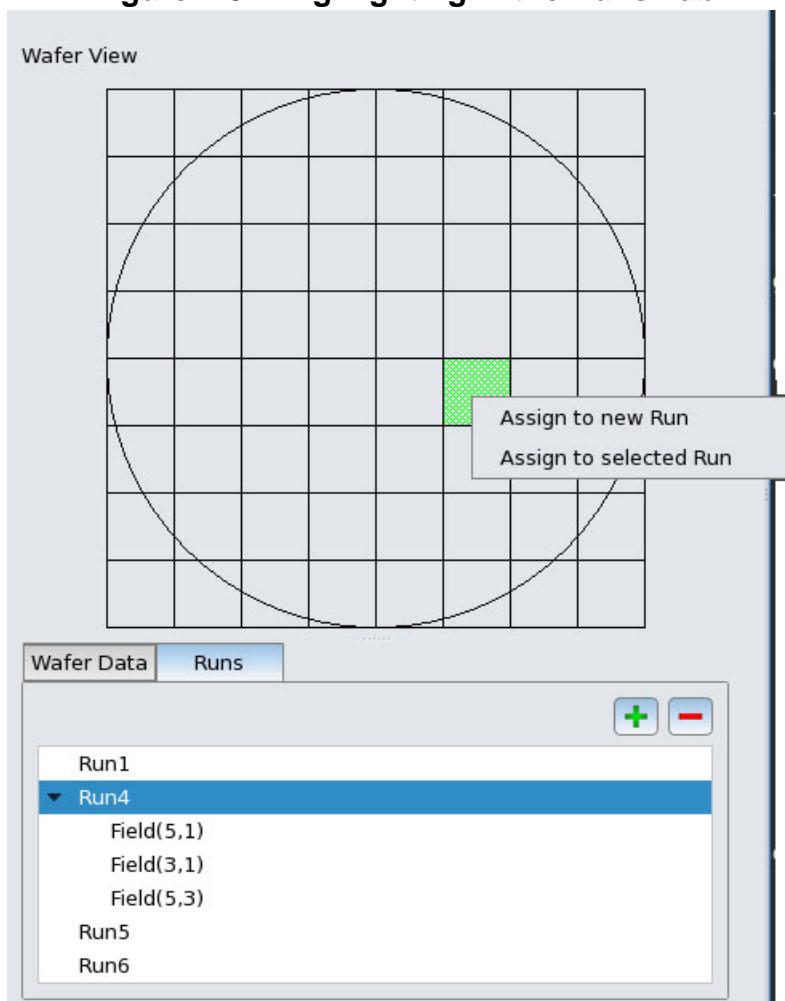


- Select markers from the Sites Table, then in the Wafer Map Manager click the green plus button in the **Runs** tab to add marker groups as individual Runs.

Each new run appears in the **Runs** tab. You can delete a Run by clicking the red minus button in the **Runs** tab.

5. In the Wafer View pane of the Wafer Map Manager, select fields of the wafer map, then right-click and select either **Assign to Selected Run** or **Assign to New Run** to map the fields to a run, or add to a new run. The fields appear under the selected Run in the **Runs** tab.

Figure 2-37. Highlighting in the Runs Tab



6. In the Wafer Map Manager, edit any of the parameters listed in the **General**, **Wafer Data**, **AMAT DAP** or **Hitachi IDW** tabs.
7. In the Job Configuration area of the main CMI window, select **Output Options** and enable **Amat Output** or **Hitachi Output**.
8. Click the **Generate Output** button (the green circle with white triangle) in the CMI button bar.
9. To save a wafer map session, select **File > Save Session**. To load an existing wafer map session, select **File > Open Session**. To create a new session, select **File > New Session**.
10. To exit the wafer map manager, select **File > Exit**.

Results

The resulting output is an AMAT DWM or Hitachi IDW file.

Overlaying SEM Images and Design Clips

You can load SEM images in (.tiff) format and compare them to the design clip generated from CMi. You can also attach this image to a marker and the SEM image path is added to the output Marker points in the CSV file. This assists in diagnosis and verification of the validity of the model against what is printed on the wafer.

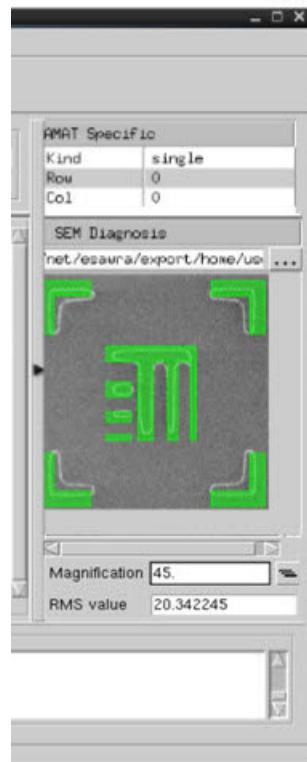
Prerequisites

- Invoked Calibre WORKbench and the Calibre Metrology Interface
- A mask layout with markers (see “[Marker Operations](#)” on page 32 for information on markers)
- A SEM image
- Run the Calibre Metrology Interface and generate GDS or OASIS reference layout clips (Refer to “[CMi Workflow Overview](#)” on page 22 for an overview of this process)

Procedure

1. In the Sites table of CMi, select a marker, then click the Browse button in the SEM Diagnosis area to load a SEM image. A SEM image appears in the SEM Diagnosis display area.

Figure 2-38. CMi With SEM Loaded



2. In the SEM Diagnosis area, you can change the magnification value to run the alignment process with an updated magnification value by pressing the Realignment button. This button can be found to the right of the Magnification field. You can also view the Root Mean Square (RMS) value of the alignment between any given SEM image and its respective design polygons and contour specified around the measurement point.
3. Select **File > Export Markers > CSV**. This generates an output CSV file that attaches the SEM image path to the marker object in the CSV file.

Automatic Metrology Algorithm Identification

The Calibre Metrology Interface (CMi) is a tool that assists generation of correct and complete SEM machine recipes. The metrology algorithm is one of the setup parameters that must be defined per measurement location.

CMi supports a flow that helps you automatically identify metrology algorithms. The flow uses a pattern matching (PM) engine, which is used to examine each measurement location in the design space against a pre-populated database of patterns.

There are increasing challenges of new technology nodes driving more photo layers, shrinking design rules, cross-process coupling, and new modeling techniques. This has caused a substantial increase in the number of measurement points and algorithm types. Because of this, a faster and more automated setup of parameters for those measurement locations is needed.

CMi uses the Calibre pattern matching flow to achieve the automatic generation of algorithms per measurement location. The idea is to use the pattern matching engine that examines each measurement location in the design space against a pre-populated library of patterns. The pattern matching library is constructed by a set of patterns that mostly describe the feature profile of each metrology algorithm.

Each pattern in the library has a property that defines which algorithm it represents. While examining the pattern library for a match, if a match is found, the property value associated with the pattern is assigned to the measurement location as the metrology algorithm; otherwise, if the pattern cannot be matched with the current library, you assign an algorithm to the new pattern and add it to the library.

Automatic Metrology Algorithm Identification Flow	70
Pattern Library Creation	73
Using Automatic Identification in CMi	78
Pattern Library Specification for Automatic Metrology Algorithm Identification ...	82
Multiple Algorithm Matches for the Same Feature	84
Resolving Individual Algorithm Conflicts.....	87
Modifying Algorithm Conflict Resolution Behavior	87
Automatically Selecting Alignment Markers Using Pattern Matching	90

Automatic Metrology Algorithm Identification Flow

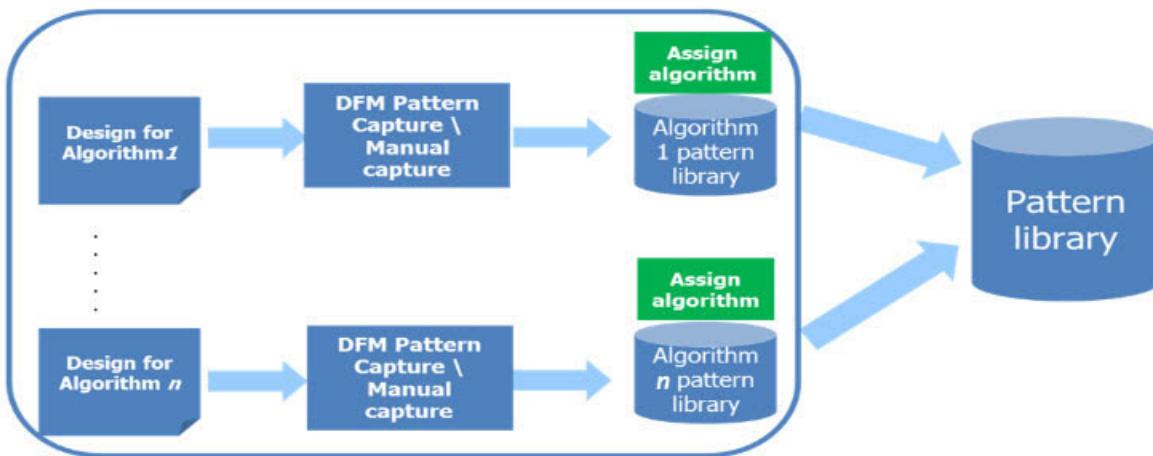
The overall process of setting up automatic metrology algorithm identification can be summarized in two phases.

Procedure

1. Create a pattern library. A pattern library is a collection of one or more patterns created within the Pattern Library Manager or captured using one of the pattern capture operations. When the pattern library is filled with the patterns you want to search for and ready to be included in a pattern matching Calibre run, the pattern library is then exported from the Pattern Library Manager to create an SVRF DMACRO pattern file.

There are different methods to create pattern libraries. See “[Pattern Library Creation](#)” on page 73 for further information.

Figure 2-39. Pattern Creation Flow

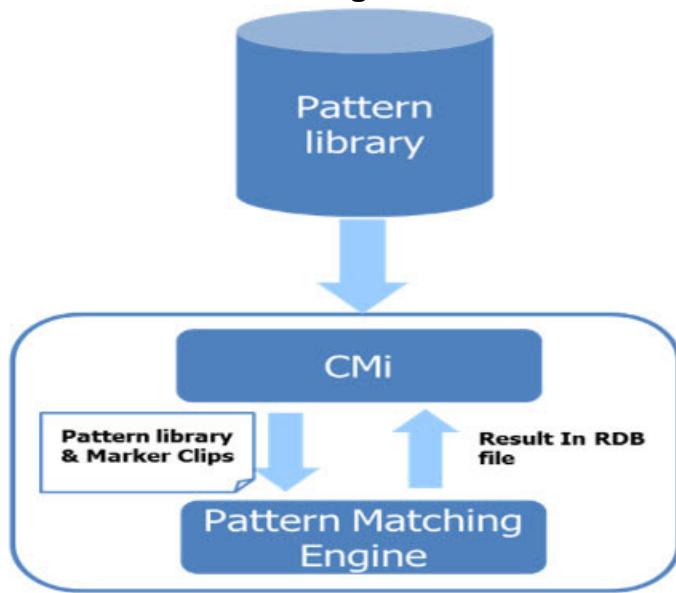


2. Use CMi to set up and run the Calibre pattern matching engine to identify the metrology algorithm.

The created pattern matching library is used as input to CMi. The CMi interface eases the process of running the Calibre pattern matching engine and automatically sets the proper Metrology Algorithm values after each run.

See “[Using Automatic Identification in CMi](#)” on page 78 for further information.

Figure 2-40. CMi Automatic Algorithm Identification Flow



Pattern Library Creation

There are different methods available to build pattern matching libraries: using the DFM Pattern Capture command in a script, or use the Pattern Library Manager in Calibre WORKbench.

There are specific rules when creating a pattern matching library for the automatic metrology algorithm identification process. See “[Pattern Library Specification for Automatic Metrology Algorithm Identification](#)” on page 82 for details on the library specification.

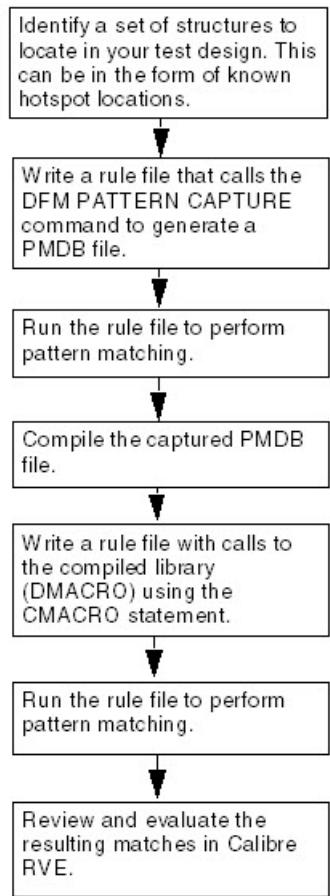
DFM Pattern Capture Command	73
Using the Pattern Library Manager	75
Library Creation Flow Example	77

DFM Pattern Capture Command

You can use the SVRF DFM Pattern Capture command that to create a library of patterns from a layout. Patterns can be captured either by defining a set of hotspots and a halo around the hotspot, or by defining a mask on a masking layer to define the extent of the pattern.

The flow to define a DFM Pattern Capture command illustrated in [Figure 2-41](#).

Figure 2-41. DFM Pattern Capture Command Flow



The DFM Pattern Capture command enables a faster method for the creation of pattern libraries. It gives the user the ability to attach a property to the captured patterns. It is used to attach the Metrology Algorithm Property to each pattern in the library. Also, this function enables the assignment of global constraints to all edges in the pattern.

The following is an example of DFM Pattern Capture in a rule file:

```
LAYOUT PRIMARY "*"
LAYOUT SYSTEM GDS
LAYOUT PATH "line_ave_algorithm.gds"
layout injection control disable all

DFM LICENSE calibreelfd
DRC MAXIMUM RESULTS all
DRC RESULTS DATABASE "result_dfm.gds" GDS PSEUDO
DRC SUMMARY REPORT "result_dfm.rpt" HIER

FLAG SKEW YES
FLAG ACUTE NO
FLAG OFFGRID YES
PRECISION 1 8000
RESOLUTION 1

layer target_1 18
layer hotspot_1 900

target_1 { COPY target_1 } DRC CHECK MAP target_1 0
hotspot_1 { COPY hotspot_1 } DRC CHECK MAP hotspot_1 1
//mask_1 { COPY mask_1 } DRC CHECK MAP mask_1 2

metal_c = DFM PATTERN CAPTURE layer_target target_1
    layer_hotspot hotspot_1
    pattern_halo 0.5
    max_search 2
    max_length 1
    outfile iso_ave_alg.pmdb
    pattern_name_prefix line_ave
    pattern_cglobal 0.2
    pattern_prop Algorithm 0 integer
    pattern_marker_method hotspot

metal_c { COPY metal_c } DRC CHECK MAP metal_c 20
dfm_min_1 {
    DFM RDB metal_c "capture.rdb" checkname metal_c NOEMPTY NOPSEUDO
}
```

See the [Calibre Pattern Matching Reference Manual](#) for complete information on the DFM Pattern Capture command and the pdl_lib_mgr batch tool.

Using the Pattern Library Manager

You can create a pattern matching library using the Pattern Library Manager in Calibre WORKbench.

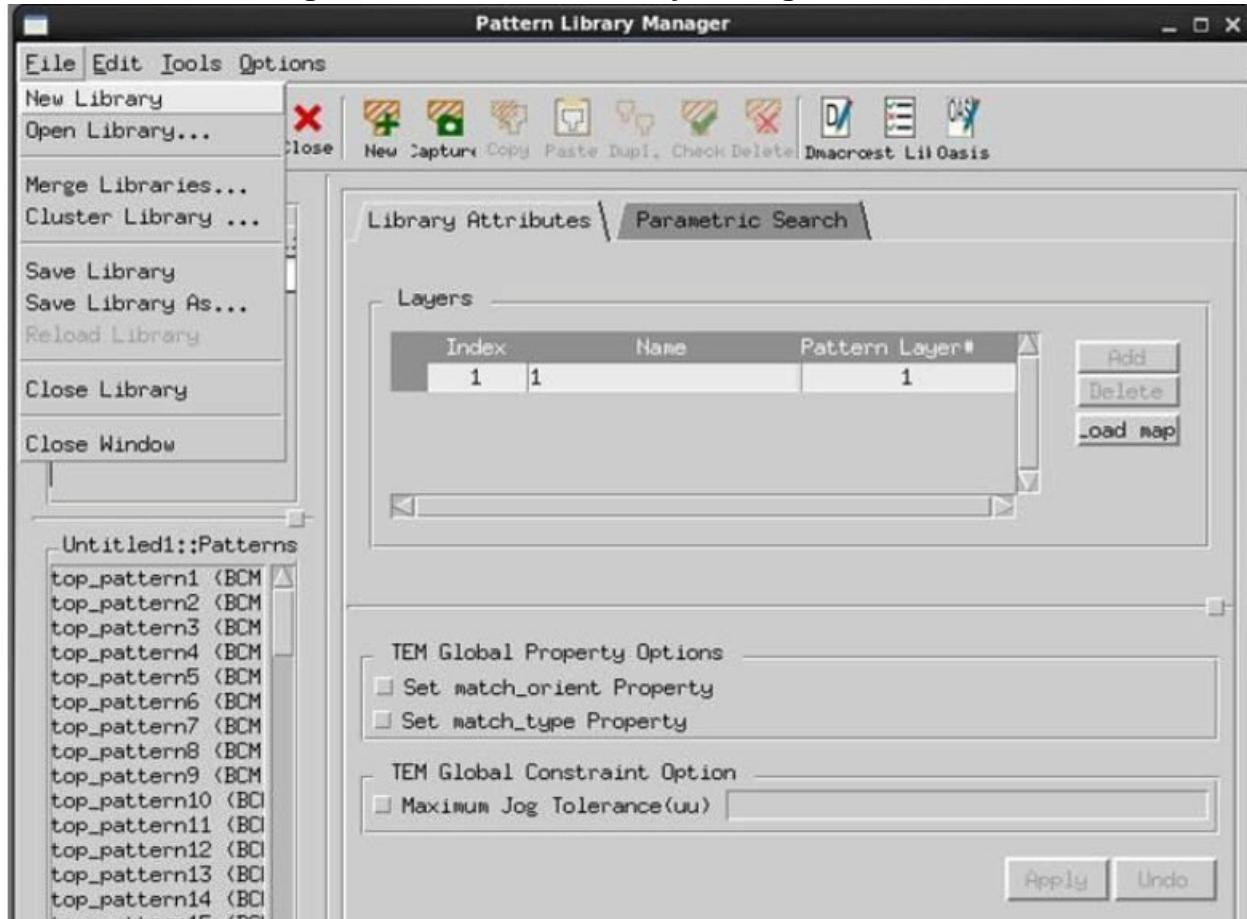
Prerequisites

- You must have valid licensing for Calibre WORKbench and the Calibre Pattern Matching.

Procedure

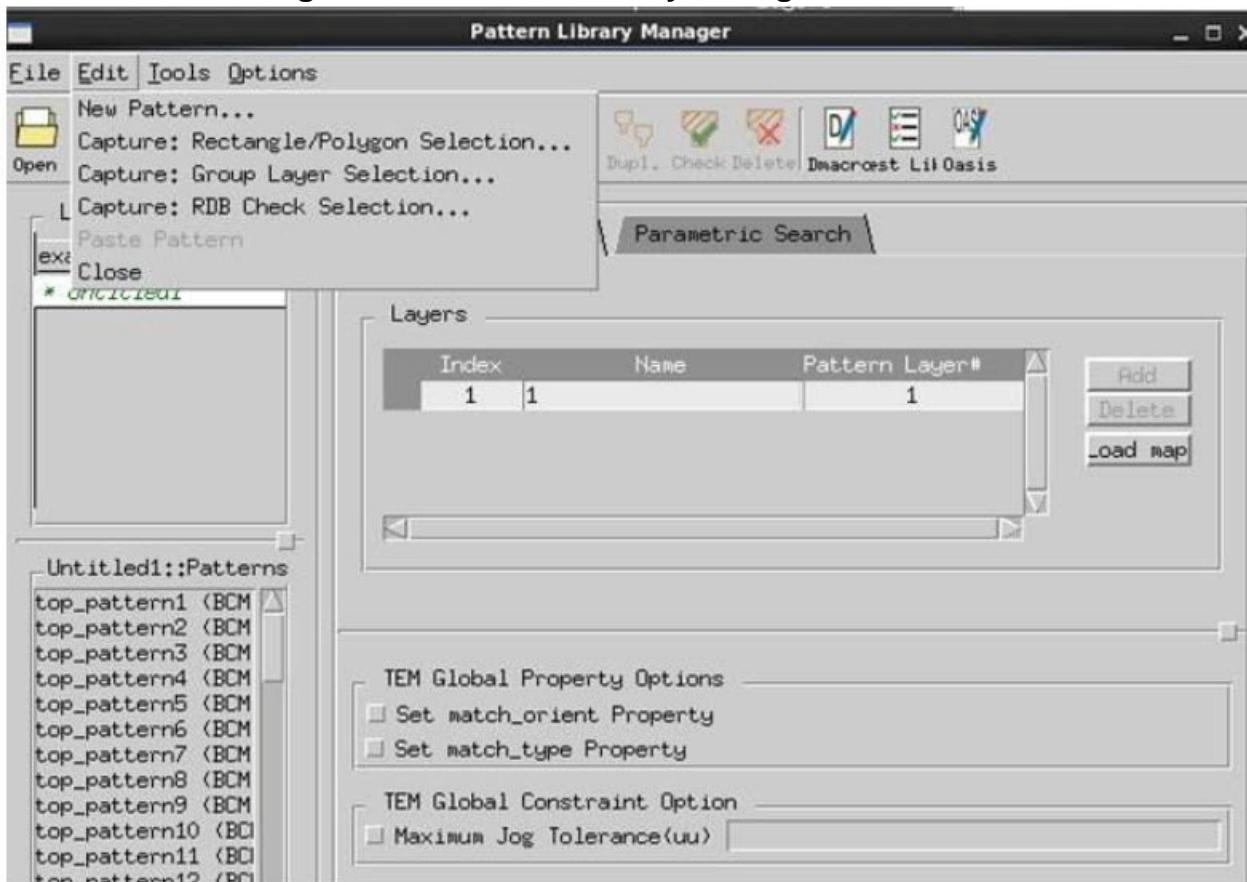
1. Invoke Calibre WORKbench and select **Verification > Run PM Capture**.
2. Invoke the Pattern Library Manager
3. To create a new library, select **File > New Library**.

