When schematics were originally captured on paper, it was often on a single sheet of paper large enough to fill a big drafting table, and reproduced by a dedicated large-format copier. Times have changed, now schematics are captured on a desktop PC, stored on a server, and printed on a small format laser printer.

This change means that even a simple design can be more easily displayed and understood if it is presented on multiple schematic sheets. Even when the design is not particularly complicated, there can be advantages in organizing it across multiple sheets.

For example, the design may include various modular elements. Maintaining these modules as individual documents allows several designers to work on a project at the same time. Breaking the design into logical modules greatly enhances the readability of the design too, an important consideration for those that need to read and interpret the schematic later in the life of that product. Another advantage is that when a design is structured over a number of sheets with fewer components on each, small format printing, such as laser printers, can be used.

There are two decisions to make if you plan to spread your design over multiple sheets:

- the structural relationship of the sheets, and
- the method employed for electrical connectivity between the circuitry on those sheets.

Your choice will vary according to the size and type of each project, and your personal preferences.

This article focuses on the structural relationships between the sheets, how it works and the tools and techniques available to create a multi-sheet design. To learn more about how the connectivity is created, refer to the Creating Connectivity article.

**Flat or Hierarchical Design**

As mentioned, as the designer you need to decide how the schematic sheets are organized, and how
the connectivity is established between those sheets. These are not separate decisions though, as you choose the structure you will also need to choose how the connections between those sheets are going to be created.

There are two approaches to structuring a multi-sheet design: either flat or hierarchical.

The technique used to connect a child sheet to the parent sheet is the same for both flat and hierarchical designs - it is how the connectivity is created that determines if it is a flat or hierarchical design.

**Flat Design**

You can think of a flat design as if a large schematic sheet has been cut up into a number of smaller sheets - in a flat design all sheets exist on the same level. The connectivity in a flat design is created directly from any sheet to any another sheet - this type of connectivity is referred to as horizontal connectivity.

The use of a top sheet is optional in a flat design. If one is included, it will have a sheet symbol for each of the sheets in the design, but cannot include any wiring. There can be any number of sheets in a flat design.
Both of the images above show a flat design, the version on the left does not have a top sheet, the version on the right does. For a small design which only has two or three schematic sheets in it, you might decide that a top sheet does not add any value. Once the sheet count gets higher, a top sheet can help the reader understand the functionality of the circuit design from the way that the logical blocks (Sheet Symbols) are arranged on the sheet. All sheets in the design appear at the same level in the Projects panel, because there is no hierarchy.

Hierarchical Design

An hierarchical design is one which has Sheet Symbols to create the parent-child type relationships between the sheets, and, the connectivity is through the Sheet Entries in those Sheet Symbols - not directly from the Ports on one sheet to the Ports on another sheet.

As in a flat design, the child sheet is identified by defining its filename in the sheet symbol. In an hierarchical design that child sheet can also include sheet symbols, referencing lower-level sheets, thus creating another level in the hierarchy. The image below shows an hierarchical design, with 3 levels in the hierarchy.
In an hierarchical design, the structure shown in the tree is determined by the parent-to-child relationships created by the sheet symbols. These relationships are worked out and updated whenever the design is compiled.

In an hierarchical design, a signal on a child sheet leaves the sheet via a Port, which connects upward to a matching Sheet Entry on the parent sheet. The parent sheet includes wiring that carries that signal across to a Sheet Entry in another Sheet Symbol, it then travels down to a matching Port on the second child sheet, as shown in the image below.
This parent-child sheet structure can be defined to any depth, and there can be any number of sheets in an hierarchical design.

The connectivity between the sheets is determined by the **Net Identifier Scope**. This is set in the **Options** tab of the **Options for Project** dialog. To learn more about creating connectivity, read the **Creating Connectivity** article. Note that the Net Identifier Scope includes an Automatic option, unless you have unusual connectivity requirements, this option is a good choice.

Another advantage of hierarchical design is that it provides the platform for a delivery of a sophisticated design reuse system. This system is delivered in 2 ways, depending on how the data is stored, either: file-based or Altium Vault-based.

