Design Directives are objects that are placed on the schematic during design capture, providing a way of specifying instructions to be passed to other parts of the software. A variety of Design Directives are available, for use in the following two ways:

- Directives associated with the compilation of source schematic documents.
- Directives used to pass information defined on a schematic sheet through to the PCB.

The following sections take a closer look at these areas and the associated directives.

###Compilation-related Directives

Designs evolve over time and are captured in stages. As each stage is bedded down, it's not uncommon to want to check them in isolation to the rest of the design. Compilation of an individual schematic document (or the entire project) at intermittent stages in the capture process will often yield a number of error messages, caused by circuitry that is yet to be captured, or interface wiring between circuit fragments that are still incomplete. Such messages are of no real value since they only create noise around the real information. The quickest and easiest way to suppress these compilation errors is by placing No ERC or Compile Mask directives.

####No ERC Directive

Object page: No ERC

The No ERC directive is placed on a node in the circuit to suppress all reported Electrical Rule Check warnings and/or error violation conditions that are detected when the schematic project is compiled. Use a No ERC directive to deliberately limit error checking at a certain point in the circuit that you know will generate a warning (such as an unconnected pin), while still performing a comprehensive check of the rest of the circuit.

The No ERC directive supports a number of different styles and can be displayed in any color. Use this ability to reflect the design intent for this point in the circuit.
The No ERC directive has two modes of operation:

1. **Suppress All Violations** - in this mode, all possible warnings and/or error conditions are suppressed. The directive is often referred to as a **Generic No ERC** directive, in this mode.
2. **Suppress Specific Violations** - in this mode, only the selected warnings or error conditions are suppressed; any other warnings or errors will be detected and reported. The directive is often referred to as a **Specific No ERC** directive, in this mode.

Suppressed errors can be displayed in the **Messages panel** by enabling the **Report Suppressed Errors in Messages Panel** option, on the **Error Reporting tab** of the **Project Options - Error Reporting** dialog. This feature can be used in the final stages of design to ensure that no critical errors have been inadvertently suppressed.

**Example Usage**

How many times have you encountered a warning about a net 'not having a driving source', only to find that the message can be safely ignored? Perhaps an input pin is fed from a connector, the pin of which is nominally passive and the driving signal only present when an external cable is plugged in? Maybe the net is sourced from a pull-up resistor or switch, again passive in nature? One of the following strategies could be adopted to resolve this warning:

- You could change the electrical characteristic of a source pin on the net. This is a fix rather than suppression, but as it involves a change to a pin's default mode of operation, it could create trouble further down the track. For example, consider wiring changes made to a design, in which the graphical display of pin direction is not enabled. Such changes might result in an output being connected to a pin of a passive device. If the pin of that device has been set electrically as an output (to alleviate previous driving source warnings), then you will have created a connection violation.
- You could set the report mode for the associated violation check - defined on the **Error Reporting tab** of the **Options For Project dialog** - to **No Report**. This disables the check of this particular violation, but you would also not be able to catch any genuine errors elsewhere in the design.
- The third (and arguably best) option is to place a No ERC directive on the net. You are not changing the design in any way, other than to suppress warning message 'noise' that you know is not a problem.

**Example Usage**

Place No ERC directives on nets you know will cause 'no driving source' warnings.
Placing ERC Directives

A No ERC directive can be placed into a schematic document in a number of ways:

- Place a generic No ERC directive by choosing the **Place » Directives » Generic No ERC command** from the main menus, by clicking the button on the **Wiring** toolbar, or by right-clicking in the workspace, and selecting **Place » Directives » Generic No ERC command**.

- Place a specific No ERC directive on a point in the circuit that is already showing a violation, by right-clicking over a violating object in the workspace (highlighted by a wavy colored line) and choosing the **Place NoERC to Suppress command**, from the context menu.