Figure 2-42. Pattern Library Manager File Menu



4. You can add patterns to the library using options from the **Edit** menu of the Pattern Library Manager:
 - Create a pattern manually and add it to the library
 - Add patterns from RDB checks.
 - Add patterns using a group layer selection
 - Add patterns from polygons selected from Calibre WORKbench.

Figure 2-43. Pattern Library Manager Edit Menu



5. The following are known limitations:
 - The Capture options in the Pattern Library Manager are slower than the DFM Pattern Capture command.
 - The pattern property supports only numeric values
 - The Pattern Library Manager allows you to add constraints and properties to the added patterns.
6. Complete details on the Pattern Library Manager can be found in the [Calibre Pattern Matching User's Manual](#).

Library Creation Flow Example

Using either the DFM Pattern Capture command or the Pattern Matching Manager in Calibre WORKbench, you can create pattern libraries.

Procedure

1. Build a separate library for each metrology algorithm using the DFM Pattern Capture command.

2. Merge the algorithm specific libraries into a single library either through the pdl_lib_mgr batch tool:

```
pdl_lib_mgr merge  
input filename [input filename2] ... output filename
```

or through **File > Merge Libraries** in the Pattern Matching Manager.

3. Compile the generated library to generate the DMACRO (an encrypted version of the pattern library in SVRF format) for use inside CMi using the pdl_lib_mgr batch tool:

```
pdl_lib_mgr compile  
input filename output filename:
```

4. Complete information on the pdl_lib_mgr tool is located in the *Calibre Pattern Matching Reference Manual*.

Using Automatic Identification in CMi

CMi users run the automatic algorithm identification flow by just selecting compiled encrypted SVRF version of the DMACRO pattern library.

CMi automatically generates a layout of merged clips of the measurement points. The pattern matching run generates the resulting RDB file that contains all the matched patterns. CMi examines each measurement point against the result of matched patterns in the RDB file, if exist the measurement point is assigned the algorithm property attached to the matched pattern.

Procedure

1. Invoke CMi.
2. In the Job Configuration area, click **Metrology Algorithm** under the Site Automatic Adjustment section.
3. Browse and select a compiled pattern library.
4. In the Metrology Algorithm options, choose an Algorithm Conflict mechanism:
 - Override — Overrides the user-specified algorithm for measurements, using the automatic algorithm instead. This is the default.
 - Keep — Keeps the user-specified algorithm and run the pattern matching on measurement points with the NULL algorithm value only.
 - Show Conflict — Highlights the measurement point with conflicts in the table. You can resolve the conflict by right-clicking the highlighted row and choosing to use either the pattern matching result or user-specified value.

5. Specify Calibre run settings in the Run Settings options:
 - Run Hier — Specifies a hierarchical run of Calibre. This is the default.
 - Turbo — Specifies a multi-threaded run.
 - Threads — Specifies the number of threads to be used.
6. Click the **Check Marker Algorithm and Type** button in the CMi button bar. The pattern matching run initiates.
7. If a conflict appears, right-click on the highlighted row and choose to resolve the conflict. [Figure 2-44](#) shows a row highlighted with yellow because the user value for the metrology algorithm is different than the pattern matching result.

Figure 2-44. Conflict Highlighted

Name	Feature_type	Position_x	Position_y
Ke_100_300	space	150,792	92,316
Ke_200_600	space	157,792	92,316
Se_200_400	space	165,079	92,316
Se_100_200	space	172,680	92,316
pitch_07A9_B12	width	2527,400	1777,000
pitch_08A9_B12	width	2673,000	1777,400
pitch_07B0_H1	width	2844,400	1700,000
pitch_08B0_H1	width	3025,000	2019,400
spitch_11A6_G10	space	1737,400	2863,000
spitch_12A6_G10	space	1882,000	3112,400
pitch_07A6_G10	space	425,400	43,000
Edit algorithms			
Delete markers			
Resolve Algorithm Conflict			

Log

```
Rendering Region objects in layout...
...Assembling Done in 1 second(s) with (2) Warning(s) and (0) Error(s).
Ready for user input (59 marker(s) rendered).
```

8. Choose either to keep the value calculated automatically, or to use the user value.

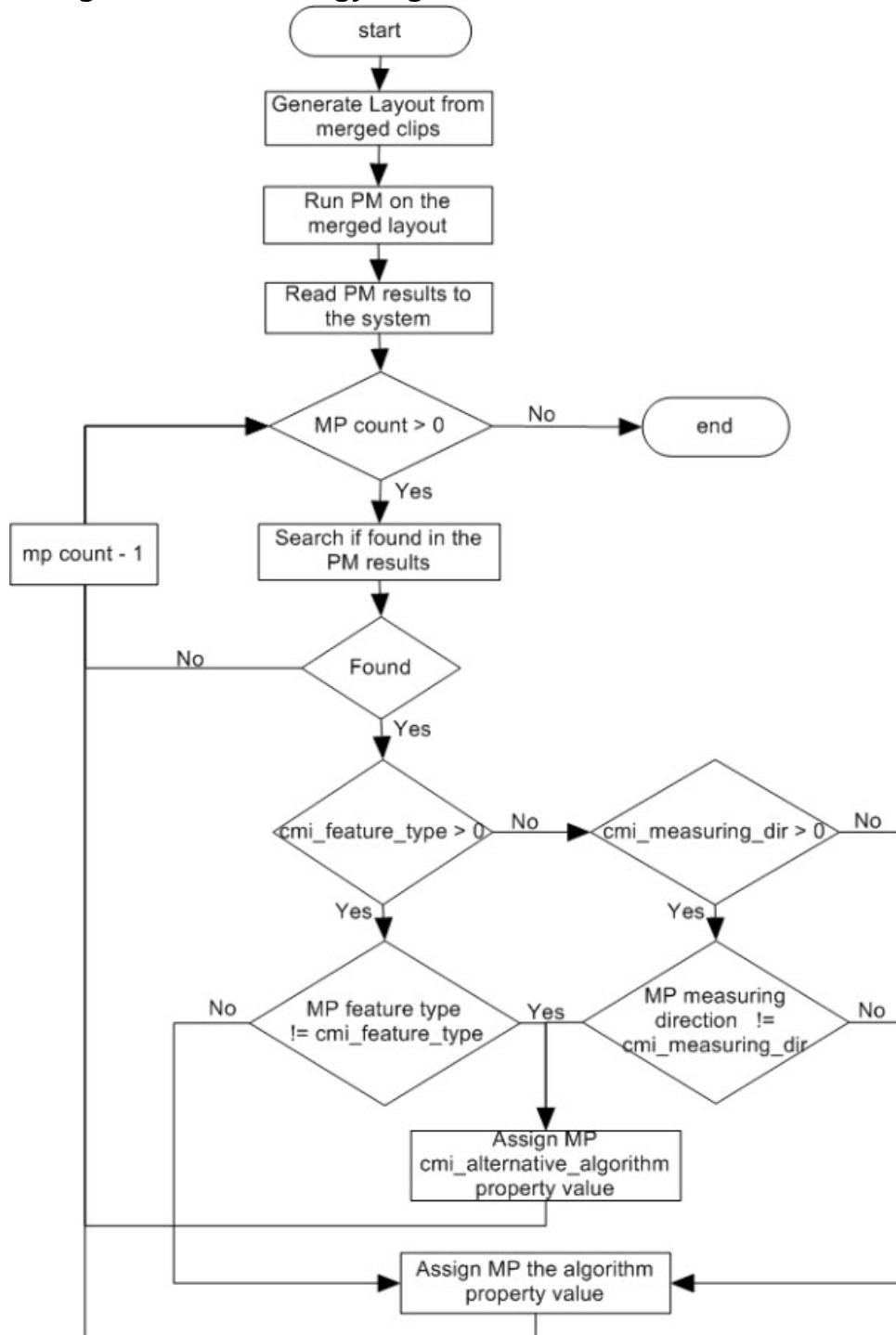
Right-click on the row with the conflict and select the **Resolve Algorithm Conflict** menu item. A pop up dialog box appears with options to either to select the automatic identified value or the user-defined value. Markers that are not assigned an algorithm are highlighted in red as shown in [Figure 2-46](#)

Figure 2-45. Markers Not Assigned Algorithms Highlighted

position_y	Orientation	Target_value	Algorithm	Ce
.825	90	3	min	
.556	90	3	min	
2.122	90	3	min	
3.258	90	3	min	
0.298	90	3	min	
.819	90	3.0	min	
.793	90	3.0	min	
.21	90	3.0	min	
0.723	90	3.0	min	
7.584	90	3.0	min	
3.616	90	3.0	min	
.737	0	1.1	contour	
.37	0	1.1	contour	
.77	90	3	ave	
.825	90	3	ave	
0.556	90	3	ave	

9. When the CMi run is initiated:
10. CMi generates clips with a user-defined size around each measurement point, and merging the generated clips in a single layout to reduce pattern matching run time.
11. CMi runs pattern matching against the generated layout from the merged clips.
12. CMi reads the RDB file generated from the pattern matching run, then examines each measurement point if exists within the extent of any of the pattern matching results.
If the measurement point is found, CMi reads the properties attached to the pattern matching result.
If the `cmi_feature_type` property is defined, then CMi checks if the measuring point is measuring Space or Width.
If the feature type of the measuring point equals the default property value, the measurement point is assigned the `Algorithm` property value; else, it is assigned the `cmi_alternative_algorithm` property value.
The same is applied to `cmi_measuring_dir`. If neither `cmi_feature_type` nor `cmi_measuring_dir` are assigned, then the measurement point is assigned the `Algorithm` property value.
13. The following flow chart illustrates this flow in CMi.

Figure 2-46. Metrology Algorithm Automatic Identification Flow



Pattern Library Specification for Automatic Metrology Algorithm Identification

A pattern library is a collection of one or more patterns created within the Pattern Library Manager or captured using one of the pattern capture operations. When the pattern library is filled with the patterns you want to search for and ready to be included in a pattern matching Calibre run, the pattern library is then exported from the Pattern Library Manager to create an SVRF DMACRO pattern file.

Complete details on the pattern libraries can be found in the *Calibre Pattern Matching User's Manual* and *Calibre Pattern Matching Reference Manual*.

There are specific rules when creating a pattern library for the automatic metrology algorithm identification process.

- A library per layer should be created.
- Patterns names will reflect the feature type of the measurement and the metrology algorithm. For example:

line_avg_1

- An “Algorithm” property must be attached to each pattern. This stores the value corresponding to the metrology algorithm value to be used for the corresponding pattern. The property value is a number. A mapping file that stores the “Algorithm” number and its corresponding Metrology Algorithm String is used to overcome the current limitation that pattern properties should be numbers.

An example map file is as follows:

Figure 2-47. Example Map File

0	ave
1	min
2	max
3	contour
4	contact
5	image
6	ellipse

- Add three more properties to the pattern matching result markers to enable assignment of the same pattern different algorithm. Also, enable the use of the BBOX pattern matching marker type without considering the position of the matched marker.

The algorithm of such patterns varies according to the measurement feature type (width or space) or the measurement direction (horizontal or vertical). The three properties are:

- cmi_alternative_algorithm — Specifies an alternative algorithm value. Expected values are the same as the Algorithm property.

- `cmi_feature_type` — Specifies the default feature type (width/space) for measuring the pattern. If defined, it indicates the need for extra processing in CMi to determine the feature type of the measurement point. Expected pattern property values are 1 for space and 2 for width. If the measurement point feature type equals the property value, then the measurement point is assigned the default algorithm; else, it is assigned to the alternative algorithm.

The following is an example:

```
algorithm = min
cmi_feature_type = 1(space)
cmi_alternative_algorithm = max
```

Use Case 1:

This measurement point is assigned the MIN algorithm as the measurement point measures the default feature type (space).

Use Case 2:

This measurement point is assigned the MAX algorithm as the measurement point is measures width, which is not the default feature type (space).

- `cmi_measuring_dir` — Specifies the default direction of measurement (X/Y) for the pattern. If defined, it indicates the need for extra processing in CMi to determine the direction of measurement for the measurement point. Expected values are 1 for horizontal measurement and 2 for vertical. If the measuring direction of the measurement equals the property value then the measurement point is assigned the default algorithm else is assigned the alternative one.

The following is an example:

Use Case 1:

```
algorithm = min
cmi_measuring_dir = 1
cmi_alternative_algorithm = max
```

In this case, the pattern is assigned the MAX algorithm (the alternative algorithm), and not the MIN algorithm as the measuring direction (the green marker) is 90, which is not equal to the `cmi_measuring_dir` value (1).

Use Case 1:

```
algorithm = max
cmi_measuring_dir = 1
cmi_alternative_algorithm = min
```

In this case, the pattern is assigned the MAX algorithm (the algorithm property) as the measurement point direction is equal to the `cmi_measuring_dir` property value.

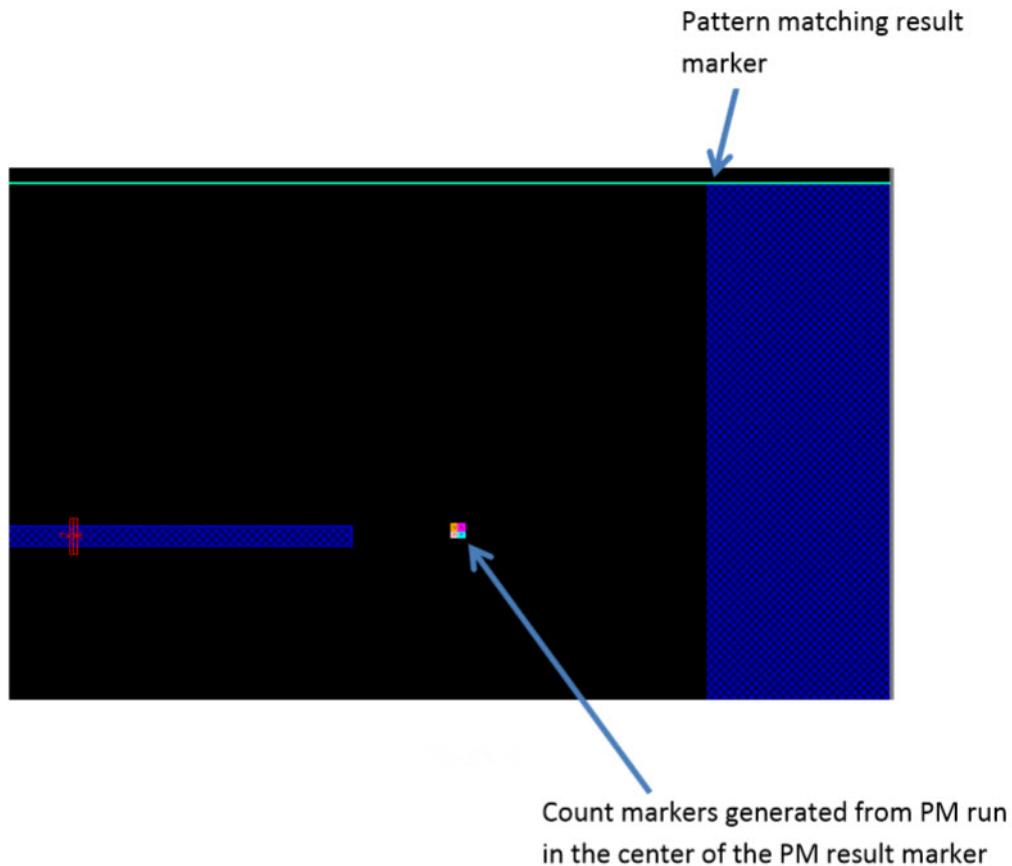
Pattern matching is executed on a single layer that is the result of the Boolean operations in CMi.

Multiple Algorithm Matches for the Same Feature

It is common to have a certain feature in the design to be matched with more than one pattern from the pattern library. Currently, CMi gets the first pattern matching result marker and assigns an algorithm according to the property attached to this pattern, ignoring the possibility of having different matches. CMi detects the existence of multi-match patterns and gives you the ability to manually select the algorithm value from the pattern matching results.

The Pattern Matching utility provides a count marker utility that can be used to help in the case where a feature in the design is matched by more than one pattern.

Figure 2-48. Pattern Matching Run With Count Markers



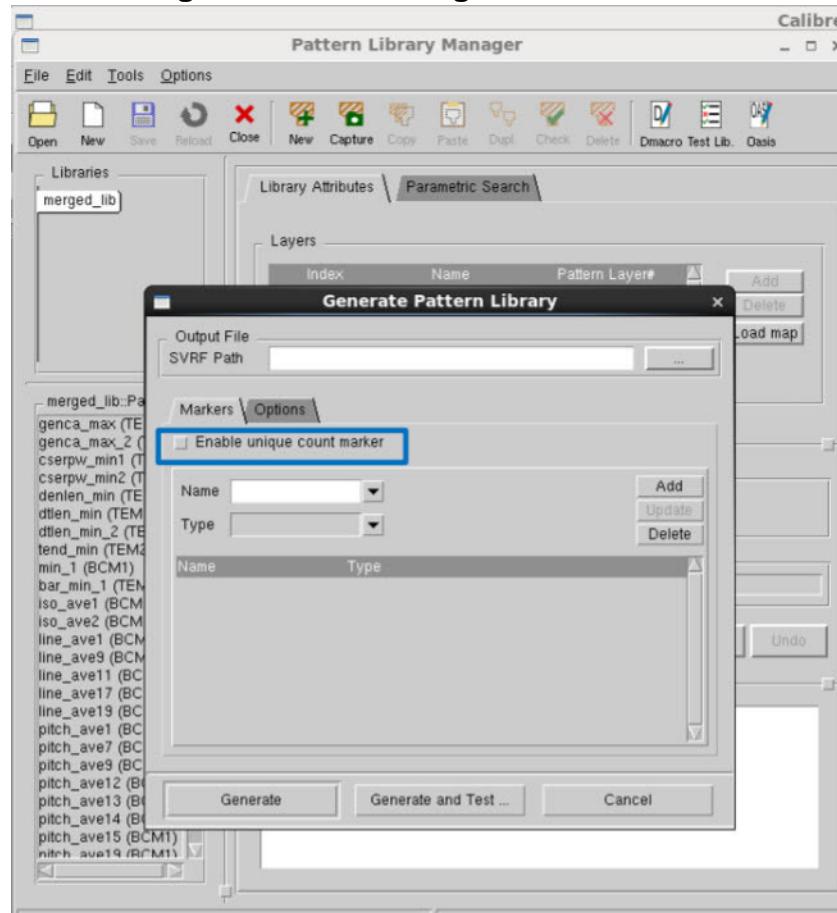
To enable the generation of the count marker during pattern matching run, you can use either the Pattern Library Manager GUI or the `pdl_lib_mgr` batch tool to export a pattern library to use in a Calibre run:

- If you are using the GUI, and you want to also generate a resulting count of matches, click the Enable unique count marker
- If you are using the `pdl_lib_mgr` batch tool and you want to also generate a resulting count of matches, use the `pdl_lib_mgr compile gen_count_layer` option.

Mutli-Match Detection

CMi uses the count markers generated by a pattern matching run to detect the occurrence of conflicts. CMi attaches each count marker result to one of the pattern matching marker results, assigning the pattern matching result different algorithm property values as shown in the following figure:

Figure 2-49. Enabling Count Markers



In the normal CMi run, CMi checks the measurement point position if it is within any of the pattern matching markers extents as shown in the following figure:

Figure 2-50. Pattern Matching Markers

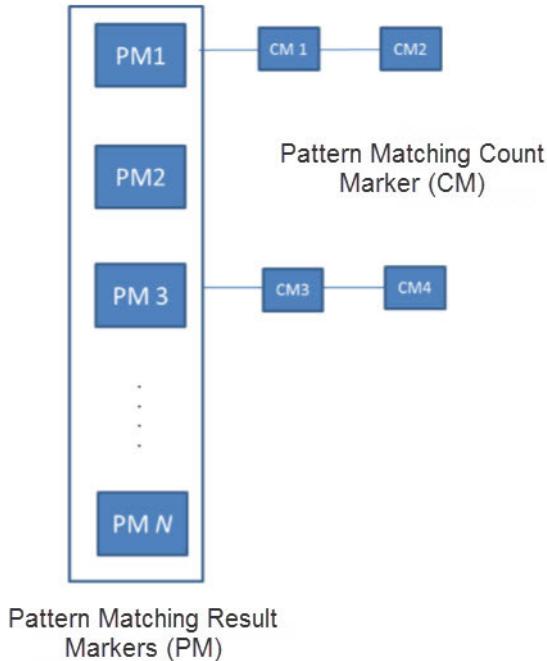
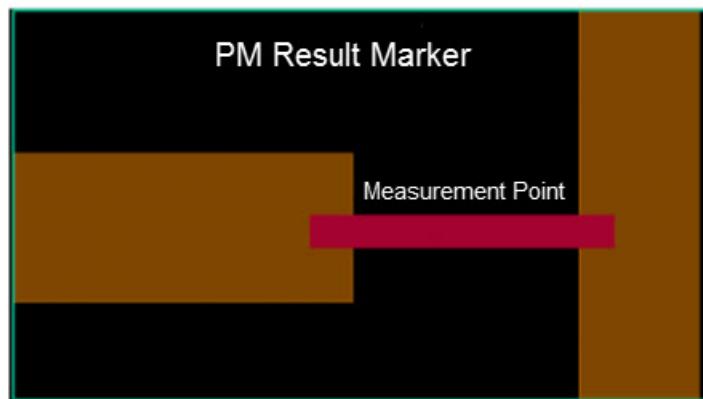


Figure 2-51. Pattern Matching Result Marker



Note

-  The marker is highlighted with any value for the “Algorithm Conflict” job parameter value.
-

Resolving Individual Algorithm Conflicts

CMi can highlight markers with more than one algorithm identified during the automatic algorithm assignment process. You can then manually choose between the identified algorithms to assign to the marker.

