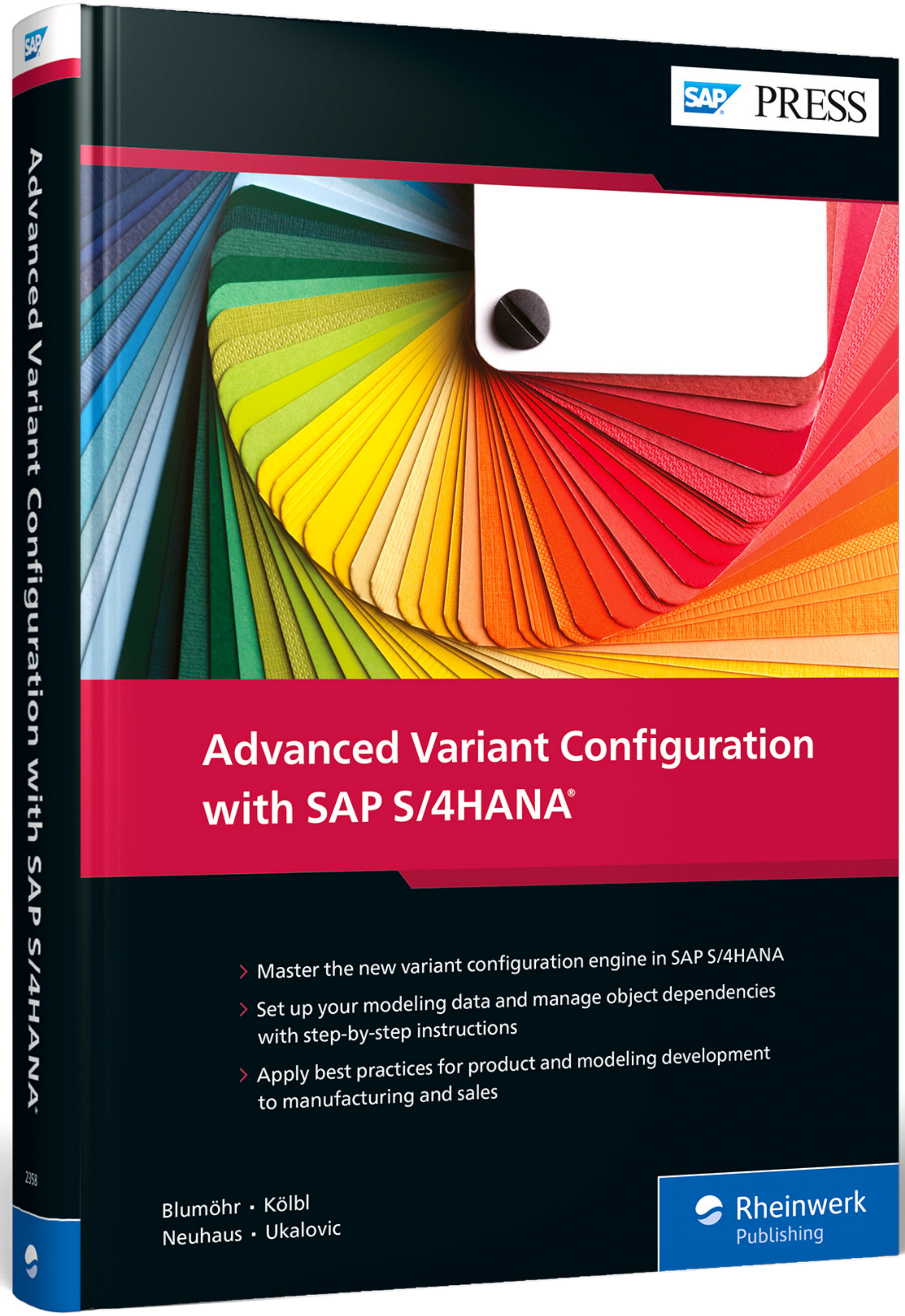

SAP PRESS
Advanced Variant Configuration with SAP S/4HANA
› Master the new variant configuration engine in SAP S/4HANA
› Set up your modeling data and manage object dependencies with step-by-step instructions
› Apply best practices for product and modeling development to manufacturing and sales
Blumöhr · Kölbl
Neuhaus · Ukalovic
Rheinwerk Publishing
Advanced Variant Configuration with SAP S/4HANA

Reading Sample

This sample chapter provides an introduction to creating your first configurable material. Using a short example, the chapter provided some early insights into working with configurable materials as well as an overview of essential modeling steps in variant configuration. Specific topics covered include the material master, variant classes, the valuation interface, and object dependencies.

Blumöhr, Köbl, Neuhaus, Ukalovic

Advanced Variant Configuration with SAP S/4HANA

567 pages | 10/2023 | $99.95 | ISBN 978-1-4932-2358-9

www.sap-press.com/5628

Chapter 3
Creating Your First Configurable Material

Before diving more deeply into modeling in Chapter 4, let's whet your appetite in this brief chapter with an overview of creating a configurable material.

Variant configuration is a tool that enables you to map products offered in a large number of different variants within an SAP system. Think about buying a new car. This example illustrates two typical features of variant configuration:

- Most of the time, you can't find your dream car at the dealer right away. Often, so many variants of the desired vehicle model exist in terms of shape and equipment that the manufacturer does not initiate production until a specific sales order has been received. Production without a sales order and thus sale from stock is also possible in the variant configuration; in this case, we call these *material variants*.
- When you buy a car, you generally don't choose a material number from a catalog; instead, you pick a model first. According to the catalog, the model is available in numerous variants. Wherever variability is offered in the catalog, the desired variant is defined via what are called *properties*.

Section 3.1 first explains the basic principles that enable you to classify the steps in the subsequent sections. Section 3.2 presents an example, and its modeling steps are covered in the remaining sections. Each model in the variant configuration starts with the configurable material master (Section 3.3).

For the valuation interface in variant configuration, we'll need characteristics in the material master and a configuration profile. This master data is created in Section 3.4, while Section 3.5 covers the valuation interface where you can check the result of these first steps.

To have the appropriate master data available in planning and production, work must still be carried out on the bill of materials (BOM) and routing, as described in Section 3.6. Finally, in Section 3.7, we'll show you how pricing is adjusted.

3.1 Basic Principles

The diagram shown in Figure 3.1 approaches the components of and steps for modeling. Since you cannot maintain separate master data for each variant, you can group all variants under one *material number*. Thus, you can use a configurable material master for an enormous number of variants. For a product like the car mentioned earlier, this diversity of variants is not yet extreme. If you multiply the possibilities of the sales catalogs of any manufacturer, you'll derive a variant number between millions and trillions, that is, between 10^6 and 10^{12}. However, some models, for instance, from the field of wind power or plant engineering, might result in a significantly larger number of variants, up to 10^{24} (septillions).

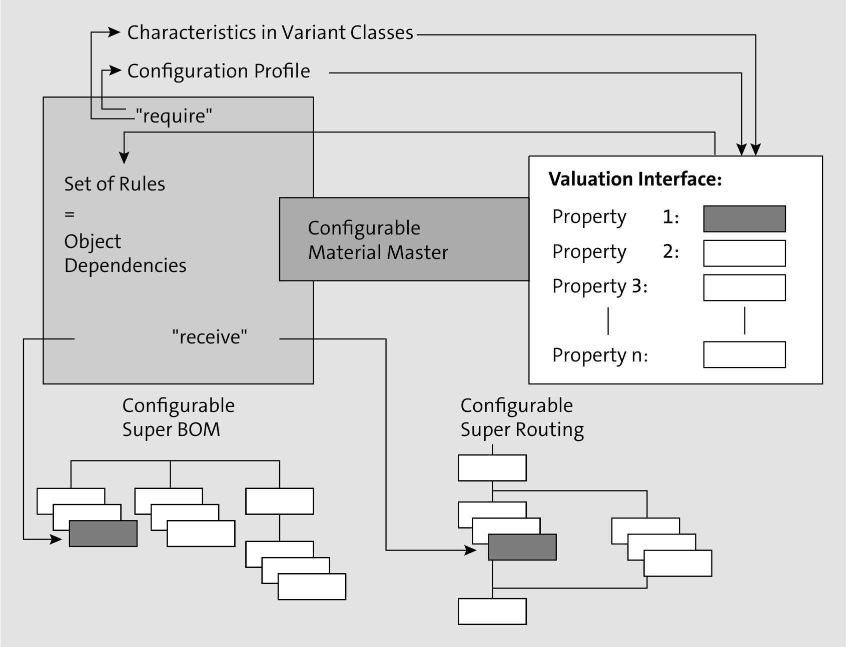


Figure 3.1 Simplified Model of Variant Configuration

At the center of the modeling process in an SAP system is the *configurable material master*, as shown in Figure 1.1. A material master is configurable if you select the **Material is configurable** flag under the **Basic data 2** tab.

You can assign a *valuation interface* to a configurable material master. This valuation interface can then be used to query your customer's request regarding possible properties. This step is referred to as *configuring*. With regard to our car example, these properties would be related to engine, transmission, paint color, heaters, seats, etc.

To obtain a valuation interface, you'll define characteristics, variant classes, and a configuration profile.

The properties in the valuation interface are mapped via *characteristics*. To assign these characteristics to the material master, they are first collected into special classes, called *variant classes*. These variant classes are linked to the material master.

In addition, a *configuration profile* is required. The configuration profile contains important control settings and object dependencies for the configuration process. Without characteristics in variant classes and the configuration profile, you have no valuation interface and thus no possibility for configuration.

With the valuation interface created in this way, you could already configure a product, for example, in a sales order. However, to map not only the sale but also the production of the product, you'll need a BOM and a routing for the production order.

Since you are using only one material master for the entire scope of variability, you cannot create separate BOMs and routings for each variant. For this reason, you'll create a configurable *super BOM* and a *configurable super routing* for the one configurable material, as shown in Figure 1.1.

The BOM structure can have multiple levels; that is, you can contain BOMs within BOMs. The entire manufacturing process can also consist of multiple levels. Thus, the production of assemblies is carried out in advance via your own production orders before the actual production of the product ordered by the customer takes place. Thus, in addition to multiple BOMs, multiple routings can exist as well.

"Super" means that the BOM and routing must include everything that may be needed at some point. With regard to our example from the automotive industry, the super BOM contains, among other things, all the engines with which the car could be optionally equipped. You can delete or change elements from the super BOM or super routing via object dependencies, but you cannot add them via object dependencies.

"Configurable" therefore means that you can assign object dependencies and, by doing so, control the explosion of the BOM and routing, depending on the configuration.

But what is meant by "explosion"? We'll need a special BOM and a special routing for a special configuration. A configured BOM is generated from the configurable super BOM and the individual components are thus removed or modified. The same applies to the routing.

BOM explosion and routing explosion are controlled by *object dependencies*. Object dependencies are therefore the set of rules that guide the configuration (that is, the value assignment of the characteristics).

In our example, if you've chosen one of the six engines offered, the five engines not required for your production order must be removed from the super BOM, which must include all six engines. All six BOM items with the engines therefore each have an object dependency, or more precisely, a selection condition: "Leave the engine in the exploded BOM only if exactly this engine is selected." Thus, five of the six selection conditions are not met, and these five engines disappear from the exploded BOM for the production order, and only the desired engine remains.

