Nesting Parent-Child Configurators

Configurators can help build complex products. But to create even more complex assemblies or represent modular products, you can connect many small configurators together

.Objectives

- Referenced Configurators
- Connecting and displaying configurators
- Controlling the reference at run-time
- Sharing information between referenced configurators

Referenced Configurators

A configurator can have many pages and many fields. Nothing prevents you from building one very large configurator to describe your custom product.

But such a structure can lead to problems.

Consider an example building configurator, where customers can choose the purpose of the building, and the type of roof it has.

It's nice you can build 6 different kinds of buildings. But what if your user wants not just one floor, but 2 floors? Or… two hundred?

What if they want each of these multiple floors to have a different purpose?

Using one configurator to try to manage such a dynamic product becomes difficult to maintain.

Rather than use one configurator, try decomposing your product into a collection of smaller, simple configurators that can talk to each other. Then, these configurators can be recombined in many ways.

Think of it like a configurator nested within a configurator. Or a parent configurator controlling a child configurator.

Each configurator remains a stand-alone entity: you should be able to launch a child configurator on its own without the parent. But when that child is referenced by a parent, then...

- A parent configurator can read or write to the child configurator.
- A child configurator can be used by many different parent configurators.
- And multiple instances of a child can be used by the same parent.

That's the general concept. Let's get into the specifics.

Creating the reference and defining its appearance

Follow these steps to create a reference between two configurators.

First, open the configurator which will be the parent. This configurator would have the responsibility of creating the child configurators, sending them information, and gathering data back from them.

In the parent configurator, select the “referenced configurators” entry in the design tree. Click the add button to create a new reference.

Then edit the new reference. In the “Configurator Type” property, specify which configurator can be a child for this parent.

You can select other components of this child to nest into the parent as well.

You can nest the child’s scene and scene hotspots in the parent’s scene, to build complex 3D scenes easily.

You can nest the child’s price object, so those elements are automatically added to the parent’s.

Depending on your integrations, you may see other elements from the child which you can nest into the parent.

In most cases, you can start with the default values provided.

The relationship between parent and child is almost done. All that remains is defining how the child will appear within the parent’s UI.

Will your user be adding child configurators manually? Or will your Snap rules manage the children automatically?

If you want your user to add the children manually, specify where they can do so in the “User can add” parameter.

Usually, each child will appear as a new set of pages in the parent’s UI.

If you specify the parent configurator name, you can have the child appear as a high-level page, not indented and easily visible.

Best practice is to have child pages nested under a parent page. This simplifies the UI, makes information easier for your user to find and understand, and streamlines your Snap rule code.

Maybe you don’t want your user adding children manually. Maybe it’s a fixed number of children, or maybe business rules control when children can be added or removed. To prevent your user from managing children, and make it completely automatic, leave this field blank.

Finally, you should know that your UI is not limited to children appearing as full pages. Even though this is most common, you may not want to show the entire child configurator’s UI in the parent. You may want to show only a portion of it.

After confirming your referenced configurators work well, read more about Nested Sets in the documentation.

Nested Sets allow a child configurator to appear in many other ways. Such as an expander in an accordion. Or even as a container within another container to give a table-like interface. It’s up to you to design your own optional Nested Sets, or just use the flexible default of pages.

Controlling the reference at run-time

So you’ve defined the parent-child relationship. How can your customers, or your Snap rules, use this relationship to make children appear?

If you selected places where the “user can add” children, then when the parent configurator is launched, these locations will have some new buttons.

First, an “add” button will be visible. If your user clicks this, an instance of the referenced child will be added.

If any page of the child is clicked, you’ll see some other added buttons to help your user clone or delete that specific child.

If you did not select a place where the “user can add” children, then your user can’t manage the children. Rather, you need to create Snap rules to manage the child configurators based on logic. Blocks with the word “nested” inside them help manage child configurators: you can see them all in the toolbox under “configurator… nested”.

Here are some examples of Snap blocks that can add a child…

Add a child and override its defaults…

Add many children…

And remove a child.