Procedure

1. Right-click the highlighted conflict in the Marker window and select “resolve Algorithm conflict”.

Conflicts are highlighted in yellow in the following figure:

Figure 2-52. Algorithm Conflict Highlighted in CMi

Marker_y	Orientation	Target_value	Algorithm	Conf
25	90	3	min	
3	90	3	min	
22	90	3	min	
58	90	3	min	
98	90	3	min	
19	90	3.0	min	
33	90	3.0	min	
	90	3.0	min	
23	90	3.0	min	
84	90	3.0	min	
16	90	3.0	min	
7	0	1.1	contour	
0	0	1.1	contour	

2. A pop-up dialog appears, displaying the possible algorithms to select. In the pop-up dialog box, select between the identified algorithm values and click **OK**.

The algorithm behaviors and what precedence they take in conflict resolution can be modified. This is detailed in “[Modifying Algorithm Conflict Resolution Behavior](#)” on page 87.

Modifying Algorithm Conflict Resolution Behavior

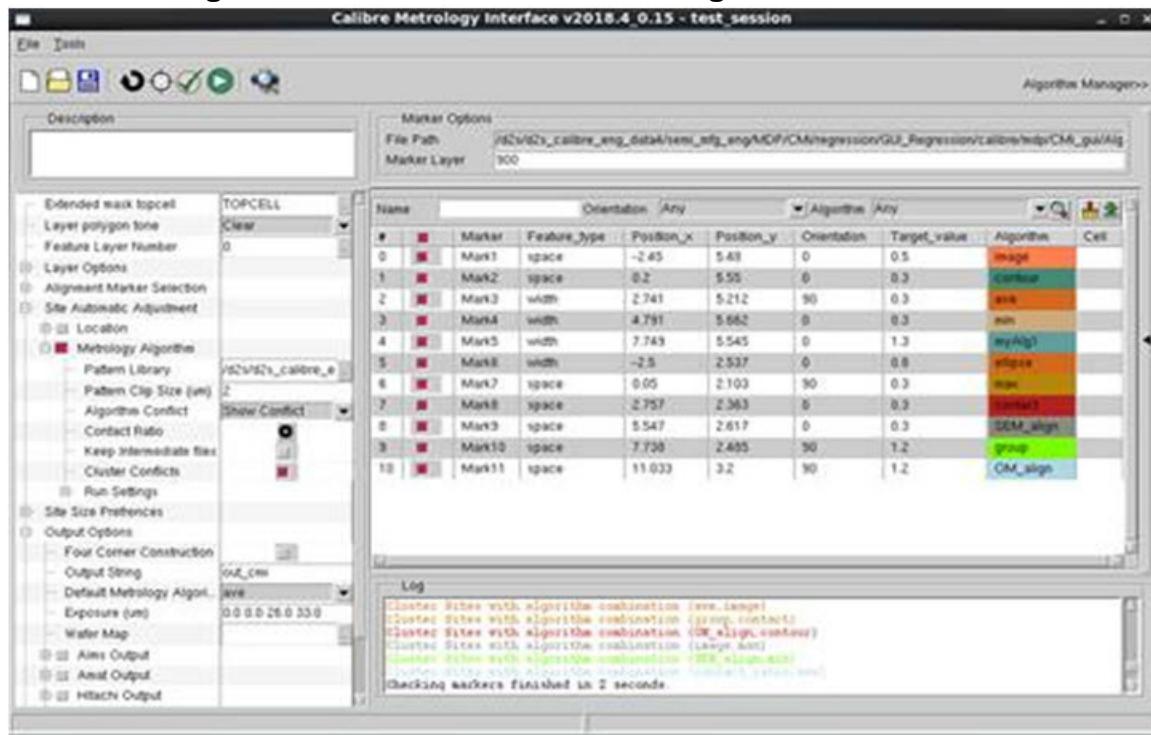
The Calibre Metrology Interface enables you to define commands to resolve conflicts in algorithm identification for sites with multiple possible pattern matching results. You can define priority and preceding commands as well as visual classification of sites with similar algorithm combinations for manual resolution and verification of the solution.

Procedure

1. In the Job Configuration area, click the radio button at **Site Automatic Adjustment > Cluster Conflicts** to enable Cluster Conflicts mode.

By enabling this mode, CMi clusters sites with similar algorithm combinations and the Algorithm field in the table are highlighted by a unique color for each algorithm cluster. The Log window also indicates what each color group means.

Figure 2-53. Sites With Common Algorithms Clustered

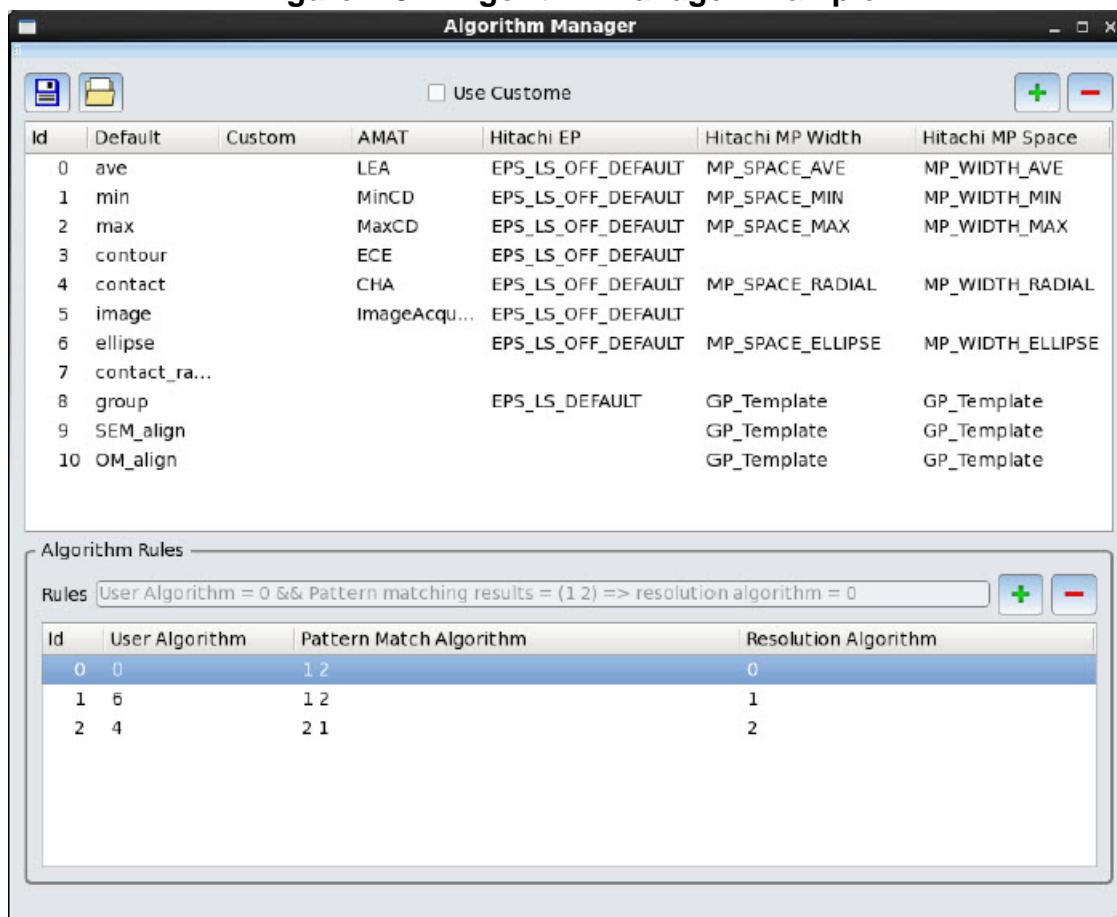


If **Cluster Conflicts** is not enabled, the algorithm conflicts are highlighted in yellow.

2. You can review the automatic conflict resolution settings for each cluster. If you want to change how CMi will resolve conflicts for a cluster, click on the Algorithm column to invoke the Algorithm Conflict Window to specify a different algorithm. Click **OK** to apply the change to the individual entry, or **Apply All** to apply the change to the entire cluster.
3. You can customize algorithm behavior using the Algorithm Manager. In the main CMi window, select **Tools > Algorithms Manager**.

In the Algorithm Manager (see “[Algorithm Manager](#)” on page 135 for complete information), you can specify the algorithm combinations and which algorithm is to be automatically assigned to a site in case of a conflict.

Figure 2-54. Algorithm Manager Example



The Algorithm Manager contains two panes:

- The Algorithm table at the top of the Algorithm Manager lists the individual algorithms, including IDs, default algorithm used, custom names, and AMAT and Hitachi template information. From here, you can add, modify, or delete individual algorithms.
- The Algorithm Rules section enables you to define different algorithm combinations (user algorithms and algorithms found by pattern matching) and define the user-required choice for those combinations. The Rules list only accepts Algorithm ID numbers that are defined in the Algorithm Manager.

After you make modifications, you can review the automatic resolution results through the manual conflict resolution process again.

Automatically Selecting Alignment Markers Using Pattern Matching

The Calibre Metrology Interface uses Calibre's pattern matching capability to find the local alignment marker structures in the design, allowing you to prepare a pattern library that includes all the structures. CMi then prepares a layout from the original design, where it includes every measurement point within a clip of a size equal to the search region. In the case of having nearby measurement points on overlapped search regions, CMi can identify these situations to avoid missing the local alignment markers.

The metrology process requires the definition of alignment markers; the local alignment markers are required to be close to the measurement points in the design for the SEM machine to correctly capture the feature. Typically, designers define a set of features or patterns that indicate the position of the local alignment markers. The process of manually identifying the position of local alignment markers in current designs can be a difficult and time-consuming process. The Calibre Metrology Interface enables you to automate the identification of these markers based on pre-defined structures. The local alignment mark is taken at the same magnification (or close to the same magnification) as for the measurement points. It must be close to the measurement and is done for all measurements, not just once for the entire wafer.

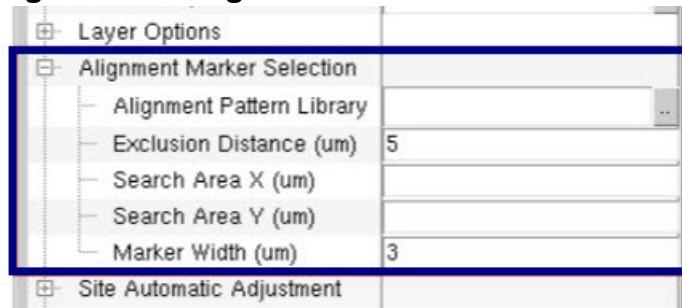
Procedure

1. To set up the automatic selection of local alignment markers, go to the Job Configuration area in CMi.

Enter the following the Alignment Marker Selection options

- Local alignment structure pattern encrypted library (DMACRO)
- Exclusion distance (um)
- Search area (X,Y coordinates)
- Marker width

Figure 2-55. Alignment Marker Selection in CMi



2. Click the target icon in the CMi tool bar.

Results

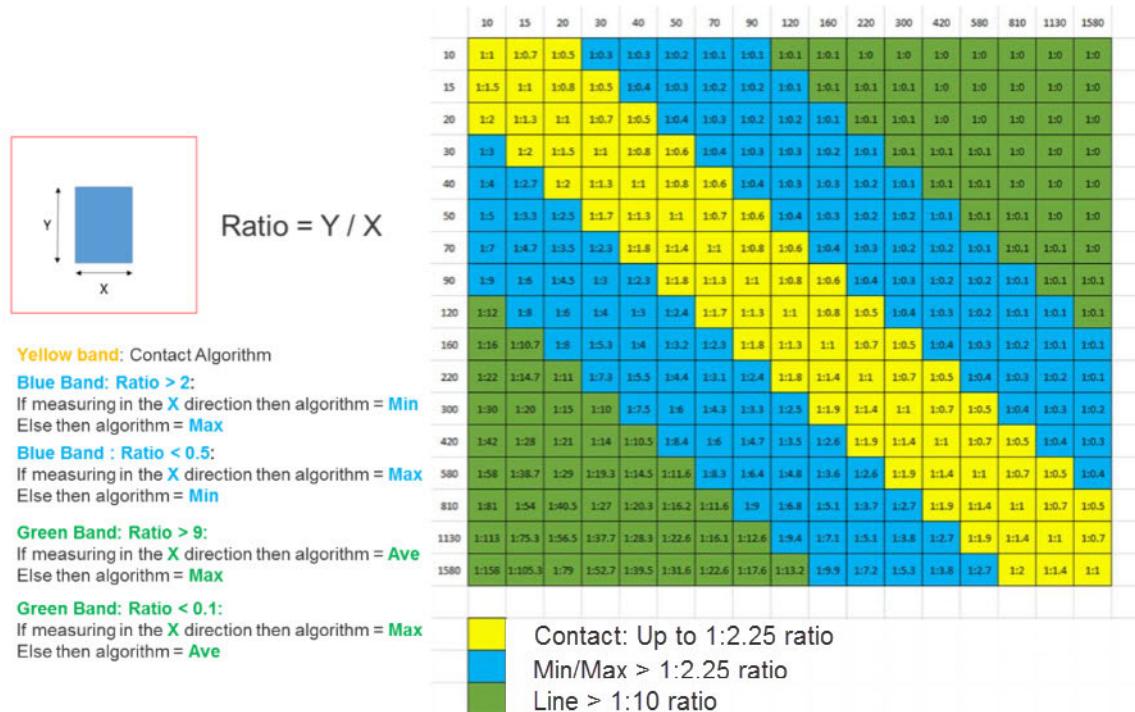
CMi runs the pattern matching engine using the given pattern library. If a match occurs, CMi automatically generates the marker and adds it to the layout and the markers table with the SEM_align metrology algorithm assigned.

Defining Contact Pattern Ratios

The Calibre Metrology Interface enables you to define a single contact pattern that could be assigned multiple algorithm values according to the pattern size ratio to improve the consistency in recipe generation. This allows you to set the ratios and corresponding algorithms according to your own specifications.

The following figure shows an example contact pattern and the ratio calculation as well as the different ratio ranges. This figure also defines the attached algorithm for each ratio range in the horizontal and vertical measurement direction.

Figure 2-56. Contact Pattern and Ratio Calculation



You define ratio ranges and assigned algorithms to a contact pattern through the Contact Ratio Editor in CMi. This tool gives you the ability to define different ratio rules for the Contact pattern.

Procedure

- To invoke the Contact Ratio Editor, in the Job Configuration area of CMi, click the **Contact Ratio** radio button under the Metrology Algorithm section to enable it.

Click the edit icon (the black wheel icon to the right of **Contact Ratio**).

Figure 2-57. Contact Ratio Option

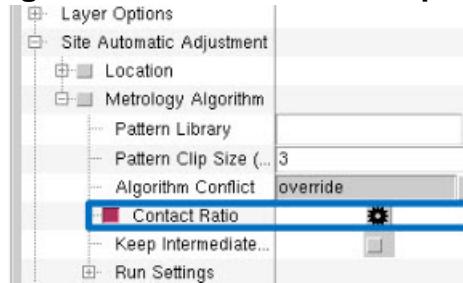
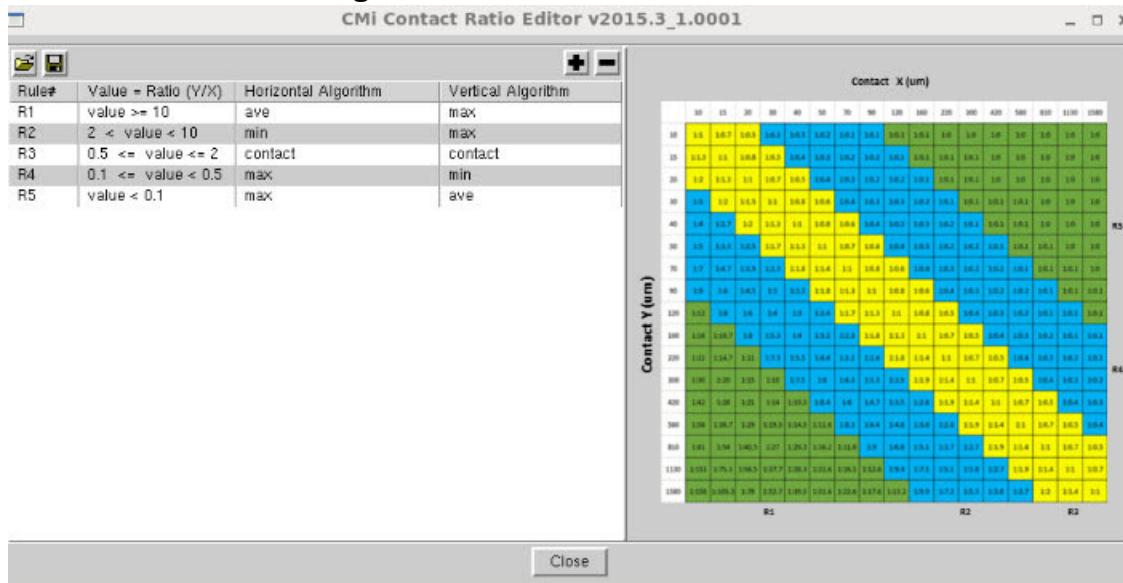


Figure 2-58. Contact Ratio Editor



2. Each row in the Contact Ratio Editor allows you to define rule name and the ratio range by defining an inequality. You can also assign the algorithms for each pattern according to the measurement direction.

The defined rules should be reflected in the Automatic algorithm identification results that use the contact pattern; this is done by defining TVF functions in the SV RF used for running pattern matching. The following is an example:

```
proc find_algorithm { in_y in_x } {
    set x [$in_x]
    set y [$in_y]
    set ratio [expr $y/$x]
    if {$ratio >= 10} {
        set algorithm 0
        return $algorithm
    } elseif {$ratio < 10 && $ratio > 2} {
        set algorithm 1
        return $algorithm
    } elseif {$ratio <= 2 && $ratio >= 0.5} {
        set algorithm 4
        return $algorithm
    } elseif {$ratio < 0.5 && $ratio >= 0.1} {
        set algorithm 2
        return $algorithm
    } elseif {$ratio < 0.1} {
        set algorithm 2
        return $algorithm
    }
}
```

3. These rules can be saved in an XML file format that can be loaded and used in different sessions by clicking the floppy disk icon at the top left of the Contact Ratio Editor. Saving a session generates an XML file that saves the contact ratio rules. Loading session will automatically load the contact ratio rule file.

You can optionally load alternate contact ratio sessions by clicking the folder icon in the upper left corner of the Contact Ratio Editor. The following is an example output XML file.

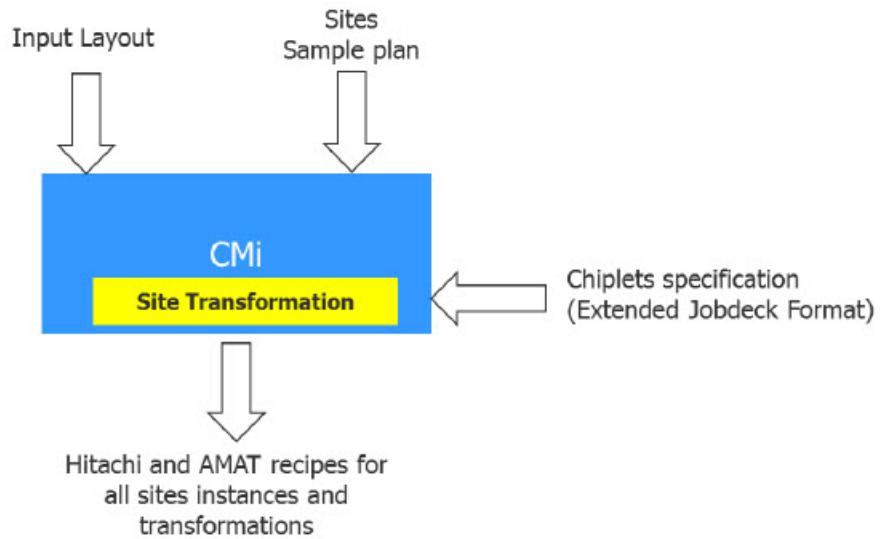
```
<?xml version="1.0" encoding="UTF-8"?>-<CMi_Contact_Ratio_Rules>
  -<Rules>
    -<Rule>
      <upper_ratio/>
      <upper_ratio_operator/>
      <lower_ratio>10</lower_ratio>
      <lower_ratio_operator>=</lower_ratio_operator>
      <horizontal_algorithm>0</horizontal_algorithm>
      <vertical_algorithm>2</vertical_algorithm>
    </Rule>
    -<Rule>
      <upper_ratio>10</upper_ratio>
      <upper_ratio_operator/>
      <lower_ratio>2 </lower_ratio>
      <lower_ratio_operator/>
      <horizontal_algorithm>1</horizontal_algorithm>
      <vertical_algorithm>2</vertical_algorithm>
    </Rule>
    -<Rule>
      <upper_ratio>2</upper_ratio>
      <upper_ratio_operator>=</upper_ratio_operator>
      <lower_ratio>0.5 </lower_ratio>
      <lower_ratio_operator>=</lower_ratio_operator>
      <horizontal_algorithm>4</horizontal_algorithm>
      <vertical_algorithm>4</vertical_algorithm>
    </Rule>
    -<Rule>
      <upper_ratio>0.5</upper_ratio>
      <upper_ratio_operator/>
      <lower_ratio>0.1 </lower_ratio>
      <lower_ratio_operator>=</lower_ratio_operator>
      <horizontal_algorithm>2</horizontal_algorithm>
      <vertical_algorithm>1</vertical_algorithm>
    </Rule>
    -<Rule>
      <upper_ratio>0.1</upper_ratio>
      <upper_ratio_operator/>
      <lower_ratio/>
      <lower_ratio_operator/>
      <horizontal_algorithm>2</horizontal_algorithm>
      <vertical_algorithm>0</vertical_algorithm>
    </Rule>
  </Rules>
</CMi_Contact_Ratio_Rules>
```

Managing Chiplets

You can use CMi to generate metrology recipes for sites in different chiplet placements. Chiplets are designs that utilize far smaller microscopic circuitry than currently used to create electronic components such as microprocessors and memory.

