```
icc2_shell> man set_floorplan_halo_rules

2.  Synopsys Commands                                   Command Reference
                          set_floorplan_halo_rules

NAME
       set_floorplan_halo_rules
              Defines a halo floorplan rule in the design.

SYNTAX
       status set_floorplan_halo_rules
              -from_object_types from_type_list
              -to_object_types to_type_list
              -to_lib_cells lib_cells
              -sides side_list
              -type inner | outer
              -name rule_name
              [-shielding_object_types type_list]
              [-shielding_lib_cells lib_cells]
              [-must_enclose]
              [-follow_rotations]
              [-ignore_rotate90]
              [-layers layer_list]
              [-to_layers layer_list]
              [-forbidden_list distance_list]
              [-forbidden_ranges {{low high} {low1 high1} ... }]
              [-max distance]
              [-min distance]
              [-offset distance]
              [-step distance]
              [-valid_list distance_list]
              [-valid_ranges {{low high} {low1 high1} ... }]
              [-offset_ranges {{low high} {low1 high1} ... }]

   Data Types
       from_type_list list
       to_type_list   list
       lib_cells      collection
       side_list      list
       rule_name      string
       type_list      list
       layer_list     list
       distance_list  list
       low            float
       high           float
       low1           float
       high1          float
       distance       float

ARGUMENTS
       -from_object_types from_type_list
              Specifies the list of "from" object types for the halo floorplan
              rule. These type of objects forms the halo around other objects
              specified with -to_object_types or library cells specified with
              -to_lib_cells. Valid values for this option are core_area,
              placement_blockage, routing_blockage, shape and std_cell_area.
              This is a mandatory option.

       -to_object_types to_type_list
              Specifies the list of "to" object types for the halo floorplan
              rule. These type of objects are surrounded by other objects
              specified with -from_object_types. Valid values for this option
              are hard_macro, placement_blockage, routing_blockage,
              soft_macro, shape and std_cell_area. This option is mutually
              exclusive with -to_lib_cells and either one of them must be
              specified.

       -to_lib_cells lib_cells
              Specifies the collection of library cells for the halo floorplan
```

rule. These library cells are surrounded by other objects specified with -from_object_types. This option is mutually exclusive with -to_object_types and you must specify one or the other.

-sides side_list
Specifies the sides or directions from which the "from" object surrounds the "to" object. Spacing is checked between the objects. Valid values are all, bottom, horizontal, left, right, top and vertical. The horizontal argument includes both left and right. Similarly, the vertical argument includes both bottom and top. This is a mandatory option.

-type inner | outer
Specifies the halo type for the distance check. -type inner specifies the inner region of surrounding object to surrounded object needs to be checked. -type outer specifies the outer region of surrounding object to surrounded object needs to be checked. This is a mandatory option.

-name rule_name
Specifies the name of the halo floorplan rule. This is a mandatory option.

-shielding_object_types type_list
Specifies the collection of object types for the halo floorplan rule as shielding objects so that the rule is not applied when these object types form a shield between "from" and "to" objects. Valid values for this option are hard_macro and std_cell_area. This is an optional option.

-shielding_lib_cells lib_cells
Specifies the collection of library cells for the halo floorplan rule as shielding objects so that the rule is not applied when these lib cells form a shield between "from" and "to" objects.

-must_enclose
Specifies that the "from" object must completely enclose the "to" object from all sides. This is an optional option.

-follow_rotations
Specifies whether the sides specified by the -sides option should follow the rotations of the library cells, that is, if meaning of horizontal or vertical should change when the library cell has a 90-degree rotation. This option must be used together with -to_lib_cells or -to_object_types hard_macro or -to_object_types soft_macro. This is an optional option.

-ignore_rotate90
Specifies whether this rule can be ignored for library cells with a 90-degree rotation. This option must be used together with -to_lib_cells or -to_object_types hard_macro or -to_object_types soft_macro. This is an optional option.

-layers layer_list
Specifies the routing layers to be considered for the routing_blockage or shape in from object type. This option must be used together with -from_object_types routing_blockage or -from_object_types shape. This is an optional option.

-to_layers layer_list
Specifies the routing layers to be considered for the routing_blockage or shape in to object type. This option must be used together with -to_object_types routing_blockage or -to_object_types shape. This is an optional option.