Object dependencies are also used to control interactive configurations. For this case, you can assign object dependencies to characteristics (and their values) and to the configuration profile. This object dependency on the configuration profile and characteristics (characteristic values) is effective while you're entering your requirements in the sales order. The object dependency for the "desire" is therefore different from the object dependency for the BOM and routing that controls the "getting," both in terms of types and processing times.

"Object dependency for the desire" also means that you must not combine your wishes arbitrarily. For example, you cannot order a convertible with a sunroof, and not every engine can be combined with every transmission. You can also force valuations: The luxury version must include the chrome package. Or you can force a valuation sequence: First, make the version selectable and then the engine.

These options illustrate the model of variant configuration. We have omitted some aspects to more simply present the interrelationships. For example, pricing or allowing special requests might go beyond what is offered in the catalog. But enough theory! Let's now illustrate all these basic concepts with a simple example.

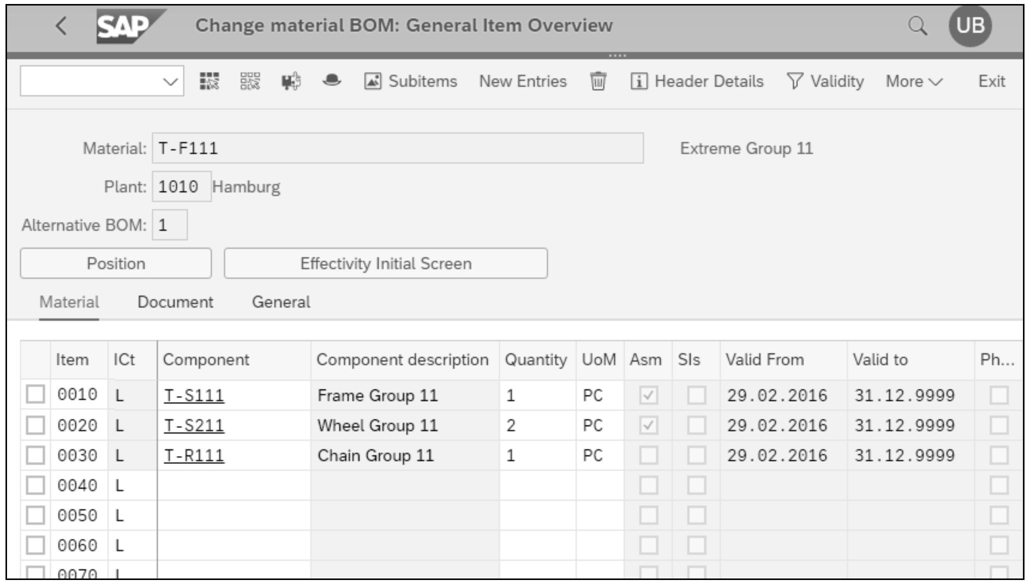
3.2 A First Example

To prevent our example from becoming too extensive, we won't start from square one. Our starting point is a non-configurable product, the "Extreme Group 11" bicycle.

Figure 3.2 shows the BOM for this bicycle, in the transaction for changing material BOMs. The three components that make up the bike—the frame, wheel, and chain—should suffice for now. With prior knowledge, you might see immediately that neither the material nor the BOM is configurable, which is evident from the fact that object dependencies are missing. Between the **UoM** (unit of measurement) and **Valid From** columns, only two columns exist, namely, the **Asm** (assembly) flag and the **SIs** (subitems) flag. The **OD** (object dependencies) flag for configurable materials and configurable BOMs is missing.

In this context, compare this screen with the screen shown later in Figure 3.5, in which the **OD** flag is visible. Furthermore, the **Class** tab (also shown later in Figure 3.5) is also missing for our non-configurable product, which only has the **Material**, **Document**, and **General** tabs.

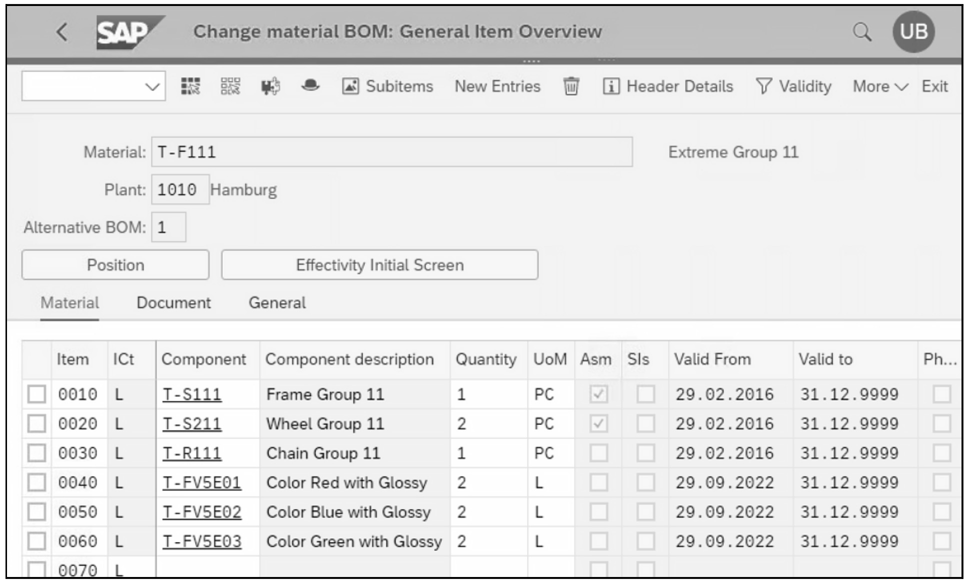
In the future, this bike will be offered in three colors—red, blue, and green. For this purpose, three new items with corresponding material numbers for the three colors are added to the BOM, as shown in Figure 3.3. (Refer to Chapter 4 for more detailed information on BOM maintenance.)



Material: T-F111 Extreme Group 11
Plant: 1010 Hamburg
Alternative BOM: 1

	Item	ICt	Component	Component description	Quantity	UoM	Asm	SIs	Valid From	Valid to	Ph...
☐	0010	L	T-S111	Frame Group 11	1	PC	☑	☐	29.02.2016	31.12.9999	☐
☐	0020	L	T-S211	Wheel Group 11	2	PC	☑	☐	29.02.2016	31.12.9999	☐
☐	0030	L	T-R111	Chain Group 11	1	PC	☐	☐	29.02.2016	31.12.9999	☐
☐	0040	L					☐	☐			☐
☐	0050	L					☐	☐			☐
☐	0060	L					☐	☐			☐
☐	0070	L					☐	☐			☐

Figure 3.2 Non-Configurable Starting Point for Our Example



Material: T-F111 Extreme Group 11
Plant: 1010 Hamburg
Alternative BOM: 1

	Item	ICt	Component	Component description	Quantity	UoM	Asm	SIs	Valid From	Valid to	Ph...
☐	0010	L	T-S111	Frame Group 11	1	PC	☑	☐	29.02.2016	31.12.9999	☐
☐	0020	L	T-S211	Wheel Group 11	2	PC	☑	☐	29.02.2016	31.12.9999	☐
☐	0030	L	T-R111	Chain Group 11	1	PC	☐	☐	29.02.2016	31.12.9999	☐
☐	0040	L	T-FV5E01	Color Red with Glossy	2	L	☐	☐	29.09.2022	31.12.9999	☐
☐	0050	L	T-FV5E02	Color Blue with Glossy	2	L	☐	☐	29.09.2022	31.12.9999	☐
☐	0060	L	T-FV5E03	Color Green with Glossy	2	L	☐	☐	29.09.2022	31.12.9999	☐
☐	0070	L					☐	☐			☐

Figure 3.3 Bike Available in Three Colors

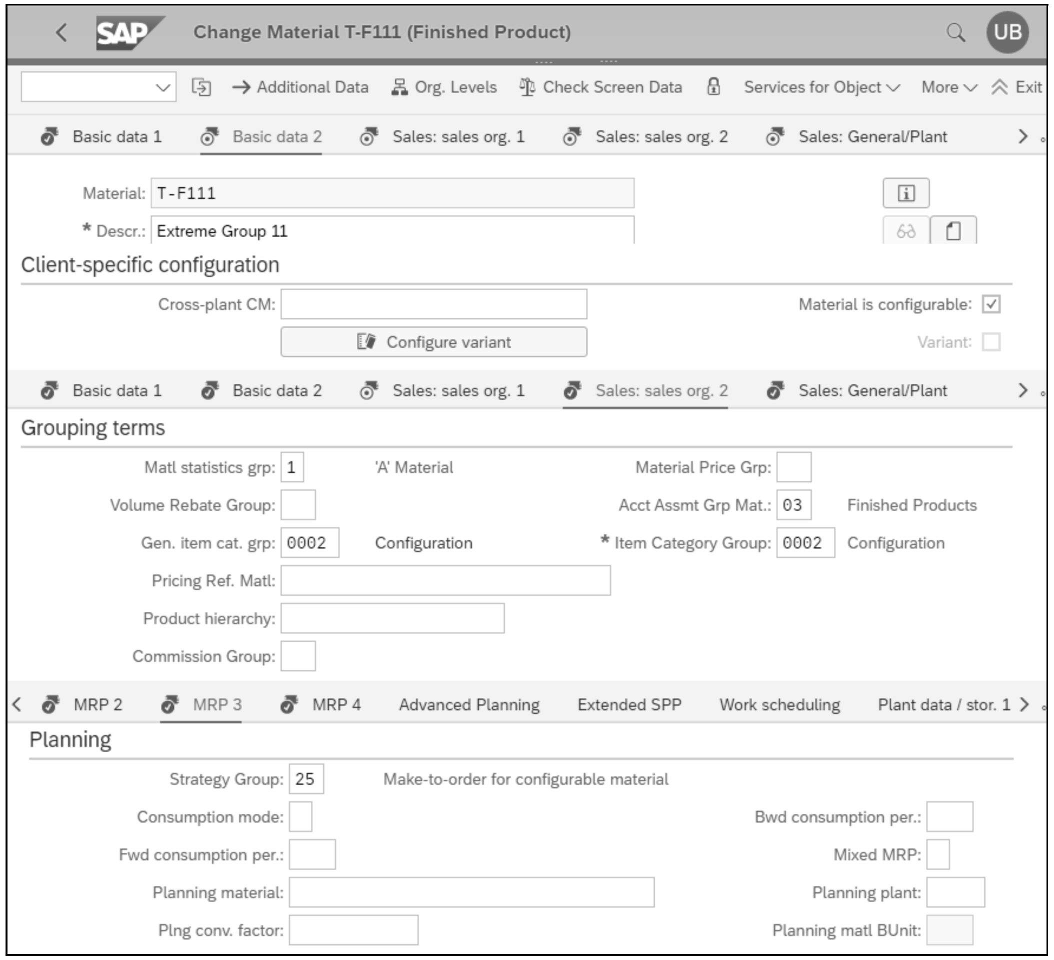
We'll fix the "issue" of the non-configurable BOM in the next section. We'll ensure that the bike and thus the material master of the bike are configurable.

3.3 The Configurable Material Master

The configurability of the BOM is controlled in the *material master*. If the material master of the BOM header material is configurable, the BOM is also configurable.

Since in our example the material master already exists, we only need to make it configurable.

