





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.

-  **“Creating Your First Configurable Material”**
-  **Contents**
-  **Index**
-  **The Authors**

Blumöhr, Kölbl, 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).

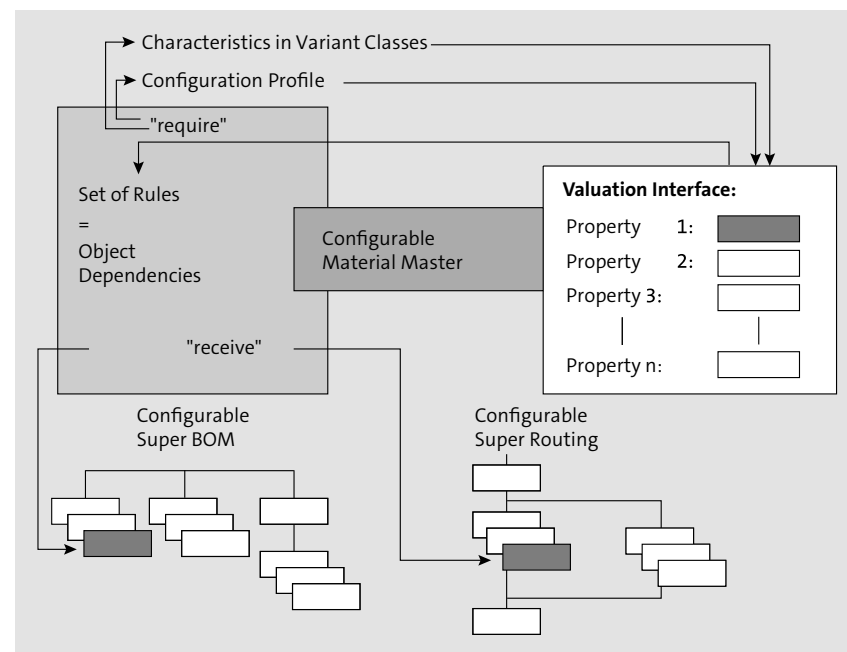


Figure 3.1 Simplified Model of Variant Configuration

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** (sub-items) 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.)

Item	ICt	Component	Component description	Quantity	UoM	Asm	SIs	Valid From	Valid to	Ph...
0010	L	T-S111	Frame Group 11	1	PC	<input checked="" type="checkbox"/>	<input type="checkbox"/>	29.02.2016	31.12.9999	<input type="checkbox"/>
0020	L	T-S211	Wheel Group 11	2	PC	<input checked="" type="checkbox"/>	<input type="checkbox"/>	29.02.2016	31.12.9999	<input type="checkbox"/>
0030	L	T-R111	Chain Group 11	1	PC	<input type="checkbox"/>	<input type="checkbox"/>	29.02.2016	31.12.9999	<input type="checkbox"/>
0040	L					<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
0050	L					<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
0060	L					<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
0070	L					<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>

Figure 3.2 Non-Configurable Starting Point for Our Example

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

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.

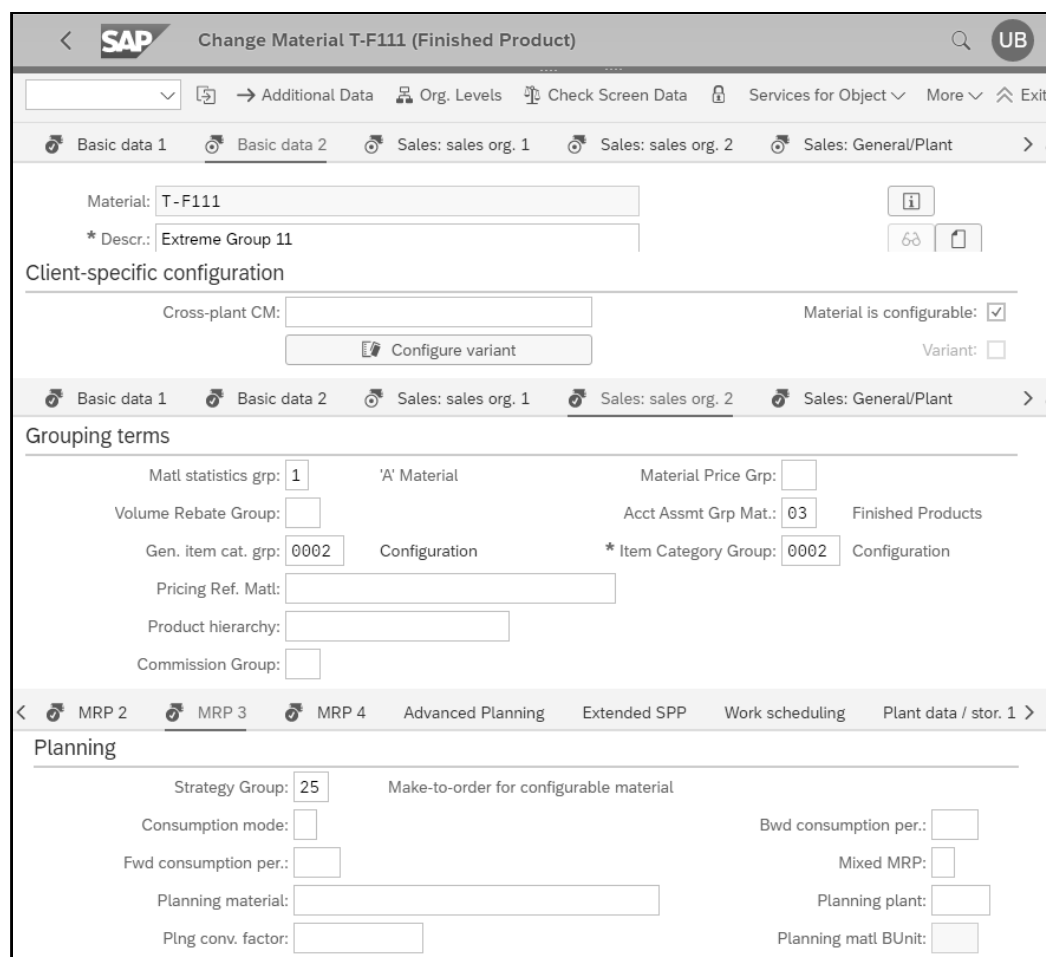


Figure 3.4 Configurable Material Master

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.

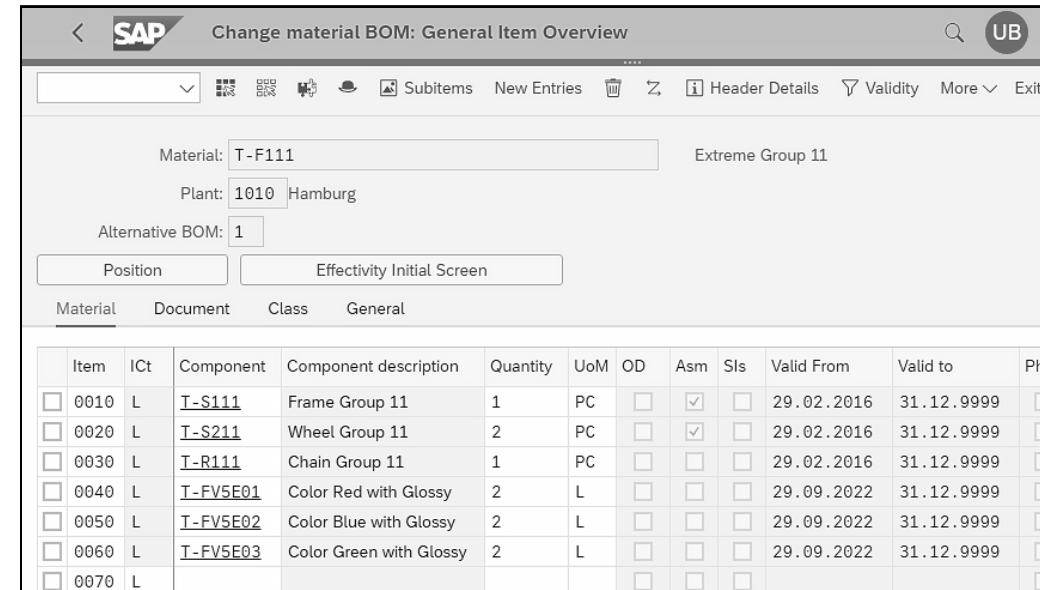


Figure 3.5 Configurable BOM

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 CLO2 (Class Management), as shown in Figure 3.6.