- Place a specific No ERC directive on a point in the circuit that is already showing a violation, by right-clicking on a warning/error in the **Messages panel**, choosing the **Place Specific NoERC for this violation command**, then jumping straight to that point in the schematic and placing a No ERC directive configured to suppress that warning/error.

![Using the right-click context menu to place a specific No ERC directive.](image-url)
Using the right-click context menu in the Messages panel to place a specific No ERC directive.

The command will only be available if the message is a Net-related compiler violation.

**Editing**

During placement, and while the No ERC object is still floating on the cursor, the following editing actions can be performed:

- From the No ERC Properties panel. This method of editing uses the associated Properties panel mode to modify the properties of an object.
During placement, the **No ERC** mode of the **Properties** panel can be accessed by pressing the **Tab** key.

After placement, the **No ERC** mode of the **Properties** panel can be accessed in one of the following ways:

- Double-click on the placed directive.
- Placing the cursor over the directive then right-click and choose **Properties** from the context menu.
- If the **Properties** panel is already active, select the directive.

The properties can be accessed prior to entering placement mode from the Schematic – Defaults page of the **Preferences** dialog. This allows the default properties for the object to be changed, which will be applied when placing subsequent objects.

- From the **No ERC dialog**. This method of editing uses the No ERC dialog to modify the violation types and connection errors of a Specific No ERC object.
The dialog can be accessed by clicking **Specific Violations** in the **Suppressed Violations** region of the **Properties** panel in **No ERC** mode (mentioned above).
The Specific No ERC directive can be configured to target multiple violations to support circuits that will generate multiple errors/warnings.

- From the SCH List and SCH Filter panels, A List panel allows you to display design objects from one or more documents in tabular format, enabling quick inspection and modification of object attributes. Used in conjunction with appropriate filtering - by using the applicable Filter panel or the Find Similar Objects dialog - it enables the display of just those objects falling under the scope of the active filter - allowing you to target and edit multiple design objects with greater accuracy and efficiency.
Deactivating

Rather than deleting a No ERC directive, it can be made inactive (disabled in the eyes of the Compiler). This state can be changed by toggling the directive's **Active** property - available through any of the methods of editing. An inactive No ERC directive will appear grey in the workspace.
If you need to temporarily remove use of a No ERC directive, render it inactive, rather than deleting it.

**Compile Mask Directive**

*Object page: Compile Mask*

No ERC directives are great for suppressing a low number of violating pins, ports, sheet entries, or nets within a design. But in some cases, it may be desirable to remove an entire section of the design; including components. Use a Compile Mask directive ([Place » Directives » Compile Mask command](#)) or click the Compile Mask button ([ ](#)) in the directives drop down on the **Active Bar** to effectively hide the area of the design it contains from the Compiler, allowing you to manually prevent error checking for circuitry that may not yet be complete and you know will generated compile errors. This can prove very useful if you need to compile the active document, or project, to check the integrity of the design in other specific areas, but do not want the 'noise' of compiler-generated messages associated with unfinished portions of the design.

As its name suggests, this directive instructs the Compiler to ignore any objects that fall *completely* within the bounds of the defined mask. Place the mask exactly as you would a note or rectangle object.

Consider the example schematic circuitry in the following image, where the wiring to the LCD1 device is not yet complete. Compiling just this schematic ([Project » Compile <Project>](#)) will result in numerous violation messages (also shown), each of which is caused by the incomplete circuitry. Hover over the image to see the effect of placing a Compile Mask directive around the incomplete circuitry. These violations will be ignored by the Compiler, while the rest of the circuit on the
schematic - which is completely wired - is checked. Notice that objects that are truly masked - those that completely fall within the bounding rectangle of the mask - will appear greyed-out.

A compile mask can be displayed in either expanded (full frame) or collapsed (small triangle) modes. These modes correspond to the mask being enabled and disabled respectively. Toggle the display mode by clicking on the top-left corner of a placed compile mask. While compile masks can be rotated or mirrored along the X or Y axis, this has no effect on the orientation of the design circuitry within.