In our example, we've set the **Material is configurable** flag under the **Basic data 2** tab in the material master (through the maintenance Transaction MM02), as shown in Figure 3.4.



Material: T-F111

* Descr.: Extreme Group 11

Client-specific configuration

Cross-plant CM:

Material is configurable: ☑

Variant: □

Grouping terms

Matl statistics grp: 1 'A' Material

Material Price Grp:

Volume Rebate Group:

Acct Assmt Grp Mat.: 03 Finished Products

Gen. item cat. grp: 0002 Configuration

* Item Category Group: 0002 Configuration

Pricing Ref. Matl:

Product hierarchy:

Commission Group:

Planning

Strategy Group: 25 Make-to-order for configurable material

Consumption mode:

Bwd consumption per.:

Fwd consumption per.:

Mixed MRP:

Planning material:

Planning plant:

Plng conv. factor:

Planning matl BUnit:

Figure 3.4 Configurable Material Master

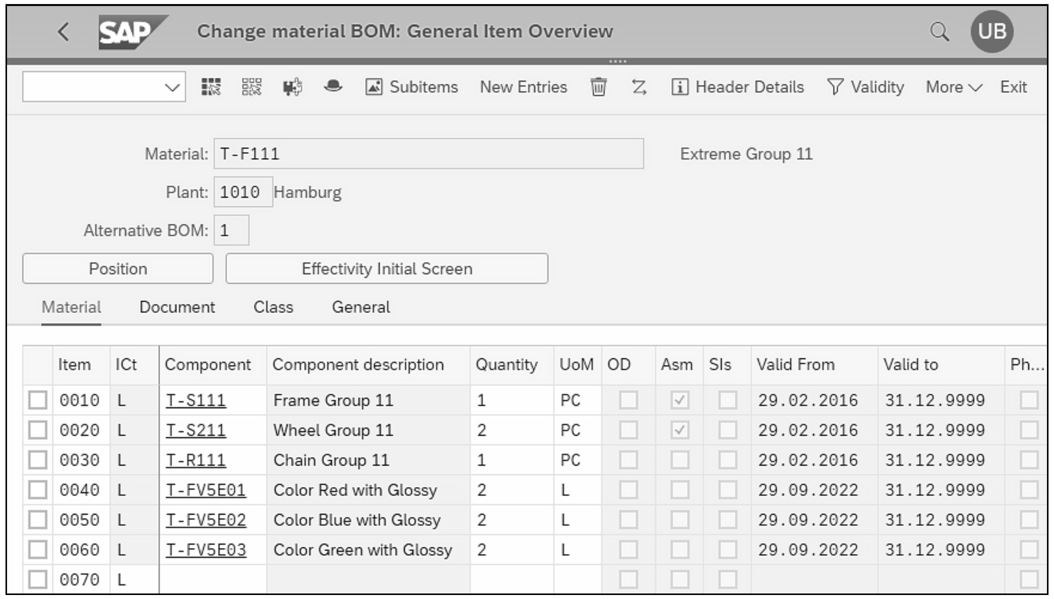
Do Not Do This!

We strongly advise against tampering with the **Material is configurable** flag. You can set the flag afterwards. However, the flag cannot be deselected later. Despite urgent advice against it, we'll set the flag now anyway to keep this example manageable and short.

Two other changes are required in material master maintenance: Under the **Sales: sales org. 2** tab, you must first ensure that the sales document recognizes an item for this material as configurable and is controlled accordingly. In a standard system, you would use item category group 0002.

Moreover, under the **MRP 3** tab, the **Strategy Group** field (here, 25) controls that the sales documents should generate corresponding requirements that take the configuration and make-to-order (MTO) production into account.

If you look at the BOM again after changing the material master, the BOM is now also configurable, as shown in Figure 3.5. You'll see the **OD** column for object dependencies and the **Class** tab. Recall that object dependencies are used to turn a configurable super BOM into a configured BOM according to a customer's requirements, that is, according to their configuration. The **Class** tab allows you to use *class nodes*, which you'll learn more about in Chapter 4.



Material: T-F111 Extreme Group 11

Plant: 1010 Hamburg

Alternative BOM: 1

	Item	ICt	Component	Component description	Quantity	UoM	OD	Asm	SIs	Valid From	Valid to	Ph...
□	0010	L	T-S111	Frame Group 11	1	PC	□	☑	□	29.02.2016	31.12.9999	□
□	0020	L	T-S211	Wheel Group 11	2	PC	□	☑	□	29.02.2016	31.12.9999	□
□	0030	L	T-R111	Chain Group 11	1	PC	□	□	□	29.02.2016	31.12.9999	□
□	0040	L	T-FV5E01	Color Red with Glossy	2	L	□	□	□	29.09.2022	31.12.9999	□
□	0050	L	T-FV5E02	Color Blue with Glossy	2	L	□	□	□	29.09.2022	31.12.9999	□
□	0060	L	T-FV5E03	Color Green with Glossy	2	L	□	□	□	29.09.2022	31.12.9999	□
□	0070	L					□	□	□			□

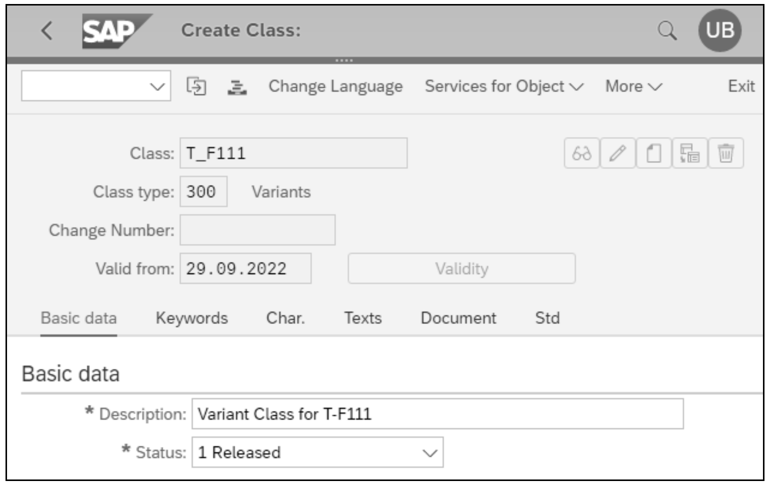
Figure 3.5 Configurable BOM

3.4 Variant Classes, Characteristics, and Configuration Profiles

Next, you need a valuation interface. Take another look at the screen shown in Figure 3.1—it's about creating the box to the right of the material master box.

Your customers should be able to choose one of the three colors in which the product will be offered. As described in Section 3.1, you'll need a *variant class* with *characteristics* and a *configuration profile* to create the desired valuation interface.

A variant class can be created via the Manage Class app in SAP Fiori or via Transaction CL02 (Class Management), as shown in Figure 3.6.



Class: T_F111

Class type: 300 Variants

Change Number:

Valid from: 29.09.2022

* Description: Variant Class for T-F111

* Status: 1 Released

Figure 3.6 Creating a Variant Class

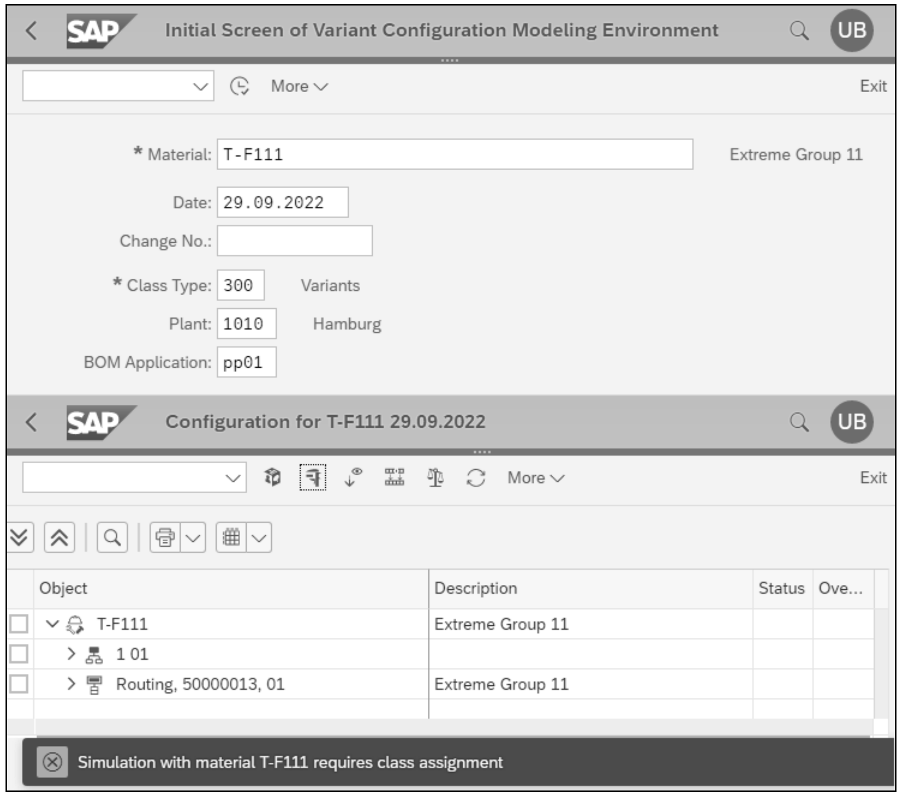
For our purposes, we'll simply give the variant class a name ("T_F111"), assign a class type (300), and provide a description (here "Variant class for T-F111"). The class also needs a status, but you can set up the system to suggest statuses.

Creating a variant class is not enough to create a valuation interface. For the next steps, we use the VC modeling environment, which can be called via Transaction PMEVC (which stands for the *Product Modelling Environment Variant Configuration*) or in the VC Modeling Environment app in SAP Fiori.

As shown in Figure 3.7, you can start this VC modeling environment with at least a material number and a class type. To load the BOM and the routing as well, you must also maintain the **Plant** field ("1010") and the **BOM Application** field ("PP01") on the initial screen.