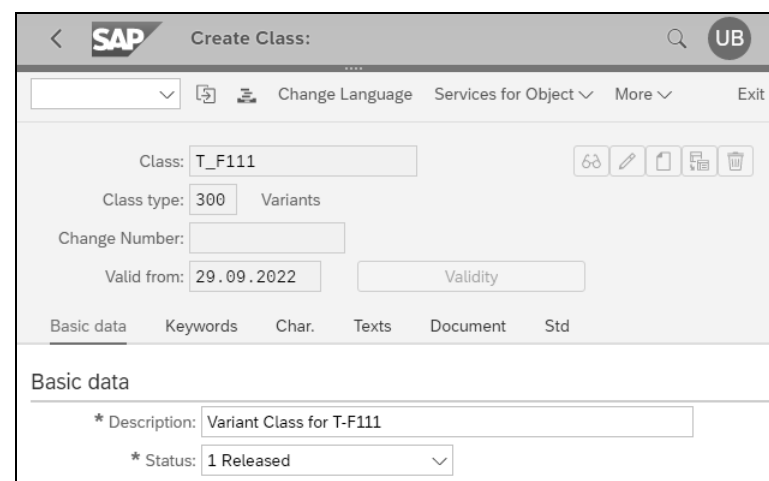


Figure 3.6 Creating a Variant Class

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

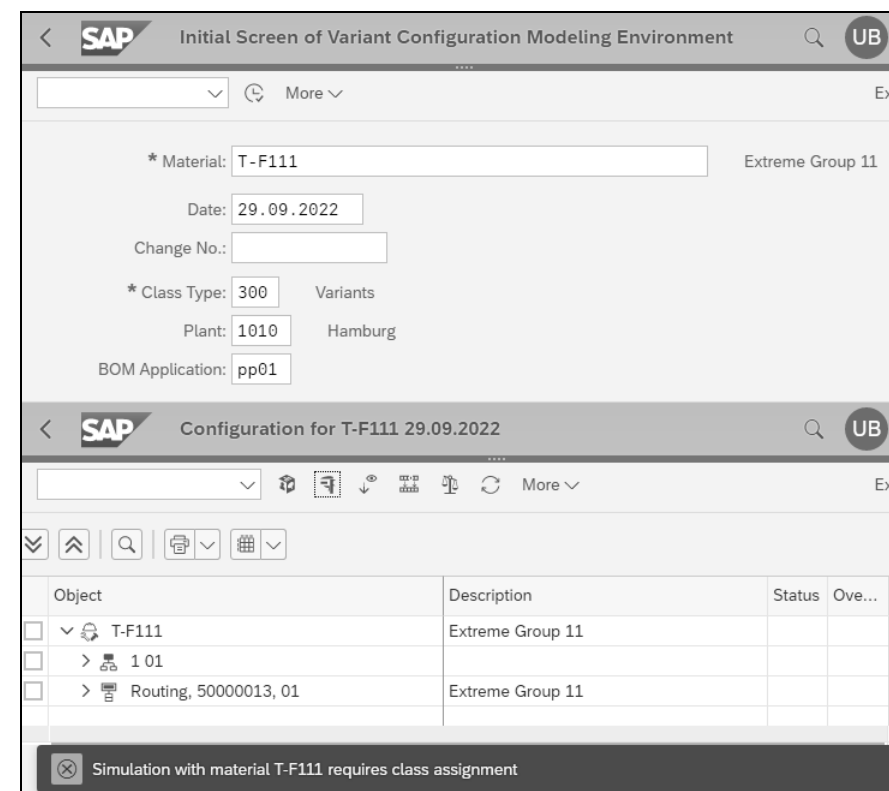




Figure 3.7 VC Modeling Environment: First Attempt

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.

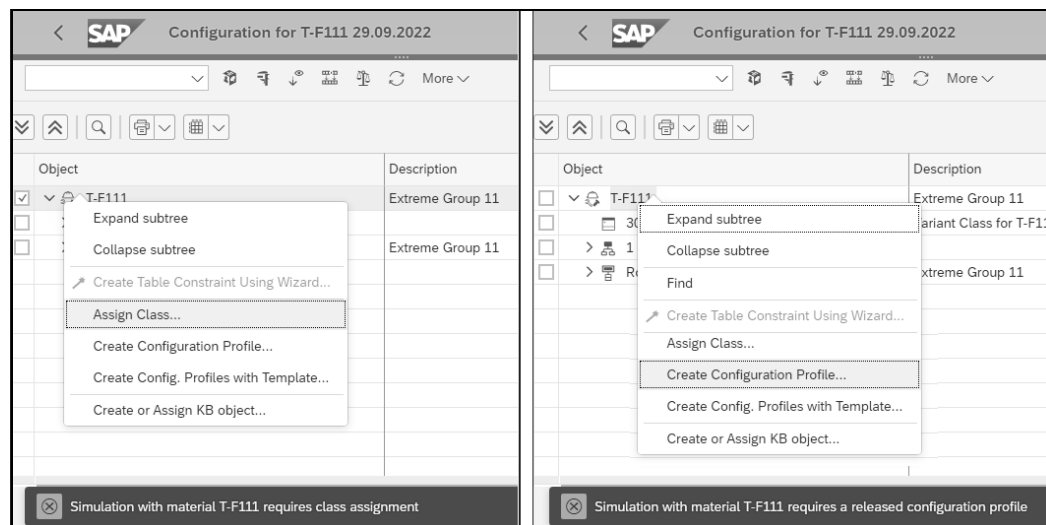


Figure 3.8 VC Modeling Environment: Second Attempt, Assigning a Class and Creating a Configuration Profile

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.

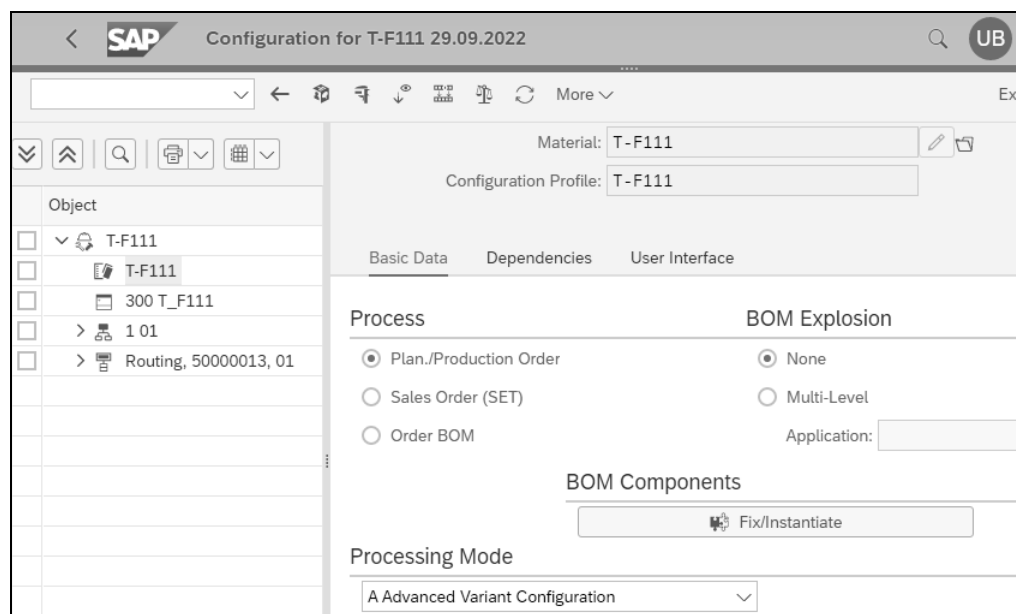


Figure 3.9 VC Modeling Environment: Configuration Profile

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.

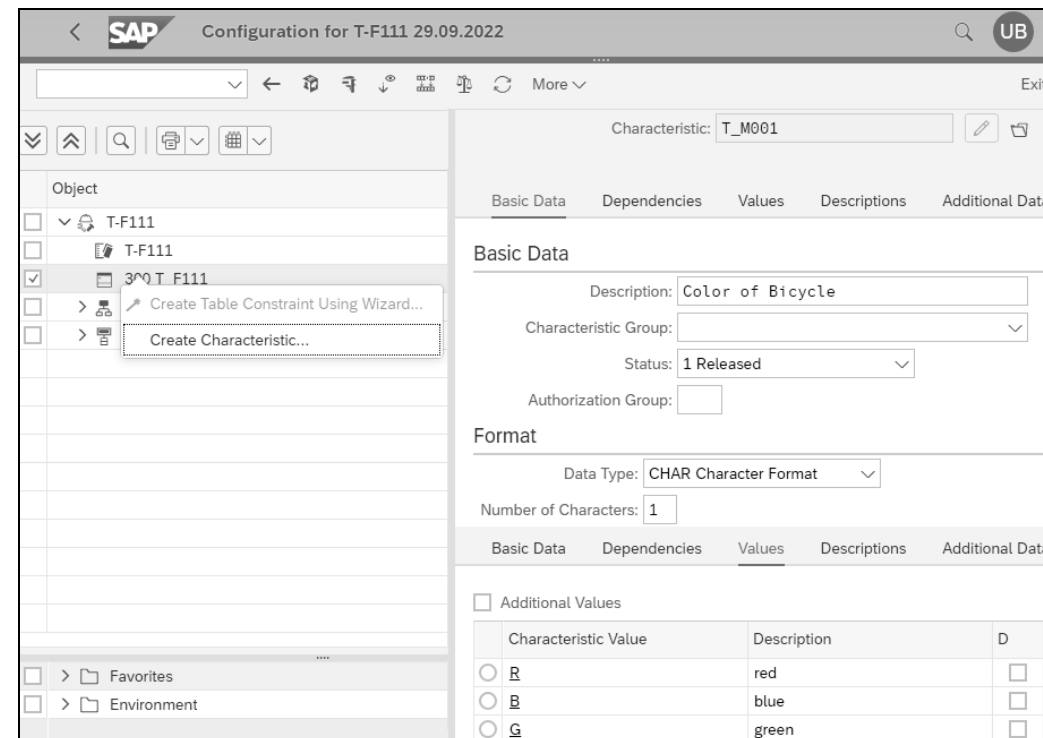


Figure 3.10 VC Modeling Environment: First Characteristic

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.

