Altium Designer introduces a robust multi-channel design system that even supports channels nested within other channels. Many designs contain repeated circuitry. One board might duplicate the same section thirty-two times, or perhaps contain four identical banks with eight sub-channels each. Designers have long endured the difficulties of "flattening" such designs at the schematic level to correlate perfectly with the PCB layout. While the initial chore of copying-and-pasting schematic sections is relatively easy, correcting or updating the schematic project afterwards can quickly escalate into a heavy burden. Altium Designer offers true multi-channel design, meaning that you can reference single sheets in your project repeatedly. Required changes need only be made once; recompiling the project then propagates the changes through each instantiation. And Altium Designer not only supports multiple channels, it allows them to be nested.

Consider the schematic representation of a sixteen-button keypad shown in Figure 1.

![Figure 1. A typical keypad circuit, consisting of only two different components.](image)

This design can be captured and wired up in Altium Designer's schematic editor in just a couple of minutes. But suppose that you wanted to use a different symbol for the button? What if you wanted to add a parameter to each one? Or what if you wanted to expand the design from four by four to ten by twenty?

Even with the help of group editing tools, these tasks are tedious and error-prone. Suppose, however, that you represented the entire circuit with just two components, one resistor and one switch, each on its own schematic sheet. A Repeat statement made within a sheet symbol instructs Altium Designer's compiler to create virtual clones (channels) from a single sheet, as shown in the sheet symbol in Figure 3. The resistor sheet could refer to the same switch sheet four times to create a row, then a top sheet could refer to the resistor's sheet four times, making four rows. Multiple channels are nested within multiple channels, and the matrix of connectivity is established accordingly.

Now the designer's tasks are made easy. Any modifications to the properties of the switches or
resistors in this design would only need to be made once, instead of over and over again. Any expansion of the connection grid may be done by simply modifying the Repeat statements within the sheet symbols.

Getting up to speed with multi-channel design means first learning how Altium Designer establishes connectivity across such projects.

**Multi-Channel Connectivity**

Unlike other multi-sheet design projects, you have no choice when it comes to your Net Identifier scope; you must use the Hierarchical (sheet symbol <-> port) method. Leaving this setting at Automatic in the Project Options dialog will be fine, but it supposes that you understand that the only way to communicate signals to a repeated sheet is from sheet entries on the parent sheet down to matching ports on the child sheet. The reason for this restriction is that sheet entries, unlike ports or net labels, have been designed to handle the channel instantiations created through Repeat statements.

There are two types of sheet-to-sheet connectivity available in multi-channel design: nets that are common to all channels, and nets that are unique to each. Both of these types are demonstrated in our keypad example above. The repeated switch in each row has one pin that connects directly to the resistor. This is a common net, meaning that the same node in each channel is tied to a single node on the parent sheet, and therefore to one another. Visually, it could be shown as in Figure 2. This is the simplest method of net distribution in Altium Designer's multi-channel design. It is made intuitively by matching the port name on the channel sheet with the name of the sheet entry above. The other node on the switch, however, is not common on the switch level, there is a unique net from each switch, which is then connected to the corresponding switch in each row. To convey a unique connection up from each switch in the row, a Repeat statement is included on the sheet entry. Figure 3 demonstrates both types of net connectivity.

The Repeat keyword in sheet symbol's Designator field has three parameters; SheetSymbolDesignator, FirstInstance, and LastInstance. It is important to note that the FirstInstance parameter needs to be indexed at a value equal or greater than 1. For example, Repeat(Row,1,4)
Notice the difference between the two sheet entries. The first one (Res) matches the name of its corresponding port exactly, the other (Col) applies a Repeat command to its corresponding port’s name. This syntax means that there are four nets in the bus Col[1..4], with one of these bus nets going to each of the four Switch sheets.

This allows four individual nets to be extracted from the four repeated sheets, first by a labeled wire, then by a labeled bus, as shown in Figure 3. At that point, the signals are free to be connected to separate nodes as necessary, or (as in this case), to be sent further up the hierarchical ladder through ports.

**Tip:**

In order to use Multi-Channel Design it is necessary to add the respective Ports to the sheets and sheet entries to the sheet symbols.

---

**Figure 4.** The top sheet in the design, this time the Repeat statement instantiates the entire row, 4 times.

Once you have built up your multi-channel project in accordance with Altium Designer’s connectivity structure, you may turn your attention to the way names are assigned to channels and their components.

**Channel Designations**

Channels are virtual sheets, and so Altium Designer's synchronizer will infer a rooms from each channel, just as it would for any other sheet. Before it can do so, however, the compiler must generate unique names for all channels and for all of their constituent components. These names are driven through the Multi-Channel tab in the Project Options dialog, where an interactive diagram shows how your choices will affect component and room designations if you are using nested channels.