The bottom half of Figure 3.7 shows our current model, which consists of three objects at this time:

- Material number **T-F111**
- BOM **1 01**
- **Routing 50000013, 01**


SAP
Initial Screen of Variant Configuration Modeling Environment
UB
More
Exit

* Material: T-F111 Extreme Group 11

Date: 29.09.2022

Change No.:

* Class Type: 300 Variants

Plant: 1010 Hamburg

BOM Application: pp01

SAP
Configuration for T-F111 29.09.2022
UB
More
Exit

Object	Description	Status	Ove...
T-F111	Extreme Group 11		
1 01			
Routing, 50000013, 01	Extreme Group 11		

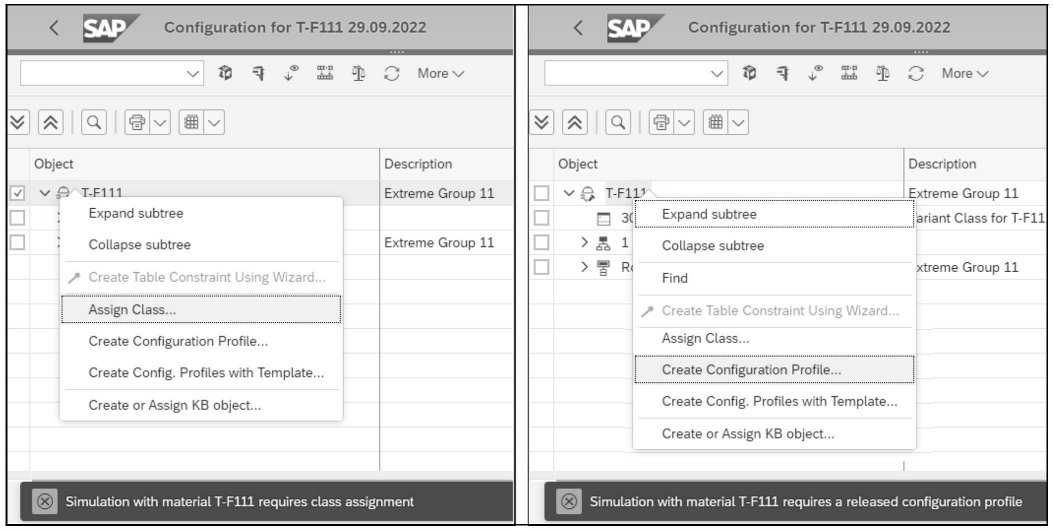
Simulation with material T-F111 requires class assignment

Figure 3.7 VC Modeling Environment: First Attempt

The VC modeling environment also includes a simulation capability. Access this feature by clicking the (**Test**) icon. The left side of Figure 3.8 shows an error message regarding the missing class assignment. In our example, the "Simulation with material T-F111 requires class assignment" error message appears. The variant class has been created but is not yet assigned to a configurable material master.

You can solve this problem by assigning the class to material number T-F111 via the context menu by right-clicking on the first line; in our case, we'll assign the `T_F111` class with class type 300.

Our next attempt to test the evaluation interface, by clicking the (**Test**) icon again, causes a new error message: "Simulation with material T-F111 requires a released configuration profile," as shown in the bottom-right of Figure 3.8. Now, you can also create a configuration profile via the context menu by right-clicking on the line with the material number.

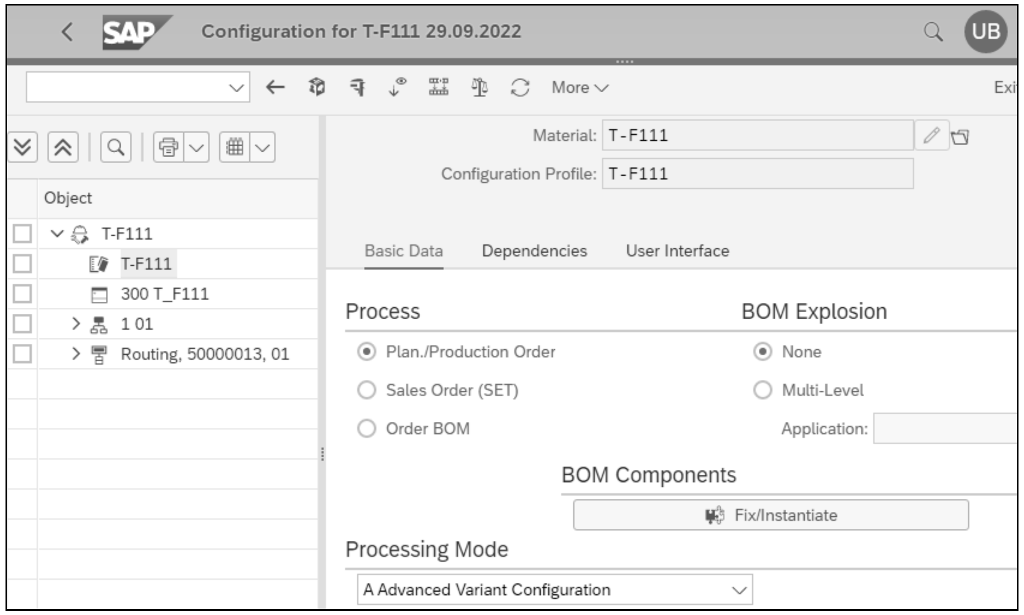

Configuration for T-F111 29.09.2022
More
Object
Description
T-F111
Extreme Group 11
Extreme Group 11
Expand subtree
Collapse subtree
Create Table Constraint Using Wizard...
Assign Class...
Create Configuration Profile...
Create Config. Profiles with Template...
Create or Assign KB object...
Configuration for T-F111 29.09.2022
More
Object
Description
T-F111
Extreme Group 11
ariant Class for T-F11
xtreme Group 11
Expand subtree
Collapse subtree
Find
Create Table Constraint Using Wizard...
Assign Class...
Create Configuration Profile...
Create Config. Profiles with Template...
Create or Assign KB object...

Simulation with material T-F111 requires class assignment

Simulation with material T-F111 requires a released configuration profile

Figure 3.8 VC Modeling Environment: Second Attempt, Assigning a Class and Creating a Configuration Profile

You can skip the **Create configuration profile** popup window by pressing `Enter`. Then, you'll see the view shown in Figure 3.9. Select the **A Advanced Variant Configuration** option from the **Processing Mode** dropdown list. Aside from that step, the default settings are sufficient for our purposes.



Material: T-F111

Configuration Profile: T-F111

Process: ● Plan./Production Order ○ Sales Order (SET) ○ Order BOM

BOM Explosion: ● None ○ Multi-Level

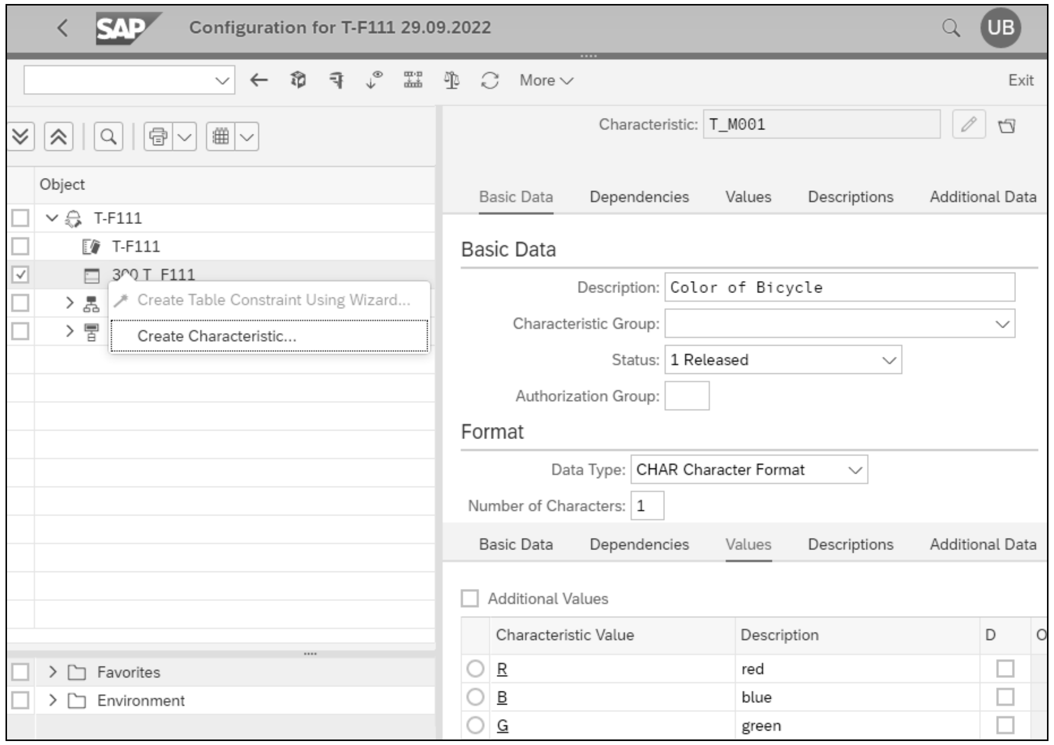
Application:

Processing Mode: A Advanced Variant Configuration

Figure 3.9 VC Modeling Environment: Configuration Profile

The settings we've made so far still don't enable a customer to select a color because a characteristic is missing. The left side of Figure 3.10 shows how you can create characteristics in the VC modeling environment via the context menu on the line of the class ("300 T_F111").

The right side of Figure 3.10 shows the required characteristic. For this purpose, a name was assigned in the top line in the **Characteristic** field ("T_M001"). Furthermore, the characteristic requires a language-dependent description in the **Description** field as well as a status.



Characteristic: T_M001

Description: Color of Bicycle

Characteristic Group:

Status: 1 Released

Authorization Group:

Data Type: CHAR Character Format

Number of Characters: 1

□ Additional Values

	Characteristic Value	Description	D
○	R	red	□
○	B	blue	□
○	G	green	□

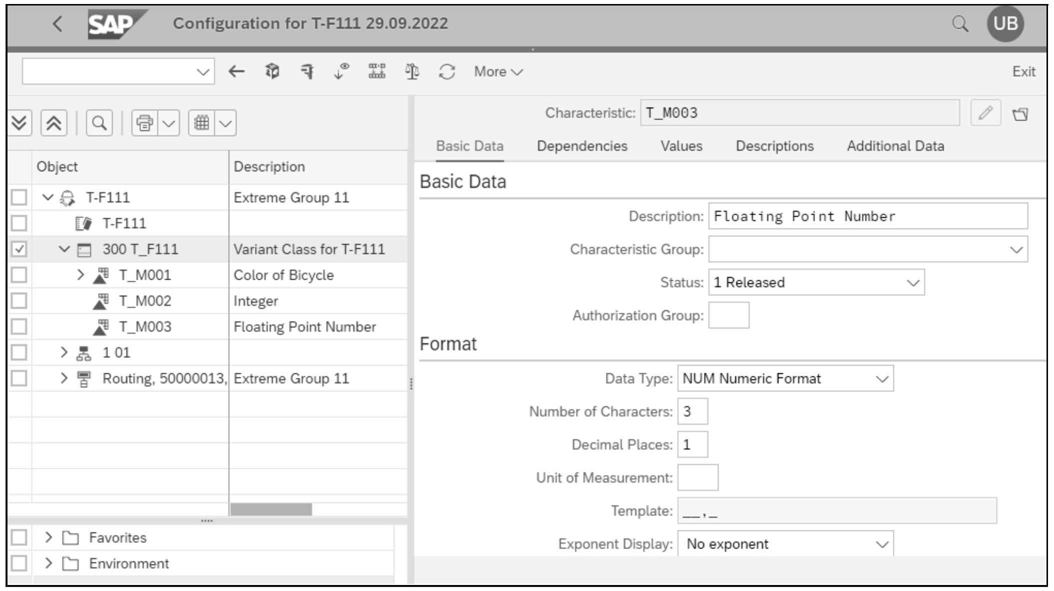
Figure 3.10 VC Modeling Environment: First Characteristic

The status can be assigned automatically via a default value (**1 Released**). Each characteristic must have a data type. The data type controls what kind of values the characteristic can have. If numeric values are not required, select the **CHAR Character Format** from the **Data Type** dropdown list. This character format requires specifying the number of digits in the **Number of Characters** field. This specification determines the number of digits of the characteristic value.

Besides the **CHAR Character Format** data type option, the **NUM Numeric Format** data type is also important in variant configuration. The number of digits must also be specified in this case.

Furthermore, you can assign language-dependent descriptions to characteristic values in the **Description** column (for instance, “red,” “blue,” and “green”).