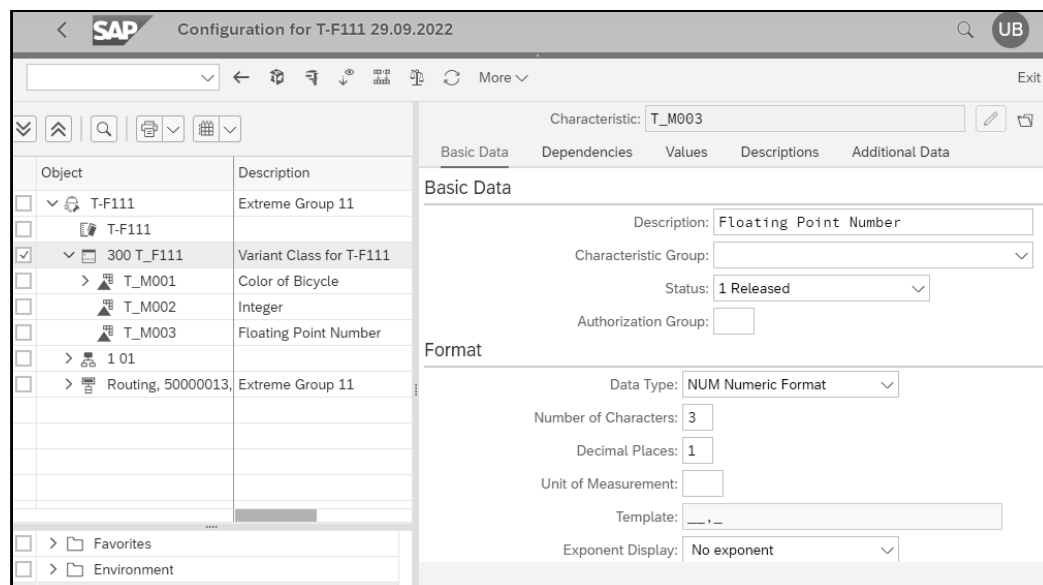


Figure 3.11 Other Characteristics in the VC Modeling Environment






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.

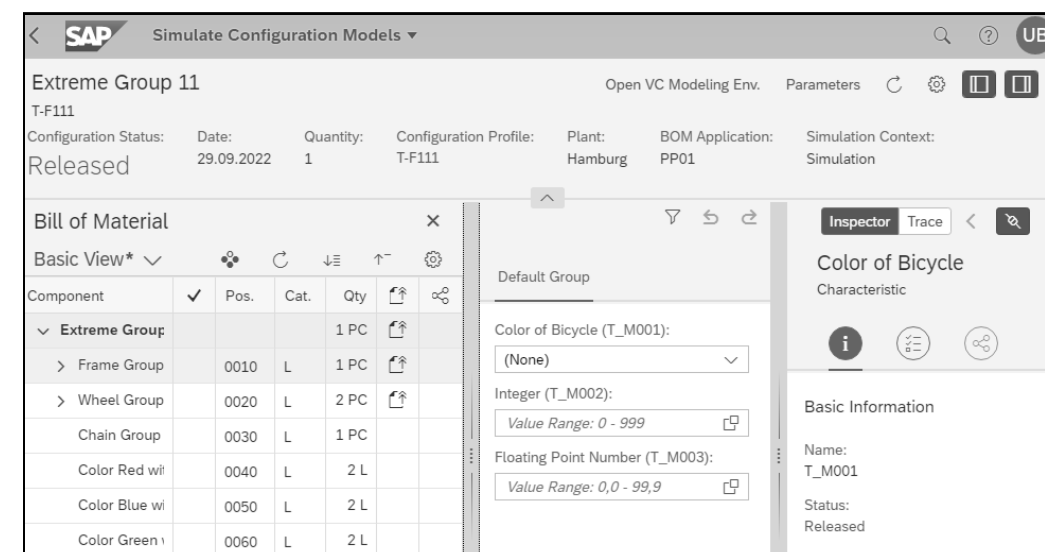



Figure 3.12 First Simulation

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.

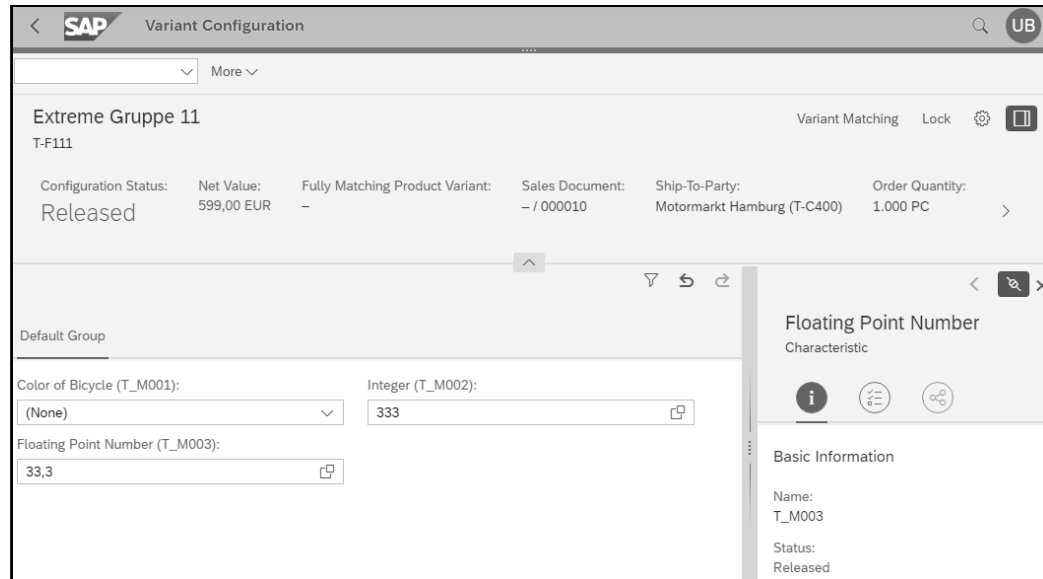


Figure 3.13 Our First Sales Order

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.

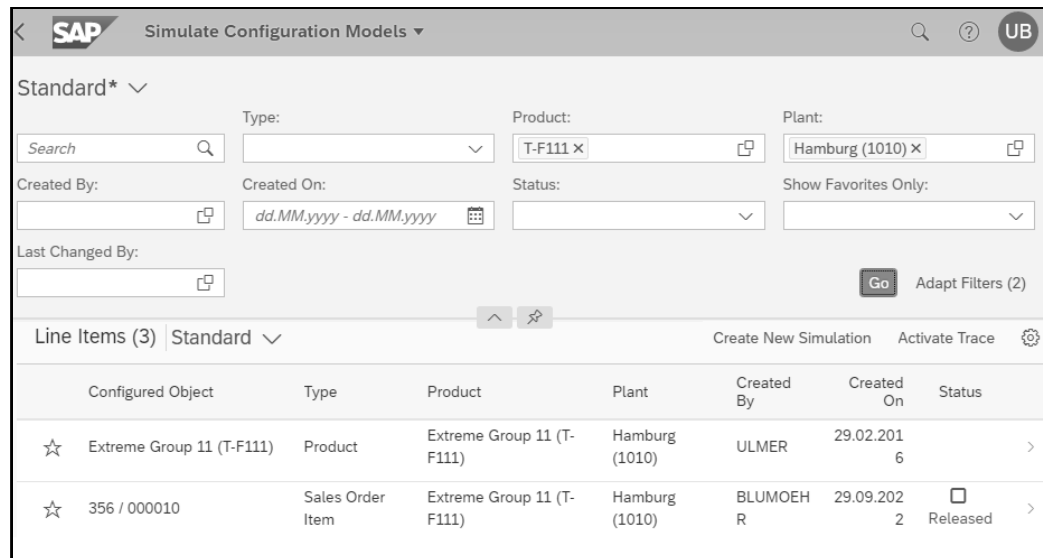


Figure 3.14 SAP Fiori: Simulate Configuration Models App

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.