**Application to Simulatable Designs**

Because all elements of a design covered by a Compile Mask directive are invisible to the design compiler, they will be omitted from the design. This feature can be put to great use when simulation is included as part of the design flow.

Voltage and Current sources are necessary elements when running circuit simulations, but they have no place on the completed PCB. By applying a small amount of planning to the structure of the circuit, it is usually possible to group all simulation-specific components in one section of the design - a section that can then be easily covered by a Compile Mask directive.

When the circuit is used for simulation, the Compile Mask directive is disabled to reveal the simulation-specific components. Once the circuit is verified and ready for inclusion in the design, the Compile Mask directive can be re-enabled, so that the simulation-specific components are excluded from the design. If the design should ever need to be changed again in the future, another simulation pass can be quickly executed prior to sign-off by disabling the Compile Mask directive (to reveal the simulation-specific components again).
Initially, the Compile Mask directive is disabled, making the circuit ready for simulation. Roll over the image to show the Compile Mask directive enabled, thereby hiding the simulation sources from the design.

**PCB-related Directives**

*Object page:* Parameter Set

As a Unified Design Environment, Altium Designer provides the ability for PCB requirements to be defined prior to laying out the board. This is achieved by adding and specifying parameters to objects placed on the schematic sheet(s).

For certain schematic design objects - such as components, sheet symbols, ports, etc - this involves adding the relevant parameter(s) as part of that object's properties. For net objects such as wires and buses, parameters cannot be added directly as a property of the wire or bus. Instead, the parameters required to hold the information are specified using dedicated design directives.

The following information can be specified, using directives, and will be transferred to the appropriate PCB-based definitions during design synchronization:

- PCB layout constraints
- Differential pairs
- Net classes

By including design directives within the Schematic, design engineers can specify explicit design constraints, and it ensures the Schematic remains the master record of the design. Any amendments to the design would be carried out on the schematic side only, and pushed across to the PCB. This can become particularly important when multiple people are working on the design - especially if they are geographically separated. Rather than attempting to communicate with one another through chains of emails, or phone calls, the person capturing the design can ensure that particular constraints are
indeed used during the layout phase.

**Placing PCB Layout Directives**

At the heart of this functionality is the Parameter Set directive.

These are essentially user-defined Parameter Set objects, which can be associated to a net object within a schematic design. Place a PCB Layout directive on a wire, bus, or signal harness, to define one or more design constraint targeting the associated net(s). When a PCB is created from the schematic, the information in the PCB layout directive is used to create relevant PCB design rules. The information specified by a PCB Layout directive is applied only to the net (or set of nets) to which the directive is connected.

This acts as a container for any number of parameters targeting the net that the Parameter Set directive is attached to. A default Parameter Set directive, one that is devoid of parameters, can be placed (Place » Directives » Parameter Set), with the relevant parameter(s) being added to it after. The following sections take a closer look at using these parameter-based directives. Both user-defined (Parameter Set) and pre-defined (Differential Pair) parameter set directives are available. The only difference between an empty parameter set and a pre-defined parameter set is that the pre-defined parameter sets include a parameter, as will be described below.

**Placing Parameter Set Directives**

Place a directive of this type by choosing the Place » Directives » Parameter Set command from the main menu, or when right-clicking within the workspace. When placing a default parameter set directive, there will be no existing parameters. A parameter set is a design directive that allows design specifications to be associated to a net-type object within a schematic design. For example, use a parameter set to declare two nets to be members of a differential pair. It is the presence of specifically named parameters in the parameter set that the software uses to determine which design directive you are placing.