As shown in Figure 3.11, we’ve created and assigned two additional characteristics for our example. Strictly speaking, these characteristics are not required since the characteristic for the color would have sufficed. The T_M002 and T_M003 characteristics have also been given names, descriptions, statuses, and data types. In this example, the numeric format was used as the data type. These two characteristics were not necessary here; we just wanted to show that the numeric format also exists.



Object	Description
T-F111	Extreme Group 11
T-F111	
300 T_F111	Variant Class for T-F111
T_M001	Color of Bicycle
T_M002	Integer
T_M003	Floating Point Number
1 01	
Routing, 50000013,	Extreme Group 11

Characteristic: T_M003

Basic Data

Description: Floating Point Number
Characteristic Group:
Status: 1 Released
Authorization Group:

Format

Data Type: NUM Numeric Format
Number of Characters: 3
Decimal Places: 1
Unit of Measurement:
Template: __,_
Exponent Display: No exponent

Figure 3.11 Other Characteristics in the VC Modeling Environment

[+]

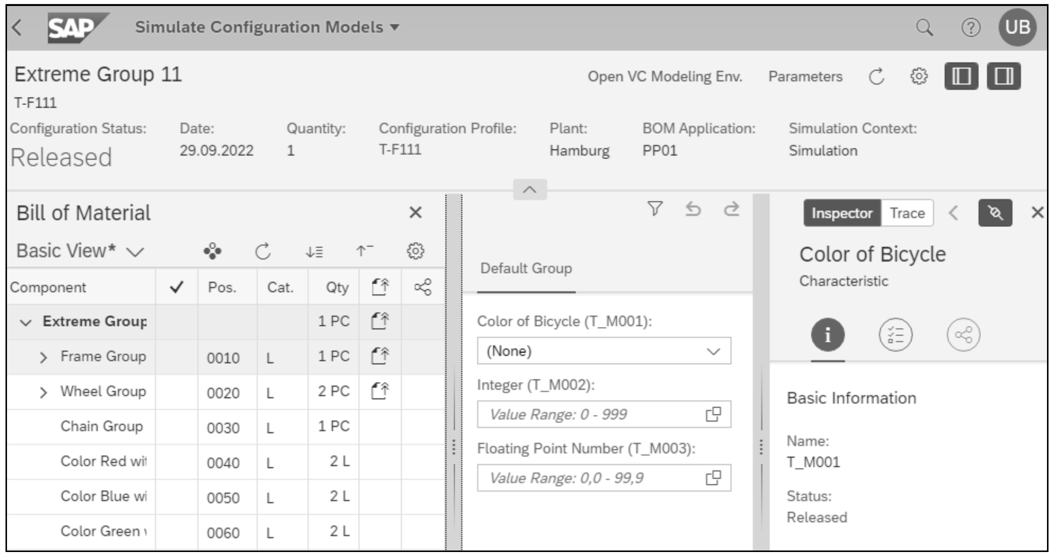
Numeric Format

The *numeric format* distinguishes between integers and floating point numbers, which differ in terms of decimal places, as shown in Figure 3.11. An integer has no decimal places. A floating point number has decimal places greater than zero. These numbers are also stored completely differently in the database and are treated differently in calculations. You’ll learn more about this topic in Chapter 5.

3.5 The Valuation Interface of the Extended Variant Configuration

With our settings so far, you’ve created the prerequisites for a valuation interface, as described at the beginning of the chapter (and shown earlier in Figure 3.1).

You can call the simulation feature from the VC modeling environment by clicking the (**Test**) icon, as shown in Figure 3.12. In the simulation, you’ll see the valuation interface of the *extended variant configuration*. This interface consists of a header area that you can close by clicking the (**Close**) icon. In the middle, at the bottom, you’ll see the actual characteristic valuation. Figure 3.12 shows two more areas to the left and right of the characteristic valuation. Whether these areas are displayed is controlled by the buttons in the top-right corner.



Extreme Group 11
T-F111

Configuration Status:	Date:	Quantity:	Configuration Profile:	Plant:	BOM Application:	Simulation Context:
Released	29.09.2022	1	T-F111	Hamburg	PP01	Simulation

Component	✓	Pos.	Cat.	Qty
Extreme Group				1 PC
Frame Group		0010	L	1 PC
Wheel Group		0020	L	2 PC
Chain Group		0030	L	1 PC
Color Red wi		0040	L	2 L
Color Blue wi		0050	L	2 L
Color Green		0060	L	2 L

Color of Bicycle (T_M001): (None)
Integer (T_M002): *Value Range: 0 - 999*
Floating Point Number (T_M003): *Value Range: 0,0 - 99,9*

Color of Bicycle
Characteristic

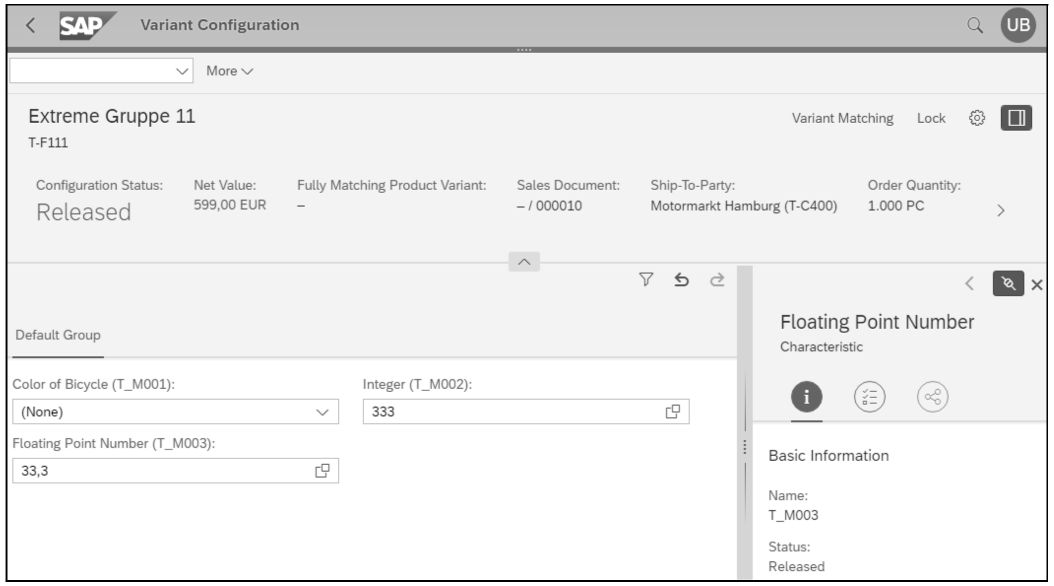
Name: T_M001
Status: Released

Figure 3.12 First Simulation

After looking at the simulated valuation interface, you can see the valuation interface in a sales order for the same material shown in Figure 3.13. What are the commonalities? What are the differences? The actual screen is quite similar to the simulation view. You can tell that you’re in a sales order because the *sales document* term appears in the header. The sales document is not yet saved and therefore does not yet have a number. However, position number 000010 already appears.

In addition, a net value is displayed as part of pricing. This net value is still independent of the configuration. To determine how you can combine the two, refer to Section 3.7 for more information on configuration and prices.

In the simulation view (shown earlier in Figure 3.12), we highlighted the two buttons. In the sales order, only the right button exists, and therefore only the right screen area is available. The left screen area, which represents the BOM, does not exist in the sales order. This omission stems from the fact that we have not set a BOM explosion in the configuration profile (shown earlier in Figure 3.9). This omission has implications for sales orders, but not for simulations.



Extreme Gruppe 11
T-F111

Configuration Status: Released
Net Value: 599,00 EUR
Fully Matching Product Variant: –
Sales Document: – / 000010
Ship-To-Party: Motormarkt Hamburg (T-C400)
Order Quantity: 1.000 PC

Color of Bicycle (T_M001): (None)
Integer (T_M002): 333
Floating Point Number (T_M003): 33,3

Floating Point Number
Characteristic

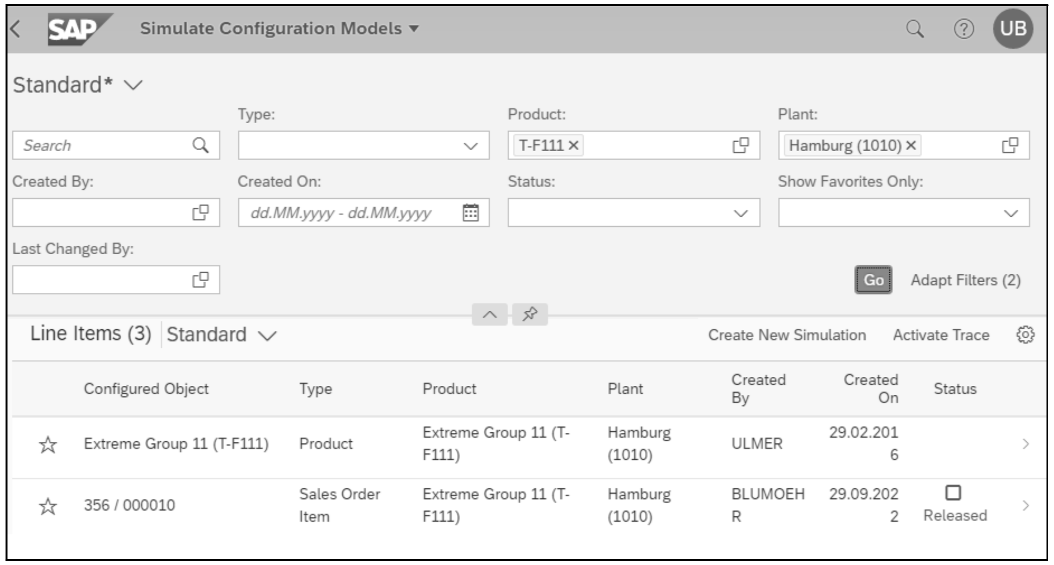
Basic Information

Name:
T_M003

Status:
Released

Figure 3.13 Our First Sales Order

The simulation environment can be called not only via the VC modeling environment, that is, by clicking on the ⊡ (**Test**) icon. You can also use the Simulate Configuration Models app in SAP Fiori to call the extended variant configuration directly.