- The file-based system is called Device Sheets, where you place an existing schematic from a library of Device Sheets directly into the design being created. To learn more about Device Sheets, refer to the **Device Sheets** article.
- The Vault-based system is called Managed Sheets, where you place an existing schematic from an Altium Vault, directly into the design being created. To learn more about Managed Sheets, refer to the **Managed Sheets** article.

**Creating a Multi-Sheet Design**

As soon as you add a second schematic sheet to your project, you've created a multi-sheet design. If you are planning on creating a flat design without a top sheet, you simply keep adding schematic
sheets to the project, and confirm that the Net Identifier Scope is set correctly.

If you want to use sheet symbols to reference lower-level sheets, you can either place the sheet symbol and manually edit it to correctly reference to the lower-level sheet, or you can use the various built-in commands to help, as described below.

**Referencing the Child Sheet**

It is the **Filename** property of the Sheet Symbol that references the lower-level sheet. Note that this field should only include the schematic filename, not the path to that file's location (this location data is actually stored in the Project file).

![Sheet Symbol](image)

The **Filename** property links this sheet symbol to the child schematic sheet.

Note that the child schematic does not have to be stored in the same folder as the parent schematic. If it is stored in a folder below the parent's storage folder then relative file referencing is used, if it is stored in another location then absolute file referencing is used - take care when relocating the files in such a project, or use the Project Packager to ZIP the files, and resolve file paths as it does.

**Creating Hierarchy**

The software includes a number of commands that allow you to build your multi-document, hierarchical structure quickly and efficiently. The commands you use will depend on your personal design methodology - which can be broadly classified as top-down or bottom-up. These commands will be more efficient than creating the hierarchy manually, as they handle all the elements that are needed in the the process, such as adding Sheet Entries, creating new schematic sheets, placing Ports, and so on.
Use this command to build the hierarchy in a top-down fashion:

- **Design » Create Sheet From Sheet Symbol** – use this command to create a new schematic sheet below the nominated sheet symbol. Ports are added to the child sheet to match any Sheet Entries found in the Sheet Symbol. Don't worry if you have not included all of the Sheet Entries in the Sheet Symbol yet, if more are added over time you can re-synchronize the Sheet Entries and Ports, as discussed below.

Use this command to build the hierarchy in a bottom-up fashion:

- **Design » Create Sheet Symbol From Sheet or HDL** – use this command to create a symbol from the nominated schematic sheet (or VHDL or Verilog file). To use this command, first switch to the sheet that will hold the new Sheet Symbol, then launch the command. The Sheet Symbol will include a Sheet Entry to match each Port it finds. If Ports or Sheet Entries are added or removed at a later stage they can be re-synchronized, as discussed below.

Use this command to reorganize how the circuitry is placed in the design:

- **Edit » Refactor » Move Selected Sub-circuit to Different Sheet** – use this command to move the selected components and wiring to a different sheet in the project. The *Choose Destination Document* dialog will open, allowing you to choose any existing schematic in the project. This command is described in more detail below.

In case you are wondering why there are dedicated commands for moving components from one sheet to another, these are provided because the standard Cut & Copy commands automatically reset the Unique Identifier in each component.

Restructuring the Design

The process of design is often unstructured and organic, the designer could be formulating ideas for multiple parts of the design at the same time, capturing sections as their ideas evolve. That means that what started out as a well organized, neatly laid out set of schematics can become crowded and poorly organized. While you can Cut, Copy and Paste to reorganize the schematic design, this is not always the best approach. Why? Because as each component is placed it is assigned a unique identifier, and this identifier is automatically reset whenever a component is Cut/Copied and Pasted. This UID management is done to ensure that there is only one instance of each UID used in the design, as it is the key field that links the schematic component to the PCB component. The Cut/Copy/Paste approach is fine if the design has not been transferred to the PCB editor, but if it has, then it is better to use the refactor tools.

Moving a Sub-circuit to Another Sheet

The easiest way to move a section of circuitry from one sheet to another is to select it, then run the **Edit » Refactor » Move Selected Sub-circuit to Different Sheet** command (also available via the right-click menu when there is a selection). The *Choose Destination Document* dialog will open, after you select the target sheet and click **OK** that sheet will appear, with the sub-circuit floating on the cursor, ready to position.
A selected section of circuitry can easily be moved to a different sheet in the project using the right-click `Move Selected Subcircuit to Different Sheet` command.