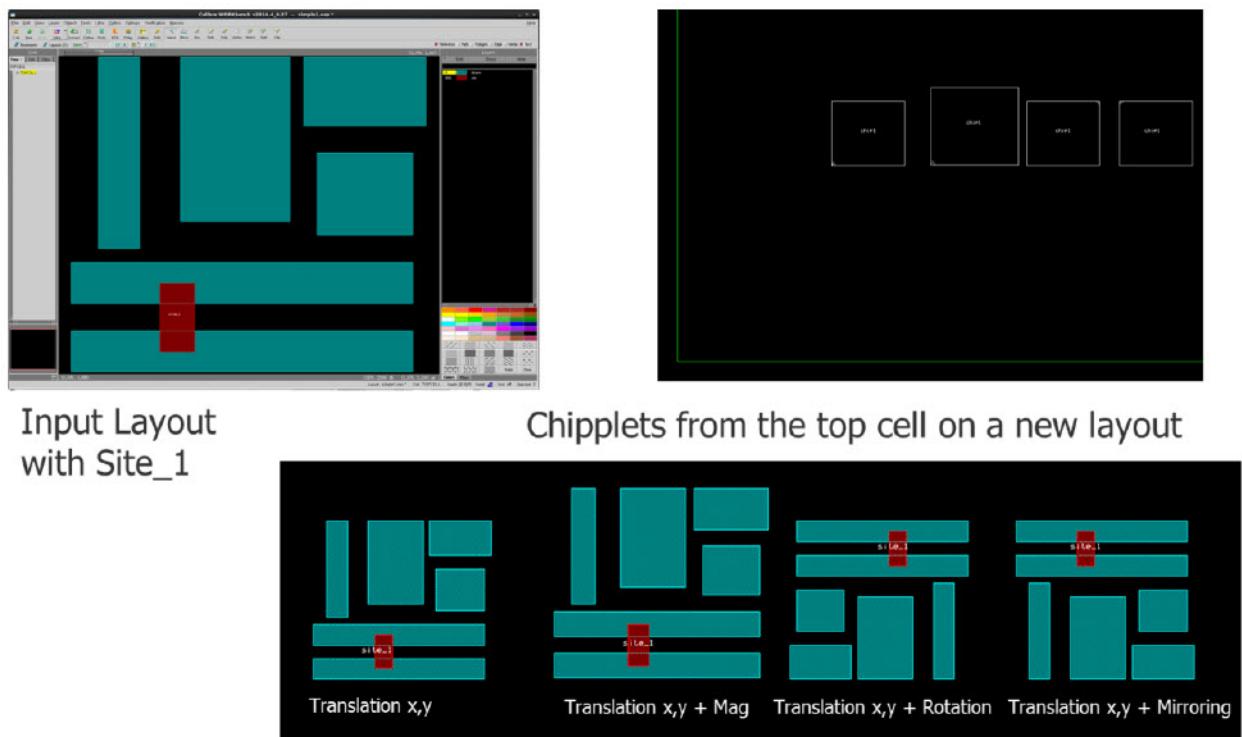
Use the CMi Chiplet Manager to specify different placements and transformations of chiplets to the input layout, enabling the generation of metrology recipes for the same site in different transformations.

Figure 2-59. Chiplet Manager Work Flow



CMi accepts a user specification of the chiplets transformation either through a specific UI or through extended job deck format. According to the specification, the transformation matrix is applied on the input sites and generates the metrology recipe accordingly.

Figure 2-60. Generating a Recipe With Chiplets



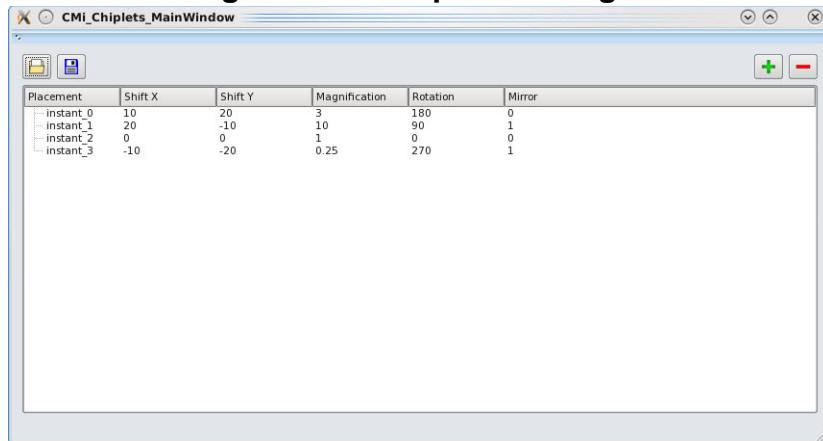
Prerequisites

- A layout file (containing the design to create chiplet transformations from) loaded into Calibre WORKbench.
- An extended job deck where the placement of the chiplets are defined. This is used for saving a loading purposes.

Procedure

1. Perform the steps described in “[CMi Workflow Overview](#)” on page 22. You should have a list of markers defined (or imported from a previous session) and run options in the Job Configuration area specified.
2. In CMi, select **Tools > Chiplet Manager**. The CMi Chiplets Manager appears.

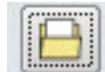
Figure 2-61. Chiplets Manager



3. In the Chiplets Manager, you can perform the following:

- Load chiplet transformations — Click the Open File button to load an extended job deck containing chiplet transformation settings. See “[Chiplets Manager](#)” on page 132 for information on the extended job deck.

Figure 2-62. Open File



- Add a new chiplet transformation settings — Click the Add (+) button to add a new entry field representing a chiplet, then manually enter settings for shift x, shift y, magnification, rotation, and mirror transforms.

Figure 2-63. Add Chiplet Transforms



- Delete an entry — Select an entry and click the Delete (-) button.

Figure 2-64. Delete Chiplet Transforms



- Modify an existing entry — Click on an entry and you can manually edit the shift X, shift Y, magnification, rotation, and mirror transforms.

You can also load a CMi session file, and if the session includes the job deck file (chiplets session) the Chiplets Manager appears automatically.

4. Save your transformation changes to an output file by clicking the Save icon.

Figure 2-65. Save Transforms

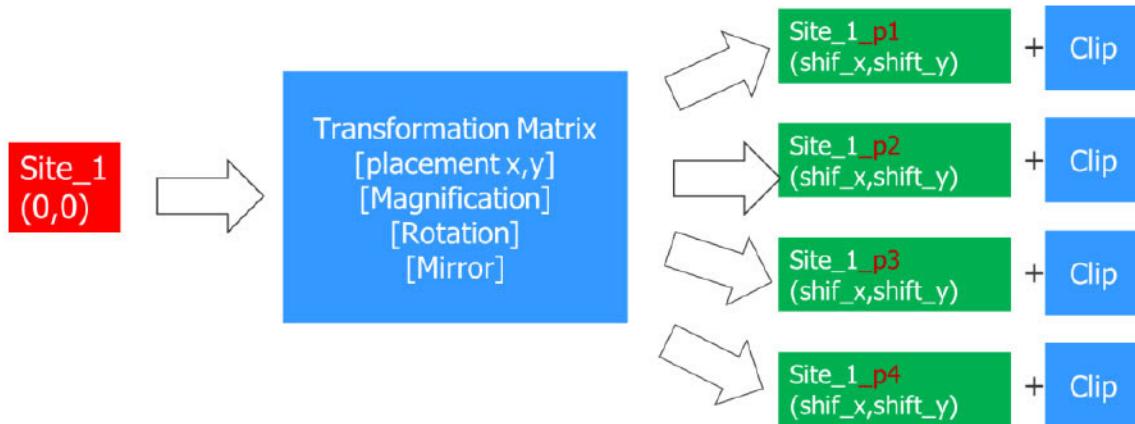


Saving the CMi session automatically saves the settings in the Chiplets Manager as well.

Results

The metrology recipes created by this process output the sites defined by the chiplets specification using the transformation matrix. CMi outputs the site IDs in a recipe using the input site ID as a prefix, then appending “_placement#” to each entry. The generated layout clips (GDS or OASIS) are captured for each chiplet placement. The following figure illustrates how the input sites are generated in the output.

Figure 2-66. Chiplet Output Flow



Chapter 3

CMi GUI Reference

The Calibre Metrology Interface is a single GUI interface with several functional sections.

The CMi Window.....	100
Assemble Jobdeck Options	107
Layer Options.....	109
Alignment Marker Options.....	110
Site Automatic Adjustment Options	111
Site Size Preferences	115
Output Options	116
Display Preferences	120
Sites Table and Properties	121
Layer Processor Window	125
Chiplets Manager.....	132
Algorithm Manager.....	135
Allowed Characters in Cell Names.....	137

The CMi Window

To access: Invoke CMi from Calibre WORKbench or Calibre MDPview, using the **Tools > Calibre Metrology Interface** option.

The Calibre Metrology Interface contains several functional regions to perform metrology operations. Show or hide the Machine Area and the SEM Diagnosis Tool by clicking on the arrow bar to the right of the Sites Table.

Figure 3-1. Calibre Metrology Interface Main Window

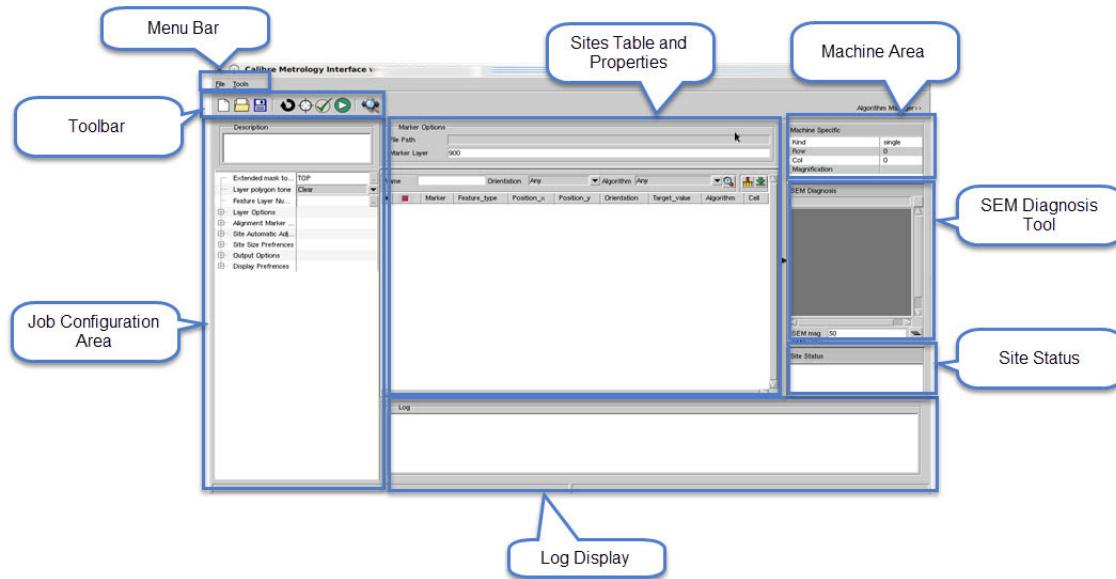
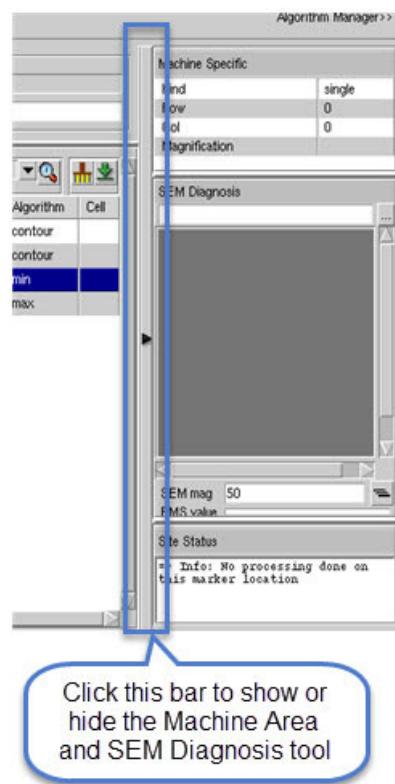


Figure 3-2. Show or Hide Bar



Objects

Table 3-1. Calibre Metrology Interface Fields

Field	Description
Menu Bar	
File	
New Session	Open a new CMi session.
Open Session	Open an existing CMi session file.
Save Session	Saves settings of the current CMi session to a file.
Save Session As	Saves CMi session settings under a user-selected name.
Load Jobspec	Loads a Jobspec Tcl file.
Save JobSpec Template	Saves the jobspec in the specified file name as a template for later use. Note that layout or site information is not saved. The recommended file name format is <i>jobspec*.tcl</i> . The name <i>jobspec.tcl</i> in the startup directory is automatically loaded on startup.

Table 3-1. Calibre Metrology Interface Fields (cont.)

Field	Description
Import Markers	<p>Imports markers from a specified source. Supported formats include:</p> <ul style="list-style-type: none"> • Marker Layer: Directly from a marker layer. • CSV: The most common marker input format for general application. Markers, Regions, and Searches may be imported from a CSV format table. This table format is like the CSV output format. It is easiest to create an input file by editing a previous output file. • SS: Spreadsheet format for interfacing from other test pattern metrology tools. • Gauges: A gauge is a drawn line along which a CD or a space measurement is performed. A gauge data file (.gd) is similar to a spreadsheet (.ss) file in that both contain CD or space measurements. • Rdb: ASCII format written from a Calibre job, commonly used for the OPCVerify hotspot flow. • XY List: Enables the import of a list of marker XY coordinates that represents the measurement location. The tool automatically identifies the best measurement point around the specified coordinates and calculates the measurement direction and the feature type width and space. The XY list is in CSV format, as in the following example: <pre style="margin-left: 20px;">1487.23, 95867.7 201, 95667.7 301, 4567.932</pre> <p>When you import markers, a dialog box appears that allows you to select the source.</p> <p>If you are importing from a marker layer, you can optionally select Find Hierarchical Markers from this dialog box. When this option is enabled, marker cells are created over the pattern files to allow drawing of hierarchical markers.</p>
Export Markers	<p>Exports markers to one of the following file types:</p> <ul style="list-style-type: none"> • CSV: CSV file. • SS: Spreadsheet format for interfacing from other test pattern metrology tools. • GG: Gauges file.
Tools	
Reload Markers	<p>Reloads markers into CMi, updating any changes that have occurred. This option only appears if you have markers already loaded into CMi.</p>

Table 3-1. Calibre Metrology Interface Fields (cont.)

Field		Description
	Scan Markers Algorithm and Type	<p>Scans all markers in the layout and checks the feature type of each marker (either width or space) and the algorithms assigned to the markers in this job.</p> <p>Note the following:</p> <ul style="list-style-type: none"> If the Location is enabled (in the Site Automatic Adjustment section), then the validation process is run on the loaded markers in the Table View and updates the feature type, Position_X, Position_Y and Target_value (if not set). Unvalidated markers are highlighted in red. If the Location is deactivated, then the feature type of each marker is automatically updated. If Metrology Algorithm (in the Site Automatic Adjustment section) is enabled then the Automatic algorithm identification is run. <p>The information gathered during scanning is used to highlight the required Hitachi templates in the Algorithm Manager. The scan results appears in the feature type column in the marker's entry in the Sites table.</p>
	Assemble Jobdeck	<p>Invokes the Assemble Jobdeck Options dialog box.</p> <p>See “Assemble Jobdeck Options” on page 107 for complete details.</p> <p>When an Assemble is initiated, any open layouts will be closed, and any previous manual drawing will be lost.</p> <p>You may want to save the layout at this point. Manual drawing can be saved. For mebes_job input data, an OASIS format can be saved for faster subsequent runs.</p> <p>If the layout appears empty during review, then there may be an error in the specification of the Marker, Region, or Measurement layers. For example, specifying the Measurement Point Output Layer to be the input layer will erase the mask data in the current working session.</p>
	Layer Processing (Booleans)	Opens the Layer Processor Window to define boolean operations for specified layers.
	Wafer Map Manager	Generates a wafer map. See “ Editing a Wafer Map ” on page 63 for further information.
	Chiplets Manager	Invokes the Chiplets Manager dialog box. This enables you to create a metrology recipe for sites defined in multiple chiplets from a layout, so sites within a chip can be found in different placements in the design with different transformations. See “ Chiplets Manager ” on page 132 for details.

Table 3-1. Calibre Metrology Interface Fields (cont.)

Field		Description
	Generate Outputs (Machine Inputs)	Generates CMi output. all output files. This generates all output files (and is required after all marker entry and algorithm edits). When the job has completed, you can review the Measurement Point results in the layout (click the View Results button).
Toolbar		
	New Session 	Open a new CMi session. This is the same as the File > New Session option.
	Open Session 	Open an existing CMi session file. This is the same as the File > Open Session option.
	Save Session 	Saves settings of the current CMi session. This is the same as the File > Save Session option.
	Reload Markers to Layout 	Reloads markers, updating any changes that have occurred. This is the same as the Tools > Reload Markers option.
	Locate Alignment Markers 	Runs a pattern matching engine to locate local alignment markers. See “ Automatically Selecting Alignment Markers Using Pattern Matching ” on page 90for further information.
	Scan Markers Algorithm and Type 	Scans all markers in the layout and check the feature type of each marker (either width/space) and the algorithms assigned to the markers in this job. This is the same as the Tools > Scan Markers Algorithm and Type option.
	Generate Output 	Generates CMi output based on inputs defined in the Job Configuration Area and Sites table. This is the same as the Tools > Generate Outputs (Machine Inputs) option.
	View Results 	Opens a dialog to open an output file (created from the previous run). Various viewers handle the variety of output file types - text, image, layout, and spreadsheet. It is recommended that you have installed a program for viewing CSV files as a spreadsheet (such as Open Office), and an image viewing tool. The environment variable CMI_BROWSER_PATH defines the program to view .htm documents.

Table 3-1. Calibre Metrology Interface Fields (cont.)

Field	Description
Job Configuration Area	Specifies job configuration parameters.
Description	Allows you to enter description of the purpose of the session. This field is an arbitrary comment indicator that is loaded/saved with the session file.
Extended mask topcell	<p>The name of the top cell for the mask must be specified. This must be an existing cell name from the mask file. All results are generated with respect to the context of this cell, including the origin and magnification of the coordinate system.</p> <p>A common value for MEBES job deck files is “TOP.” If this field is blank, then one of the layout topcell names will be automatically entered at Assembly. If there are multiple topcells in the layout, then correct this value and re-assemble.</p> <p>The top cell name has restrictions for characters allowed. The allowed and not allowed characters are described in “Allowed Characters in Cell Names” on page 137.</p>
Layer Polygon Tone	<p>Sets the tonality of the layer to be measured. The layer tonality is used to determine the feature type to be measured, either width or space.</p> <p>If set to Clear, all targets measured on a polygon are considered to be Width type and all other in-between polygons are Space type. If set to Dark, then all targets measured on the polygon are considered Space type and all other in-between polygons are Width.</p> <p>See “Layer Polygon Tone” on page 29 for further information.</p>
Feature Layer Number	<p>This layer is used for validation of the markers and determination to target CD. This specification controls the same parameter with one in the Site Automatic Adjustment section. Datatype is specified as a decimal layer value, for example, “2.1”.</p> <p>Multiple feature layers are allowed, separated by spaces, for example, “123 124 7.” They are treated as if the layers had been merged into one. You can also input multiple layers by clicking the “..” button to invoke the Input Layers List dialog box (see Figure 3-12).</p>
Layer Options	Contains controls to define layer parameters for processing Click the plus sign (+) to expand the list of options. See “ Layer Options ” on page 109 for complete information.
Site Automatic Adjustment	Contains controls to define measurement parameters. Click the plus sign (+) to expand the list of options. See “ Site Automatic Adjustment Options ” on page 111 for complete information.

Table 3-1. Calibre Metrology Interface Fields (cont.)

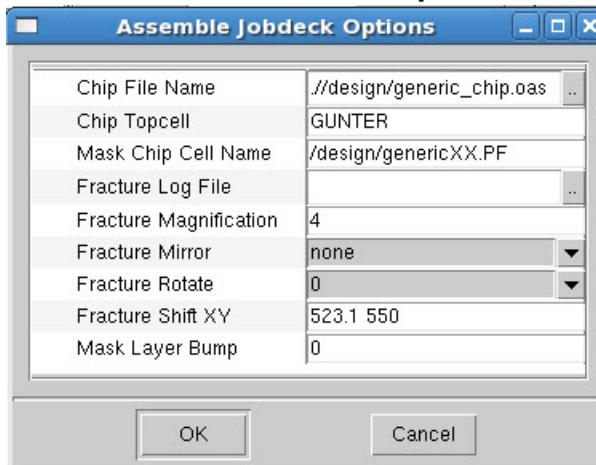
Field	Description
Site Size Preferences	Contains controls to specify visual display parameters for site markers using image size, group size, SEM_align size, and OM_align size. These options are used if no global window width or size were specified in the Display Preferences section or in the CSV file. See “ Site Size Preferences ” on page 115 for complete information.
Output Options	Contains controls to specify the output format for the Calibre Metrology Interface. Click the plus sign (+) to expand the list of options. See “ Output Options ” on page 116 for complete information.
Display Preferences	Contains controls to specify global window width and size display preferences for site markers. If you do not specify a window width or size (either using these options or in a loaded CSV file), CMi will use the options set in Site Size Preferences. See “ Display Preferences ” on page 120 for complete information.
Sites Table and Properties	Contains the Sites table and associated marker properties. This area allows you to view and manage marker and site information in the layout. You can edit the site properties by clicking on a properties field. Refer to “ Sites Table and Properties ” on page 121 for complete information.
Machine Area	Sites properties related to specific machine recipes area. Currently, this is AMAT format only. The properties listed are: <ul style="list-style-type: none"> • kind: Allows you to designate multiple measurements per marker for AMAT SEM (see “Designate Multiple Measurements Per Marker” on page 48). • rows: Lists the number of rows in an array • cols: Lists the number of columns in an array
SEM Diagnosis	Overlay SEM images in (.tiff) format and compare them to the design clip generated from CMi. See “ Overlaying SEM Images and Design Clips ” on page 68 for further information.
Site Status	Displays site status information.
Log Display	Displays any warning or error generated during a CMi run.

Assemble Jobdeck Options

To access: In the CMi window, select **Tools > Assemble Jobdeck**

The Assemble Jobdeck Options dialog box contains controls for the Assembly operation.