Type:
Product: T-F111 ×
Plant: Hamburg (1010) ×
Created By:
Created On: dd.MM.yyyy - dd.MM.yyyy
Status:
Show Favorites Only:
Last Changed By:

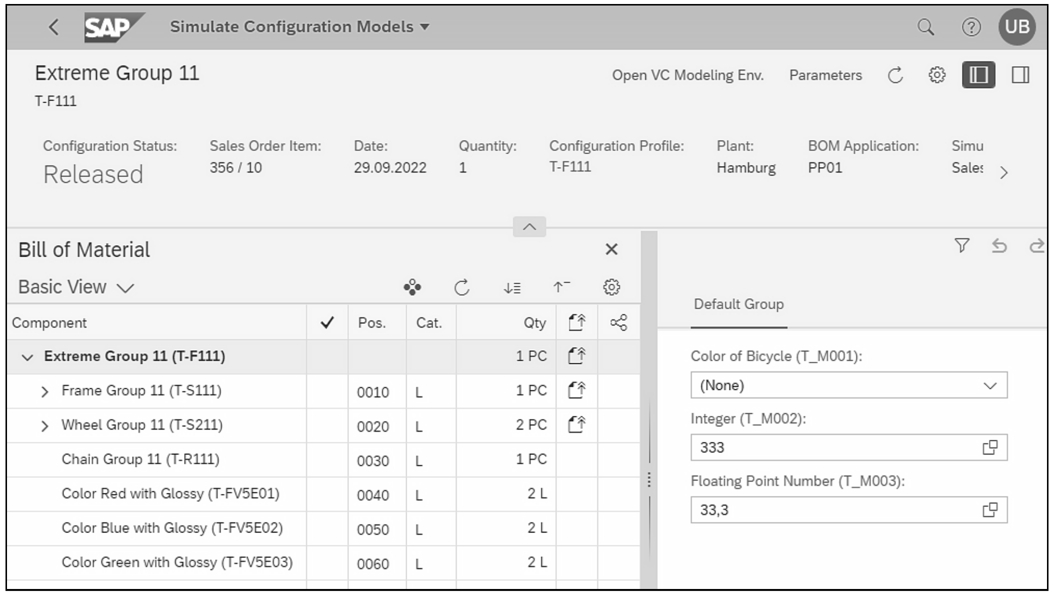
	Configured Object	Type	Product	Plant	Created By	Created On	Status
☆	Extreme Group 11 (T-F111)	Product	Extreme Group 11 (T-F111)	Hamburg (1010)	ULMER	29.02.2016	
☆	356 / 000010	Sales Order Item	Extreme Group 11 (T-F111)	Hamburg (1010)	BLUMOEHR	29.09.2022	□ Released

Figure 3.14 SAP Fiori: Simulate Configuration Models App

Now, the initial screen shown in Figure 3.14 will display. In the upper area, we filtered by our product **T-F111** (which is a material master) and the plant **Hamburg (1010)**. Start the

search by clicking the **Start** button. In the lower half of the initial screen, you'll find two entries in the search results: One represents the product, and the other is the sales order we just created.

You can click the **>** arrow on the right of a line to enter the simulation, as shown in Figure 3.15. The presentation corresponds to the information shown earlier in Figure 3.12 but now includes the valuation from the sales order.



Extreme Group 11
T-F111

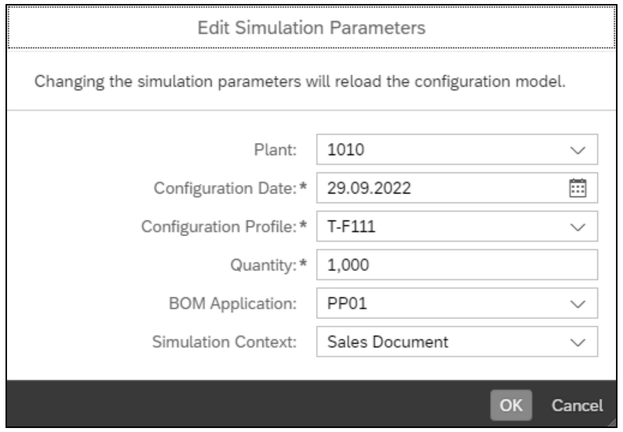
Configuration Status: Released
Sales Order Item: 356 / 10
Date: 29.09.2022
Quantity: 1
Configuration Profile: T-F111
Plant: Hamburg
BOM Application: PP01
Simu
Sale:

Component	✓	Pos.	Cat.	Qty
Extreme Group 11 (T-F111)				1 PC
Frame Group 11 (T-S111)		0010	L	1 PC
Wheel Group 11 (T-S211)		0020	L	2 PC
Chain Group 11 (T-R111)		0030	L	1 PC
Color Red with Glossy (T-FV5E01)		0040	L	2 L
Color Blue with Glossy (T-FV5E02)		0050	L	2 L
Color Green with Glossy (T-FV5E03)		0060	L	2 L

Color of Bicycle (T_M001): (None)
Integer (T_M002): 333
Floating Point Number (T_M003): 33,3

Figure 3.15 Simulation for Our Sales Order

The sales order is shown in the header. Click the **Parameters** button to set the parameters for the parts list in the popup window shown in Figure 3.16.



Edit Simulation Parameters

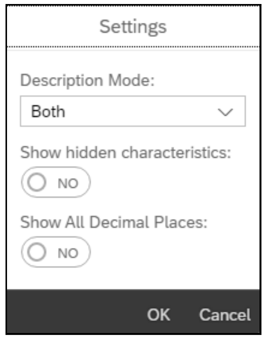
Changing the simulation parameters will reload the configuration model.

Plant: 1010
Configuration Date:* 29.09.2022
Configuration Profile:* T-F111
Quantity:* 1,000
BOM Application: PP01
Simulation Context: Sales Document

Figure 3.16 Editing Simulation Parameters

By clicking on the ⚙ (**Settings**) icon, you can specify, for example, via the **Both** option, that both the description and the language-independent name (for material masters, characteristics, and characteristic values) should be displayed, as shown in Figure 3.17.

The BOM shown in Figure 3.15 on the left is an already configured BOM. However, although the red color was requested, all three colors are provided according to the configured BOM, which is a bug that we'll fix in the next section, through object dependencies.



Settings

Description Mode: Both

Show hidden characteristics: ○ NO

Show All Decimal Places: ○ NO

Figure 3.17 Settings for the Simulation

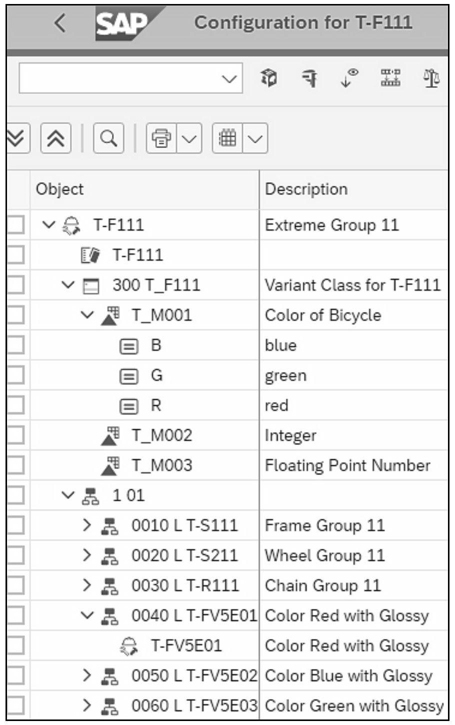
3.6 Object Dependencies for Planning and Production

At the end of the previous section, why did the system incorrectly display all three colors, although only one color was required? The answer is that *object dependencies* are missing from the BOM. Object dependencies control which BOM items are required depending on characteristic value assignments, that is, on the customer requirement. A BOM item without object dependencies is always required, regardless of the characteristic value assignment. Thus, you must assign object dependencies to the three BOM items of the colors, which ensures that only the chosen BOM item is used, depending on the color requested.

The easiest way to create object dependencies is in the VC modeling environment. The left side of Figure 3.18 shows the tree structure has been exploded to ensure the following:

- The characteristic values (B, G, and R) on which the necessary BOM items depend are visible.
- The BOM items (e.g., 0040 L T-FV5E01), whose necessity we want to control, are also visible. Initially, only the material master (e.g., T-FV5E01) is assigned to these items.

You can maintain object dependencies by dragging and dropping a characteristic value to the appropriate BOM item. Thus, in this case, you should move the value R (for red) to the BOM item 0040 L T-FV5E01.



Object	Description
T-F111	Extreme Group 11
T-F111	
300 T_F111	Variant Class for T-F111
T_M001	Color of Bicycle
B	blue
G	green
R	red
T_M002	Integer
T_M003	Floating Point Number
1 01	
0010 L T-S111	Frame Group 11
0020 L T-S211	Wheel Group 11
0030 L T-R111	Chain Group 11
0040 L T-FV5E01	Color Red with Glossy
T-FV5E01	Color Red with Glossy
0050 L T-FV5E02	Color Blue with Glossy
0060 L T-FV5E03	Color Green with Glossy

Figure 3.18 VC Modeling Environment: Characteristic and BOM

Figure 3.19 shows the result: On the left side of the tree structure, another entry 0000000290 exists under BOM item 0040 L T-FV5E01. This entry is the object dependency that we just created.

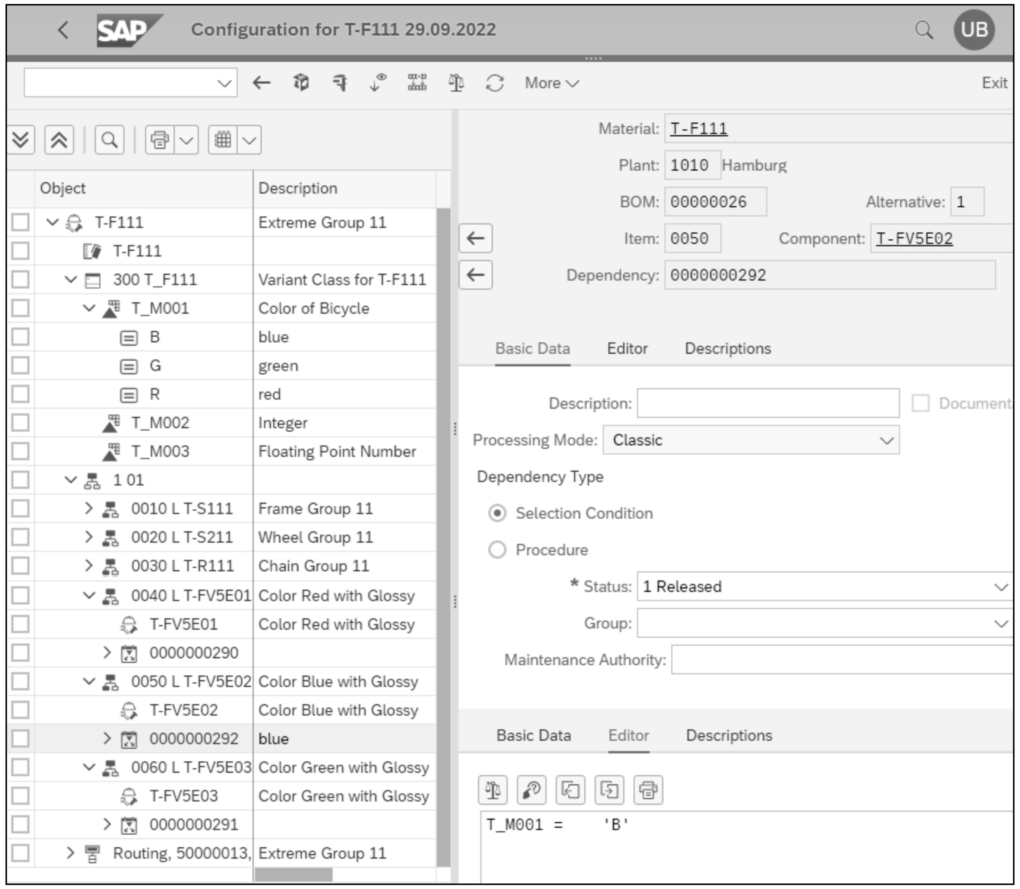
Repeat this process for the colors green and blue. The value G is drawn to BOM item 0060 L T-FV5E03 and creates object dependency 0000000291. Value B is drawn to BOM item 0050 L T-FV5E02 and creates object dependency 0000000292.