**Synchronizing the Ports and Sheet Entries**

If you have moved components and wiring as part of restructuring the design, then you may also need to re-synchronize the child sheet to its Sheet Symbol, so that each Port has a matching Sheet Entry. This is done using the **Synchronize Sheet Entries and Ports** command, which is available in the:

- **Design » Synchronize Sheet Entries and Ports menu** - choose this command to analyze all of the Sheet Symbols in the entire design, the dialog will include a tab for each Sheet Symbol in the design, in accordance with the **Only Show unmatched sheet symbols** option at the bottom of the dialog.
- **Right click » Sheet Symbol Actions » Synchronize Sheet Entries and Ports** - choose this command to analyze only the Sheet Symbol under the cursor.

For both of these commands, the **Synchronize Ports to Sheet Entries dialog** will open. It will list the already matched Ports/Sheet Entries on the right side of the dialog, with the unmatched Ports and Sheet Entries listed in two columns on the left side of the dialog.

The **Synchronize Ports to Sheet Entries** dialog is used to ensure that the Sheet Entries match with the Ports on the child sheet.
Note the two tabs, that means there are two Sheet Symbols that have Sheet Entry / Port mismatches in this design.

Focusing on the mismatches displayed on the left of the dialog, the idea is to select the Sheet Entry in the first column, then the correct Port in the second column, then click the correct button to update one of them so they are synchronized (and move to the list on the right of the dialog). The buttons function as follows:

- ![Sheet Entry to Port](image) - use the Sheet Entry properties, and push them to the selected Port.
- ![Port to Sheet Entry](image) - use the Port properties, and push them to the selected Sheet Entry.

If multiple Sheet Entries are selected in the left-hand column, the software will synchronize each Sheet Entry with the adjacent Port in the second column. If there is no adjacent Port (or Sheet Entry), a new one is created.

If the tool adds Sheet Entries or Ports, they will need to be correctly positioned on the schematic sheet when it has finished.

---

**Multi-Channel Design**

It is not unusual for an electronic design to included repeated sections of circuitry. It might be a stereo amplifier, or it might be a 64 channel mixing desk. This type of design is fully supported in Altium Designer, by a feature-set known as *multi-channel design*.

The two decoder channels of a video multiplexer, using a multi-channel design structure means the decoder channel is only captured once.

In a multi-channel design you capture the repeated circuit once, then instruct the software to repeat it the required number of times. When the design is compiled it is expanded out in memory, with all components and connectivity repeated the required number of times, in accordance with a user-defined channel naming scheme.
The logical design that you capture is never actually flattened, the source always remains as a multi-channel schematic. When you transfer it to PCB layout, the physical components and nets are automatically stepped out the required number of times. You have full access to the standard cross-probing and cross-selecting tools available for working between the schematic and the PCB. There is also a tool in the PCB editor to replicate the placement and routing of one channel across all other channels, with the ability to easily move and re-orient an entire channel.

**The Unified Data Model**

To understand multi-channel design, it helps to understand how the design data is managed. A fundamental element of the software is the Unified Data Model (UDM). When the project is compiled, a single, cohesive model is created, which sits central to the design process. Data within the model can then be accessed and manipulated using the various editors and services within the software. Rather than using a separate data store for each of the various design domains, the UDM is structured to accommodate all information from all aspects of the design, including the components and their connectivity.

The Unified Data Model makes all of the design data available to all of the editors, and helps deliver sophisticated features like multi-channel design.

The UDM, in combination with the hierarchical design system, is leveraged to deliver the multi-channel design capabilities. The “channel” is the circuitry within a Sheet Symbol - below this Sheet Symbol can be a single sheet, or it can be an entire branch of the project structure, containing other sub-sheets. You can also create channels within channels, in a 2-level multi-channel design the upper channels are referred to as banks, with the lower level ones referred to as channels.

Because this full, PCB-ready description exists in memory it is then possible to repeat a section of circuitry, as long as there is a systematic way to handle repeated objects, such as component designators and nets. The systematic naming is defined in the Multi-Channel tab of the Options for Project dialog, as discussed below.