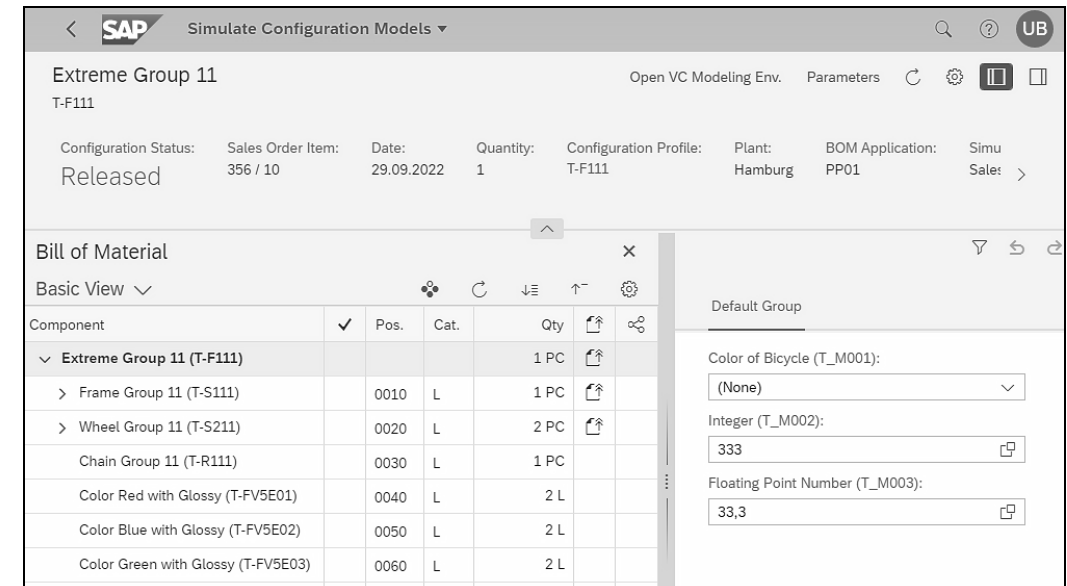


Figure 3.15 Simulation for Our Sales Order

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.

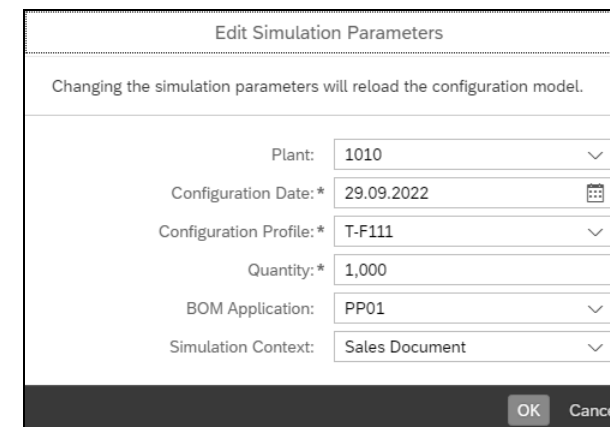



Figure 3.16 Editing Simulation Parameters

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.

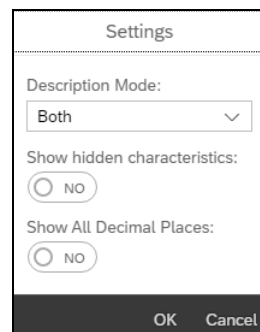


Figure 3.17 Settings for the Simulation

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.

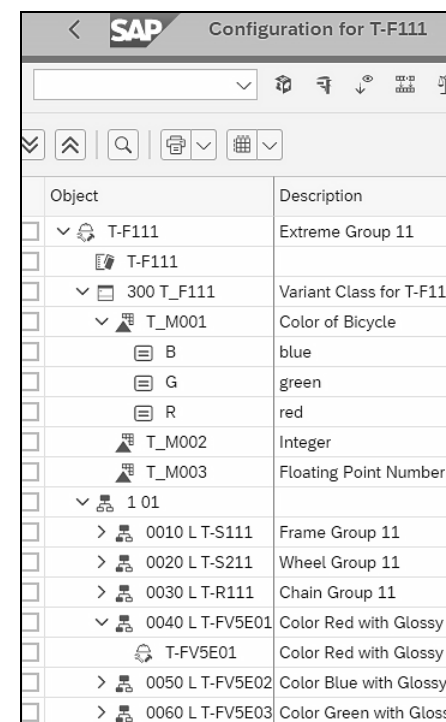


Figure 3.18 VC Modeling Environment: Characteristic and BOM

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.

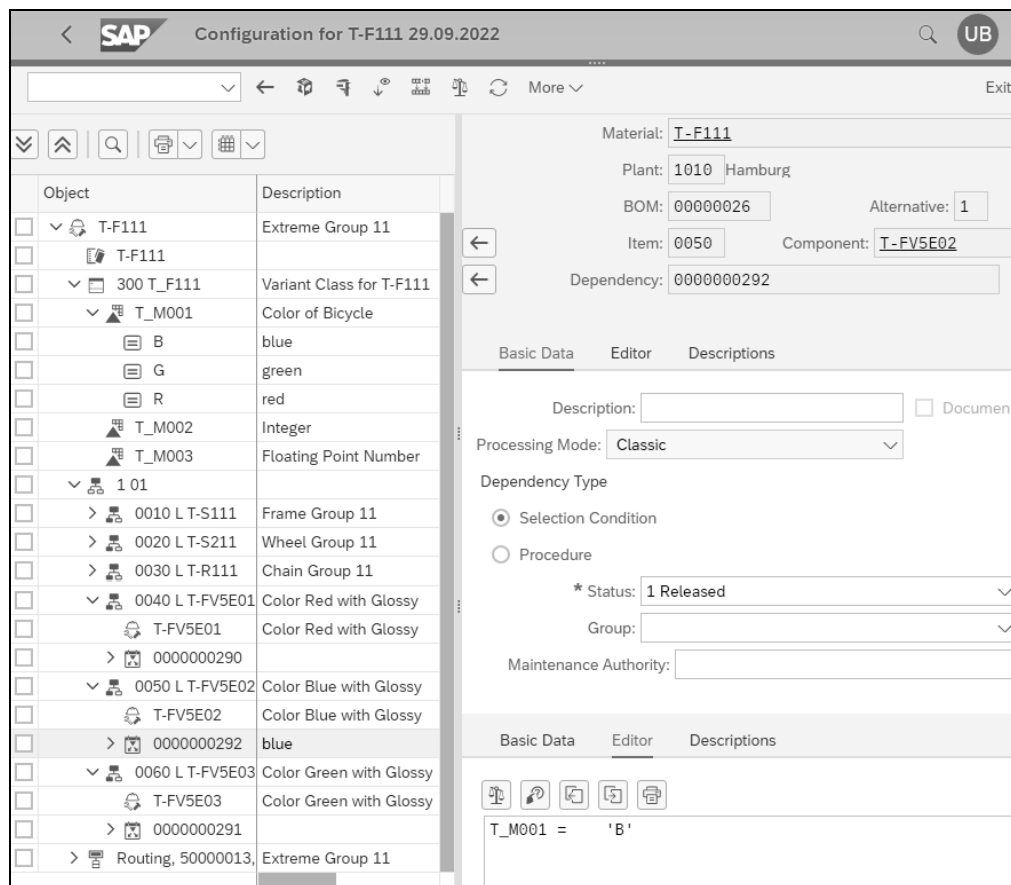


Figure 3.19 Maintaining Object Dependencies via Drag and Drop

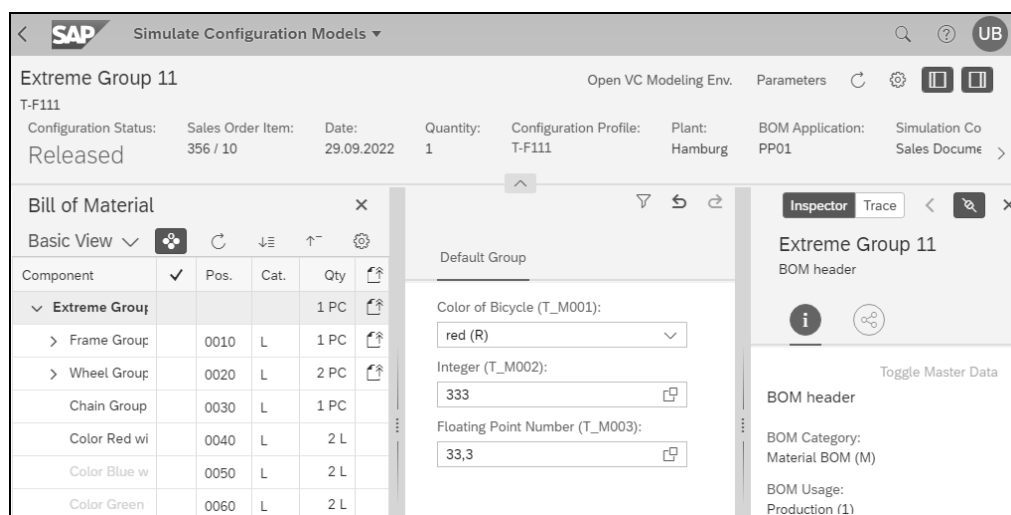


Figure 3.20 SAP Fiori: Simulate Configuration Models App

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.