Figure 3-3. Assemble Jobdeck Options Dialog Box



Objects

Table 3-2. Assembly Options

Field	Description
Chip File Name	<p>One OASIS or GDS-format chip layout may be overlaid onto the mask layout. This provides data for additional layers, and the chip hierarchy. Additional layers may be useful for visual reference when drawing markers, or for importing graphical marker locations from the designer. Drawn layers that are different from the mask data (for example, pre- OPC) may be useful in determining the target CD.</p> <p>The chip hierarchy may be useful to add markers which repeat in many locations across the mask.</p> <p>When this option is used, carefully check the overlay of the data to the mask file pattern layer. If the data or transformations are not correctly specified, then the chip data will not be aligned properly to the mask, and the CMi job can produce incorrect results.</p>
Chip Top cell name	The name of the cell in the chip file. The top cell name has restrictions for characters allowed. The allowed and not allowed characters are described in “ Allowed Characters in Cell Names ” on page 137.
Mask Chip cell name	The corresponding name of the chip cell in the mask layout. After assembly, the number of instances of the given cell in the mask is indicated in the center gray box.

Table 3-2. Assembly Options (cont.)

Field	Description
Fracture Log File	Loads the transform values from a specified Calibre log file. The fracture log (the <code>log_file_name</code> specified in the Calibre FRACTURE command, not the Calibre log file) contains the values for Fracture Magnification, Fracture Mirror, Fracture Rotate, and Fracture Shift.
Fracture Magnification	After assembly, the approximate magnification is indicated in the center gray box for reference. This value is determined from the extents of the data, and may be less than the correct value due to the presence of other layers and data outside the fracture extent.
Fracture Mirror	This value allows only none and x transformations. For y, xy, or yx, the rotation and shift must be adjusted.
Fracture Rotate	The choices are 0, 90, 180, or 270 degrees counter-clockwise.
Fracture Shift X Y	After assembly, the approximate shift is indicated in the center gray box for reference. This value is determined from the extents of the data, and may be less than the correct value due to the presence of other layers and data outside or not touching the fracture extent. It may also be incorrect due to chip rotation by 90 or 270 degrees. The correct value can be obtained from the lower-left coordinate of the fracture extent.
Mask Layer bump	The Bump value is added to the layer numbers in the job deck to avoid collisions with the chip layer numbers. It is common to use a value like 1000.

Layer Options

To access: In the CMi window, click **Layer Options** in the Job Configuration Area

The Layer Options section contains controls to define layer parameters for processing.

Objects

Table 3-3. Layer Options

Field	Description
Enable Layer Processing	Enables layer processing. This is used in conjunction with the Layer Processor window. See “ Layer Processor Window ” on page 125.
Region Outline layer number	This layer is used to input and construct regions from groupings of measurements. The layer may contain rectangles and text. Other shapes will be ignored with a warning. Text on this layer will be assigned as region names when contained within shapes. CSV regions are rendered on this layer during assembly.
Measurement (Output) layer number	When you generate CMi output, this layer is used to draw the resulting measurement locations. Any input data on this layer will be erased. You can examine these shapes after run for accuracy of the results.
Capture Field (Output) layer number	This layer is used for rendering the Capture Field boundaries when generating CMi output. The field may be left blank. Data on this layer will be erased and re-rendered during the output generation.
Alignment Image (Output) Layer Number.	This layer is used for rendering the alignment image boundaries points during output generation. The field may be left blank.
Failed Markers Layer Number	The sites that fail validation are rendered on this layer to allow browsing and diagnosis. The field may be left blank.
Layer Property File Name	The layer file is used to maintain visibility, color, and fill settings. The Calibre WORKbench Layer >Save Layer Properties command can be used to generate a layer file. It is suggested to adjust the layer properties and save the file after output generation. The specified file is loaded during both assembly and output generation in subsequent jobs.

Alignment Marker Options

To access: In the CMi window, click **Alignment Marker Selection** in the Job Configuration Area

The Alignment Marker section contains controls to define parameters for automatic selection of alignment markers.

Objects

Table 3-4. Alignment Marker Selection Options

Field	Description
Alignment Pattern Library	Specifies Local alignment structure pattern encrypted library (DMACRO).
Exclusion Distance (um)	Specifies the exclusion distance (in um).
Search Area X (um)	Specifies the marker search area along the X-axis (in um).
Search Area Y (um)	Specifies the marker search area along the Y-axis (in um).
Marker Width (um)	Specifies the width of the marker (in um).

Site Automatic Adjustment Options

To access: In the CMi window, click **Site Automatic Adjustment** in the Job Configuration Area

The Site Automatic Adjustment section defines parameters for automatic site parameter identification, including location and metrology algorithm.

Objects

Table 3-5. Site Automatic Adjustment Options

Field	Description
Location	<p>Location validation is the process of examining the layout at the Marker positions to generate Measurements. In most applications, this should be enabled.</p> <p>Unvalidated measurements are transferred directly from Markers without regard for the layout shapes. This may be useful for special applications where the measurement positions and sizes have been accurately generated by other means. For example, non-validated markers can be drawn manually between simulation contours.</p> <p>The following occurs when the Scan Markers Algorithm and Type button is clicked:</p> <ul style="list-style-type: none">• If the Location is enabled (in the Site Automatic Adjustment section), then the validation process is run on the loaded markers in the Table View and updates the feature type, Position_X, Position_Y and Target_value (if not set). Unvalidated markers are highlighted with orange and red color.• If the Location is deactivated, then the feature type of each marker is automatically updated.
Override Target CD	Overrides the target CD values, which is performed with the check algorithm and type step processing. This allows you to control whether or not the target is to be overridden or not using the new calculation methods. This enables you to perform calculations of a site's target value even if the target value has already been defined in the input CSV file. By default, this option is enabled.

Table 3-5. Site Automatic Adjustment Options (cont.)

Field	Description
Override Direction	<p>Overrides the default CD orientation set by CMi (detect and adjust the orientation favoring the shorter dimension). Enable this option to measure using the longer dimension instead.</p> <ul style="list-style-type: none"> • If this option is enabled, orientation of all 6 markers are fixed for any CD (1,3,5,10 um). • If this option is not enabled, orientation of all 6 markers are not changed for any CD (1,3,5,10 um).
Validation Type	<p>Two validation method choices are available:</p> <ul style="list-style-type: none"> • Touching Edges: Requires exactly two perpendicular edges touching or contained in the marker region. This method is commonly used for hand-drawn or csv markers. • Center Point: Locates two perpendicular edges nearest to the center of the marker extent.
Validate On	<p>Two choices are available for validation:</p> <ul style="list-style-type: none"> • Layer: • Contour: <p>The Measure Layer must be specified. If this layer has been specified for other purposes (Marker, Region, or Measurement), then the data on that layer is erased. This specification controls the same parameter with one in the Layer Options section.</p>
Maximum CD	<p>Specifies the maximum CD acceptable for this technology. The default value is 3 um.</p>
Metrology Algorithm	<p>When enabled, the pattern matching flow runs if the required parameters are set. See “Automatic Metrology Algorithm Identification” on page 70 for complete information.</p>

Table 3-5. Site Automatic Adjustment Options (cont.)

Field	Description														
Pattern Library	<p>Specifies the compiled encrypted SVRF version of the DMACRO pattern library file. This is a required parameter; CMi issues a warning if the library has not been specified.</p> <p>The Algorithm Mapping file should exist in the same path of the selected pattern matching library. If it does not exist, the default values as shown in the following figure are used:</p> <p style="text-align: center;">Figure 3-4. Map File Defaults</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>ave</td></tr> <tr><td>1</td><td>min</td></tr> <tr><td>2</td><td>max</td></tr> <tr><td>3</td><td>contour</td></tr> <tr><td>4</td><td>contact</td></tr> <tr><td>5</td><td>image</td></tr> <tr><td>6</td><td>ellipse</td></tr> </table>	0	ave	1	min	2	max	3	contour	4	contact	5	image	6	ellipse
0	ave														
1	min														
2	max														
3	contour														
4	contact														
5	image														
6	ellipse														
Pattern Clip Size	Specifies the size of the clip generated around each measurement point. The merged clips layout is used as an input to the pattern matching run. The default value is 3 um.														
Algorithm Conflict	<p>Specifies the conflict resolution mechanism if the pattern matching result contradicts the manually-set algorithm. One of the following can be specified:</p> <ul style="list-style-type: none"> • Override — The automatic algorithm identification overrides the user-specified algorithm for measurements. • Keep — Keeps the user-specified algorithm and run the pattern matching on measurement points with the NULL algorithm value only. • Show Conflict — Highlights the measurement point with conflicts in the table. You can resolve the conflict by right-clicking the highlighted row and choosing to use either the pattern matching result or user-specified value. 														
Contact Ratio	Invokes the Contact Ratio Editor, which allows you to define different ratio rules for a contact pattern. Refer to “ Defining Contact Pattern Ratios ” on page 91 for further information.														

Table 3-5. Site Automatic Adjustment Options (cont.)

Field	Description
Keep Intermediate Files	If enabled, retains pattern matching intermediate files. The files are: <ul style="list-style-type: none"> • Pattern matching run log file • Layout that the pattern matching ran on • Pattern matching run results RDB files
Cluster Conflicts	Specifies a mode to display and resolve conflicts in algorithm identification in the Sites Table. When Cluster Conflicts is enabled, the Sites Table color codes the sites based on algorithm and the Log window indicates what each color group means. If Cluster Conflicts is not enabled, the column color is yellow where conflicts occur. Refer to “ Modifying Algorithm Conflict Resolution Behavior ” on page 87 for further information.
Run Settings	Defines the run options required to run the Calibre pattern matching engine: <ul style="list-style-type: none"> • Run Hier — Specifies a hierarchical run of Calibre. • Turbo — Specifies a multi-threaded run. • Threads — Specifies the number of threads to be used.

Site Size Preferences

To access: In the CMi window, click **Site Size Preferences** in the Job Configuration Area

The Site Size Preferences section defines the size preferences for sites.

Objects

If you load a CSV file with no defined size or window width value (specified by the Display Preferences options as documented in “[Display Preferences](#)” on page 120), CMi uses the Site Size Preferences parameters to draw the visual markers for each site.

Table 3-6. Site Size Preference Options

Field	Description
Image Size (um)	Specifies the visual marker size for sites using the Image algorithm in um.
Group Size (um)	Specifies the visual marker size for sites using the Group algorithm in um.
SEM_align Size (um)	Specifies the visual marker size for sites using the SEM_align algorithm in um.
OM_align Size (um)	Specifies the visual marker size for sites using the OM_align algorithm in um.

Output Options

To access: In the CMi window, click **Output Options** in the Job Configuration Area

The Layer Options section contains controls to specify the output format for the Calibre Metrology Interface.

Objects

Table 3-7. Output Options

Field	Description
Four Corner Construction	Constructs a four-corner square around the SEM/OM target. See “ Constructing a Four Corner Shape ” on page 62 for further information.
Output String	The name to be used for the results directory and the prefix of various output files.
Default Metrology Algorithm	The value for metrology algorithm which is applied to all sites which are not specified (indicated as “<undefined>” in the Sites table and New Marker dialog box). Refer to the section “ CSV Table Construction ” on page 49 for a list of the possible algorithms. Applying the most common value in this field can reduce the work for setting the algorithm values, especially when markers which have been imported from a text file.
Exposure (um)	Specifies the exposure field (in ums) of the chip as four numeric values, separated by spaces.
Wafer Map	Loads a wafer map file. See “ Editing a Wafer Map ” on page 63 for further information.
Tolerance Default (um)	This default is used when no tolerance value is specified in the CSV Marker object. A common value is 0.04 for 4% tolerance. The tolerance value for each measurement is passed to the metrology system for use in filtering and analyzing the results. It does not otherwise affect the operation of CMi.
Capture Width (um)	The width of a capture field image which represents the SEM field-of-view. This default value is applied to sites where the field is not explicitly given (ungrouped measurements of type ave, min, max, or contact).
Capture Tolerance (um)	The tolerance of the position of a capture field image. This value is subtracted from the boundary for the effective capture field size.

Table 3-7. Output Options (cont.)

Field	Description
Mask Background in Setup File	The mask background (clear or dark). This is only indicated from the value specified in the setup file, and cannot be changed directly from the CMi GUI.
Image Width (um)	The required DFM value for the size of simulated alignment images is 10 um. Other applications may allow a different size. A warning is issued for other values.
Pixel Size (um)	The required DFM value for the resolution of simulated alignment images is 0.045 um. Other applications may allow a different resolution. A warning is issued for other values.
AMAT Output	An enable button for AMAT format output.
Master Mask Name	The master mask name is transferred to the AMAT output file. CMi does not verify or otherwise use this value.
Second Mask Name	An empty value for a second mask name designates “Functional Usage” mode. The second mask name, when non-blank, designates “Poly-on-active” mode. This causes an additional mask name to be defined in the AMAT output file using the Additional Clip Layers.
Exposures	The exposures values, when non-blank designates “Double Exposure” mode. The list of values are applied sequentially to the multiple feature layers. Typically, this is a sequence of “1” and “2” with the same number values as feature layers.
Image Content	The Image Content value is transferred to the AMAT output file. CMi does not verify or otherwise use this value.
Device Type	The Device Type value is transferred to the AMAT output file. CMi does not verify or otherwise use this value.
Hitachi Output	An enable button for Hitachi format output.
Version	Select the version: 1.02, 02.00, or 9

Table 3-7. Output Options (cont.)

Field	Description
Truncate Site Names	<p>Activates or deactivates truncation of site names. The Calibre Metrology Interface, by default, truncates target names to a 10-character limit.</p> <p>CMi also generates a CSV file that maps the truncated target names to the original names. This file is in ASCII format and maintains the same sequence of the input targets.</p> <p>The CSV output map file is generated in the output folder with a name using the following format:</p> <p>\$output_string_target_name_map.csv</p> <p>This file is sorted with the Index value marker and uses the following format:</p> <p>Id, Short Name, Name</p>
Keep Right Most Characters	If truncation is enabled, you can preserve the right-most two characters in the target name by clicking this radio button.
Design Rotation	Specifies the design orientation (up, right, bottom or left).
Exposure X (um) Exposure Y (um)	Displays the exposure field size for the X and Y axis (calculated by the Exposure entry in the general output options). If there were less than 4 numbers in the Exposure entry, the values of Exposure X or Y will be in zeroes.
Maximum Magnification	Specifies the maximum magnification value when the feature size is smaller than a ratio of the FOV (specified using the “FOV ratio” field).
Minimum Magnification	Specifies a minimum magnification value when the feature size is greater than or equal to ratio of the FOV (specified using the “FOV ratio” field).
Image Capture Magnification	Specifies the image capture magnification value in cases where an Image algorithm is used.
MROI Magnification	Specifies the magnification value of the Multiple Region of Interest (MROI) EP marker. This is required if MROI markers are used (they are used with the Group algorithm).
SEM Align Magnification	Specifies the magnification value if the SEM_align algorithm is used.
OM Align Magnification	Specifies the magnification value if the OM_align algorithm is used.

Table 3-7. Output Options (cont.)

Field	Description
FOV Ratio	Specifies the ratio of the (Field of View) FOV that will be compared with the feature size. This ratio will be used to determine whether minimum or maximum magnification settings are used. The default value is $2/3 = 0.66$.
Layout Clip Output	An activation button for layout clip output.
Merge Clips	Activate the button to create a single merged layout clip file. A deactivated button creates individual layout clip files for each site.
Clip Format	Choice of GDSII or OASIS clip format.
Layout Clip Size	The width and height of the layout clips, in microns, surrounding the center of the measurement site.
OM Clip Size (um)	<p>Specifies the dimensions of a Global Alignment (GA) Optical Microscope (OM) clip from a target and merge this clip with the other target clips.</p> <p>This generates a GDS square clip with a size equal to the OM Clip Size value around the center of the marker with the “OM” algorithm. The generated GDS clip will be merged with other SEM GDS clips from targets, and will be in the final merged clips layout.</p> <p>Markers with “OM” algorithms will not be validated and will not be output in the Hitachi file.</p>
Additional Clip Layers	Additional layers from the layout to be included in layout clips. The feature layers, the clip extent, and the measurement site are automatically included.
Visible Under Layer	Additional layers from the layout are to be included in the AMAT or Hitachi profiles.
CSV Output	An enable button for a CSV output file. This file is used for viewing the results and is available for editing and as an input to subsequent runs (read text objects from file). This is recommended to preserve hand-drawn markers.

Display Preferences

To access: In the CMi window, click **Display Preferences** in the Job Configuration Area

The Display Preferences section specifies the global visual display parameters for site markers, including window width and size.

Objects

If you load a CSV file with no defined size or window_width value CMi uses the Size Site Preferences parameters (as documented in “[Site Size Preferences](#)” on page 115) instead to draw the visual markers for each site.

Table 3-8. Display Preferences Options

Field	Description
Window Width (um)	Specifies the window width display for site markers in um.
Size (um)	Specifies the site marker display size in um.

Sites Table and Properties

To access: Center of the CMi window, beneath the Marker Options pane

The Sites table allows you to view and manage marker and site information in the layout. This includes the name, x-y position, size, orientation, width, target value, tolerance, algorithm, magnification, and cell used.

Figure 3-5. Sites Table and Properties

Marker Options

File Path: /home/choi/ekit_ret/Hitachi_1/marker_ex.csv
Marker Layer: 900

Sites Table

#	Marker	Feature_type	Position_x	Position_y	Orientation	Target_value	Algorithm	Cell
0	Mark8	width	0.655	22.818	90	2.292	ave	
1	Mark10	width	0.573	15.033	90	2.292	min	
2	Mark11	width	1.802	1.596	90	2.292	min	
3	Mark12	width	1.229	-11.784	90	2.294	ave	
4	Mark13	width	3.768	-28.341	90	2.292	max	
5	Mark14	width	3.194	-40.392	90	2.292	ave	
6	Mark16	width	32.168	23.841	90	2.292	min	
7	Mark17	width	32.258	16.056	90	2.294	ave	
8	Mark18	width	32.250	2.619	90	2.294	max	
9	Mark19	width	32.659	-10.761	90	2.292	ave	
10	Mark20	width	34.623	-27.318	90	2.294	min	
11	Mark21	width	34.541	-39.369	90	2.292	max	
12	Mark30	space	1.909	9.737	0	0.455	contour	
13	Mark31	space	15.985	1.870	0	0.455	contour	

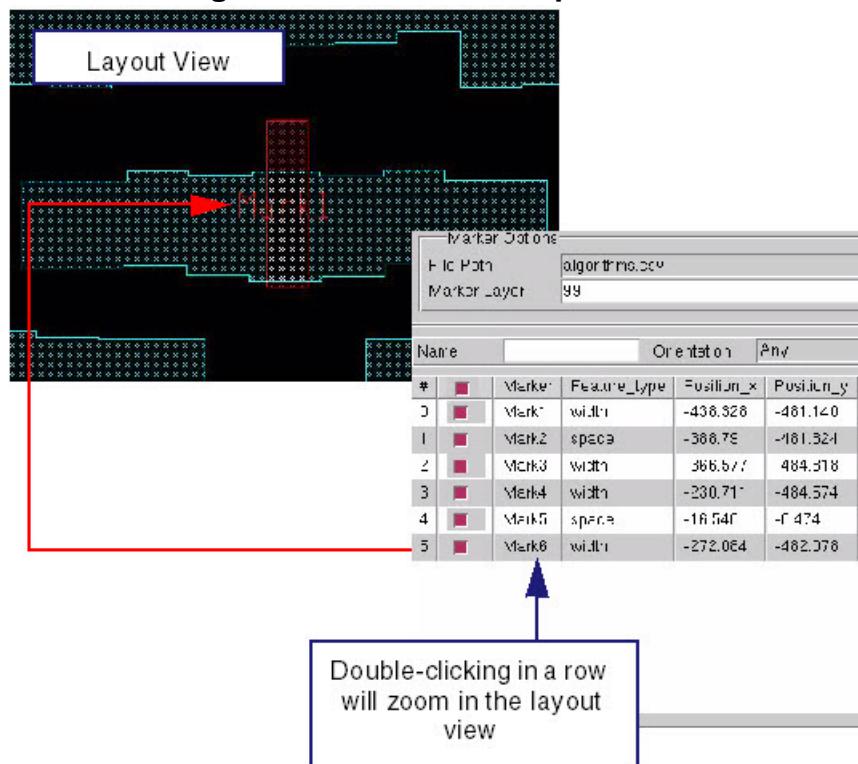
Description

When a CMi session is loaded, the markers form the layout or text files are added to the table. Whenever you add a new marker using the New Marker dialog box, an entry is automatically added to the bottom row of the table as well as visually on the layout.

Algorithms can be scanned to be input by clicking the **Scan Markers or Algorithm Type** button, or be set for each individual marker in the table.

You can double-click on a table row to zoom to the marker location the layout (see [Figure 3-6](#)). The column width in the table can be adjusted. Place the cursor over a vertical divider. The cursor shape will change to a double-headed arrow. Left-drag the divider to the desired position.

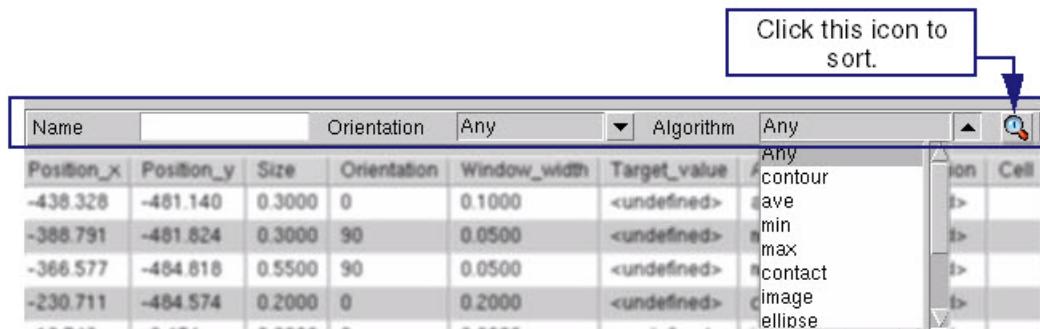
Figure 3-6. Sites Table Operation



Sort Sites Table Entries

You can sort and filter entries using the filters at the top of the Sites table. The table entries can be sorted by name, orientation, and algorithm. Enter or select a criteria from one of the three filters, then click the magnifying glass icon.

Figure 3-7. Sorting the Sites Table



Objects

Table 3-9. Sites Table Fields

Fields	Description
Marker Options	Specifies the marker file path and marker layer number.

Table 3-9. Sites Table Fields (cont.)