-forbidden_list distance_list
Specifies a list of distances by which the "from" object cannot enclose the "to" object. This option is mutually exclusive with -valid_list. Values in the distance_list cannot be negative.

This is an optional option.

-forbidden_ranges {{low high} {low1 high1} ... }
Specifies a list of distance ranges between which the "from" object cannot enclose the "to" object. The enclosing distance must **not** lie **within** any of low **and** high in the specified list of ranges. This option is mutually exclusive **with** -valid_ranges. Values cannot be negative. This is an optional option.

-max distance
Specifies the maximum distance by which the "from" object can enclose the "to" object. The distance cannot be greater than **this** value. The distance cannot be negative. If -min is also specified, **this** value must be greater than the min value. This is an optional option.

-min distance
Specifies the minimum distance by which the "from" object can enclose the "to" object. The distance cannot be less than **this** value. The value specified cannot be negative. If -max is also specified then **this** value must be less than the max value. This is an optional option.

-offset distance
Specifies a **parameter** in distance calculation between the "from" **and** "to" objects. This option must be used together **with** -step. This **implies** that the distance has to be an **integer** multiple of the -step value plus the -offset value. This option is mutually exclusive **with** -offset_ranges. The value specified cannot be negative. This is an optional option.

-step distance
Specifies a **parameter** in distance calculation between the "from" **and** "to" objects. This option must be used together **with** -offset. This **implies** that the distance has to be an **integer** multiple of the -step value plus the -offset value **or** distance has to be in range of an integral multiple of step value plus offset_ranges value. The value specified must be greater than zero. This is an optional option.

-valid_list distance_list
Specifies a list of distance by which "from" object can surround the "to" object. This option is mutually exclusive **with** -forbidden_list. Values specified cannot be negative. This is an optional option.

-valid_ranges {{low high} {low1 high1} ... }
Specifies a list of distance ranges between which the "from" object can enclose the "to" object. The distance must lie **within** any of low **and** high in the specified list of ranges. This option is mutually exclusive **with** -forbidden_ranges. Values specified cannot be negative. This is an optional option.

-offset_ranges {{low high} {low1 high1} ... }
Specifies a list of distance ranges. This **implies** that the distance has to be in range of an integral multiple of step value plus offset_ranges value. Values specified can't be negative. This option must be used along **with** -step. This is an optional option.

DESCRIPTION
The set_floorplan_halo_rules command defines a named halo floorplan rule in the current **design**. The defined rule is persistent. If another floorplan rule by the same name exists then the command errors out.

There is a difference between the object **type** core_area **and** std_cell_area. The core_area object **type** means core boundary region without cutting out any blockages **and** is typically applicable **for** top level whereas std_cell_area object **type** means core boundary region

after cutting out all blockages and is typically applicable for block level.

If a halo rule is defined for a library cell and another halo rule is defined for a hard macro then the halo rule defined for the library cell takes precedence over the halo rule defined for hard macro when checks are done for that library cell.

If the measured value falls inside valid range or is a member of the valid list then there is no violation given by check_floorplan_rules regardless of other constraints like min, max, and so on. If this measured value is outside valid range or list then a violation is reported if other constraints are specified and they are not met or if no other constraints are specified.

EXAMPLES

The following example creates a halo rule by name h1 to check how hard macros and soft macros are surrounded by the core area. The check is done for all sides for the inner region of core area. The distance can be one of 5, 7, 9 and 15. Also the distance must be an integer multiple of 3 plus 13.

```
prompt> set_floorplan_halo_rules -name h1 -from_object_types core_area \
   -to_object_types {hard_macro soft_macro} -must_enclose \
   -sides {all top} -type inner -valid_list {5 7 9 15} -offset 13 \
   -step 3
```

SEE ALSO

remove_floorplan_rules(2)
report_floorplan_rules(2)
set_floorplan_area_rules(2)
set_floorplan_enclosure_rules(2)
set_floorplan_extension_rules(2)
set_floorplan_exception_rules(2)
set_floorplan_forbidden_rules(2)
set_floorplan_length_rules(2)
set_floorplan_spacing_rules(2)
set_floorplan_width_rules(2)

Version S-2021.06-SP5
Copyright (c) 2022 Synopsys, Inc. All rights reserved.

icc2_shell>