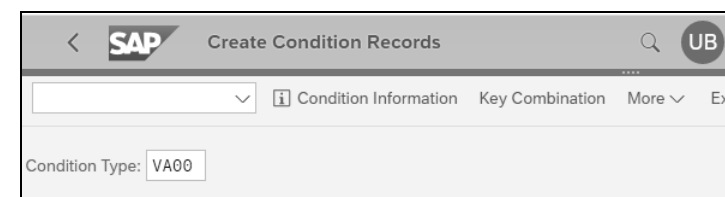


Figure 3.21 Creating Condition Records for Pricing: Initial Step

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.

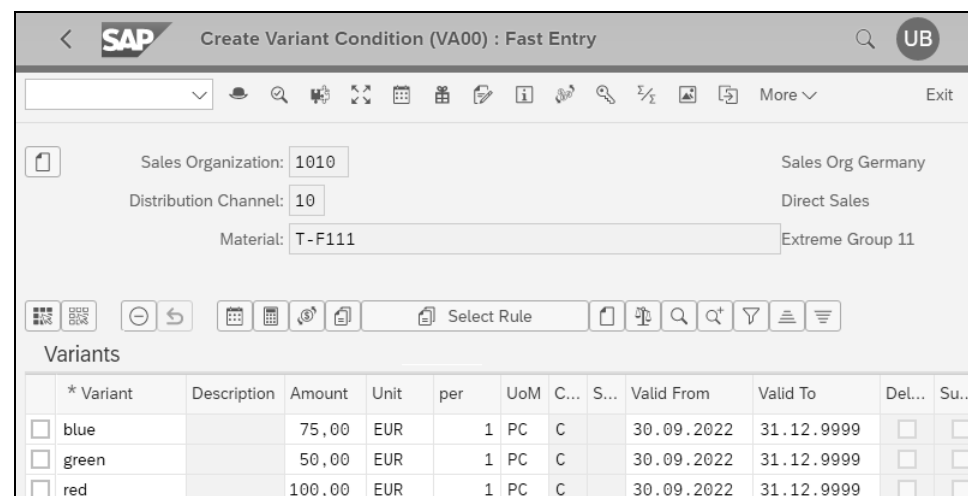


Figure 3.22 Creating Condition Records for Pricing: Fast Entry

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.

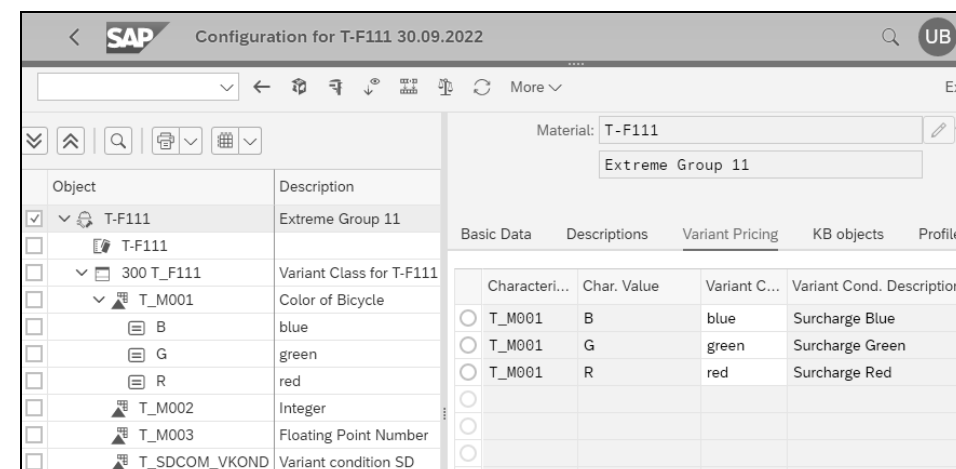


Figure 3.23 Condition Records for Pricing: Assigning Condition Records

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.

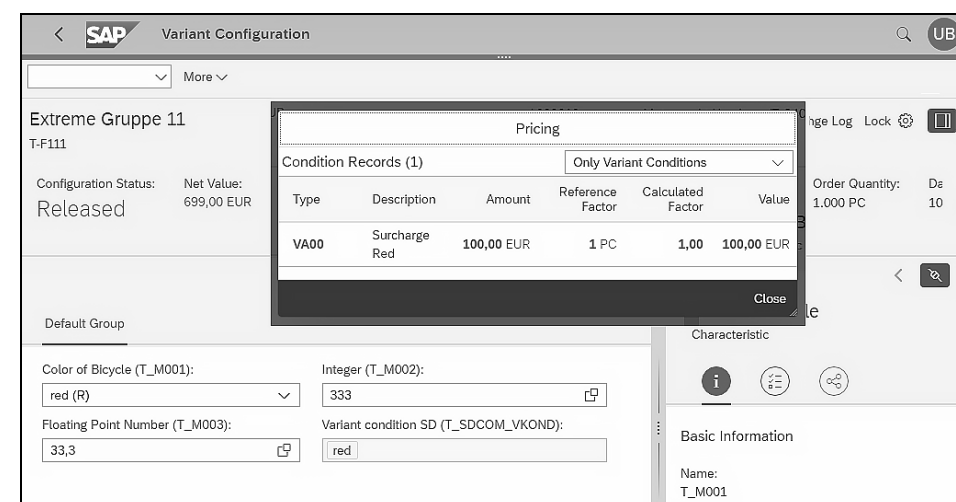


Figure 3.24 Pricing with Variant Conditions in the Sales Order

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

Foreword	17
Preface	21

Part I Basic Principles of Variant Configuration

1 Product Configuration with SAP 27

1.1 Generation Change: From SAP ERP to SAP S/4HANA	28
1.1.1 Comparing SAP ERP and SAP S/4HANA	28
1.1.2 The SAP Fiori User Interface	30
1.1.3 SAP HANA or SAP S/4HANA?	30
1.1.4 Modes of Operation	31
1.1.5 The Digital Core	31
1.1.6 Simplification	32
1.2 Generation Change: From LO-VC to AVC	32
1.2.1 SAP S/4HANA: The Simplification List in LO-VC	33
1.2.2 AVC in SAP S/4HANA Cloud	34
1.3 Which SAP Configurators Are Available?	39
1.3.1 Internet Pricing and Configurator	39
1.3.2 Solution Sales Configurator	39
1.3.3 SAP CPQ	40
1.3.4 SAP Advanced Variant Configuration and Pricing Microservices	40
1.4 Product Configuration with AVC in an End-to-End Process	41
1.5 Managing Product Diversity	47
1.6 Manufacturing Scenarios for Configurable Products	50
1.7 Summary	52

2 The Path from LO-VC to AVC in SAP S/4HANA 53

2.1 AVC and LO-VC: An Either-Or?	53
2.2 Ways of Transformation	57

2.3	Basic Principles of the Transformation	58
2.4	Transition Workspaces	60
2.5	Transition Workbench	62
2.5.1	Structure of the Transition Workbench	62
2.5.2	Analysis of the LO-VC Model	63
2.5.3	Creating the AVC Model	65
2.5.4	Transforming a Constraint: An Example	66
2.6	Comparing the Results	67
2.7	Transferring the Material Variants	70
2.8	An Exemplary Approach to Switching from VC to AVC (Greenfield versus Brownfield)	72
2.8.1	Why Switch to AVC?	72
2.8.2	Brownfield versus Greenfield	73
2.8.3	Step 1: Scoping	73
2.8.4	Step 2: Planning	74
2.8.5	Step 3: Implementation	75
2.8.6	Step 4: Testing	75
2.8.7	Step 5: Go-Live	76
2.8.8	Step 6: Hypercare	76
2.9	Summary	76

Part II Basics of Modeling

3	Creating Your First Configurable Material	79
3.1	Basic Principles	80
3.2	A First Example	82
3.3	The Configurable Material Master	84
3.4	Variant Classes, Characteristics, and Configuration Profiles	86
3.5	The Valuation Interface of the Extended Variant Configuration	90
3.6	Object Dependencies for Planning and Production	94
3.7	Configuration and Prices	97
3.8	Summary	100

4	Master Data in the Modeling Context	101
4.1	Variant Configuration Modeling Environment	101
4.2	AVC Simulation Environment	110
4.3	Tools from the Classification System	119
4.3.1	Characteristic Maintenance	119
4.3.2	Class Maintenance	129
4.3.3	Classification	133
4.3.4	Search and Evaluation Options in the Classification System	139
4.4	Material Master, Bill of Materials, and Routing of the Configurable Material	145
4.4.1	Material Master of the Configurable Material	145
4.4.2	Super Bill of Materials of the Configurable Material	154
4.4.3	Super Routing for the Configurable Material	157
4.4.4	Production Versions	159
4.5	Configuration Profile and Configuration Scenarios	162
4.5.1	Overview of a Configuration Profile	162
4.5.2	Configuration Profiles in Detail	166
4.5.3	Configuration Scenarios at a Glance	170
4.5.4	Planned/Production Order without BOM Explosion Scenario	171
4.5.5	Planned/Production Order with BOM Explosion Scenario	174
4.5.6	Sales Order (SET) Scenario	177
4.5.7	Order BOM Scenario: The Engineer-to-Order Process	181
4.5.8	Order BOM Scenario: Order Bill of Materials and Order Routing	191
4.6	Characteristic Groups as User Interface Design	197
4.7	Summary	202
5	Object Dependencies	203
5.1	Basic Principles	203
5.1.1	Types and Assignment of Object Dependencies	204
5.1.2	Procedural and Declarative Character of Object Dependencies	209
5.1.3	Local and Global Object Dependencies	209
5.1.4	Status of Object Dependencies	215
5.1.5	Processing Mode of Object Dependencies	216
5.1.6	Object Dependencies in Classification and Variant Configuration	217
5.1.7	Execution Sequence of Object Dependencies	218
5.1.8	Basic Rules of the Syntax	221