Fields		Description
	File Path	Specifies the path to the CSV file containing marker information.
	Marker Layer	Specifies the layer number containing the markers.
Sorting Bar		Filters entries according to a specified criteria: marker name, orientation, or algorithm.
	 Sort	Sorts markers based on marker name, orientation, and algorithm.
	 Clear All Highlights	Clears all marker highlights.
	 Re-Import Markers	Re-imports markers from a file.
Sites Table		Lists all markers and related information.
	Marker	Specifies the marker name.
	Feature_type	Specifies the feature type.
	Position_x	Specifies the marker position along the x-axis.
	Position_y	Specifies the marker position along the y-axis.
	Size	Specifies the marker size.
	Window_width	Specifies the width of the window.
	Target_width	Specifies the width of the target.

Table 3-9. Sites Table Fields (cont.)

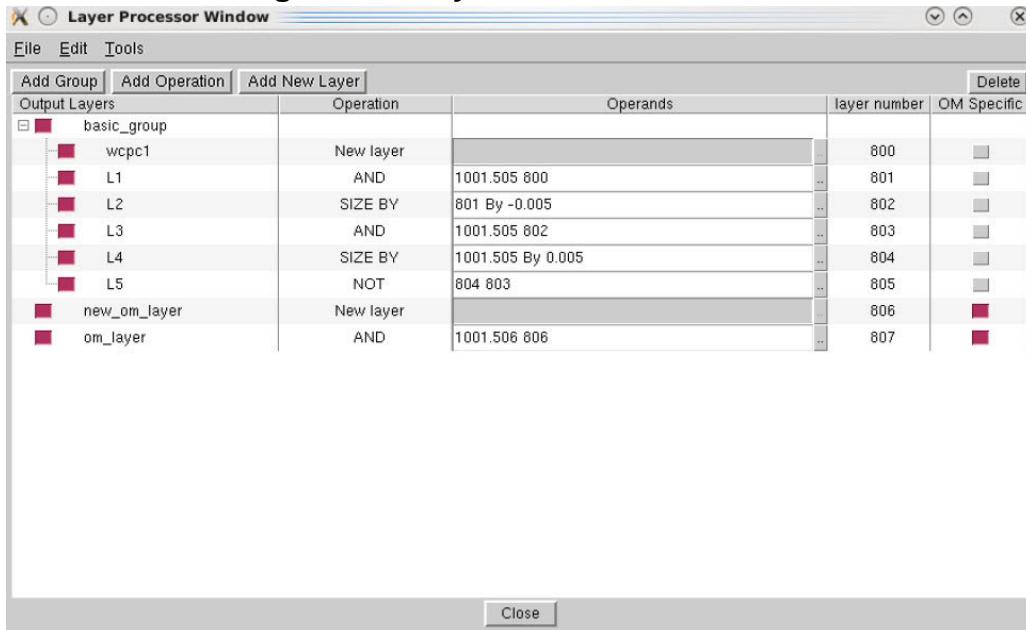
Fields	Description
Algorithm	<p>Specifies the algorithm. This can be one of the following:</p> <ul style="list-style-type: none"> • ave: Takes the average of measurements within the marker, across the window width. • min: Measures the minimum width. This is useful for finding pinches in hot spots. • max: Measures the maximum width. This is useful for finding bulges in hot spots. • contacts: Measures a two-dimensional hole (for contacts). • image: Gets the image only. • ellipse: Measure a two-dimensional elliptical area (similar to contacts). • contour: Gets the image fit contours to the field. • group: Specifies the field-of-view and designate the measurements inside to be extracted from that image. • undefined: Applies the Default Metrology Algorithm value. <p>You can scan the layout for the algorithm types for each marker by clicking the ScanMarker Algorithm and Type button.</p>
Magnification	Specifies the magnification setting.
Cell	Specifies the cell name that the marker is located in.

Layer Processor Window

To access: In the CMi window, select **Tools > Layer Processing (Booleans)**

The Layer Processor window enables you to define boolean operations on multiple layers (more than two) from the original layers in the layout or from derived intermediate layers from previous boolean operations. Prior to the **Assemble for Review** phase, this window will be empty. Following **Assemble for Review**, the input layer list will automatically be filled in. You can also load a previously saved session by clicking the **Load** button

Figure 3-8. Layer Processor Window



Objects

Table 3-10. Controls for the Layer Processor Window

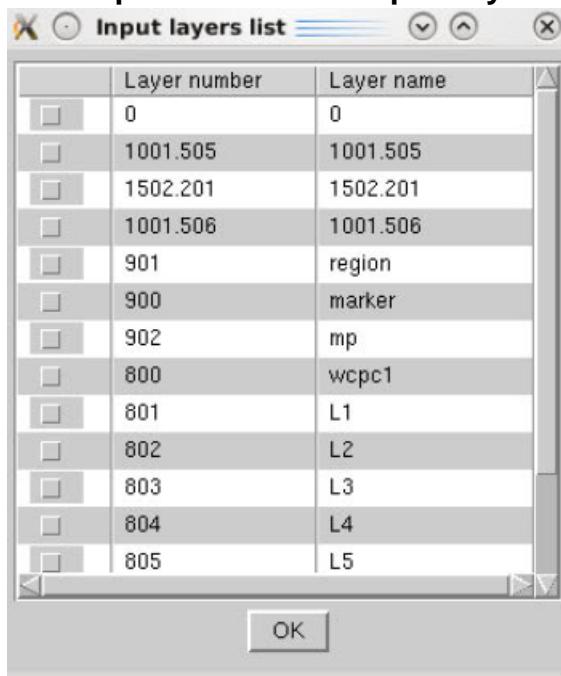
Item		Description
File		Specifies file-based operations.
	Save	Saves the session to a session file.
	Load	Loads a previously saved session file.
Edit		Specifies editing operations.
	Clear	Clears the Layer Processor window.
Tools		Specifies utilities available for the Layer Processor window.
	Add New Layer	Generates a new layer for the layout and draws a box with user-defined dimensions around each marker in the layout (see Figure 3-13 on page 129).

Table 3-10. Controls for the Layer Processor Window (cont.)

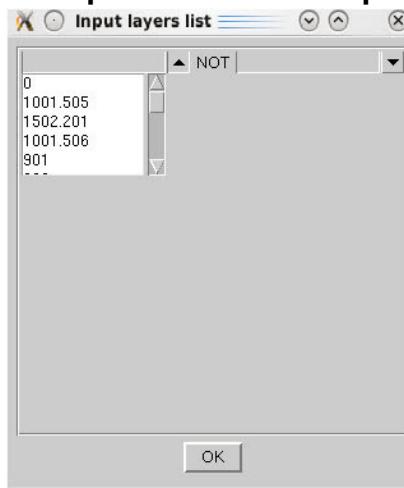
Item	Description
Operations Table Area	<p>Lists the defined operations. You can edit the operations defined for a layer by clicking on the row entry. You can modify the output layer or group name, operation, operands, and the output layer number.</p> <p>You can browse for specific layers by clicking the “...” button for each row. This invokes the Input Layers List dialog box.</p> <p>You can also specify that the boolean operations be processed on OM_align markers only by clicking the OM Specific radio button.</p>
Add Group	Generates a tree with a parent node (Group Name) and a child node (Operation). The group is a method to collect operations in a tree structure and does not affect the processing of the operations.
Add Operation	Adds a specified boolean operation.
Add New Layer	Generates a new layer for the layout and draws a box with user-defined dimensions around each marker in the layout (see Figure 3-13 on page 129)
Delete	Deletes the selected group or operation.
Close	Closes the window.

Usage Notes

The Input Layers List dialog box has different views according to the selected operation. If the selected operation is AND/OR, the Select Layers dialog box lists all the available layers for processing in a table format.

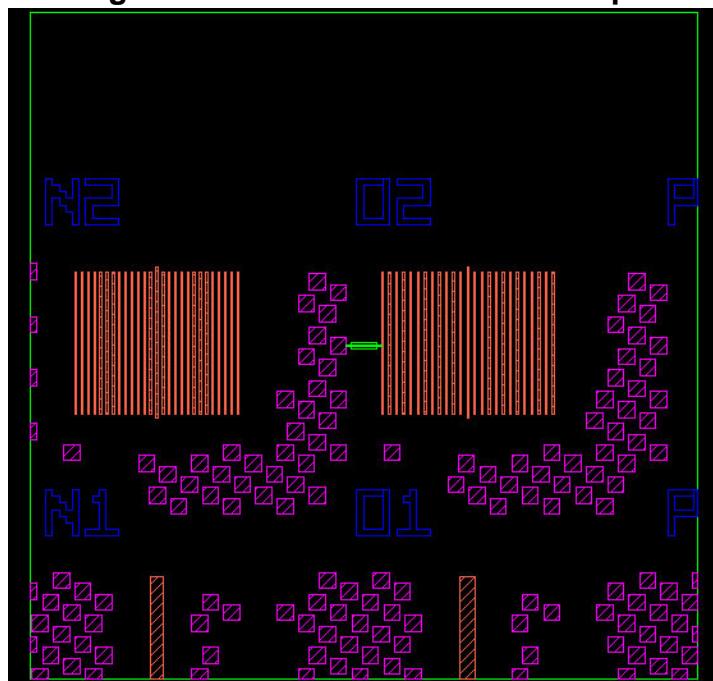
Figure 3-9. AND/OR Operations in the Input Layers List Dialog Box

If the selected operation is NOT and XOR, the Input Layers List dialog box provides two operands for input. The NOT and XOR operations apply to just two layers and keeps the order of layers (see [Figure 3-10](#)).

Figure 3-10. NOT and XOR Operations in the Input Layers List Dialog Box

The boolean operations are applied during the CMi output generation stage on small clips centered by a marker with the dimensions of the GDS clip size attribute (see [Figure 3-11](#) for an illustration). This saves processing run time as opposed to applying the booleans on the full chip.

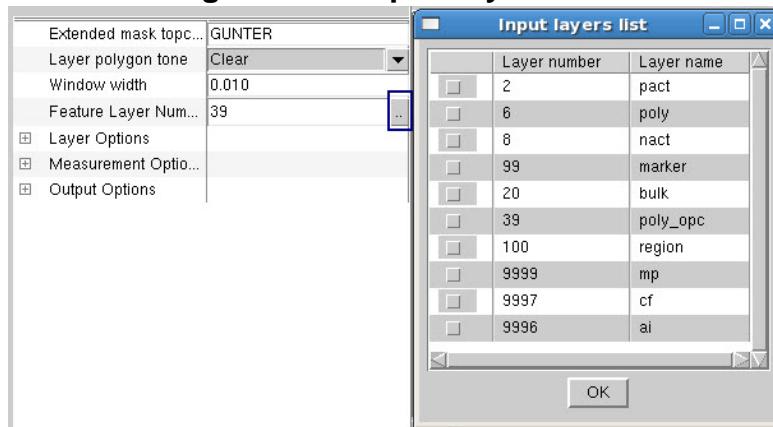
Figure 3-11. Generated Boolean Clips



The generated clips include all original layers specified as feature layers and all intermediate layers generated from the boolean processing.

You can select feature layers from both original layers or from the results of the boolean operations. Multiple input layers can be selected by clicking the “..” button in the “Feature layer number” field to invoke the Input Layers List dialog box. The tool then verifies your input and issue an error message if the layer numbers do not exist in the layout or not defined as an output from the boolean operations.

Figure 3-12. Input Layers List



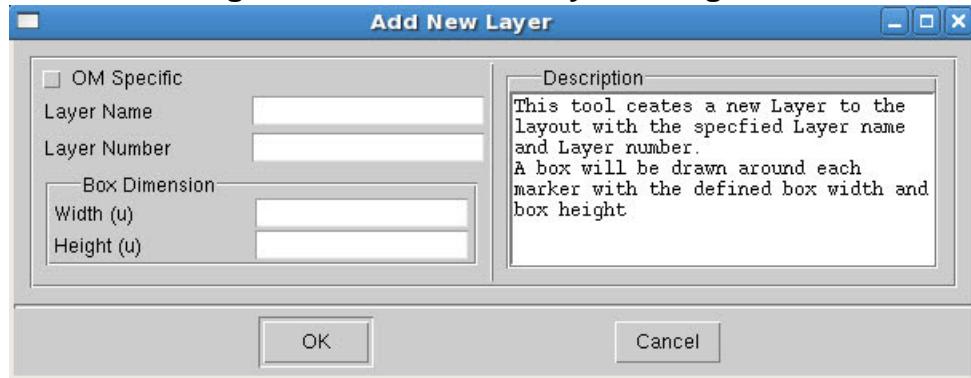
When you click the **Add New Layer** button or select **Tools > Add New Layer**, the Add New Layer dialog box appears. The Add New Layer dialog box generates a new layer for the layout and draws a box with user-defined dimensions around each marker in the layout. You can

define the new layer name and number, the box dimensions to be drawn around each marker, and whether the layer is OM-specific.

The Add New Layer dialog box allows you to generate two new layers one for regular marker algorithm and a specific new layer only for the OM_align algorithm marker.

The new added layer will be available for Boolean operation selection in the Layer Processor window. The defined layers are saved in a file generated for the Layer Processor window. More information on the process of defining boolean operations can be found in “[Defining Boolean Operations on Layers](#)” on page 60.

Figure 3-13. Add New Layer Dialog Box



The following is an example Boolean XML definition file.

```
<?xml version="1.0" encoding="UTF-8"?>
-<CMi_booleans_defintion>
-<Operations>
-<Operation>
<group>basic_group</group>
<operation_name>wcpcl</operation_name>
<operation>NEW</operation>
<input_layer1/>
<input_layer2/>
<output_layer>800</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>basic_group</group>
<operation_name>L1</operation_name>
<operation>AND</operation>
<input_layer1>1001.505</input_layer1>
<input_layer2>800</input_layer2>
<output_layer>801</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>basic_group</group>
<operation_name>L2</operation_name>
<operation>SIZE BY</operation>
<input_layer1>801</input_layer1>
<input_layer2/>
<size_value>-0.005</size_value>
<output_layer>802</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>basic_group</group>
<operation_name>L3</operation_name>
<operation>AND</operation>
<input_layer1>1001.505</input_layer1>
<input_layer2>802</input_layer2>
<output_layer>803</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>basic_group</group>
<operation_name>L4</operation_name>
<operation>SIZE BY</operation>
<input_layer1>1001.505</input_layer1>
<input_layer2/>
<size_value>0.005</size_value>
<output_layer>804</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>basic_group</group>
<operation_name>op_5</operation_name>
```

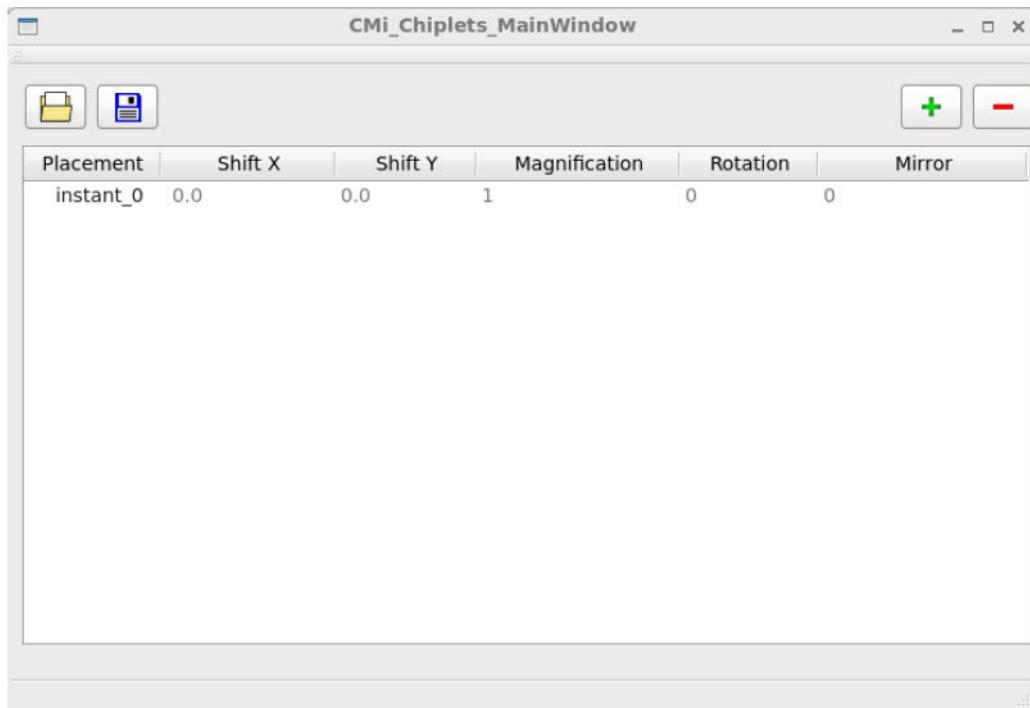
```
<operation>NOT</operation>
<input_layer1>804</input_layer1>
<input_layer2>803</input_layer2>
<output_layer>805</output_layer>
<om_operation>0</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>No_group</group>
<operation_name>new_om_layer</operation_name>
<operation>NEW</operation>
<input_layer1/>
<input_layer2/>
<output_layer>806</output_layer>
<om_operation>1</om_operation>
<enable>1</enable>
</Operation>
-<Operation>
<group>No_group</group>
<operation_name>om_layer</operation_name>
<operation>AND</operation>
<input_layer1>1001.506</input_layer1>
<input_layer2>806</input_layer2>
<output_layer>807</output_layer>
<om_operation>1</om_operation>
<enable>1</enable>
</Operation>
</Operations>
</CMi_booleans_defintion>
```

Chiplets Manager

To access: In the CMi window, select **Tools > Chiplets Manager**

The Chiplets Manager enables you to create a metrology recipe for sites defined in multiple chiplets from a layout, so sites within a chip can be found in different placements in the design with different transformations.

Figure 3-14. Chiplets Manager Window



Description

The CMi Chiplets Manager supports generation of metrology recipes (Hitachi and AMAT) for sites in different chiplet placements. This is done by specifying sites in the input design layout and specifying different placements of the design in CMi by defining the following:

- Placement in the X direction
- Placement in the Y direction
- Magnification
- Rotation
- Mirroring

The output metrology recipes include all the different placements of the input site as specified in the chiplets definition in an extended job deck. For example:

```
SLICE 1, 17
* GROUP 0,1
RETICLE
*
*
*DESIGN_RULE_TnM_BY_GENIE
OPTION MA, PA
MTITLE 1, GENE123
*
CHIP ID001, *MAIN
$ (1, simple1-oa-ss, SF=1.1, BX=0.1, BY=0.1, UX=27.00, UY=34.00, AG=180.0,
LY={8}, DT= {0})
ROWS 0.0/0.00
*CHIP ID01, *MAIN
*$ (1, simple1-oa-ss, SF=1.0, BX=0.1, BY=0.1, UX=27.00, UY=34.00, LY={8},
DT= {0})
*ROWS 15000.000/18900.000

CHIP ID002, *MAIN2
$ (1, testpat-tr-gd, SF=1.0, BX=0.98, BY=2.1, UX=97.2, UY=193.9, LY={0},
DT= {0})
ROWS 75000.000/78900.000

*
END
```

Extended job decks support the following commands:

- TC — Topcell name. If there is only one topcell in the OASIS file, this option is not required.
- SF — Scale factor. The scale that is applied to the input OASIS file. The default scale factor is 1.
- BX., BY, UX, UY — The bottom left X coordinate, bottom left Y coordinate, upper-right X coordinate, and upper-right Y coordinate of the pattern, respectively. The coordinates are supplied in microns, after the scale factor is applied.
- AG — Angle of rotation. This is supplied in 90-degree increments, counter-clockwise. The default value is 0.
- LY — Layers to be processed. The default behavior is that all layers in the file are processed.
- DT — Datatypes to be processed. The default behavior is that all datatypes are processed.
- ROWS — Defines the X/Y placement.

In the extended job deck, the name of the input design should follow the MEBES pattern convention of 9 letters followed by a period (.) then two characters.

Objects

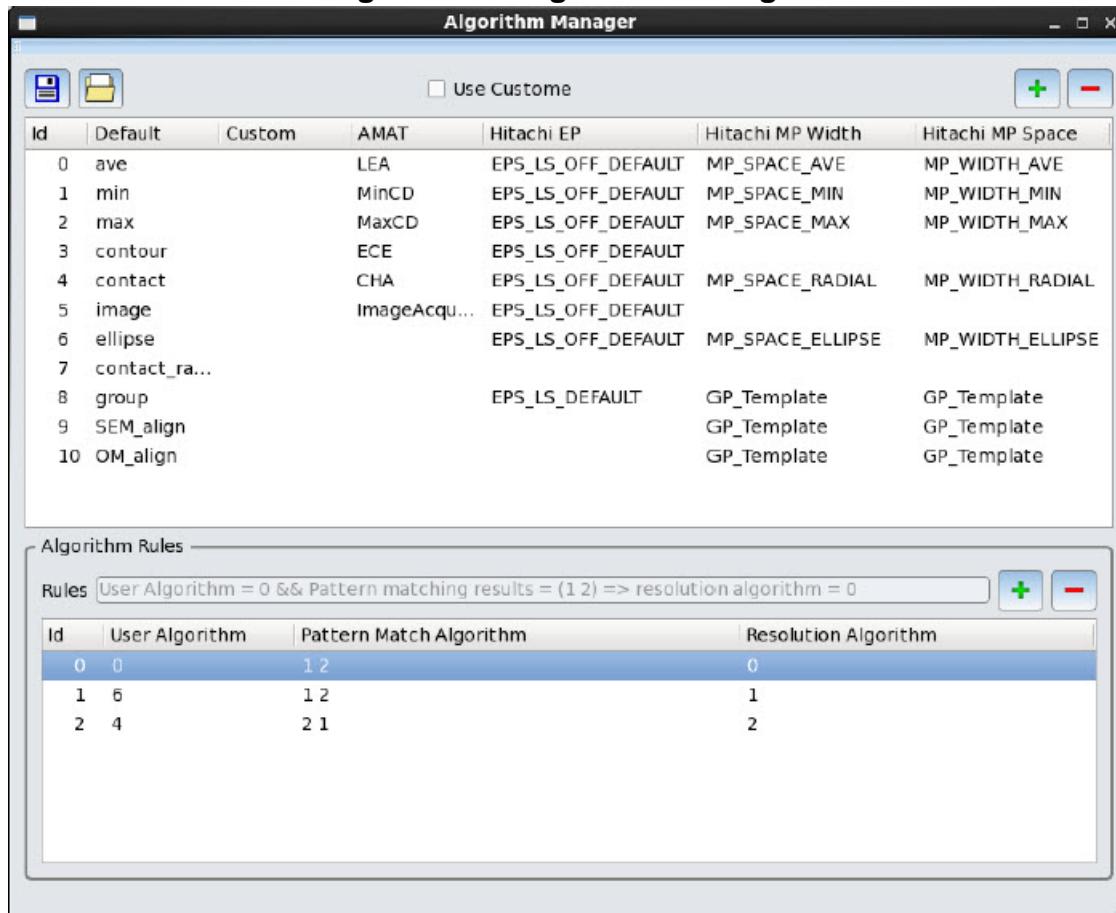
Objects	Description
Open File 	Opens an extended job deck.
Save 	Saves the current settings to an extended job deck..
Add 	Add a new entry to define a chiplet's transform settings.
Delete 	Delete a chiplet.
Placement	Indicates the chiplet placement location.
Shift X	Indicates a placement shift in the X-axis direction.
Shift Y	Indicates a placement shift in the Y-axis direction.
Magnification	Indicates the magnification factor.
Rotation	Indicates the rotation angle.
Mirror	Indicates if the design is mirrored. 0 means the design is not mirrored, 1 means the design is mirrored.

