

Milkyway™ Environment Variables and Attributes

Version O-2018.06, June 2018

SYNOPSYS®

Copyright Notice and Proprietary Information

© 2018 Synopsys, Inc. All rights reserved. This software and documentation contain confidential and proprietary information that is the property of Synopsys, Inc. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Synopsys, Inc., or as expressly provided by the license agreement.

Destination Control Statement

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

Disclaimer

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

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at <http://www.synopsys.com/Company/Pages/Trademarks.aspx>.

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

Free and Open-Source Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

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

Copyright Notice for the Command-Line Editing Feature

© 1992, 1993 The Regents of the University of California. All rights reserved. This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright Notice for the Line-Editing Library

© 1992 Simmule Turner and Rich Salz. All rights reserved.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following restrictions:

1. The authors are not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.

Contents

check_mw_lib_mode	4
external_pin_file	5
mw_lib_lock_mode	7
mw_open_design_with_gui	8
mw_support_hier_fill_view	9
mwlib_second_power_on_top	11
skip_reference_library_read_lock	13
via_region_attributes	14

check_mw_lib_mode

Determines whether to check libraries for consistency when they are opened.

TYPE

boolean

DEFAULT

true

DESCRIPTION

Determines whether to check libraries for consistency when they are opened. The default value is *true*. When a library and its reference libraries are initially opened, tool checks for consistency between the main library's and its reference libraries' technology and settings. If discrepancies are found, warning messages are displayed in the window. However, tool continues to open the library and its reference libraries.

SEE ALSO

external_pin_file

Defines for the **auExtractBlockagePinVia** command to only extract external pins defined in the external pin file during BPV.

TYPE

string

DEFAULT

N/A

SYNTAX

```
set external_pin_file {db_file_name_list | external_pin_text_file}
```

1. {**db_file_name_list**} is a list that specifies one or more valid db file names
2. {**external_pin_text_file**} is a list of text files specifies the external pins for cells
The external pin text file format is as below:

```
<cell_name> {<pin_name> <pin_name> ...}
```

- (1). Each line specifies all the external pins for one cell or multiple cells.
- (2). Pin name is separated by blank.
- (3). <cell_name> can support wildcard for cells having same external pins.

Here we only support wildcard in the suffix using "*". The format is : <name_of_cell>*

For example, If there are two cells named "and1" and "and2" and there is no other cell with "and" prefix. They have same external pins A, B, O; we can define the external pins as below:

```
And* {A B O}
```

DESCRIPTION

Defines for the **auExtractBlockagePinVia** command to only extract external pins defined in the external

pin files during BPV. (Extract means text associated with geometries in the cells will be generated as a physical pin). If customer doesn't specify this variable, all the pins will be extracted during BPV. This variable can only be set efficiently before "Extract Blockage".

NOTE: This variable only works in Milkyway.

EXAMPLES

For example, if MW library is named "design", and its cells can be found in two DB files named "standard_cell1.db" and "standard_cell2.db", then the Milkyway command is,

```
Milkyway> set external_pin_file {standard_cell1.db standard_cell2.db}
Milkyway> auExtractBlockagePinVia
```

For example, if MW library is named "design" and the external pin file is exPin.txt, then the Milkyway command is,

```
Milkyway> set external_pin_file {exPin.txt}
Milkyway> auExtractBlockagePinVia
```

SEE ALSO

mw_lib_lock_mode

Determines what kind of mode should be used to lock library.

TYPE

string

DEFAULT

new

DESCRIPTION

Determines the mode in which a library is locked when it is in use. The default value is *new*. The choice of a lock mode is limited to the following strings:

- new, the default
- old
- both
- neither

The *old* lock mode is used to provide some protection for the library against pre 2003.09 binaries, which only lock the "lib" file. If a library is not a legacy library, it's recommended to switch off the *old* mode.

SEE ALSO

mw_open_design_with_gui

Controls whether MW design will be opened with GUI widget or not.

TYPE

Integer

DEFAULT

1

DESCRIPTION

This variable controls whether MW design will be opened with GUI widget or not.

By default (1), an MW design will be opened with GUI widget. You can set this variable to be zero before running **open_mw_cel** if you want to open a design without GUI.