Basically, you have two choices to make. The first is whether to use a flattened or a hierarchical naming convention for your rooms and components. (Hierarchical here refers to the hierarchy of channels, so this decision only matters if your design contains nested channels.) The second is whether the instantiation index should be numeric or alphabetic. (One option actually allows for both,
but, once again, this only becomes apparent in the naming of nested channels.)
The format you specify for your component designators may include the channel/room name you have stylized, but it doesn't have to. Usually, your designator format should include both a prefix and a suffix reference for both components and channels. Two of the seven reserved keywords do this automatically: $Component and $RoomName.
$Component is the concatenated form of $ComponentPrefix $ComponentIndex. For example, a component part designated U3A consists of a prefix (U) and an index (3A). Similarly, $RoomName is also a concatenated form of a channel prefix and suffix, but its exact form will vary depending upon your design and the room naming style you have chosen. If your design does not include nested channels, then the hierarchical portion of the style is irrelevant, and only the numeric or alphabetic portion of the style will alter how $RoomName describes your channel notations. But if you do have nested channels in your design, and you use a hierarchical room naming style, $RoomName cannot simply be replaced by $ChannelPrefix and either $ChannelAlpha or $ChannelIndex. This is because it builds a special, multi-level prefix from the channel names used in the Repeat statements in your design.

Figure 5. The 16 switches, copied and pasted into a single image to show their designations.

Suppose that, in our keyboard project, we had selected Numeric Name Path in the Room Naming Style, a comma for the Level Separator, and for the Designator Format we had simply written $RoomName. The sixteen instantiated buttons after compiling the design would be annotated as shown in Figure 5.

Had we chosen instead to use the Alpha Name Path, our designators would have been RowA,PBA; RowA,PBB, etc. Had we chosen the Mixed Name Path, they would have been RowA,PB1; RowA,PB2, and so on. But if we had chosen a flat alternative, the reference to the Row channel would be ignored, and the designators would simply range between PB1 and PB16, or PBA and PBP (numeric or alphabetic). Of course, it is not usual practice to leave out any reference to the component itself in the Designator Format field, and is only acceptable in this case because each channel contains just one component.

**Displaying the designators on the PCB**

It can be difficult positioning the designator strings in a multi-channel design, as they can end up being 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 of course necessitate 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.

**Parametric multi-channel design**

Certain multi-channel designs have repeated circuitry but require that specific components have a different value in each channel. This capability is supported, it is referred to as parametric multi-channel design. Parametric components are defined by declaring their value as a parameter of the parent sheet symbol, and then referencing that sheet symbol parameter in a component parameter, on the target component on the channel schematic, as shown in Figure 6.

![Image of parametric component definition](image)

Figure 6. Specify the value of the parametric component as a parameter of the parent sheet symbol (as shown in the Sheet Symbol dialog to the right), then reference that parameter via a parameter in the target component (as shown in the component dialog above).

Note that this means that you cannot use the single sheet symbol with a Repeat statement approach to define the multi-channel structure, instead you must place a sheet symbol for each instance of the channel (where all of these sheet symbols reference the same channel schematic sheet). The parametric component definition is then specified in each of these sheet symbols.

**Room for PCB Designers**

Multi-channel design benefits everybody, not just schematic designers. The benefits to the PCB designer extend far beyond the initial grouping of channeled components into their respective rooms alongside the Board Outline. Powerful new room-based tools will be particularly appreciated during multi-channel design layout.

Altium Designer’s Situs autorouter has a single-room setting, whereby it can route all connections wholly contained within a room. Conversely, the unrouting tool has a room option. Corresponding components in each channel are all assigned the same Channel Offset value, meaning that they may be targeted together for selection or group edits.

More impressive than all these, however, is the ability to copy formatting from one room to another. This allows you to make changes to the placement of components in one room, then copy the room's
formatting to the other rooms. If the source and destination rooms belong to the same channel class (meaning that they were cloned from the same schematic strand), then the updates can be extended to all class members at once. What information you copy can be limited to the room's size and shape, or can be extended to footprint arrangement and even routing. Additional controls let you determine how far the routes previously present in such room will be ripped up.

Numerous other tools have been added to the PCB editor which will allow board designers to get the most out of multi-channel design. While using the Move command in the Design » Rooms submenu, for instance, you may press the L key, which flips whatever is being moved to the opposite side of the board. You will find that the whole room, including relative placement and routing, remains intact after the move.

The ultimate promise of multi-channel design is the same for the board designer as it is for the schematic engineer: you no longer have to spend your time do the same thing twice - or thirty two times, for that matter.

**What to do next**

To learn more about the benefits of multi-channel and nested multi-channel designs, study the example projects in the \Examples\Reference Designs\ folder of your installation, including the:

- LedMatrixDisplay
- Multi-Channel Mixer
- Peak Detector
- PortSwitcher

Refer to the tutorial, [Creating a Multi-channel Design](https://techdocs.altium.com/display/ADOH1/Multi-Channel+Design+Concepts) for step-by-step instructions on building a multi-channel design.

Refer to the article, [Connectivity and Multi-Sheet Design](https://techdocs.altium.com/display/ADOH1/Multi-Channel+Design+Concepts) for more information on complex designs.

**Source URL:** https://techdocs.altium.com/display/ADOH1/Multi-Channel+Design+Concepts