Algorithm Manager

To access: In the CMi window, select **Tools > Algorithms Manager**

The Algorithm Manager enables you to define custom names to the current default metrology algorithms used in a CMi flow. It also gives you the ability to add new custom metrology algorithm names and attach them to output AMAT and Hitachi algorithms.

Figure 3-15. Algorithm Manager



Description

The Algorithm Manager enables you to choose either the default metrology names or custom names corresponding to each default algorithm. You can add new custom algorithms and they are identified when loading a CSV file. If you use the new algorithm IDs in the pattern matching library, they are designated for the automatic algorithm identification flow.

Using the Algorithm Manager, you can edit the algorithm ID, Custom name, AMAT value from a selected set (LEA, MinCD, MaxCD, ECE, CHA, ImageAcquisition) and Hitachi template names.

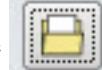
The Algorithm Manager session can be saved as an XML file by clicking the Save icon in the Algorithm Manager, or it can be part of the CMi session when saving a session. If there is an

algorithm session file in the CMi session, CMi automatically opens the Algorithm Manager and loads the algorithm session file.

Note

 CMi issues an error message if duplicate custom or ID values are already used.

Objects

Object	Description
Algorithm Table	The top portion of the Algorithm Manager lists the individual algorithms, including IDs, default algorithm used, custom names, and AMAT and Hitachi template information.
Save 	Saves the current settings on the Algorithm Manager to an XML session file.
Open File 	Opens an Algorithm Manager XML session file.
Use Custom	Directs CMi to use custom algorithm names; otherwise, the default names are used.
Add 	Add a new entry to the Algorithm Manager table. For each entry, double click a column to create the custom name corresponding to each of the default metrology algorithms, as well as selecting which AMAT algorithm and Hitachi template names are used.
Delete 	Delete an entry in the Algorithm Manager table.
Algorithm Rules	Allows you to define different algorithm combinations (user algorithms and algorithms found by pattern matching) and define the user-required choice for those combinations. The Rules list only accepts Algorithm ID numbers that are defined in the Algorithm Manager.
Rules	Displays the condition each row defines and the resolution algorithm that is used.
Add 	Add a new entry to the Algorithm Rules list. For each new entry, double-click on a column to enter a custom user algorithm number, pattern match algorithm, or resolution algorithm. The entry in the Rules field automatically updates to reflect your changes.
Delete 	Delete an entry in the Algorithm Rules list.

Allowed Characters in Cell Names

The **Extended Mask top cell** and **Chip top cell name** fields have restrictions of the allowed characters.

Figure 3-11 lists the allowed characters.

Table 3-11. Characters Allowed/Not Allowed in Cell Names

Allowed	Not Allowed
+	=
- (dash)	,
*	{
&	}
^	(
%)
#	"
\$ (dollar)	/
	\
_ (underscore)	
:	
.	
: (colon)	
<	
]	
>	
[
'	
`	
! (exclamation)	
? (question)	

Chapter 4

CMi Command Reference

Calibre Metrology Interface sessions can be configured through batch commands and Tcl functions.

The specific CMi functions that can be configured through this command line interface are:

- Local Alignment marker selection using pattern matching
- SEM image overlay alignment scoring
- Metrology measurement point automatic adjustment
 - Measurement point location adjustment
 - Metrology Algorithm identification using pattern matching
- Generate layout clips
 - Single layout for all clips
 - Separate clips per measurement point
- Generate Hitachi/AMAT metrology recipes

Executing Tcl Scripts for CMi	139
CMi Tcl Script Examples	140
CMi Tcl Commands.....	142

Executing Tcl Scripts for CMi

CMi executes custom Tcl scripts through the Calibre WORKbench layout viewer.

Prerequisites

- Refer to the *Calibre DESIGNrev Reference Manual* for a complete description of how Calibre layout viewers, including Calibre WORKbench, use Tcl-based scripts.

Procedure

Use one of the following methods to execute a Tcl script in a Calibre layout viewer:

Table 4-1. Executing Tcl Scripts in CMi

If you want to...	Do the following:
Execute a Tcl script in batch mode, by including the pathname for the script when invoking the layout viewer	<code>calibredrv my_script.tcl</code> <code>calibredrv my_script.tcl -gui</code>
Execute a Tcl script in interactive mode, by sourcing the script from the command line	<code>source my_script.tcl</code>

CMi Tcl Script Examples

The Calibre Metrology Interface uses Tcl-based batch scripts that enable you to customize various tool behaviors including sessions, output, clip generation, and automatic algorithm identification.

There are several different operations that can be modified by the CMi custom Tcl scripting.

SEM Image Scoring

The following command block example shows a CMi session being created and layout clips of 3.5 um being generated and output to CSV format.

```
set cmi_session [CMiSession create]
$cmi_session openSession -session test_session_working
set layout_config [$cmi_session getLayoutClipConfig]
$layout_config configure -clip_size 3.5
$cmi_session sites generateLayoutClips -configure $layout_config
$cmi_session sites writeResultsToCSV
```

Hitachi Sites Output

The following example shows how Hitachi format marker sites are configured and output.

```

set layout [layout create 32RX2A_ECE.g11]
set cmi_session [CMiSession create]
$cmi_session layout setLayout $layout
$cmi_session configure -tonality clear -max_cd 5 \
    -feature_layer "13 14 342 349 351 352 353 355 394" \
    -default_metrology ave -output_string out_cmi
$cmi_session sites readSamplePlan -type csv -file ./
Markers_with_target.csv
set hitachi_config [$cmi_session getHitachiOutputConfig] |
$hitachi_config configure -design_rotation Down \
    -truncate_sites_name 1 -keep_right_most_char 1 \
    -max_mag 300000 -min_mag 150000 -image_mag 200000 \
    -sem_align_mag 2000 -om_align_mag 104 -fov_ratio 0.66
$cmi_session sites writeResultsToHitachi -configure $hitachi_config
$cmi_session sites writeResultsToCSV

```

Clip Generation with Filters

The following example demonstrates layout clip generated with certain site types filtered.

```

set layout [layout create METROLOGY001_clip_1to1_LL.OAS]
set cmi_session [CMiSession create]
$cmi_session layout setLayout $layout
$cmi_session configure -output_string "gds_merge" -feature_layer 1
$cmi_session sites readSamplePlan -type csv -file "test_sepeed.csv"
set layout_config [$cmi_session getLayoutClipConfig]
$layout_config configure -clip_size 35 \
    -clip_format GDS -merge_clips 1 -gds_layers "200 300"
set sites_filter [$cmi_session getSitesFilter]
$sites_filter configure -orientation 90 -metrology_algorithm ave
$cmi_session sites generateLayoutClips -configure $layout_config -filter
$sites_filter

```

Sites Algorithm Identification

The following example shows the configuration of the CMi automatic algorithm identification engine from a command block.

```

set cmi_session [CMiSession create]
set layout [layout create testpattern.gds]
$cmi_session layout setLayout $layout
$cmi_session readJobspec -jobspec testpattern_jobspec.tcl
$cmi_session configure -output_string "batch_out" -feature_layer 0
$cmi_session sites readSamplePlan -type csv \
    -file "testpattern_markers2.csv"
set algorithm_identification [$cmi_session \
    getAlgorithmIdentificationConfig]
$algorithm_identification configure -pattern_library big_lib_dmacro.svrf \
    -pattern_clip_size 3 -algorithm_conflict keep
$cmi_session sites automaticAlgorithmIdentification -configure $alg_conf
$cmi_session sites writeResultsToCSV

```

CMi Tcl Commands

The Calibre Metrology Interface supports Tcl-based batch commands used to create custom scripts to modify its operations.

The following table lists commands supported for CMi batch scripting, organized into functional groups, including:

- **CMi Session Commands** — These commands are used for the creation and configuration of the CMi session.
- **Sites Commands** — These commands are for reading, processing, and writing of sites, depending on their configuration.
- **Layout Commands** — These commands configure the working layout and Boolean operations to apply for further site processing.
- **Filter Commands** — These commands define the filter object and filtering criteria.
- **Layout Clip Configuration** — These commands configure the layout clipping object.
- **Output Configuration** — These commands configure output objects.
- **Automatic Algorithm Identification** — These commands configure the automatic algorithm identification object capability of CMi.
- **Alignment Marker Selection** — These commands configure the alignment marker selection object.

Table 4-2. Summary of CMi Commands

Command	Description
CMi Session Commands	
CMiSession create	Creates a new CMi session.
\$cmi_session configure	Configures the CMi session with initial session parameters.
openSession	Loads a previously-defined session file into CMi.
readJobspec	Loads a previously-defined CMi job configuration file in XML format.
readWaferMapSession	Loads an XML wafer map session file in CMi.
getAlgorithmIdentificationConfig	Retrieves the configuration object used for algorithm identification.
getAmatOutputConfig	Retrieves the configuration object used to define AMAT sites output

Table 4-2. Summary of CMi Commands (cont.)

Command	Description
<code>getHitachiOutputConfig</code>	Retrieves the configuration object used define Hitachi sites output.
<code>getLayoutClipConfig</code>	Retrieves the configuration object used to define a clip layout.
<code>getSitesFilter</code>	Retrieves the configuration object for the sites filter.
<code>getLocateAlignmentConfig</code>	Retrieves the configuration object for the local alignment locator.
Sites Commands	
<code>sites adjustLocation</code>	Specifies adjustments for center, orientation, and feature types for sites.
<code>sites automaticAlgorithmIdentification</code>	Runs automatic algorithm identification for sites.
<code>\$cmi_session sites generateLayoutClips</code>	Generates CMi GDS or OASIS layout clips.
<code>sites locateAlignmentSites</code>	Locates alignment marker sites as defined by a Tcl configuration object.
<code>sites readSamplePlan</code>	Loads a sample plan from different sources by defining the file format and the file path.
<code>sites writeResultsToAMAT</code>	Outputs marker sites in AMAT format.
<code>sites writeResultsToCSV</code>	Outputs sites in CSV format.
<code>sites writeResultsToHitachi</code>	Outputs marker sites in Hitachi format.
Layout Commands	
<code>layout runBooleans</code>	Applies Boolean operations to a working layer for site processing.
<code>\$cmi_session layout setLayout</code>	Specifies the layout handle name.
Filter Commands	
<code>\$sites_filter configure</code>	Defines the sites filter object and the filtering criteria.
Layout Clip Configuration .	
<code>\$layout_clip_config configure</code>	Defines the layout clip object.
Output Configuration	
<code>\$amat_config configure</code>	Configures AMAT parameters for CMi output.

Table 4-2. Summary of CMi Commands (cont.)

Command	Description
\$hitachi_config configure	Configures Hitachi parameters for CMi output.
Automatic Algorithm Identification	
\$algorithm_identification configure	Defines the configuration of the automatic algorithm identification object.
Alignment Marker Selection	
\$sites_locator configure	Configures the alignment marker selection object.

CMiSession create	146
\$cmi_session configure	147
openSession.	148
readJobspec	149
readWaferMapSession	150
getAlgorithmIdentificationConfig	151
getAmatOutputConfig.	152
getHitachiOutputConfig	153
getLayoutClipConfig	154
getSitesFilter.	155
getLocateAlignmentConfig	156
sites adjustLocation	157
sites automaticAlgorithmIdentification	158
\$cmi_session sites generateLayoutClips	159
sites locateAlignmentSites	160
sites readSamplePlan.	161
sites writeResultsToAMAT	162
sites writeResultsToCSV	163
sites writeResultsToHitachi	164
layout runBooleans	165
\$cmi_session layout setLayout	166
\$layout_clip_config configure.	167
\$sites_filter configure	169
\$amat_config configure	170
\$hitachi_config configure	172
\$algorithm_identification configure.	174

\$sites_locator configure..... 176

CMiSession create

CMi Tcl Commands

Creates a new CMi session.

Usage

```
set cmi_session [CMiSession create]
```

Arguments

- *cmi_session*

Specifies a reference name for the CMi session.

Description

This command creates a new CMi session and is typically at the start of any Tcl scripting block that modifies CMi session parameters.

Examples

The following example shows CMiSession create initiating a new Tcl scripting block to specify parameters for a new CMi session.

```
set cmi_session [CMiSession create]
$cmi_session openSession -session test_session_working
set layout_config [$cmi_session getLayoutClipConfig]
$layout_config configure -clip_size 3.5
$cmi_session sites generateLayoutClips -configure $layout_config
$cmi_session sites writeResultsToCSV
```

\$cmi_session configure

CMi Tcl Commands

Configures the CMi session with initial session parameters.

Usage

```
$cmi_session configure [-tonality {clear | dark} -max_cd max_cd_um -feature_layer layers -  
default_metrology metrology_algorithm -output_string output_string]
```

Arguments

- *cmi_session*
A required argument that references the CMi session name created from the [CMiSession create](#) command.
- *-tonality {clear | dark}*
An optional argument that sets the default layer tonality used for the CMi session, either clear or dark.
- *-max_cd max_cd_um*
An optional argument that specifies the maximum CD in um.
- *-feature_layer layers*
An optional argument that specifies the feature layers to be measured by CMi.
- *-default_metrology metrology_algorithm*
An optional argument that specifies the metrology algorithm to be applied to all sites that are not specified. The algorithms are described in “[CSV Table Construction](#)” on page 49.
- *-output_string output_string*
An optional argument that specifies the name to be used for the results directory and the prefix of various output files.

Description

This command allows you to specify initial settings for your CMi session. All arguments of this object can be printed using a cget Tcl command

Examples

```
$cmi_session config -tonality clear -max_cd 5 -feature_layer "13 14 342  
349 351 352 353 355 394" -default_metrology ave -output_string out_cmi
```

openSession

CMi Tcl Commands

Loads a previously-defined session file into CMi.

Usage

\$cmi_session openSession -session session_path

Arguments

- **\$cmi_session**

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-session session_path**

A required argument that specifies the path name to the session file.

Description

This command instructs CMi to load the settings from a previously-defined session file. A CMi session file defines the paths of the inputs to CMi design layout, job specification template file, and site sources.

Examples

```
$cmi_session Opensession -session test_session_working
```

readJobspec

CMi Tcl Commands

Loads a previously defined CMi job configuration file in XML format.

Usage

```
$cmi_session readJobspec -jobspec file_path
```

Arguments

- **\$cmi_session**

A required argument that references the CMi session name created from the [CMiSession create](#) command.

- **-jobspec *file_path***

A required argument that specifies the path to a previously defined CMi job specification file in XML format.

Description

This command loads a previously-defined XML-based job configuration file into CMi. The settings in the file appear in the Job Configuration Area of CMi.

Examples

```
$cmi_session readJobspec -jobspec test_job_file
```

readWaferMapSession

CMi Tcl Commands

Loads an XML wafer map session file in CMi.

Usage

```
$cmi_session readWaferMapSession -file wafer_session.wmp
```

Arguments

- **\$cmi_session**

A required argument that references the CMi session name created from the [CMiSession create](#) command.

- **-file wafer_session.wmp**

A required argument that specifies the name of a wafer map session file (with the *.wmp* extension) with the settings from the Wafer Map Manager utility.

Description

Wafer data is often used to complete SEM recipes for machine vendor middle-ware to speed up and enhance the metrology process cycle. Wafer data parameters loaded from the CSV file can be viewed and edited through the Wafer Map Manager utility. Settings in this CMi subtool can be saved into a *.wmp* file and reloaded using this command.

Examples

```
$cmi_session readWaferMapSession -file wafer_test.wmp
```

getAlgorithmIdentificationConfig

CMi Tcl Commands

Retrieves the configuration object used for algorithm identification.

Usage

```
set algorithm_identification [$cmi_session getAlgorithmIdentificationConfig]
```

Arguments

- *algorithm_identification*

A required argument that specifies the name of the algorithm identification object.

- *cmi_session*

A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command retrieves the Tcl object that defines the parameters for the automatic algorithm identification engine. The Tcl object is defined by the [sites automaticAlgorithmIdentification](#) command.

Examples

```
set algorithm_identification [$cmi_session \
    getAlgorithmIdentificationConfig]
$cmi_session sites automaticAlgorithmIdentification \
    -configure $algorithm_identification
```

getAmatOutputConfig

CMi Tcl Commands

Retrieves the configuration object for the AMAT sites output.

Usage

```
set amat_config [$cmi_session getAmatOutputConfig]
```

Arguments

- ***amat_config***
A required argument that specifies the name of the AMAT output configuration object.
- ***cmi_session***
A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command retrieves the Tcl object containing parameters for AMAT sites output configuration, created by the [\\$amat_config configure](#) command.

Examples

```
set amat_config [$amat_config getAmatOutputConfig] |  
$amat configure -dmaster_mask_name -second_mask_name -Exposures 1 \  
-image content -device_type 13 14 342 349 351
```

getHitachiOutputConfig

CMi Tcl Commands

Retrieves the configuration object for the Hitachi sites output.

Usage

```
set hitachi_config [$cmi_session getHitachiOutputConfig]
```

Arguments

- ***hitachi_config***

A required argument that specifies the name of the Hitachi output configuration object.

- ***cmi_session***

A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command references the Tcl object containing parameters for Hitachi sites output configuration (created by the [\\$hitachi_config configure](#) command).

Examples

```
set hitachi_config [$cmi_session getHitachiOutputConfig] |  
$hitachi_config configure -design_rotation Down \  
-truncate_sites_name 1 -keep_right_most_char 1 \  
-max_mag 300000 -min_mag 150000 -image_mag 200000 \  
-sem_align_mag 2000 -om_align_mag 104 -fov_ratio 0.66  
$cmi_session sites writeResultsToHitachi -configure $hitachi_config  
$cmi_session sites writeResultsToCSV
```

getLayoutClipConfig

CMi Tcl Commands

Retrieves the configuration object used to define a clip layout.

Usage

```
set layout_clip_config [$cmi_session getLayoutClipConfig]
```

Arguments

- *layout_clip_config*

A required argument that specifies the name of the layout clip configuration object.

- *cmi_session*

A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command references a Tcl object containing parameters for clip layout configuration. The configuration object is defined by the [\\$layout_clip_config configure](#) command.

Examples

```
set layout_clip_config [$cmi_session getLayoutClipConfig]
$layout_clip_config configure -clip_size 3.5
$cmi_session sites generateLayoutClips -configure $layout_config
$cmi_session sites writeResultsToCSV
```

getSitesFilter

CMi Tcl Commands

Retrieves the configuration object used for the sites filter configuration.

Usage

```
set sites_filter [$cmi_session getSitesFilter]
```

Arguments

- *sites_filter*

A required argument that specifies the name of the sites filter object.

- *cmi_session*

A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command retrieves the Tcl configuration object that defines the parameters for filtering sites. The configuration object is created by the [\\$sites_filter configure](#) command.

Examples

```
set sites_filter  [$cmi_session getSitesFilter]
$sites_filter configure -orientation 90 -metrology_algorithm ave
$cmi_session sites generateLayoutClips -configure $layout_config -filter
$sites_filter
```

getLocateAlignmentConfig

CMi Tcl Commands

Retrieves the configuration object used for the local alignment locator.

Usage

```
set sites_locator [$cmi_session getLocateAlignmentConfig]
```

Arguments

- **sites_locator**
A required argument that specifies the name of the sites locator object.
- **cmi_session**
A required argument that references the CMi session name created from the [CMiSession create](#) command.

Description

This command retrieves the Tcl object that defines the parameters for the local alignment locator.

Examples

```
sites_locator [$cmi_session getLocateAlignmentConfig]
```

sites adjustLocation

[CMi Tcl Commands](#)

Specifies adjustments for center, orientation, and feature types for sites.

Usage

```
$cmi_session sites adjustLocation [-filter $sites_filter]
```

Arguments

- ***cmi_session***

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-filter *\$sites_filter***

An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command is used to adjust the site center, orientation, and feature type for all types except for those specified by the -filter object.

sites automaticAlgorithmIdentification

CMi Tcl Commands

Runs automatic algorithm identification for sites.

Usage

```
$cmi_session sites automaticAlgorithmIdentification -configure $algorithm_identification  
[-filter $sites_filter]
```

Arguments

- *cmi_session*
A required argument that references CMi session name created from the [CMiSession create](#) command.
- **-configure \$algorithm_identification**
A required argument that runs the automatic algorithm identification flow (see “[Using Automatic Identification in CMi](#)” on page 78).
- **-filter \$sites_filter**
An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command runs automatic algorithm identification, configured by a Tcl object created by the [\\$algorithm_identification configure](#) command, for all sites except those defined in the **-filter** object.

Examples

```
set algorithm_identification [$cmi_session \  
    getAlgorithmIdentificationConfig]  
$cmi_session sites automaticAlgorithmIdentification \  
    -configure $algorithm_identification
```

\$cmi_session sites generateLayoutClips

CMi Tcl Commands

Generates CMi GDS or OASIS layout clips.

Usage

```
$cmi_session sites generateLayoutClips -configure $layout_clip_config [-filter $sites_filter]
```

Arguments

- ***cmi_session***

A required argument that references CMi session name created from the [CMiSession create](#) command.