NOTE: This variable only works in Milkyway.

SEE ALSO

`open_mw_cel(2)`

mw_support_hier_fill_view

Affects the behavior of the **copy_mw_cel**, **remove_mw_cel**, and **save_mw_cel** commands when they operate on a hierarchical FILL view.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

This variable affects the behavior of the **copy_mw_cel**, **remove_mw_cel**, and **save_mw_cel** commands when they operate on a hierarchical FILL view. This type of view created by the **signoff_metal_fill** command in the IC Compiler tool.

By default, this variable is set to **true**, which causes the affected commands to consider whether a FILL view is implemented hierarchically and to act accordingly, as described below for each command. If you set this variable to **false**, each command carries out its operation on the specified view without considering whether it is a hierarchical FILL view.

copy_mw_cel Command

The **copy_mw_cel** command copies a specified FILL view hierarchically when you use the **-hierarchy** option in the command OR when the **mw_support_hier_fill_view** variable is set to **true** (even if the **-hierarchy** option is not used). The command copies all lower-level FILL views as well as the top-level FILL view. If copying any of these views would overwrite an identically named view in the destination library, the whole copy operation is canceled and you get an MWUI-1065 error message.

If the variable is set to false AND the **-hierarchy** option is not used, the **copy_mw_cel** command copies only the top-level FILL view, and you get a MWUI-137 warning message.

remove_mw_cel Command

The **remove_mw_cel** command removes a specified FILL view hierarchically when you use the **-hierarchy** option in the command OR when the **mw_support_hier_fill_view** variable is set to **true** (even if the **-hierarchy** option is not used). The command removes all lower-level FILL views as well as the top-level FILL view.

If the variable is set to false AND the **-hierarchy** option is not used, the **remove_mw_cel** command removes only the top-level FILL view, and you get a MWUI-137 warning message.

save_mw_cel Command

The **save_mw_cel** command saves a specified FILL view hierarchically when the **mw_support_hier_fill_view** variable is set to **true**. The command saves any open and modified lower-level FILL views as well as the top-level FILL view.

If the variable is set to false, the **save_mw_cel** command saves only the top-level FILL view, and you get a MWUI-137 warning message.

SEE ALSO

copy_mw_cel(2)
remove_mw_cel(2)
save_mw_cel(2)
MWUI-137(n)
MWUI-1065(n)

mwlib_second_power_on_top

Indicates whether the second power pin is at the top of the cell during library preparation of a double-height, dual-power standard cell.

TYPE

Boolean

DEFAULT

false

GROUP

mwlib_variables

DESCRIPTION

By default, a standard cell has one primary power pin and one primary ground pin, located on opposite ends of the cell (top and bottom). When the cell is placed in a row, the power and ground pins are aligned with the power and ground rails of the row.

An alternative type of standard cell has double the standard height, two different power pins (for example, VDD and VDDS) at opposite ends of the cell, and a single ground pin at the center of the cell. During library cell preparation, you need to specify the location of the second power pin (for example, VDDS) before you set the place-and-route boundary for the cell.

If the second power pin is at the top, set the **mwlib_second_power_on_top** variable to **true**. Otherwise, leave the variable set to **false** (the default). Make this selection before you create the place-

and-route boundary using "Cell Library > Set PR Boundary" or the `auSetPRBdry` command.

During library cell preparation, these are the key steps for setting the place-and-route boundary for a double-height, dual-power cell:

- Set the **`mwlib_second_power_on_top`** variable to **`true`** if the second power is at the top, or leave it set to **`false`** if the second power rail is at the bottom.
- Open the Set PR Boundary dialog box (choose Cell Library > Set PR Boundary or execute the `auSetPRBdry` command in Scheme mode).
- In the dialog box, enter the Milkyway library name.
- Set "Apply To" to "Only Cell(s)" and enter the name of the double-height cell to which you are applying the boundary.
- Set "Height" to "specify" and enter the full height of the cell, which must span two power supply rails.
- Set "Multiple (2x, 3x)" to "based on marked cell type".
- Set "Tile Name" to "specify" and enter the unit tile name in the library.
- Click OK to close the dialog box and create the place-and-route boundary.