**Creating a multi-channel design**

A design is multi-channel when a section of the circuitry is repeated. This is done at the sheet symbol level, either by:

- placing multiple Sheet Symbols that all reference the same child schematic, or
- configuring a single Sheet Symbol to repeat the referenced child schematic the required number of times
These two approaches to creating a multi-channel design are shown in the images above. In the image on the left there are four sheet symbols that all reference the same child sheet (PortIO.SchDoc). In the image on the right, the InputChannel.SchDoc is repeated 8 times and the OutputChannel.SchDoc is repeated 2 times, by the presence of the Repeat keyword in the Designator field of the Sheet Symbol.

When the project is compiled, repeated component designators and net names are resolved by applying a standard naming scheme. For example, one naming scheme is to add a channel index to the repeated components and net identifiers (Net Labels and Ports), as shown in the highlights in the images below.

Note the tabs that appear across the bottom of the schematic when the project is compiled. The Editor tab contains the logical design as you have captured it, the other tabs (COUT1 and COUT2) appear once the project is compiled, they contain the physical design that will be transferred to the PCB editor. There will be a tab for each physical channel.
The compiled project, showing the tab for the logical view, and a tab for each channel. Note how the repeated designators and net identifiers are managed.

The channel naming scheme is defined in the Multi-Channel tab of the Options for Project dialog, press F1 over the dialog for more information.

The Repeat Keyword

As mentioned, a channel is repeated by placing multiple sheet symbols that reference the same child sheet, or by including the Repeat keyword in the Sheet Symbol's Designator field.

The Repeat statement defines the channel designator and the number of channels. Note how the Sheet Symbol is drawn as a set of stacked symbols, to signify repeated channels.
Each channel is identified by a channel designator, which comes from the Sheet Symbol \textbf{Designator}. When the design is channelized by placing multiple Sheet Symbols, the channel designator is the \textbf{Designator} value defined for each Sheet Symbol. If the design is channelized by the use of the Repeat keyword, the channel designator is the ChannelIdentifier+ChannelIndex defined by the Repeat keyword.

The syntax of the Designator field when the Repeat keyword is used, is as follows:

\begin{verbatim}
Repeat(<ChannelIdentifier>,<ChannelIndex_1>,<LastChannelIndex_n>)
\end{verbatim}

Note that the index range must start at 1, starting at 0 (zero) is not supported. When the Repeat keyword is used, the Sheet Symbol is drawn as a set of stacked Sheet Symbols.

\textbf{Multi-Channel Naming}

The concept of being able to capture once and then repeat - multi-channel design - is delivered by building on the software’s unified data model (UDM). Repeated components are named using a systematic naming scheme, which is configured in the \textbf{Multi-Channel tab} of the \textit{Options for Project} dialog, as shown below.

The dialog includes an upper section used to control the naming of the Rooms, and a lower section used to control the naming of the components within those Rooms. At the Room level, there are 2 flat naming styles and 3 hierarchical naming styles, typically you would only need to choose an hierarchical naming style if the design has channels within channels. Otherwise, a flat Room naming style is shorter and easier to understand.

For the component naming, the \texttt{$Component$ChannelAlpha} or the \texttt{$Component_$ChannelIndex} option will give the shortest, and most easily interpreted component designation. It is also possible to construct your own designator naming scheme, using the \textbf{available keywords}.

Repeated components are managed by applying a systematic naming scheme, chosen in the Multi-Channel tab of the Options for Project.
The Role of the Room

A Room is a PCB design object used to define an area on the board, which can then be used in two ways:

1. **Contain objects** - while a room is placed like any polygonal shape object, it is actually created as a Placement design rule. Part of a Room's definition is to specify the objects that must be contained within that room, which is often components. When that room is moved, all the components within the room move too.

2. **To scope other design rules** - as well as being a rule in their own right, rooms can also be used to scope other design rules. For example, one routing width design rule defines the routing width for a class of nets, then a higher-priority design rule, with a scope of Room, can specify a different width to be applied to that class of nets, within that Room.

Rooms work very well with multi-channel design. They can be created automatically as the design is transferred from the schematic editor to the PCB editor, based on options in the Class Generation tab of the Options for Project dialog, with a room for each Sheet Symbol. As well as clustering the components in that channel, the room can then be used in the naming of the components within that room. Rooms and their role in the board design process are discussed further in the Multi-Channel PCB Design section of this article.