In addition to user-defined parameter directives, a rule-based parameter directive is defined from the Choose Design Rule Type dialog, accessible from a parameter's associated properties dialog. Access involves the following:

1. Press Tab before placing the Parameter Set, or double-click an already placed Parameter Set directive, to display the the Parameter Set mode of the Properties panel.
2. Next, click the Add button in the Rules section of the the Parameter Set mode of the Properties panel to select a rule from the Choose Design Rule Type dialog.
Use the *Choose Design Rule Type* dialog to choose the rule that you wish to add as a rule parameter to the directive. Double-clicking on a rule type will give you access to the relevant *Edit PCB Rule (From Schematic)* dialog, from where you can define the constraints for the rule.
Specifying the constraints for a chosen rule.

The entry for the parameter’s Rules region will be the rule type chosen, along with the specified constraints. The following image illustrates the defined width constraint rule parameters for a Parameter Set directive. To display the rule within the workspace, click the visibility ( viện ) icon within the Rules region.
Multiple rule constraints defined for a particular net, courtesy of a Parameter Set directive.

When the design is transferred to the PCB, through the synchronization process, the relevant design rules will be created, based on the information contained within a directive. The word Schematic is used in the name for each generated rule, to distinguish the source of that rule.
Generated design rules on the PCB side.

Remember that multiple parameters can be added to the same Parameter Set directive, allowing for a neater schematic.

**Placing Differential Pair Directives**

A Differential Pair directive allows you to define a differential pair object on the schematic. Attach a directive of this type to both the positive and negative nets of the intended pair or cover the pair with a blanket object to target multiple nets with a single directive. Place a directive of this type by choosing the **Place » Directives » Differential Pair command** from the main menus. The directive contains a single parameter entry, with **Name**: DifferentialPair and **Value**: True. Each pair of directives (one for the positive net, one for the negative) of this type will yield a single differential pair object when transferred to the PCB.
Multiple parameters can be added to the same parameter set object. In the image below, the Differential Pair directive was placed to touch the edge of the blanket. It was then edited to include a **Class Name** parameter and a **Rule** parameter. When the design is synchronized with the PCB, these additional elements will be created:

- Eight PCB differential pairs
- A PCB net class called **ROCKET_IO_LINES**, containing the 16 named nets that are under the blanket
- A Differential Pair Routing design rule, with a scope of `InNetClass('ROCKET_IO_LINES')`

By using a blanket, only one Differential Pair directive is required. As well as defining the differential pairs, it also specifies a Net Class and a Differential Pair Routing rule. These nets will become members of that Net Class, and that Net Class will be used to scope the Differential Pair Routing rule.

When the design is transferred to the PCB, through the synchronization process, a single differential pair object is created from each pairing of directives. The name of a generated differential pair object on the PCB side will be the root name for the net pair on the schematic. For example directives added to `D_N` and `D_P` on the schematic, will generate a differential pair object on the PCB with the name D. Each resulting differential pair object will be added to the default Differential Pair class: `<All Differential Pairs>`. You can rename differential pair objects on the PCB side only.

Generated differential pair objects can be quickly verified using the **PCB panel**, configured in **Differential Pairs Editor mode**.
By attaching a differential pair directive to the perimeter of a **Blanket Object**, you can quickly create differential pair objects based on differential nets within the confines of that blanket.

### Placing Net Class Directives

Net Class directives enable you to create user-defined net classes on the schematic. When a PCB is created from the schematic, the information in a Net Class directive is used to create the corresponding Net Class on the PCB. To make a net a member of a net class, attach a Net Class directive to the relevant wire, bus, or signal harness, and set the directive’s `ClassName` parameter to the name of the desired class. The **Generate Net Classes** option (for User-Defined Classes) must be enabled on the **Class Generation** tab of the *Project Options* dialog to use this feature.

If a Net Class directive has been defined for a net, then any PCB design rules that are also created by that parameter set object will have a rule scope of Net Class, when the design is transferred to the PCB editor. A Net Class directive can be created from your placed Parameter Set directive by adding a class that must have its value set to the required PCB Net Class.