5.1.9	Syntax Elements	224
5.2	Variant Tables	228
5.2.1	Creating a Table Structure	228
5.2.2	Linking a Variant Table to a Database Table or to a Custom Business Object	230
5.2.3	Contents of a Variant Table	232
5.2.4	Accessing a Variant Table	234
5.2.5	Variant Tables for Value Restrictions via Constraints	234
5.2.6	Variant Tables for Deriving Values via Procedures	236
5.2.7	Variant Tables in Conditions	238
5.3	Object Dependencies for Interactive Configuration (High-Level Configuration)	240
5.3.1	Use Constraints	240
5.3.2	Exclusive Syntax for Constraints in Detail	242
5.3.3	Constraints in the Trace	250
5.3.4	Procedures	253
5.3.5	Exclusive Syntax for Procedures in Detail	256
5.3.6	Procedures in the Trace	258
5.3.7	Facet Changes with Preconditions and Selection Conditions	259
5.3.8	AVC: Business Add-Ins	263
5.3.9	Principles of AVC Modeling for Good Performance	268
5.3.10	Constraints and the Multiple Use of Classes	270
5.4	Object Dependencies for a Bill of Materials and Routing (Low-Level Configuration)	271
5.4.1	Selection Conditions in a Bill of Materials and Routing	271
5.4.2	Class Nodes in Bills of Materials	274
5.4.3	Procedures in Bills of Materials and Routings	278
5.5	Summary	281
6	Pricing	283
6.1	Pricing in Sales	283
6.1.1	Step 1: Reference Characteristic with Reference to Structure SDCOM-VKOND	284
6.1.2	Step 2: Determine the Condition Type and Create a Variant Condition	286
6.1.3	Step 3: Check the Costing Sheet	288
6.1.4	Step 4: Assign Variant Conditions	288
6.1.5	Checking the Result of Our Modeling Steps	291

6.2	Pricing on the Basis of Sales Order Costing	295
6.3	Pricing in Purchasing	296
6.3.1	Step 1: Check the Costing Sheet	302
6.3.2	Step 2: Determine the Condition Type and Create Variant Conditions	302
6.3.3	Step 3: Assign the Reference Characteristic	305
6.3.4	Step 4: Assign Variant Conditions	305
6.4	Microservice for Pricing on SAP Business Technology Platform	306
6.4.1	How the Pricing Service Works	307
6.4.2	Administration	309
6.4.3	Extensibility	310
6.5	Summary	311
7	Material Variants	313
7.1	Material Master	314
7.2	Bill of Materials	319
7.3	Routing	323
7.4	Pricing	328
7.5	Variant Matching	328
7.6	Summary	336
 Part III Integration		
8	Integrating Variant Configuration in Platforms via Knowledge Bases and Runtime Versions	339
8.1	Architecture	339
8.2	Master Data for Configuration Integration: Knowledge Bases and Runtime Versions	342
8.2.1	Replicating Master Data from SAP ERP and SAP S/4HANA Systems	342
8.2.2	The Delta List in Relation to the Use of Microservices and LO-VC	343
8.2.3	Differences between Configuration Models and Knowledge Bases in the Delta List	345
8.2.4	Selecting the Knowledge Base at Runtime	345

8.3	Tips for Generating Knowledge Base Runtime Versions	346
8.3.1	Objects of the Knowledge Base	346
8.3.2	Generating New Runtime Versions	347
8.3.3	Regenerating Existing Runtime Versions	348
8.3.4	Configuration with Runtime Versions during Changes to the Product Model	350
8.3.5	The Suitability of a Product Model for Generating Knowledge Bases and Runtime Versions	352
8.3.6	The Compatibility of the Product Model	353
8.3.7	Changes for the Application of Configuration Models in Sales and Distribution	353
8.3.8	Tips for Creating Runtime Versions	353
8.3.9	Recommendations for Naming Conventions	354
8.4	Instructions for (Re)generating Runtime Versions	365
8.4.1	When Do New Runtime Versions Need to Be Created?	366
8.4.2	When Do Existing Runtime Versions Need to Be Regenerated?	367
8.4.3	Runtime Versions in Connection with Change Numbers	367
8.4.4	How Often Should You (Re)generate Runtime Versions?	368
8.4.5	Innovations in the Use of AVC	370
8.4.6	Filtering Unneeded Runtime Versions	372
8.5	Using the Configuration Microservice in SAP BTP	373
8.5.1	Calling the Configuration Service	373
8.5.2	Administration	374
8.5.3	Extensibility	375
8.6	Variant Configuration in SAP Commerce Cloud	375
8.7	Integrated Variant Configuration in SAP CPQ	379
8.7.1	SAP CPQ as a Leading System	379
8.7.2	SAP CPQ as an Add-On for Integrated SAP ERP or SAP S/4HANA Variant Configuration	380
8.8	Summary	386
9	New Integration Aspects for Variant Configuration	387
9.1	Master Data Distribution with Product Data Replication	387
9.1.1	Challenges and Opportunities	389
9.1.2	Setting Up and Customizing Product Data Replication	392
9.1.3	Additional System Preparation for Product Data Replication	402
9.1.4	Replication Workbench: Objects and Terminology	404

9.1.5	Creating a Configuration Definition and Folder	405
9.1.6	Creating and Exploding the Baseline	407
9.1.7	From Distribution Order to Distribution Unit and Distribution Packet	411
9.1.8	Sending the UPS Packet	411
9.1.9	Posting the Packet	413
9.1.10	Correction Packets	414
9.1.11	Replicating Condition Records	418
9.1.12	Product Data Replication Add-On for Routings	419
9.1.13	Troubleshooting and Tips and Tricks	420
9.2	SAP Variant Configuration with SAP Teamcenter	424
9.2.1	The Strategic Partnership between SAP and Siemens	425
9.2.2	The Next Generation of SAP Teamcenter Integration	426
9.2.3	Current Status and Outlook for End-to-End Variant Configuration	427
9.2.4	A Real-Life Example: Consistent Variant Configuration of a Configurable Valve	428
9.3	Requirements Management and Documentation in Modeling: SAP Enterprise Product Development	432
9.4	Machine Learning and Artificial Intelligence for Configurable Products	437
9.4.1	Data Acquisition and Preparation	443
9.4.2	Creating Models/Scenarios and Training	444
9.4.3	Using Machine Learning Models	444
9.4.4	Machine Learning in the Industry Cloud	445
9.5	Integrating AVC with SAP Workflow Management	447
9.6	Two-Tier Scenario: Cross-System Procurement of Configured Products	453
9.7	Summary	456
10	Solutions from SAP Partners	457
10.1	3D Configuration	457
10.1.1	Terminology	458
10.1.2	Functions of a 3D Configuration Integrated into the SAP System	460
10.1.3	3D Configuration, Augmented Reality, and Virtual Reality	461
10.1.4	Real-Life Example: TRILUX GmbH & Co. KG	462
10.1.5	3D Engine	467
10.2	Artificial Intelligence-Optimized Processing of Customer Requests	470
10.2.1	Basic Principles and General Conditions	470
10.2.2	Preliminary Considerations	474

10.2.3	Examples	477
10.3	Variant Table Contents on SAP Business Technology Platform	482
10.3.1	System Architecture	482
10.3.2	The Simple Variant Table Maintenance Application from networker, solutions	483
10.4	Variant Configuration with Machine Learning	492
10.4.1	Machine Learning and Complex Formulas: Bread Baking Example	493
10.4.2	Using Machine Learning in Real Life: “Folder Gluer” Packaging Machine Example	500
10.5	Hybrid System Configuration, End-to-End Automation, and API-Based Integration with SAP Solutions	503
10.5.1	Integrated Variant Configuration with SAP Solutions and External Software Systems	504
10.5.2	Consistent User Experience with Heterogeneous Configuration Engines in Parallel Operation	505
10.5.3	Multilevel Hybrid System Configuration and Orchestration of Heterogeneous Knowledge Bases	507
10.5.4	Service-Oriented Architectures, SAP-Compatible APIs, and Complex Multilevel Data Structures	509
10.5.5	System-Neutral Configuration IDs and Configuration Lifecycle Management	510
10.5.6	End-to-End Digitization and Automation Using Workflow Management and Integration	512
10.6	Summary	513