If you prefer a flat component numbering system, it is possible to replace the systematic naming scheme by performing a Board Level Annotation. It is called Board Level Annotation because the component designators are only applied to the full, compiled design (physical design) that is destined to become the PCB.

Connectivity in a Multi-Channel Design

For a multi-channel design, set the Net Identifier Scope to Automatic, Hierarchical or Strict Hierarchical. A multi-channel design must be hierarchical because the software uses this structural model to instantiate the channels in memory when the design is compiled.

There are two different connectivity requirements that the software must support for a net connecting to a repeated channel, the net will either be:

- common to all channels, or
- unique in each channel.

The level of support for this depends on which method has been used to define the channels (multiple Sheet Symbols, or via the Repeat keyword). If a design uses multiple Sheet Symbols (an individual Sheet Symbol for each channel), then the net connectivity is explicit, as defined by the wiring placed by the designer.
This design uses the Repeat keyword to create the multiple channels.

If the design uses the Repeat keyword, then the following connectivity conventions apply.

**Passing a Net to all Channels**

If a net needs to be available in all channels it is simply wired to a Sheet Entry, as shown for the Monitor net connected to the InputChannel.SchDoc Sheet Symbol in the image above. The nets Effects, MB1 and MB2 will also be available on all of the channels. Nets in a bus are handled the same way, when a bus is connected to a Sheet Entry then each element in that bus is available on all of the channels.

**Passing a net to a Specific Channel**

To allocate a single net from a bus to each channel, the Repeat keyword is used in the Sheet Entry, as is shown for the Headphone bus in the image above, where the Sheet Entry has the name Repeat (Headphone). In this case, the net Headphone1 will connect to the channel CIN1, Headphone2 will connect to channel CIN2, and so on. It is not possible to pass an individual net to just one channel if the Repeat keyword has been used to create the channels. If this is required, you must place an individual Sheet Symbol for each channel.

**How the Nets are Named**

Ultimately, each net can only have one name on the PCB (one PCB net cannot have two names), and each net name should only be used once on the PCB (you should not have two different PCB nets with the same name). The software automatically resolves nets with multiple names to have just a single name when the project is compiled - it is important that you configure the naming options in a multi-channel design to ensure that your nets are labelled in a way that is meaningful to you. The net naming options are in the Netlist Options section of the Options tab of the Options for Project dialog.

A good approach to setting these options in a multi-channel design is to enable the Higher Level Names Take Priority option, and to also place Net Labels on all nets that connect to a channelized child sheet.
As an example, consider the images below. Note that both of the Sheet Symbols point to the sheet PCB_Decoder.SchDoc, so there are 2 channels of this circuit, identified as U_PCB_DecoderA and U_PCB_DecoderB.

The Sheet Symbol includes a Sheet Entry called TDI, which is the Test Data In line in a JTAG boundary scanning chain. TDO (Test Data Out) from DecoderA then connects to the TDI of the next device in the chain, which is in the DecoderB channel.

The two decoder channels are created by placing 2 Sheet Symbols, that both reference the same schematic, PCB_Decoder.SchDoc.

When the project is compiled, the channel tabs U_PCB_DecoderA and U_PCB_DecoderB appear at the bottom of the schematic view, these tabs show the physical design as it will be passed to the PCB editor. In the images below, the schematic for the decoder circuit as it was captured, is shown in the Editor tab on the left below, followed by the two physical channels for that schematic, U_PCB_DecoderA and U_PCB_DecoderB.

In the original captured schematic, the net was labelled TDI by the designer (first image). Note how the software has applied the higher-level net name, TDO_CONTROLLER, in the DecoderA tab (second image), as it is a higher level net name (which has been set to have priority in this design). For DecoderB there is no higher-level name defined, so the original net name, TDI, has been identified in this channel as TDI_2 (third image) because that is the naming scheme defined in Multi-Channel tab of the Options for Project dialog (the Component Naming scheme is used to identify both components and nets within each channel).
The PCB_Decoder.SchDoc schematic, the captured schematic is on the left, the compiled view of the two channels in the middle and on the right.

**Displaying the Compiled Names**

Tracing and analyzing the nets in a multi-channel design can be confusing, as the names have to change to identify nets that are repeated, but continue to be unique. To help with this, there are a number of options to control the display of compiled object names, including component Designators, Net Labels and Ports. There are also options for Sheet and Document Numbers, these will be important when you are ready to generate print-type output.