As shown in Figure 3.19, the final object dependency created is shown on right. This selection condition uses the classic processing mode and has the following syntax:

```
T_M001 = 'B'
```

With this step, `characteristic = 'characteristic value'` applies. Do not forget to save these settings.

Now, let's check the result. For this step, use the Simulate Configuration Models app in SAP Fiori once again (shown earlier in Figure 3.15). As shown in Figure 3.20, you can see the effects of object dependencies. As shown in the middle pane, if the red color is desired, the blue and green colors are made inactive on the left side of the configured BOM. This behavior would be similar if either the blue color or the green color were desired instead. Since this screen is a simulation, you can test the system's behavior without “breaking” anything.



	Object	Description
□	T-F111	Extreme Group 11
□	T-F111	
□	300 T_F111	Variant Class for T-F111
□	T_M001	Color of Bicycle
□	B	blue
□	G	green
□	R	red
□	T_M002	Integer
□	T_M003	Floating Point Number
□	1 01	
□	0010 L T-S111	Frame Group 11
□	0020 L T-S211	Wheel Group 11
□	0030 L T-R111	Chain Group 11
□	0040 L T-FV5E01	Color Red with Glossy
□	T-FV5E01	Color Red with Glossy
□	0000000290	
□	0050 L T-FV5E02	Color Blue with Glossy
□	T-FV5E02	Color Blue with Glossy
□	0000000292	blue
□	0060 L T-FV5E03	Color Green with Glossy
□	T-FV5E03	Color Green with Glossy
□	0000000291	
□	Routing, 50000013,	Extreme Group 11

Material: T-F111
Plant: 1010 Hamburg
BOM: 00000026 Alternative: 1
Item: 0050 Component: T-FV5E02
Dependency: 0000000292

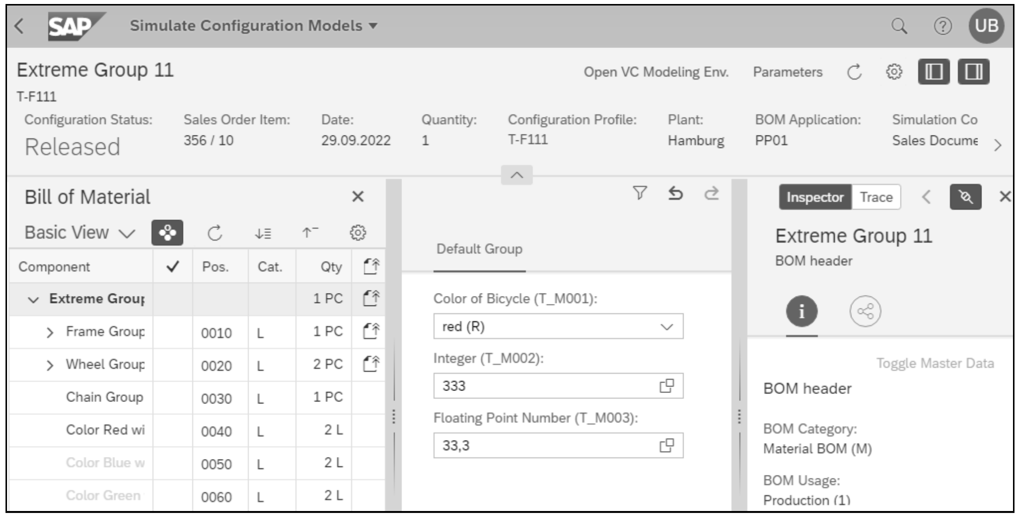
Basic Data | Editor | Descriptions

Description: □ Document
Processing Mode: Classic
Dependency Type
● Selection Condition
○ Procedure
* Status: 1 Released
Group:
Maintenance Authority:

Basic Data | Editor | Descriptions

```
T_M001 =    'B'
```

Figure 3.19 Maintaining Object Dependencies via Drag and Drop



Extreme Group 11
T-F111

Configuration Status:	Sales Order Item:	Date:	Quantity:	Configuration Profile:	Plant:	BOM Application:	Simulation Co
Released	356 / 10	29.09.2022	1	T-F111	Hamburg	PP01	Sales Docume

Bill of Material
Basic View

Component	✓	Pos.	Cat.	Qty
Extreme Grouן				1 PC
Frame Group		0010	L	1 PC
Wheel Group		0020	L	2 PC
Chain Group		0030	L	1 PC
Color Red wi		0040	L	2 L
Color Blue w		0050	L	2 L
Color Green		0060	L	2 L

Default Group
Color of Bicycle (T_M001): red (R)
Integer (T_M002): 333
Floating Point Number (T_M003): 33,3

Extreme Group 11
BOM header

BOM header
BOM Category: Material BOM (M)
BOM Usage: Production (1)

Figure 3.20 SAP Fiori: Simulate Configuration Models App

Now, nothing else stands in the way of the further logistical process. This object dependency was not required for creating the sales order. However, material requirements planning is only possible on the basis of a correct BOM, that is, a BOM with the necessary object dependencies. Now, the requirements for the required BOM items can be determined.

Now that the necessary additions have been made to the BOM (for example, for requirements planning) in the form of object dependencies, let's consider what actions are still required.

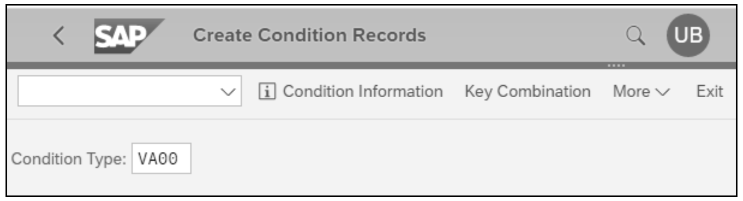
A routing is also required for production. Since we started from a complete model, the routing already exists. At this point, only an adjustment similar to adjusting the BOM would be necessary if new operations were also required for the three new color components. But you shouldn't assume that here!

3.7 Configuration and Prices

To conclude our example, let's look at *valuation-based pricing*. As shown in Figure 3.13, the net value of the product (in our example, the bike) is 599 EUR. This value is initially independent of the configuration, which means that the value is always the same, regardless of whether none (unpainted) or any of the three available colors (blue, green, or red) is selected.

Let's change all that: As shown in Figure 3.22, surcharges of 75 EUR, 50 EUR, or 100 EUR should be added to the base price of 599 EUR depending on if the blue, green, or red color is desired. For this purpose, you must create *condition records* (see Chapter 6), which you can achieve using several apps and transactions. Transaction VK11 was used in our example, which is also included in the variant configuration menu.

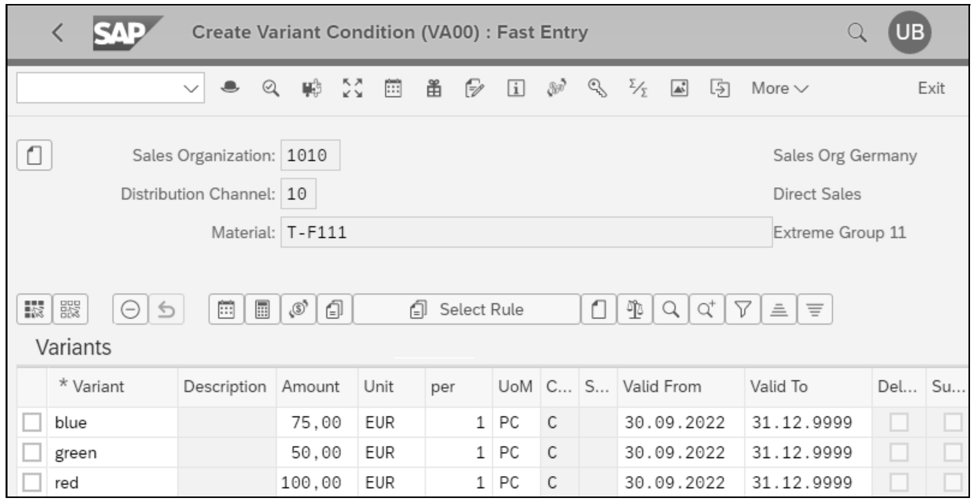
On the initial screen, a condition type called a *variant condition* must be entered in the **Condition Type** field, as shown in Figure 3.21. The VA00 condition type used in our example involves absolute surcharges (or discounts), in EUR.



Condition Type: VA00

Figure 3.21 Creating Condition Records for Pricing: Initial Step

As shown in Figure 3.22, these variant conditions were created. For this purpose, a name is assigned in the **Variant** column. In the **Amount** column, the surcharges for the different colors are maintained; the rest was filled in automatically.



Create Variant Condition (VA00) : Fast Entry

Sales Organization:	1010	Sales Org Germany
Distribution Channel:	10	Direct Sales
Material:	T-F111	Extreme Group 11

Variants

	* Variant	Description	Amount	Unit	per	UoM	C...	S...	Valid From	Valid To	Del...	Su...
□	blue		75,00	EUR	1	PC	C		30.09.2022	31.12.9999	□	□
□	green		50,00	EUR	1	PC	C		30.09.2022	31.12.9999	□	□
□	red		100,00	EUR	1	PC	C		30.09.2022	31.12.9999	□	□

Figure 3.22 Creating Condition Records for Pricing: Fast Entry

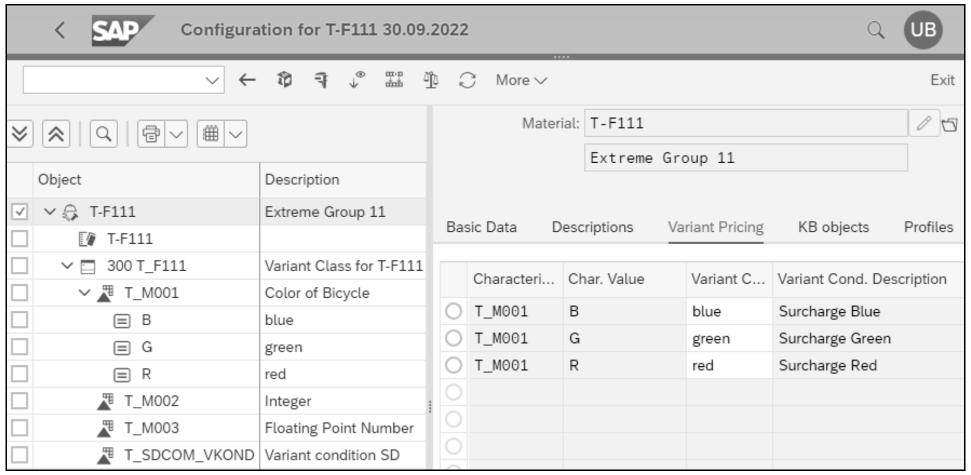
These condition records must next be assigned to the model. For this purpose, you should open the VC modeling environment by double-clicking the first line (**T-F111**) from the tree structure on the left, as shown in Figure 3.23. A detailed view is then displayed in the right-hand pane. Switch to the **Variant Pricing** tab (for variant pricing) and then click the ✎ (**Change**) icon to enter change mode. You'll have two options to choose from, as shown in Figure 3.23:

1. You can enter the desired values in the **Characteristic**, **Char. Value**, **Variant Condition**, and **Variant Cond. Description** columns.
2. You can generate a list of all characteristics values by clicking the **List** button; then, only the third and fourth columns need to be filled manually.