A common place to use these blocks is in a value rule in the parent configurator. Consider our building configurator example. The parent configurator has a number field called “floor count”.

The value rule shown here takes this number, and creates (or removes) child configurators to match it. Don’t worry: the “add nested” block is not destructive. If a child configurator already exists with that same name, it is left alone.

Since this code is in a value rule, your user simply has to change the number field on the parent, and the correct number of child configurators appear.

With these two techniques, you can have your user manually manage some child configurators, or you can have your Snap rules automatically manage the child configurators, based on business logic.

Sharing information between referenced configurators

When a parent configurator references another configurator as its child, it can send information back and forth to that child.

Remember, the parent configurator defined the relationship to the child. Therefore, it’s the parent’s job to send and receive information.

Let’s see how it’s done.

After creating a reference to a child configurator, and specifying how that child will appear, you can also apply Snap logic to the relationship. A parent can read or write to its child in 3 special events, and in the rule cycle.

First, each child configurator reference has optional events which you can use for Snap code.

Click the add symbol next to any child configurator reference, and you’ll see 3 events. You can have Snap code run when this child is added… when it’s removed… or when it’s copied.

These events aren't the only place to write Snap code for a referenced configurator. You can also add code to most rules as well.

Remember the rule cycle, which runs through a series of rules in a specific order whenever any field is edited.

When a child reference is added to a parent, this rule cycle expands to include all children. That means if any field is edited in the parent… or any child… then the complete rule cycle runs across all of them, as you would expect.

Here's an example from our building configurator of a parent reading information from its children. We're looking at a value rule in the parent.

In this value rule, the parent is looping through the array of child configurators stored under a specific page. It finds the height of each child floor, adds them all up, and then writes the resulting total height into its own Building Height field.

In this way, the parent building configurator doesn't have to know how tall it is. It simply asks all the children for their heights, and uses the sum.

The interesting Snap blocks here are the "get nested" block, which returns an array of nested configurators. And the "From nested… get" block, which can take a reference to a nested configurator and get any UI element within it, such as a field. Here, we get the field's value.

To learn more about these nested snap blocks, as well as how to create a fully-functioning parent-child relationship, see the self-directed exercise on "referenced configurators" accompanying this course.

In summary, Snap blocks are available for you to gather any info from any child… and change the attributes of any child. To your user, it looks and behaves as one big configurator.

Recap

In this course, you have learned the concept of referenced configurators.

You discovered how to connect configurators together, and control how your user will see them.

You've seen how both your customers, and your rules, can control when referenced configurators appear.

And you've examined how Snap code can move information between them.

About Epicor

Epicor Software Corporation drives business growth. We provide flexible, industry-specific software that is designed around the needs of our manufacturing, distribution, retail, and service industry customers. More than 40 years of experience with our customers' unique business processes and operational requirements is built into every solution—in the cloud, hosted, or on premises. With a deep understanding of your industry, Epicor solutions spur growth while managing complexity. The result is powerful solutions that free your resources so you can grow your business. For more information, connect with Epicor or visit www.epicor.com.

EPICOR

Corporate Office
804 Las Cimas Parkway
Austin, TX 78746
USA
Toll Free: +1.888.448.2636
Direct: +1.512.328.2300
Fax: +1.512.278.5590

Latin America and Caribbean
Blvd. Antonio L. Rodriguez #1882 Int. 104
Plaza Central, Col. Santa Maria
Monterrey, Nuevo Leon, CP 64650
Mexico
Phone: +52.81.1551.7100
Fax: +52.81.1551.7117

Europe, Middle East and Africa
No. 1 The Arena
Downshire Way
Bracknell, Berkshire RG12 1PU
United Kingdom
Phone: +44.1344.468468
Fax: +44.1344.468010

Asia
238A Thomson Road #23-06
Novena Square Tower A
Singapore 307684
Singapore
Phone: +65.6333.8121
Fax: +65.6333.8131

Australia and New Zealand
Suite 2 Level 8,
100 Pacific Highway
North Sydney, NSW 2060
Australia
Phone: +61.2.9927.6200
Fax: +61.2.9927.6298