The **`mwlib_second_power_on_top`** variable setting has no effect on the preparation of standard cells that have only one primary power pin and one primary ground pin.

SEE ALSO

`save_mw_cel(2)`

skip_reference_library_read_lock

Determines whether read lock on a reference library should be skipped.

TYPE

boolean

DEFAULT

true

DESCRIPTION

Bypasses the lock on a reference library when you are certain the reference library is read only. No one else can modify the library. If the variable is set to be *true*, read lock on reference library will be skipped. If the variable is set to be *false*, read lock on ref lib will be performed as usually. The default value is *true*.

SEE ALSO

via_region_attributes

Contains attributes related to via region.

DESCRIPTION

Contains attributes related to via region.

You can use **get_attribute** to determine value of an attribute, and use **report_attribute** to get a report of all attributes on specified object. Specified with **list_attribute -class via_region -application**, the definition of attributes can be listed.

Via Region Attributes

bbox

Specifies the bounding-box of a via region. The **bbox** is represented by a **rectangle**.

The format of a *rectangle* specification is $\{\{llx\ llx\}\ \{urx\ ury\}\}$, which specifies the lower-left and upper-right corners of the rectangle.

The data type of **bbox** is string.

This attribute is read-only.

bbox_ll

Specifies the lower-left corner of the bounding-box of a via region.

The **bbox_ll** is represented by a **point**. The format of a *point* specification is $\{x\ y\}$.

You can get the **attr_name** of a via region, by accessing the first element of its **bbox**.

The data type of **bbox_ll** is string.

This attribute is read-only.

bbox_llx

Specifies x coordinate of the lower-left corner of the bounding-box of a via region.

The data type of **bbox_llx** is double.

This attribute is read-only.

bbox_lly

Specifies y coordinate of the lower-left corner of the bounding-box of a via region.

The data type of **bbox_lly** is double.

This attribute is read-only.

bbox_ur

Specifies the upper-right corner of the bounding-box of a via region.

The **bbox_ur** attribute is represented by a point. The format of a point specification is {x y}.

You can get the **bbox_ur** of a via region, by accessing the second element of its **bbox**.

The data type of **bbox_ur** is string.

This attribute is read-only.

bbox_urx

Specifies x coordinate of the upper-right corner of the bounding-box of a via region.

The data type of **bbox_urx** is double.

This attribute is read-only.

bbox_ury

Specifies y coordinate of the upper-right corner of the bounding-box of a via region.

The data type of **bbox_ury** is double.

This attribute is read-only.

cell_id

Specifies Milkyway design ID in which a via region object is located.

The data type of **cell_id** is integer.

This attribute is read-only.

name

Specifies name of a via region object.

The data type of **name** is string.

This attribute is read-only.

number_of_points

Specifies the number of points to illustrate the boundary of a via region object.

The data type of **number_of_points** is integer.

You can refer to the attribute **points**. The list length of **points** is the value of **number_of_points**.

This attribute is read-only.

object_class

Specifies object class name of a via region, which is **via_region**.

The data type of **object_class** is string.

This attribute is read-only.

object_id

Specifies object ID in Milkyway design file.

The data type of **object_id** is integer.

This attribute is read-only.

owner_port

Specifies port name which a via region object is associated with.

The data type of **owner_port** is string.

This attribute is read-only.

points

Specifies points of the boundary of a via region. A via region can be a rectangle, a rectilinear polygon, or multiple rectangles.

When a via region is either a rectangle or a rectilinear polygon, its **points** is represented by a list of points. The last element of the list is the same as the first element.

When a via region consists of multiple rectangles, its **points** is represented by a list of points of rectangles. Every five points represent one rectangle.

The data type of **points** is string.

This attribute is read-only.

is_rotate90

Specifies whether a via region is used for the contactCode object who is rotated 90 degree.

The data type of **is_rotate90** is boolean.

This attribute is read-only.

via_master

Specifies the name of the via master associated with a via region object.

The data type of **via_master** is string.

This attribute is read-only.

SEE ALSO

```
get_attribute(2)  
list_attributes(2)  
report_attribute(2)
```