Figure 3.23 shows the T_SDCOM_VKOND characteristic on the left, in the tree structure, in the second-to-last line. This *reference characteristic* references a database field. In our example, the field is VKOND of the SDCOM structure.

This characteristic is necessary to ensure that the pricing of the sales order knows that the desired variant configurations should be considered. The characteristic thus enables the communication between variant configuration on the one hand and pricing in sales on the other. SDCOM is the name of the communication structure.

This characteristic was additionally assigned to a class. (The characteristic must be assigned to our model to be effective in the configuration. This assignment can actually only be made to the class, as no other assignment is provided in this case.) In the VC modeling environment, the detail screen of the class was displayed (see Figure 4.38 in Chapter 4, Section 4.3.1) by double-clicking on the **300 T_F111** line from the tree structure on the left, as shown in Figure 3.23.



Configuration for T-F111 30.09.2022

Material: T-F111
Extreme Group 11

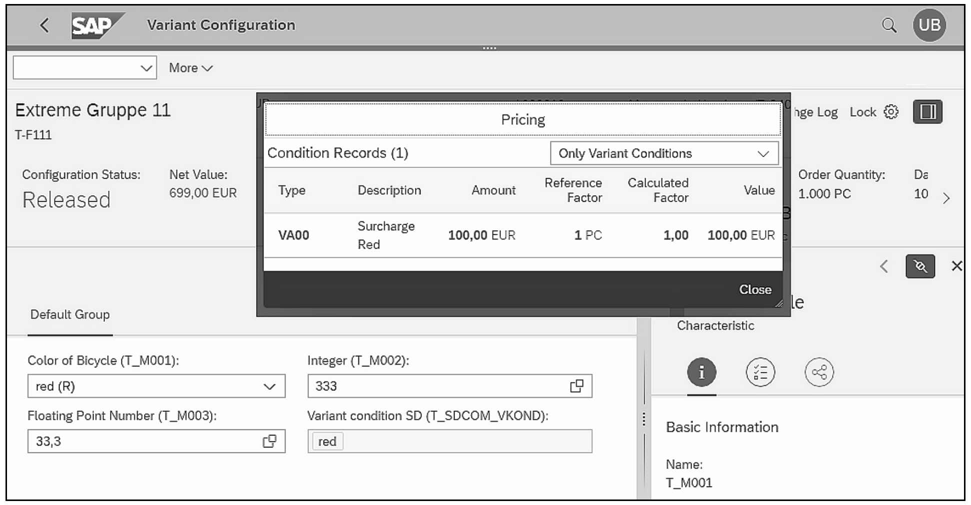
	Object	Description
☑	T-F111	Extreme Group 11
□	T-F111	
□	300 T_F111	Variant Class for T-F111
□	T_M001	Color of Bicycle
□	B	blue
□	G	green
□	R	red
□	T_M002	Integer
□	T_M003	Floating Point Number
□	T_SDCOM_VKOND	Variant condition SD

	Characteri...	Char. Value	Variant C...	Variant Cond. Description
○	T_M001	B	blue	Surcharge Blue
○	T_M001	G	green	Surcharge Green
○	T_M001	R	red	Surcharge Red
○				
○				
○				

Figure 3.23 Condition Records for Pricing: Assigning Condition Records

To maintain the reference characteristic, click the ✎ (**Change**) icon under the **Characteristics** tab to switch to change mode. Enter the name of the desired characteristic in the first empty row in the first column under the **Characteristics** tab. You can search for a matching characteristic using the database field referenced by the characteristic. If no suitable reference characteristic exists, you can create it at this point. For more details on this topic, refer to Chapter 4, Section 4.4.1.

We've completed the final modeling step. To check whether the modeling is successful, switch again to the sales order and then to the configuration, as shown in Figure 3.24. What has changed? The net value has increased by 100 EUR compared to the original value (shown earlier in Figure 3.13). By clicking on the net price, the **Pricing** popup window opens, where you'll find more details.



Variant Configuration

Extreme Gruppe 11
T-F111

Configuration Status: Released
Net Value: 699,00 EUR
Order Quantity: 1.000 PC
Da 10

Pricing

Condition Records (1) — Only Variant Conditions

Type	Description	Amount	Reference Factor	Calculated Factor	Value
VA00	Surcharge Red	100,00 EUR	1 PC	1,00	100,00 EUR

Color of Bicycle (T_M001): red (R)
Integer (T_M002): 333
Floating Point Number (T_M003): 33,3
Variant condition SD (T_SDCOM_VKOND): red

Basic Information
Name: T_M001

Figure 3.24 Pricing with Variant Conditions in the Sales Order

3.8 Summary

With this short example, we've provided some early insights into working with configurable materials. We've provided an overview of essential modeling steps in variant configuration. After this "appetizer," we'll go step by step into the implementation of variant configuration with SAP S/4HANA for advanced variant configuration (AVC) in Chapter 4 through Chapter 7.

We deliberately decided not to pursue this simple example further. Now, a sales order could be created with our material, and our model could be controlled.

Contents

Part I Basic Principles of Variant Configuration

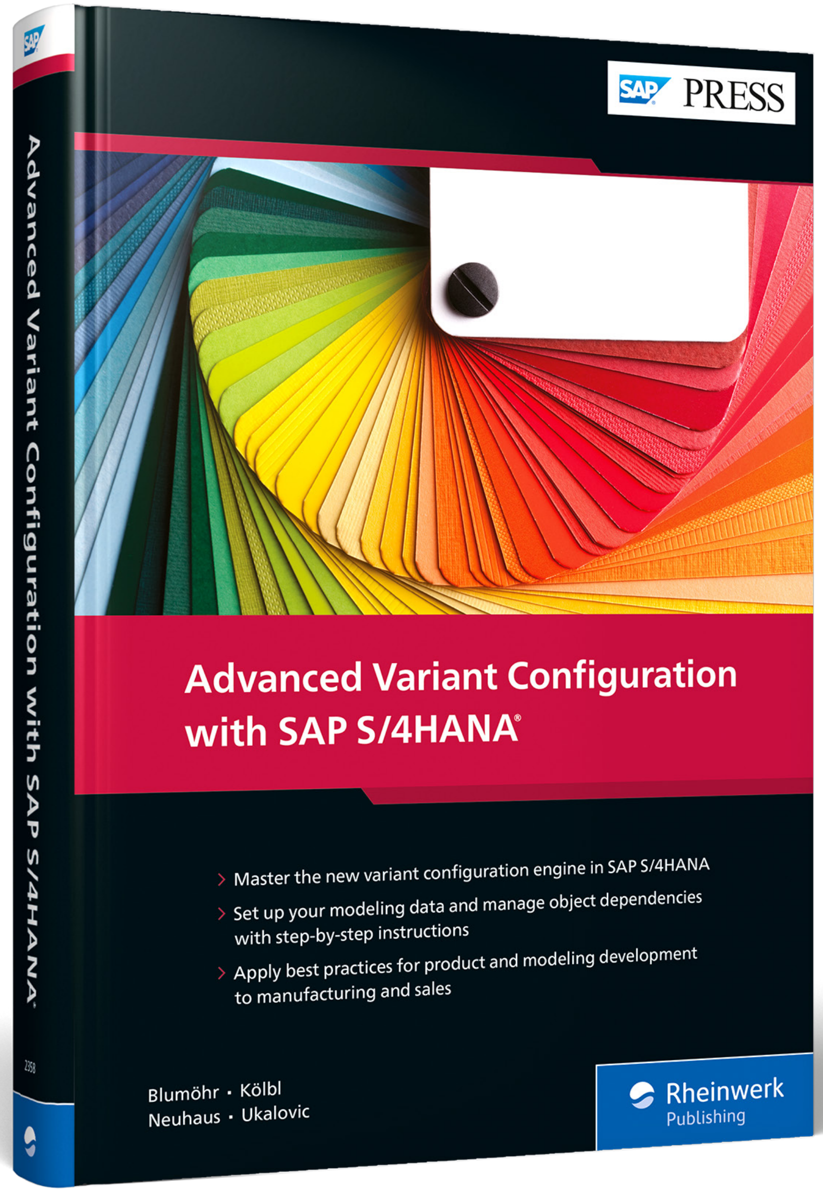
Part II Basics of Modeling

Part III Integration

Part IV Industry Best Practices

Index

SAP PRESS
First-hand knowledge.
Rheinwerk Publishing


SAP PRESS
Advanced Variant Configuration with SAP S/4HANA
Advanced Variant Configuration with SAP S/4HANA®
Master the new variant configuration engine in SAP S/4HANA
Set up your modeling data and manage object dependencies with step-by-step instructions
Apply best practices for product and modeling development to manufacturing and sales
Blumöhr · Kölbl
Neuhaus · Ukalovic
Rheinwerk Publishing

Blumöhr, Köbl, Neuhaus, Ukalovic

Advanced Variant Configuration with SAP S/4HANA

567 pages | 10/2023 | $99.95 | ISBN 978-1-4932-2358-9

www.sap-press.com/5628

Dr. Uwe Blumöhr holds a degree in mathematics and is a qualified teacher. He works for SAP as a training consultant in the area of customer and partner training. Uwe focuses on variant configuration and lifecycle data management. Since 1996, he has been responsible for all SAP training developments in the area of variant configuration worldwide; the majority of SAP's training material on the subject was developed directly by him.

Andreas Kölbl is product manager and the responsible area product owner for SAP S/4HANA for advanced variant configuration (AVC) at SAP. After graduating with a degree in mechanical engineering in 1997, he has worked for several companies as an SAP consultant. His focus was on the processes of work preparation as well as production planning and control. In this context, he supported a large number of national and international customers from various industries, with mechanical engineering always his core area. In 2014, he joined SAP as a solution architect in the product lifecycle management (PLM) area. Andreas has been part of the software development team at SAP SE in Walldorf since 2016 and initially worked as a product owner for the AVC engine. He moved to his current role in 2020.

Michael Neuhaus is the managing director of networker, solutions GmbH and TRILUX Digital Solutions GmbH and responsible for consulting as well as sales and marketing. Michael has worked in the IT industry for many years, is an experienced expert in the SAP world, and has focused on consulting and implementing SAP solutions for several years. In particular, he is an established expert in the area of variant configuration and advises variant manufacturers on the path to digital transformation. His career includes various consulting and management positions in internationally oriented consulting and manufacturing companies.

After graduating from university with a degree in mechanical engineering, **Marin Ukalovic** initially worked as a design engineer in mechanical and plant engineering. In 1999, he moved to the sales department at SAP SE and has advised customers in discrete manufacturing in the area of logistics. In 2003, he took over industry responsibility for the paper and furniture industry at SAP as a solution architect. Since moving to the industry business development team for EMEA in 2006, he has been responsible for mechanical and plant engineering in Europe. The topic of SAP variant configuration runs like a thread throughout his SAP career. From 2006, he was responsible for business development in the EMEA region for mechanical and plant engineering. In 2019, he moved back to the SAP headquarters in Walldorf and took on responsibility as the chief product owner for variant configuration, which includes AVC. At the same time, Marin has been leading the Configuration Workgroup (CWG) in his role as chairman.