While Net Classes can be created fairly easily from within the PCB editor, the logical function or grouping of Nets is usually much clearer in the Schematic, and so it makes more sense to drive the process from there.

**Important Project Option Requirements**

To ensure Schematic defined Parameter Sets are propagated to the PCB, the following options must be set in the *Options for PCB Project* dialog:

- Enable the **Generate Net Classes** option located in the **User-Defined Classes** region of the dialog’s **Class Generation** tab.
- On the dialog's **Comparator tab**, set the **Differences Associated with Nets » Extra Net Classes** checking mode to **Find Differences**.
To propagate Parameter Set directives to the PCB, two project options need to be configured. First, enable the **Generate Net Classes** option on the **Class Generation** tab. Roll over the image to show the **Comparator** tab, where you will need to set the **Extra Net Classes** option to **Find Differences**.

When the design is transferred to the PCB, through the synchronization process, the relevant net classes will be created, based on the information contained within a directive.

Attach a Net Class directive to a **Blanket object**, to create a net class whose members are the individual nets covered by that blanket. If a PCB Layout directive is also attached to that blanket, the PCB Layout directive’s rule parameters will target that net class, rather than each individual net. When importing the changes into the PCB document, this results in a single design rule being created (per parameter), with a scope set to target the net class.

**Placing Blanket Directives**

**Object page:** Blanket

Parameter Set directives can only target the specific net that they are attached to, but when combined with a Blanket directive, their scope can be expanded to cover all nets within the blanket.

Place a directive of this type by choosing the **Place » Directives » Blanket command** from the main menus. When placing a blanket, you can either define a simple rectangular shape, or a polygonal-shape. The latter gives more precise control over coverage of the required net objects on a sheet.

The blanket identifies the nets of interest - place a **Parameter Set directive** anywhere on the edge of the blanket to apply design requirements to those nets. To apply the perimeter directive to a net under a Blanket directive, an object associated to that net - a pin, a port, a net label, a power port, a wire/bus/harness segment (including both ends) - must fall within the bounds of the blanket. Note that for net identifiers, such as net labels, the hotspot must be within the blanket. If member nets do not come across into the PCB Parameter Set as expected, try adjusting the area of the blanket accordingly.

To check which nets the blanket directive will apply to, use the **Net Colors** feature to highlight them. Choose the required color from the **View » Set Net Colors** menu, then click on the perimeter of the required Blanket directive. To clear the highlighting for a specific net, use the **View » Set Net Colors »**
Clear Net Color command, then click on the net you wish to remove the coloring from. To clear net coloring from all schematic sheets, use the View » Set Net Colors » Clear All Net Colors command.

An example of using a Blanket directive to apply a Parameter Set directive to nets within the blanket.

Example usage of a blanket directive can include:

- Attaching a Parameter Set directive to a blanket object to have its rule parameters applied to each individual net covered by that blanket.
- Attaching a Parameter Set directive to a blanket object to create a Parameter Set whose members are the individual nets covered by that blanket.
- Attaching a Differential Pair directive to a blanket object to create differential pair objects based on differential nets within the confines of that blanket.

When attaching a Parameter Set directive to a blanket object, the directive's rule parameters will target that net class, rather than each individual net. When importing the changes into the PCB document, this results in a single design rule being created (per parameter), with a scope set to target the net class.

Attach a Parameter Set directive to a Blanket object, to have its rule parameters applied to each individual net covered by that blanket. If a Parameter Set directive is also attached to that blanket, the Parameter Set directive's rule parameters will target that net class, rather than each individual net. When importing the changes into the PCB document, this results in a single design rule being created (per parameter), with a scope set to target the net class.

You can also copy a perimeter Parameter Set directive and attach it to another Blanket directive or even individual wires, busses or harnesses - the result will be to add all additional nets associated with the same Parameter Set directive, to the same generated PCB Net Class.