- ***-configure \$layout_clip_config***

A required argument that generates the layout clips for all sites, except for those specified using the *-filter* option.

- ***-filter \$sites_filter***

An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command generates GDS or OASIS layout clips for all sites except those defined in the filter object with the layout clip configuration. The parameters for the layout clip are specified by the [\\$layout_clip_config configure](#) command.

Examples

```
$cmi_job sites generateLayoutClips -configure $layout_config -filter  
$sites_filter
```

sites locateAlignmentSites

CMi Tcl Commands

Locates alignment marker sites as defined by a Tcl configuration object.

Usage

```
$cmi_session sites locateAlignmentSites -configure $sites_locator [-filter $sites_filter]
```

Arguments

- **\$cmi_session**

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-configure \$sites_locator**

A required argument that specifies the alignment markers to be located using a Tcl configuration object created by [\\$sites_locator configure](#).

- **-filter \$sites_filter**

An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command locates alignment marker sites as defined by the [\\$sites_locator configure](#) command. The Calibre Metrology Interface uses Calibre's pattern matching capability to find the local alignment marker structures in the design, allowing you to prepare a pattern library that includes all the structures. See “[Automatically Selecting Alignment Markers Using Pattern Matching](#)” on page 90 for further information.

Examples

```
$cmi_session sites locateAlignmentSites -configure $sites_locator
```

sites readSamplePlan

CMi Tcl Commands

Loads a sample plan from different sources by defining the file format and the file path.

Usage

```
$cmi_session sites readSamplePlan -type {csv | ss | gg | rdb | xylist} -file file_path
```

Arguments

- ***cmi_session***

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-type {csv | ss | gg | rdb | xylist}**

A required argument that specifies the file format type: csv (CSV), ss (super spreadsheet), gg (gauge), rdb (RDB), or xylist (list of X Y marker coordinates).

- **-file *file_path***

A required argument that specifies the path name to the sample plan file.

Description

This command loads a sample plan file into CMi. Sample plan files contain marker information that can be loaded into a CMi session. Several different file formats are supported, including CSV, super spreadsheet, gauge, RDB, and an X Y list.

Examples

```
$cmi_job sites readSamplePlan -type csv -file test_sepeed.csv
```

sites writeResultsToAMAT

[CMi Tcl Commands](#)

Output marker sites in AMAT format.

Usage

```
$cmi_session sites writeResultsToAMAT -amat_config $amat_config [-filter $sites_filter]
```

Arguments

- ***cmi_session***

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-amat_config \$amat_config**

A required argument that specifies that the configuration object for AMAT format output.

- **-filter \$sites_filter**

An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command is used to output marker site information into AMAT format (based on the configuration specified by the [\\$amat_config configure](#) command), except for those sites excluded by the -filter option.

Examples

```
$cmi_job sites writeResultsToAMAT -configure $amat_config
```

sites writeResultsToCSV

CMi Tcl Commands

Outputs sites in CSV format.

Usage

```
$cmi_session sites writeResultsToCSV [-filter $sites_filter]
```

Arguments

- *cmi_session*

A required argument that references CMi session name created from the [CMiSession create](#) command.

- *-filter \$sites_filter*

An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command is used to output marker site information into CSV format, except for those sites excluded by the *-filter* option.

Examples

```
$cmi_job sites writeResultsToCSV
```

sites writeResultsToHitachi

[CMi Tcl Commands](#)

Output marker sites in Hitachi format.

Usage

```
$cmi_session sites writeResultsToHitachi -hitachi_config $hitachi_config [-filter  
$sites_filter]
```

Arguments

- ***cmi_session***
A required argument that references CMi session name created from the [CMiSession create](#) command.
- **-hitachi_config *\$hitachi_config***
A required argument that specifies that the configuration object for Hitachi format output.
- **-filter *\$sites_filter***
An optional argument that filters sites using a Tcl object with parameters defined by the [\\$sites_filter configure](#) command.

Description

This command is used to output marker site information into Hitachi format (based on the configuration specified by [\\$hitachi_config configure](#)), except for those sites excluded by the -filter option.

Examples

```
$cmi_job sites writeResultsToHitachi -configure $hitachi_config
```

layout runBooleans

Applies Boolean operations to a working layer for site processing.

Usage

```
$cmi_session layout runBooleans -boolean_file file_path
```

Arguments

- *cmi_session*

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **-boolean_file** *file_path*

A required argument that specifies the pathname to the file that defines the Boolean operations to be applied to the layout.

Description

This command is used in conjunction with [\\$cmi_session layout setLayout](#) to set a working layout and apply Boolean operations for further site processing.

Examples

```
$cmi_session layout setLayout $layout_handle
$cmi_session layout runBooleans -boolean_file /mnt/path/boolfile
```

\$cmi_session layout setLayout

CMi Tcl Commands

Specifies the layout handle name.

Usage

\$cmi_session layout setLayout \$layout_handle

Arguments

- **cmi_session**

A required argument that references CMi session name created from the [CMiSession create](#) command.

- **layout_handle**

Specifies the layout handle object name.

Description

This command is used in conjunction with the [layout runBooleans](#) command to specify a working layout and apply Boolean operations.

Examples

```
$cmi_session layout setLayout $layout_handle
$cmi_session layout runBooleans -boolean_file /mnt/path/boolfile
```

\$layout_clip_config configure

CMi Tcl Commands

Defines the layout clip object.

Usage

```
$layout_clip_config configure [-clip_format {GDS | OASIS} -clip_size um -om_clip_size um -merge_clips {0 | 1} -gds_layers layers_list -visible_under_layers layers_list -mt {0 | 1} -threads number_of_threads]
```

Arguments

- *layout_clip_config*
A required argument that specifies the layout clip configuration object name.
- *-clip_format {GDS | OASISOASIS}*
An optional argument that specifies the clip layout format, GDS or OASIS.
- *-clip_size um*
An optional argument that specifies the layout clip size in um.
- *-om_clip_size um*
An optional argument that specifies a Global Alignment (GA) Optical Microscope (OM) clip from a target and merge this clip with the other target clips size in um. This generates a GDS square clip with a size equal to the OM Clip Size value around the center of the marker with the “OM” algorithm. The generated GDS clip is merged with other SEM GDS clips from targets, and is in the final merged clips layout.
- *-merge_clips {0 | 1}*
An optional argument that, if set to 1, creates a single merged layout clip file. Setting this argument to 0 creates individual layout clip files for each site.
- *-gds_layers layers_list*
An optional argument that specifies the list of GDS layers for the clip.
- *-visible_under_layers layers_list*
An optional argument that specifies a list of additional layers from the layout that are to be included in the clip.
- *-mt {0 | 1}*
An optional argument that, if set to 1, specifies that the layout clip for a multi-threaded run. A setting of 0 indicates that the clip is for a single run.
- *-threads number_of_threads*
An optional argument that defines the number of threads needed for a multi-threaded run.

Description

This command configures the parameters for a layout clip. These parameters will then be used as a layout clip object for other layout clip commands. The configured Tcl object from this command is referenced by the [getLayoutClipConfig](#) command. This command is equivalent to the **Output Options > Layout Clip Output** options in the Job Configuration Area.

All arguments of this object can be printed using a cget Tcl command

\$sites_filter configure

CMi Tcl Commands

Defines the sites filter object and the filtering criteria.

Usage

```
$sites_filter configure [-name site_name -metrology_algorithm algorithm -orientation {0 | 90}]
```

Arguments

- *sites_filter*
A required argument that specifies the site filter object name.
- *-name site_name*
An optional argument that specifies the filter object name.
- *-metrology_algorithm algorithm*
An optional argument that specifies the algorithm used by the marker sites in the filter object. The algorithms are described in “[CSV Table Construction](#)” on page 49.
- *-orientation {0 | 90}*
An optional argument that specifies the marker site orientation, 0 or 90 degrees.

Description

This command is used to define the marker site parameters to be later used as a filter object by other CMi Tcl commands.

All arguments of this object can be printed using a cget Tcl command.

Examples

```
$sites_filter configure -orientation 90 -metrology_algorithm ave
```

\$amat_config configure

CMi Tcl Commands

Configures AMAT parameters for CMi output.

Usage

```
$amat_config configure [-master_mask_name -second_mask_name -Exposures {1 | 1 2} -  
image_content -device_type -extent {llx lly urx ury}]
```

Arguments

- **amat_config**
An required argument that specifies the AMAT configuration object name.
- **-master_mask_name**
An optional argument that specifies the master mask name to be transferred to the AMAT output file.
- **-second_mask_name**
An optional argument enabling the use of a second mask name. The second mask name, when non-blank, designates “Poly-on-active” mode. This causes an additional mask name to be defined in the AMAT output file using additional clip layers. An empty value for a second mask name designates “Functional Usage” mode.
- **-Exposures {1 | 1 2}**
An optional argument that specifies a “Double Exposure” mode. The list of values are applied sequentially to the multiple feature layers. Typically, this is a sequence of “1” and “2” with the same number values as feature layers.
- **-image_content**
An optional argument that the image content is to be transferred to the AMAT output file. CMi does not verify this value.
- **-device_type**
An optional argument that specifies that the device type is to be transferred to the AMAT output file. CMi does not verify this value.
- **-extent {llx lly urx ury}**
An optional argument that specifies the clip extent to be transferred to the AMAT output file, defined by the four corner coordinates, lower-left x, lower left, y, upper-x, and upper-right y.

Description

This command defines the parameters of an AMAT output object that will be used by other CMi commands. The parameters are referenced by the [getAmatOutputConfig](#) and [sites writeResultsToAMAT](#) commands. This command is equivalent to setting the **Output Options > AMAT Output** options in the Job Configuration Area of CMI.

All arguments of this object can be printed using a cget Tcl command.

\$hitachi_config configure

CMi Tcl Commands

Configures Hitachi parameters for CMi output.

Usage

```
$hitachi_config configure [-design_rotation {DOWN | UP | LEFT | RIGHT} -exposure_x um -  
exposure_y um -truncate_sites_name {0 | 1} -keep_right_most_char {0 | 1} -max_mag um -  
min_mag um -image_mag um -mroi_mag um -sem_align_mag um -om_align_mag {104 |  
210} -fov_ratio um]
```

Arguments

- ***hitachi_config***
A required argument that specifies the Hitachi configuration object name.
- **-design_rotation {DOWN | UP | LEFT | RIGHT}**
An optional argument that specifies the design orientation (up, right, bottom or left).
- **-exposure_x um -exposure_y -exposure_y**
An optional argument that displays the exposure field size for the X and Y axis, in um.
- **-truncate_sites_name {0 | 1}**
An optional argument that activates or deactivates truncation of site names. The Calibre Metrology Interface, by default, truncates target names to a 10-character limit.
- **-keep_right_most_char {0 | 1}**
An optional argument that, if -truncate_sites_name and this keyword are set to 1, preserves the right-most two characters in the target name.
- **-max_mag um**
An optional argument that specifies the maximum magnification value when the feature size is smaller than a ratio of the Field of View (FOV).
- **-min_mag um**
An optional argument that specifies a minimum magnification value when the feature size is greater than or equal to ratio of the FOV.
- **-image_mag um**
An optional argument that specifies the image capture magnification value in cases where an Image algorithm is used.
- **-mroi_mag um**
An optional argument that specifies the magnification value of the Multiple Region of Interest (MROI) EP marker. This is required if MROI markers are used.

- **-sem_align_mag *um***
An optional argument that specifies the magnification value if the SEM_align algorithm is used.
- **-om_align_mag {104 | 210}**
An optional argument that specifies the magnification value if the OM_align algorithm is used.
- **-fov_ratio *um***
An optional argument that specifies the ratio of the FOV that is compared with the feature size. This ratio will be used to determine whether minimum or maximum magnification settings are used. The default value is $2/3 = 0.66$.

Description

This command configures CMi output for Hitachi-format sites. This is a Tcl configuration object referenced the [getHitachiOutputConfig](#) and [sites writeResultsToHitachi](#) commands. This command is equivalent to setting the **Output Options > Hitachi Output** options in the Job Configuration Area of CMi.

All arguments of this object can be printed using a cget Tcl command.

Examples

```
set hitachi_config [$cmi_session getHitachiOutputConfig] |
$hitachi_config configure -design_rotation Down \
    -truncate_sites_name 1 -keep_right_most_char 1 \
    -max_mag 300000 -min_mag 150000 -image_mag 200000 \
    -sem_align_mag 2000 -om_align_mag 104 -fov_ratio 0.66
```

\$algorithm_identification configure

CMi Tcl Commands

Configures the automatic algorithm identification object.

Usage

```
$algorithm_identification configure -pattern_library file_path [-pattern_clip_size um -  
algorithm_conflict {override | keep}]
```

Arguments

- ***algorithm_identification***

A required argument that specifies the algorithm identification configuration object name.

- **-pattern_library *file_path***

A required argument that specifies the compiled encrypted SVRF version of the DMACRO pattern library file. The Algorithm Mapping file should exist in the same path of the selected pattern matching library. If it does not exist, the default values as shown in the following figure are used:

Figure 4-1. Defaults for Map File

0	ave
1	min
2	max
3	contour
4	contact
5	image
6	ellipse

- **-pattern_clip_size *um***

An optional argument that specifies the size of the clip generated around each measurement point. The merged clips layout is used as an input to the pattern matching run. The default value is 3 um.

- **-algorithm_conflict {override | keep}**

An optional argument that specifies the conflict resolution mechanism if the pattern matching result contradicts the manually-set algorithm. One of the following can be specified:

- override — The automatic algorithm identification overrides the user-specified algorithm for measurements.
- keep — Keeps the user-specified algorithm and run the pattern matching on measurement points with the NULL algorithm value only.

Description

This command configures the parameters for automatic algorithm identification that are used as an object by other CMi commands. The Tcl object created from this command is referenced by the [sites automaticAlgorithmIdentification](#) command. This command is equivalent to several options under the **Site Automatic Adjustment > Metrology Algorithm** section of the Job Configuration Area.

All arguments of this object can be printed using a cget Tcl command.

Examples

```
set algorithm_identification [$cmi_session \
    getAlgorithmIdentificationConfig]
$algorithm_identification configure -pattern_library big_lib_dmacro.svrf\
    -pattern_clip_size 3 -algorithm_conflict keep
```

\$sites_locator configure

CMi Tcl Commands

Configures the alignment marker selection object.

Usage

```
$sites_locator configure -pattern_library file_path [-exclusion_distance um -search_area_x um -search_area_y um]
```

Arguments

- ***sites_locator***
A required argument that specifies the site locator configuration object name.
- **-pattern_library *file_path***
A required argument that specifies the pathname to the local alignment structure pattern encrypted library (DMACRO).
- **-exclusion_distance *um***
An optional argument that specifies the exclusion distance in um.
- **-search_area_x *um* -search_area_y *um***
An optional argument pair that specifies the marker search area in X and Y directions, respectively, in um.

Description

This command defines the parameters used for alignment markers that will be used as an object by other CMi commands. The Tcl object created by this command is referenced by the [\\$cmi_session sites generateLayoutClips](#) command. This command is equivalent to the functionality provided by the **Alignment Marker Options** dialog box in the Job Configuration Area.

All arguments of this object can be printed using a cget Tcl command.

Appendix A

Calibre Metrology Interface (CMi) Error Messages

The Calibre Metrology Interface can issue error and warning messages for conditions it encounters during operations.

Error Messages	177
Warning Messages	178

Error Messages

The following table lists example error messages that can be issued by CMi.

Table A-1. CMi Error Messages

Message
“Generating PNG Images (WW) Could not open RGB file “/usr/X11R6/lib/X11/rgb.txt”; will use built-in copy. _XSERVTransmkdir: Owner of /tmp/.X11-unix should be set to root error opening security policy file ../xserver/SecurityPolicy Could not init font path element /usr/X11R6/lib/X11/fonts/misc, removing from list!” This message indicates an error while generating the MHT Review Document. The user does not have permission to write the /tmp directories indicated. This can occur if they have been created by another user. They should generally be owned by “root,” with permission for all to write in them. The solution is to request the system administrator to fix the ownership and permissions on these directories and files.
“error opening security policy file ../xserver/SecurityPolicy” This message indicates that an abnormal termination occurred for an xserver session. This can result from the interruption of a CMi job during the creation of a MHT Review Document. A solution can be to delete the file /tmp/.X111-lock, and re-start CMi. The user should take care that there is not another xserver session number 111, which is not related to a running CMi job.
“ERROR: Setup layer type ‘hidden’ not allowed. Only ‘opc’ or ‘visible’”
“ERROR: Setup file = mask_setup does not exist.”
“ERROR: No setup file specified.”
“ERROR: No measure layer number specified.”
“ERROR: Mask file name specified does not exist.”

Table A-1. CMi Error Messages (cont.)

Message
“ERROR: Variable \$var not defined.”
“ERROR: Unsupported group type = other.”
“ERROR: Image layer’34’ does not exist. Image not output.”
“ERROR: Setup file mask_setup does not exist. Image not output.”
“ERROR: Image layer 34 does not exist. Image not output.”
“ERROR: Unsupported image type: bmp.”
“ERROR: Values loaded not validated. Abort”
“ERROR: Measure Layer: (34) does not exist in layout. Abort”
“ERROR: No measurements found.”
“ERROR: No feature layer number specified. Fracture magnification value should be > 0.”

Warning Messages

The following table lists example warning messages that can be issued by CMi.

Table A-2. CMi Warning Messages

Messages
“WARNING: No JobSpec File specified.”
“WARNING: wavelength value specified in optical model != 0.193 um.”
“WARNING: Resist model does not exist.”
“WARNING: JobSpec file does not exist: jobspec1.tcl.”
“WARNING: capture_width = 3 not supported by AIMS (only 20 um).”
“WARNING: image_type=bmp not supported by AIMS (only tiff).”
“WARNING: Variable feature_type not defined. Default value (width) will be used”
“WARNING: with flatten_flag=FALSE, object=marker1 with ‘cell’ context is ignored.”
“WARNING: object=marker1 specified in MDP cell pattern1 was rendered in cell markers_pattern1.”
“WARNING: unsupported property context=cell group_type=marker object=marker1.”
“WARNING: specified cell_name=cell1 does not exist. Ignoring object=marker1”
“WARNING: specified cell_name=cell1 does not exist. Ignoring object=marker1”
“WARNING: Extended Mask topcell not found: top_cell. Check topcell name.”

Table A-2. CMi Warning Messages (cont.)

Messages
“WARNING: Layer file layers.layerprop does not exist.”
“WARNING: file=in.csv does not exist. Ignored.”
“WARNING: Unsupported file type xml.”
“WARNING: duplicate named (position_x) found in csv, renamed to marker1”
“WARNING: Rejected the polygon because it is not a box: 0 0 01 10”
“WARNING: Rejected the polygon because it has zero width: 3942 436 3942 480”
“WARNING: Rejected the polygon because it is not a rectangle: 3942 436 3942 480”
“WARNING: Rejected the wire because it is has >2 points: 0 0 01 10”
“WARNING: multiple texts for shape found, using ‘marker1’ x1=3 y1=8”
“WARNING: shape name not found for shape at x1=3 y1=8”
“WARNING: Searching for features with orientation 23 is not supported.”
“WARNING: Validation For marker with angle 23 is not supported.”
“WARNING: marker1 is too far from the center of region1 for alignment image (3.6 4.2 <= 4)”
“WARNING: marker1 is too far from the center of region for measurement (7.6 82 <= 8)”
“WARNING: Hitachi format version 02.00 requires merged GDS clips with size 30 um.”
“WARNING: Can not generate MHT review document, merged GDS or OASIS clips required.”
“WARNING: Please set CMI_BROWSER_PATH by supported browsers.”

Index

— Symbols —

\$algorithm_identification configure, 174
\$amat_config configure, 170
\$hitachi_config configure, 172
\$layout_clip_config config, 167
\$sites_filter configure, 169
\$sites_locator configure, 176

— A —

Algorithms, 51
Alignment images, 16
Alignment markers
 automatic selection, 90, 110
Allowed characters in cell names, 137
AMAT CD-SEM output, 117
Assembly, 24
 chip layout over job deck, 28
Assembly options, 107
Automatic metrology algorithm identification, 70

— B —

Boolean operations on layers, 60
Browsing failed markers, 44

— C —

Chiplets, 96
CMi
 workflow, 22
CMI_BROWSER_PATH, 105
cmi_session config, 147
CMiSession create, 146
Contact pattern ratios, 91
CSV output, 119
CSV table, 49

— D —

DFM Pattern Capture, 73
Display Preferences options, 120
Draw markers, 23

— E —

Error messages, 177

— F —

Four corner construction, 62
Fracture specification, 108

— G —

GDS clip output, 119
getAlgorithmIdentificationConfig, 151
getAmatOutputConfig, 152
getHitachiOutputConfig, 153
getLocateAlignmentConfig, 156
getSitesFilter, 155

— H —

Hitachi output, 117

— I —

Import markers, 36

— K —

Kind, 48

— L —

Layer options, 109
Layer polygon tone, 29
Layer Processor window, 125
layout runBooleans, 165
layout setLayout, 166

— M —

Marker magnification, 40
Markers
 adjusting marker size, 34
 browsing failed markers, 44
 deleting, 38
 using markers in the chip layout, 32
Measurement locations, 58
Measurements
 creating multiple per capture, 53
 debugging and optimizing, 55

grouping in a region, 54
specifying target values, 56
specifying tolerances, 56
MP templates, 31

— N —

New Marker dialog box, 24

— O —

openSession, 148
Output options, 116

— P —

Pattern library, 73, 82
Pattern Library Manager, 75

— R —

readJobspec, 149
readWaferNaoSession, 150

— S —

Search, 58
SEM images
 overlaid design clips, 68
Sessions, 25
Site Automatic Adjustment options, 111
Site Size Preferences options, 115, 120
sites adjustLocation, 157
sites automaticAlgorithmIdentification, 158
sites generateLayoutClips, 159
sites locateAlignmentSites, 160
sites readSamplePlan, 161
sites writeResultsToAMAT, 162
sites writeResultsToCSV, 163
sites writeResultsToHitachi, 164
Specifying target values, 56
Specifying tolerances, 56

— T —

Target values, 56
Tolerances, 56
Top cell, 137

— W —

Wafer map, 63
Warning messages, 177

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.