Part IV Industry Best Practices

11	Case Study: Transformation from LO-VC to AVC	519
11.1	AVC Preliminary Study	519
11.1.1	Data Model Analysis	520
11.1.2	Analysis of PFUNCTIONs	520
11.1.3	Identifying Risks and Challenges	521
11.1.4	Estimated Work Involved	522
11.2	Best Practice Transformation: A Real-Life Example	526
11.2.1	Scoping	526
11.2.2	Planning	527
11.2.3	Implementation	529

11.2.4	Testing	531
11.2.5	Go-Live	531
11.2.6	Hypercare	532
11.3	Summary	532
12	SAP Variant Configuration Communities	533
12.1	Configuration Working Group	533
12.1.1	Objective and Tasks	535
12.1.2	History	536
12.1.3	Organizational Structure	538
12.1.4	CWG Conferences	539
12.1.5	CWG Portal	541
12.1.6	CWG Sandbox Systems	542
12.2	The DSAG Variant Configuration Working Group	543
12.2.1	Interview with the DSAG Variant Configuration Working Group	544
12.3	SAP AVC Customer Co-Innovation Council	546
12.4	Summary	547
	The Authors	549
	Index	559

Index

3D configuration	457, 460
3D data model	463
3D engine	467
3D visualization	459

A

Action	207
Additional data	133
Administration	374
American Configuration Workgroup (ACWG)	536
Americas' SAP Users' Group (ASUG)	533
AND operation	205, 261
Application Link Enabling (ALE) <i>distribution</i>	390
<i>distribution model</i>	392
<i>partner agreements</i>	392, 403
Application Log	415
Application view	130, 168
Arithmetic operators	225
Artificial intelligence (AI)	437, 470
<i>optimized processing of customer requests</i>	470
Assemble-to-order (ATO)	51
Augmented and virtual reality	461
Authorization object <i>C_TCLS_BER</i>	130
<i>C_TCLS_MNT</i>	130
Automation Workflow Integration (AWI) ...	512
AVR services	504

B

BADIs	221
<i>BD_LO_VCHCLF_VCH_SD_VAR_ COPY</i>	265, 331
<i>BD_LO_VCHCLF_VCH_SD_VAR_ DATA</i>	265
<i>BD_LO_VCHCLF_VCH_SD_VAR_ PRCG</i>	265
<i>BD_VCH_HL_PRINTING</i>	265
<i>VCH_HL_MD_DOMAIN_ MODIFY</i>	219, 263, 266
<i>VCH_HL_ON_SAVE</i>	219, 264
<i>VCH_HL_POST_VALIDATE_ ASSIGN</i>	219, 264

BADIs (Cont.) <i>VCH_HL_PRE_VALIDATE_ ASSIGN</i>	218, 264
Baseline	405, 407
Baseline explosion	410
Baseline name	408
Batch job	408
Bill of materials (BOM)	45, 81
<i>alternative</i>	159
<i>assigning a value to a BOM item</i>	273
<i>BOM application</i>	395
<i>class nodes</i>	274
<i>configurable</i>	85
<i>creating</i>	320
<i>dynamic</i>	155
<i>explosion</i>	274
<i>item category</i>	156
<i>maintaining</i>	274
<i>multiple BOMs</i>	154
<i>simple</i>	154
<i>technical type</i>	154
<i>usage</i>	178
Brownfield approach	73
Bundling	55

C

Change number	367
Change service	415
Character format	89
Character format (CHAR)	125, 221–222
Characteristic	81, 89
<i>data type</i>	120
<i>name</i>	119
<i>restrictable characteristics</i>	121
<i>status</i>	119
Characteristic group	130, 168, 197
Characteristics-oriented allocation	45
Class group	130
Class in object dependencies	221
Class item	156
Class management	86
Class name	130
Class node	85, 133, 274
<i>constraint</i>	276
<i>manual replacement</i>	276
<i>procedure</i>	275
Class type	134, 166

- Classification 134
 - Classification of configurable materials 134
 - Cloud first 31
 - Communication structure SDCOM 98
 - Comparing characteristic valuations 69
 - Comparing facets 69
 - Comparing the results 67
 - Comparison operator 480
 - Comparison status 71
 - Computer vision 475
 - Condition 242
 - Condition part 244
 - Condition record 97
 - Condition record PB00 302
 - Condition technique 308
 - Condition type 97
 - determining* 286, 302
 - EK01* 296
 - overview* 304
 - PROO* 284
 - VA00* 284, 304
 - VA01* 284, 304
 - Configurability
 - of the BOM* 84
 - of the material master* 84
 - Configurable item 112
 - Configurable material 70, 357
 - Configuration definition 404–405
 - Configuration engine 268
 - Configuration folder 404–405, 408
 - Configuration model 27, 520
 - Configuration profile 81
 - AVC* 70
 - classic* 70
 - combining* 171
 - creating* 166
 - Configuration scenario 166–167
 - Configuration service 306
 - Configuration simulation CU50 194
 - Configuration structure 112
 - Configuration Workgroup (CWG) 533
 - association* 538
 - Board of Directors* 539
 - CWG blog* 541
 - CWG conference* 539
 - CWG forum* 541
 - CWG portal* 534, 541
 - CWG sandbox system* 542
 - document directory* 541
 - Executive Committee* 539
 - president* 539
 - statutes* 535
 - Configuration-ID Service (CDIS) 511
 - Configurators 39
 - Configure-to-order (CTO) process 51, 437
 - Configuring 80
 - Consistency check 240
 - Constraint 207
 - anticipatory* 247
 - class node* 276
 - condition* 242, 244–246, 248
 - inferences* 242, 248
 - objects* 242
 - parts* 242
 - restrictions* 242
 - variable* 243–244
 - Constraint net 207, 241, 362
 - Constraint-based 268
 - Correction packet 405, 414
 - Costing 295
 - Costing sheet 294, 302
 - checking* 302
 - Create product folder 405
 - Creating a configuration profile 88
 - Creating a standard order 292
- ## D
- Data model 520
 - Data type 89
 - characteristic* 120
 - Declarative object dependencies 208–209
 - Deep reinforcement learning 475
 - Default value 114
 - dynamic* 256
 - Default value key 280
 - Delta list 343
 - Dependency editor 214
 - Dependency net 207
 - Design-to-operate (DTO) process 424
 - Digital core 31
 - Digital thread 425
 - Direct assignment of variant
 - conditions* 288, 290
 - Distribution order 405, 411
 - Distribution packet 391, 405, 411
 - Distribution type 401
 - Distribution unit 405, 411
 - Document type 68

- ## E
- Edit simulation parameters 110
 - Embedded analytics for classification
 - and configuration* 122
 - Engineer-to-order (ETO) process 35, 42, 52, 156, 163, 181, 184
 - Enhancement spot
 - ES_VCHCLF_VCH_SD_VAR_COPY* 265
 - ES_VCHCLF_VCH_SD_VAR_DATA* 265
 - VCH_HL_CLOUD* 265
 - Event type linkage 401
 - Explosion profile 395, 407
 - Extended graphical rule set 466
 - External procurement 296
- ## F
- FIRE component 507
 - Flag
 - configurable assemblies only* 170
 - material is configurable* 84
 - product is configurable* 314
 - Floating point number 121
 - Framework for Object Explosion (FOX) 395, 407–408
- ## G
- Gecode constraint-solving engine 34, 55, 268
 - German-speaking SAP user
 - group (DSAG)* 533
 - Global object dependencies 210, 241
 - Golden client 388
 - Go-live phase 76, 531
 - Greenfield approach 73
 - Group 130
- ## H
- Header condition 308
 - Headless architecture 507
 - High-level configuration 203–204, 218, 240
 - High-level model 44
 - Hybrid system configuration 503
 - Hypercare phase 76, 532
- ## I
- IDoc (Intermediate Document) 390, 392, 413
 - if condition 245
- Implementation phase 75
 - Inconsistency 116, 245
 - unrecoverable* 262
 - Industry Cloud 445
 - Inferences part 67, 242, 248
 - In-list queries 225
 - Inspector 250
 - Inspector area (simulation) 115
 - Inspector Panel 36
 - Integer 121
 - Integrated automation 503
 - Intelligent product recommendation 447
 - Intelligent scenario lifecycle
 - management* 441
 - Interactive 3D configuration 459
 - Intercompany processing 46
 - Internet Pricing and
 - Configurator (IPC)* 39, 340
 - Item category 178
 - Customizing* 334
 - TAN* 334
 - Item category determination 178
 - Item category group 316
 - 0002* 85
 - 0004* 178
 - 002* 149
- ## K
- Key combination 286
 - Knowledge base 346
 - profiles* 347
 - Knowledge Base Interchange
 - Format (KBIF)* 536
- ## L
- Life cycle phase 404
 - List of permitted values 125
 - Local object dependencies 209, 241
 - Logical operator 225
 - LO-VC
 - availability in SAP S/4HANA* 53
 - AVC versus* 32
 - check object* 63
 - Low-level configuration 203–204, 220
 - Low-level model 45
- ## M
- Machine learning 437, 492
 - Maintaining variant tables 483