The display of compiled object names is configured in the [Schematic - Compiler](#) page of the Preferences dialog, and are shown in the image below.

Typically you will want the designators and Net Labels displayed, Ports are handy if you are diagnosing an issue. If the Superscript option is enabled, the current document view will include the object identifier from the non-visible view as a superscript. Configure these to suit your preferences.

Are you wondering why some of the schematic elements are dimmed in the channel (physical) tabs? The default is to show the objects that can be edited in the channel tab in the standard display strength, with non-editable objects dimmed. The dimming is set in the Schematic - Compiler page of the Preferences dialog. While it might seem frustrating to dim objects, it helps prevent you from trying to perform edit-actions that cannot be
supported, such as moving a Net Label in one channel. Set the Dim level as required.

### Resolving Multiple Net Name Errors

The schematic editor's error checking options default to flag every instance of a net with multiple names. This can occur when you intentionally change the name, for example when a net enters a sheet symbol and you prefer to use a different name within that sheet. The image below shows a number of examples of this, where the Left and Right outputs are bundled into a bus on the top sheet, but are called Left and Right in the Sheet Entries.

A bus has been used as it allows the two output channels to be represented by a single sheet symbol with a Repeat keyword. If separate wires were used for the Left and Right channels then the designer would have needed to place separate sheet symbols for the Left and Right output channels (both pointing to the same child schematic sheet), and then wire each output to its respective output sheet symbol.

You must instruct the software how to deal with multiple net identifiers. To do this, you will need to either:

- Set the **Nets with multiple names** error check to **No Report** in the **Error Reporting** tab of the **Options for Project** dialog. This is not the preferred option, as it blocks all checking of this error condition throughout the design. Alternatively, you can,
- Place a **Specific No ERC marker** on the effected nets, which is most easily done by right-clicking on an Error/Warning in the **Messages** panel and selecting the **Place Specific No ERC Marker for this violation** command. Doing this will put you into NoERC object placement mode, with a pre-configured Specific NoERC marker attached to the cursor, ready to place on the net in error. After placing, double-click to configure the style and color.
Note the small triangle circled in orange, this is a specific No ERC marker that has been placed to disable the Duplicate Net Names errors on the MB1 and MB2 nets.

Multi-Channel PCB Design

When you transfer a design from the schematic editor to the PCB editor, the components from each sheet are clustered in a PCB placement room if Room creation is enabled in the Project Options.

The big advantage of using rooms in a multi-channel design is that the PCB editor supports duplicating the placement and routing from one room (channel), to the other rooms (channels). Rooms can also be moved as if they are a single object, simplifying the process of arranging the channels on the PCB.
The Role of the PCB Room

A **Room** is a PCB design object used to define an area on the board, which can then be used in two ways:

1. **Contain objects** - while a room is placed like any polygonal shape object, it is actually created as a Placement design rule. Part of a Room's definition is to specify the objects that must be contained within that room - which is often components. When that room is moved, all the components within the room move too.
2. **To scope other design rules** - as well as being a rule in their own right, rooms can also be used to scope other design rules. For example, one routing width design rule defines the routing width for a class of nets, then a higher-priority design rule, with a scope of Room, can specify a different width to be applied to that class of nets, within that Room.

Rooms work very well with multi-channel design. They can be created automatically as the design is transferred from the schematic editor to the PCB editor, based on options in the [Class Generation tab](#) of the [Options for Project](#) dialog, with a room for each Sheet Symbol. As well as clustering the components in that channel, the room can then be used in the naming of the components within that room. Rooms and their role in the board design process are discussed further in the [Multi-Channel PCB Design](#) section of this article.

**Useful Room Manipulation Commands**

- **Design » Rooms** - this sub-menu has a number of useful commands for defining and
modifying rooms.

- **Design » Rooms » Copy Room Formats** - use this command to replicate the placement and routing of one room (channel), to the other rooms (channels)
- **Design » Rooms » Move Room** - a room can be moved by clicking and holding, anywhere where there is no other design object under the cursor. The cursor will snap to the nearest component pad, or room vertex (whichever is closer). Use this command when you cannot click and hold without clicking on a design object.
- **Edit » Select » Room Connections** - use this command to select all copper segments that start on a pad and end within the room.