**Indirect (Parameter-based) Directives**

Parameter Set directives are necessary when targeting design objects in the Schematic that can't contain parameters, but for those objects that can, design directives can be applied indirectly by adding (and defining) them as parameters to the relevant schematic object. In essence, they are parameter-based directives.
Examples of how parameter-based directives could be used would include limiting the height of a particular component, or adding a clearance constraint targeting all objects in the design. The required parameter that defines the constraint is added to the object as a rule.

When synchronized with the PCB, parameter-based directives that have been added to objects in the schematic will become PCB design rules. The scope of the corresponding PCB design rule will be determined by the nature of the object to which the parameter was first assigned. The following table summarizes the schematic parameter-to-PCB rule scope options that are supported.

<table>
<thead>
<tr>
<th>Add a Parameter (as a rule) to a...</th>
<th>From...</th>
<th>For a PCB Rule Scope of...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin</td>
<td>the Parameters tab of the Pin mode of the Properties panel.</td>
<td>Pad</td>
</tr>
<tr>
<td>Port</td>
<td>the Parameters tab of the Port mode of the Properties panel.</td>
<td>Net</td>
</tr>
<tr>
<td>Component</td>
<td>the Parameters region of the Components mode of the Properties panel.</td>
<td>Component</td>
</tr>
<tr>
<td>Sheet Symbol</td>
<td>the Parameters tab of the Sheet Symbol mode of the Properties panel, when Local is selected in the Source region.</td>
<td>Component Class</td>
</tr>
<tr>
<td>Device Sheet Symbol</td>
<td>the Parameters tab of the Sheet Symbol mode of the Properties panel, when Device is selected in the Source region.</td>
<td>Component Class</td>
</tr>
<tr>
<td>Managed Sheet Symbol</td>
<td>the Parameters region of the Sheet Symbol mode of the Properties panel, when Managed is selected in the Source region.</td>
<td>Component Class</td>
</tr>
<tr>
<td>Sheet</td>
<td>the Parameters tab of the Document Options mode of the Properties panel.</td>
<td>All Objects</td>
</tr>
</tbody>
</table>

In each case, the method of adding a rule-based parameter is the same. From the respective tab or dialog, perform the following:

1. Add a parameter as a rule.
2. Select which rule type to use.
3. Configure the constraints for the chosen rule type.

When adding design rule parameters to objects on a schematic, a unique ID is given to each rule parameter. The same IDs are given to the corresponding design rules that are created on the PCB. With this Unique ID, the constraints of a rule can be edited on either the schematic or PCB side, and the changes pushed through upon synchronization.

**Specifying Component Classes**

In a similar vein, component classes can be defined on the schematic by adding a ClassName parameter to targeted components and setting its value to the desired class name. When the design is transferred to the PCB, the defined component classes will be created.

To ensure Schematic defined Component Classes are propagated to the PCB, the following options must be set in the Options for PCB Project dialog:

- Enable the Generate Component Classes option located in the User-Defined Classes region of the dialog's Class Generation tab.
- On the dialog's Comparator tab, set the Differences Associated with Components » Extra Component Classes checking mode to Find Differences.
Once defined, classes may be locked in the **Classes** section of the respective directive object properties.

To propagate Component Classes to the PCB, two project options need to be configured. First, enable the **Generate Component Classes** option on the **Class Generation** tab. Roll over the image to show the **Comparator** tab, where you will need to set the **Extra Component Classes** option to **Find Differences**.

### Controlling the Printing of Directives

By default, all design directives are included during printing of the schematic sheets. This can, however, be changed:

- From within the [Schematic Print Properties dialog](#), when printing directly from the schematic, or as part of a configured output in an [Output Job Configuration file](#).
- From the **Schematics include** region, on the [Additional PDF Settings](#) page of the [Smart PDF wizard](#).
Control the printing of directives as required. For No ERC directives, you can opt to print certain symbol styles, while excluding others.

Source URL: