

## Using Nondefault Rules

This document presents orientations on how to use nondefault rules (NDRs) that are supported by TritonRoute (TR). Thus, the features of NDRs presented here are the ones that are currently supported in TR, but other features will be implemented latter, and some of the existing will be improved (and also need more testing).

NDRs are special (customized) rules that may be different from the default rules.

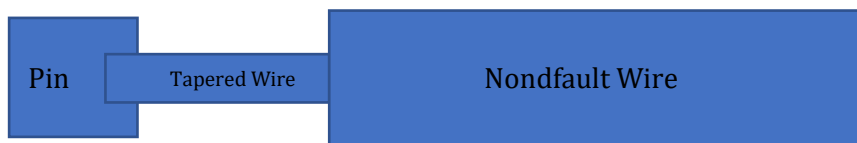
## Current support of NDRs in TR

### What you can do with NDRs:

- Increase the wire width
- Increase wire spacing
- Use special vias (needs more testing, but you can try it)

### Tapering wires and vias

A tapered wire or via uses the default rules instead of nondefault rules. It is used in the connections to pins, to allow pin access, as illustrated below.



A nondefault wire with default width but nondefault spacing can also be tapered and it will require default spacing. A tapered via will also require default spacing.

At the moment, TR always taper wires/vias within a radius of about 3 tracks from the access points of the pin (rarely, this may be violated).

We intend to add custom tapering in the future.

## Setting up NDRs in the design

In order to use NDRs you must manually write them in the lef and def files.

### NDRs in LEF Files

NDRs are defined in the LEF files. NDRs should be defined after the layers and vias definition. NDR definition take the following form:

```
[NONDEFAULTRULE ruleName
  [HARDSPACING ;]
  {LAYER layerName
    WIDTH width ;
    [SPACING minSpacing ;]
  } layerName ...
  [USEVIA viaName ;] ...
```

END *ruleName*]

Explanations of the keywords:

LAYER *layerName* ... END *layerName*

Specifies the constraints of a routing layer. Every routing layer must have a WIDTH keyword and value specified. All other keywords are optional.

SPACING *minSpacing*

Specifies the minimum spacing required by router-created shapes, using this NDR. If the spacing is given, it must be at least as large as the foundry minimum spacing rules defined in the LAYER definitions.

*Type:* Float, specified in microns

HARDSPACING

Specifies that the spacing rule used by the nondefault rule is *hard*, meaning that, if it is not obeyed, this implies in a violation. Without this keyword, the router should try to respect the spacing, but it does not need to. At the moment, TR always treats nondefault spacing as hard spacing, so this keyword has no effect on TR right now. However, if you want hard spacing and want to validate the solution of TR in an EDA tool, you should use this keyword.

USEVIA *viaName*

Specifies a previously defined via from the LEF VIA statement, that should be used by the NDR. If you don't assign any via, the default vias will be used.

*Using large vias needs more testing at this moment.*

WIDTH *width*

Specifies the required minimum width for *layerName*.

*Type:* Float, specified in microns

Example of a NDR called NDR\_1W\_3S, with wire width equal the default width and spacing equal to 3x default spacing:

```
NONDEFAULTRULE NDR_1W_3S
    HARDSPACING ;
    LAYER Metal1
        WIDTH 0.06 ;
        SPACING 0.18 ;
    END Metal1
    LAYER Metal2
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal2
    LAYER Metal3
```

```

        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal3
    LAYER Metal4
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal4
    LAYER Metal5
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal5
    LAYER Metal6
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal6
    LAYER Metal7
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal7
    LAYER Metal8
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal8
    LAYER Metal9
        WIDTH 0.07 ;
        SPACING 0.21 ;
    END Metal9
END NDR_1W_3S

```

## NDRs in DEF Files

NDRs can also be defined in def files. The syntax is the following:

```

NONDEFAULTRULES numRules ;
    {- ruleName
        [+ HARDSPACING]
        {+ LAYER layerName
            WIDTH minWidth
            [SPACING minSpacing]
        } ...
        [+ VIA viaName] ...
    ;} ...
END NONDEFAULTRULES

```

Here you can define multiple NDRs inside the NONDEFAULTRULES statement.

The meanings of the Keywords are the same from LEF. In the case of conflict of information between LEF and DEF, DEF info has priority.

### Assigning a NDR to a net

To assign a NDR to a net, it is necessary to add + NONDEFAULTRULE <NDR\_name> in the net definition.

Example of a net, named net1, with 2 pins (B from inst1689 and A from inst1989), with the NDR defined in the previous example:

```
- net1
  ( inst1689 B ) ( inst1989 A )
  + NONDEFAULTRULE NDR_1W_3S
;
```

For more information on NDRs, please see <http://free-online-ebooks.appspot.com/enc/14.17/lefdefref/LEFSyntax.html#NondefaultRule>.