### How the Components are Linked

Each schematic component links to its PCB component through a Unique Identifier (UID). The UID is assigned when the schematic component is placed on the sheet, and is then assigned to the PCB component when the design is transferred to the PCB editor. This scheme would be adequate for a simple design, but is not capable of supporting a multi-channel design, where the same schematic component is repeated in each physical channel (so the PCB components would have the same UID).

To cater for this, the UID for the PCB component is created by combining the UID of the parent Sheet Symbol with the UID of the schematic component. The syntax of the PCB UID changes slightly, depending on how the multi-channel design has been created.

For a multi-channel design created by placing multiple Sheet Symbols that all reference the same schematic sheet, then each Sheet Symbol can provide a unique ID, so the PCB UID has the format:

```
\SheetSymbolUID\SchComponentUID
```

For a multi-channel design created using the Repeat keyword, there is only 1 Sheet Symbol UID available, so the PCB UID also includes the ChannelIndex, in the format:

```
\ChannelIndex+SheetSymbolUID\SchComponentUID
```
Component links are managed via the Edit Component Links dialog (Project » Component Links command in the PCB editor).

**Displaying the designators on the PCB**

It can be difficult positioning the designator strings in a multi-channel design, as they can be quite long. As well as choosing a naming option that results in a short name, another option is to display just the original, logical component designation instead. For example, C30_CIN1 would display as C30. This would require some other notation being added to the board to indicate the separate channels, such as a box being drawn around each channel on the component overlay.

You can select between Logical and Physical designator display on the PCB in the Board Options dialog (Design » Board Options). If you choose to display the logical designators for components in a multi-channel design, these will be displayed on the PCB and in any output generated, such as prints and Gerber's. The unique physical designators, however, are always used when generating a Bill of Materials.

- If you prefer a flat component numbering system, it is possible to replace the systematic naming scheme by performing a Board Level Annotation. It is called Board Level Annotation because the component designators are only applied to the full, compiled design (physical design) that is destined to become the PCB.

  These Board Level Annotations are stored in a .Annotation file, which maps each logical designator to the assigned physical designator. This file is part of the project so you will be prompted to save it.

**Parametric Multi-Channel Hierarchical Design**

The challenge with reusing a section of design, for example linking a Sheet Symbol on your current project to your company's preferred power supply schematic, is that the values of the components are not always fixed from one design to the next.

Parametric Hierarchical Design solves this - it allows you to move the specification of the component values from the schematic sheet, into the sheet symbol that references that sheet. This capability also works perfectly with multi-channel design, allowing you to have different component values in each channel. Note that it does require that each channel has its own Sheet Symbol, as this is where the component values are stored.

For example, a graphic equalizer can have the same circuit repeated many times, with the only difference between each channel being the component values. So a capacitor might take the values 0.12µF, 0.056µF, and 0.033µF in the different channels. Implementing this is simple, since you specify these values in the Sheet Symbol referencing each channel, eliminating the need to have many similar schematics with only the component values being different.

Parametric components are defined by declaring their value as a parameter of the sheet symbol.
above, and then referencing that parameter on the target component. The image below shows the
top sheet for a graphic equalizer on the left, with the parameters of the 1KHz Sheet Symbol next to it.
The image also shows the lower-level captured schematic, and the compiled 1KHz channel.

A graphic equalizer with different capacitor and resistor values in each channel, the actual component values are defined in the Sheet Symbols, so only 1 lower-level schematic needs to be captured.

Each of those Sheet Symbol parameters is also defined as the value of a parameter in a schematic component on a lower-level sheet, as shown in the image below. When the project is compiled, the Value of each Sheet Symbol parameter is passed to the relevant schematic component, where it is then mapped into the component's Comment field.

Because the Value parameter has a value of =C2_Value, the compiler knows that it must look for the actual component value in a higher-level Sheet Symbol.

Parametric hierarchy is not limited to component values, you can parametrically reference any component parameter or any text label on the schematic sheet. You can also refer to parameters from a symbol that is many sheets up in the hierarchy, the system will search the hierarchy until it finds the matching parameter.

Download the reference design | Parametric Hierarchy\AudioEqualizer.PrjPcb for an example of how to correctly configure the Sheet Symbol and Component parameters.