- Maintenance task list 138
 - Make-to-order (MTO) process 50, 313
 - Make-to-stock (MTS) process 50–51
 - anonymous* 313
 - Manage class 130
 - Mandatory characteristic 262
 - Manufacturing scenario 50
 - assemble-to-order (ATO)* 51
 - configure-to-order (CTO)* 51–52
 - engineer-to-order (ETO)* 52
 - make-to-order (MTO)* 50
 - make-to-stock (MTS)* 50
 - Master data 47, 342
 - Master data distribution 387
 - MAT 146
 - Material in object dependencies 221
 - Material master
 - configurable* 80, 84
 - views* 145
 - Material number 80
 - Material requirements
 - planning (MRP) 46, 455
 - automatic* 46
 - element* 299
 - Material variant 313
 - converting* 71
 - global* 70
 - plant-specific* 70
 - SDCOM* 79
 - transfer to AVC* 70
 - Message type
 - IDoc* 390
 - UPSMAS* 392
 - UPSRCP* 392, 403
 - Metadomain model 426
 - Microservice 344
 - Microservices (Configuration and Pricing) 306
 - Model creation 65
 - Model once, configure
 - anywhere (MOCA) 306, 339
 - Modeling environment for variant
 - configuration 101
 - Multilevel planning in the sales order 298
- N**
- Names of variant conditions 288
 - Naming conventions 354
 - Natural language processing 475
 - Networking 536
 - Non-valuated characteristic
 - in a precondition* 205, 261
 - in a selection condition* 205, 263
 - Non-variable part 271
 - Number (integers and floating point numbers) 121
 - Number range 402
 - NWS TDS 483
 - NWS VTMV 482–483
- O**
- Object dependencies 44, 81, 94, 167, 245
 - ABAP* 221
 - advanced processing mode (AVC)* 216
 - areas of application* 207
 - assignment* 208
 - constraint* 240
 - declarative* 209
 - document* 221
 - execution sequence* 218
 - for characteristics and characteristic values* 217
 - for classes* 218
 - global* 210, 241
 - in classification* 216–217
 - local* 209, 211
 - overview* 203
 - procedural* 209
 - processing mode* 216
 - search in the classification system* 218
 - semi-declarative* 209
 - simple* 222
 - status* 215
 - valuating* 253
 - value-checking* 240, 242
 - value-setting* 240, 242
 - Object dependency wizard 106
 - Object hierarchy 65
 - Object name 65
 - Object posting status 413
 - Object search
 - in class type* 139
 - in classes* 139
 - Object type 398
 - Objects part 242, 270
 - Operator
 - arithmetic* 225
 - logical* 225
 - string operators* 225
 - Optional characteristic 262
 - OR operation 205, 263

- Order BOM 156, 181
 - creating* 193
 - fixing* 167
 - instantiating* 167
 - managing* 187
 - multilevel maintenance* 193
 - Order BOM browser 193
 - Order Engineering Worklist app 187
 - Order processing 45
 - Order task list 158, 194, 197
 - Order-related construction 45
 - Original document 68
 - Original packet 405
- P**
- Package material 177
 - Packet type 399
 - PFUNCTIONS 75
 - Planning phase 74, 527
 - Plant-specific configuration 316
 - PLM extension 393
 - Precondition 205, 260
 - non-valuated characteristic* 205
 - variant table* 261
 - Preliminary study 519
 - Price condition 44
 - Pricing 91
 - based on sales order costing* 283
 - in sales* 283
 - purchasing* 296
 - sales order costing* 295
 - valuation-dependent* 97
 - value-dependent* 283
 - Pricing APIs 309
 - Pricing microservice 283
 - Pricing service 306
 - Procedural object dependencies 208–209
 - Procedure 206, 253, 278
 - class node* 275
 - Procedures on characteristics and characteristic values 219
 - Process code 403
 - Processing mode 88, 216
 - classic* 59, 273
 - extended variant configuration* 59
 - object dependencies* 216
 - Processing sequence 253
 - Procurement 46
 - Product characteristics 47
 - Product configuration 27, 50
 - Product configuration in the process view ... 41
 - Product data replication 58
 - delta filtering* 411
 - packet* 405
 - packet posting* 413
 - packet transfer* 411
 - replication of a VC model* 411
 - Product development 44
 - Product lifecycle management (PLM) 41
 - Product management 44
 - Product model 43, 350, 353
 - AVC model* 57
 - exploding* 290
 - LO-VC model* 57
 - transforming* 57
 - Product structure 193
 - Product variant 50
 - Production 46
 - Production order change service 154
 - Production version 159, 272, 327
 - Profile name 137
 - Program planning 45
 - Project team 527
 - Purchase requisition 299
 - Purchasing 296
 - Purchasing info record 302
- Q**
- Quality management 46
- R**
- Real-time 3D visualization 458
 - Reasons for changeover 53
 - Reference characteristic 98, 125, 278
 - assigning* 305
 - case sensitivity* 285
 - SDCOM_VKOND* 284
 - Reinforcement learning 476
 - Replication table 399, 415, 422
 - Replication workbench 404–405
 - Report
 - RUPSPPOST* 414
 - RUPSSSEND* 412
 - Representational State Transfer (REST) 373
 - Requirements and stock list 297
 - REST API 505
 - RESTRICTIONS 270
 - Restrictions part 242, 245
 - Root material \$Root 59
 - Routing 81, 112
 - dynamic* 158

- Routing (Cont.)
 - material TL assignment* 327
- Runtime version 220, 275, 342, 347
 - regenerate* 365
- S**
- Sales order (SET) 91, 177
- Sales order costing 295
- Sales pricing 306
- Sales relevance 174, 178
- SAP Advanced Variant Configuration
 - and Pricing 40, 42
- SAP Business Technology
 - Platform (SAP BTP) 31, 306, 441
- SAP Business Workflow 401
- SAP Commerce Cloud 306, 353, 375, 505
- SAP CPQ 38, 40, 306, 375, 379, 505
- SAP Enterprise Product Development ... 44, 432
- SAP ERP
 - components* 28
 - database* 28
 - difference to SAP S/4HANA* 28
- SAP Fiori 29–30
- SAP Fiori app
 - Add Info Record* 302
 - Change Configuration Profile* 162
 - Change Info Record* 302
 - Create Configuration Profile* 137, 162
 - Create Material MM01* 314
 - Create Sales Orders (VA01)* 292
 - Exclusion of Characteristics for Type Determination* 335
 - Generate CDS Views for Classification/Configuration* 444
 - Maintain Bill of Material* 319
 - Maintain BOMs* 214
 - Manage Characteristic Groups* 200
 - Manage Class* 86
 - Manage Classes* 129
 - Manage Order BOM* 191
 - Manage Product Master* 135, 314
 - Order Engineering Worklist* 187
 - SAP Business Application Studio* 451
 - Simulate Configuration Models* 102, 110
 - Simulation Environment* 36
 - Stock/Requirements List* 297
 - Variant Configuration - Overview* 102
 - VC Modeling Environment* 162
 - Worklist Order Engineering* 35
- SAP Fiori launchpad 484
 - Order BOM group* 187
- SAP HANA 29
- SAP Integration Suite 454
- SAP Note
 - 1049251* 393
 - 1103146* 393
 - 1113681* 420
 - 1518479* 393
 - 1529481* 345
 - 1537346* 345
 - 1574296* 403
 - 1819856* 344–345
 - 1924092* 419
 - 2116263* 402, 421
 - 2210904* 34
 - 2214906* 34
 - 3060829* 217
 - 318927* 403
 - 651112* 345
 - 664274* 345
 - 837111* 345
- SAP PLM 512
- SAP S/4HANA
 - basic technical principles* 28
 - database* 29
 - definition* 30
 - difference to SAP ERP* 28
 - disabled functions* 33
 - modes of operation* 31
 - obsolete transactions* 34
 - roles and authorizations* 29
 - user interface* 29
- SAP S/4HANA Cloud 525
- SAP S/4HANA Cloud, private edition 524
- SAP S/4HANA Cloud, public edition 524
- SAP S/4HANA for advanced ATP 45
- SAP Solution Sales
 - Configuration (SSC) 340, 537
- SAP Teamcenter by Siemens 424
- SAP Workflow Management 447, 512–513
- SAP_VF_NUM_TO_CHAR() function 66
- SAP_WFRT (workflow user) 411
- Scoping 526
- Scoping phase 73
- Selection condition 205, 271, 273
 - non-valuated characteristic* 205
- Semi-declarative object dependencies 209
- set_pricing_factor 291
- Setting a default value 226
- Siemens Teamcenter Gateway for
 - PLM System Integration (T4ST) 425
- Simplification list 32–33
- Simulating a configuration model 92, 110

- Simulation environment 36
 - general* 110
 - header* 110
 - settings* 111
- SKEY 119
- Software as a service (SaaS) 31
- Solution order 55
- Solution quote 55
- Source compatibility 65
- Source profile 70
- Special functionality 240, 248
- Standard network 138
- Start logo (configuration profile) 169
- Stateful service 308
- Stateless service 307
- Status of a characteristic 119
- Status of object dependencies 215
- Strategy group 85, 152, 316
- String concatenation 225
- String operator 225
- Structure
 - MAAPV* 147
 - SDCOM* 284
 - VCSD_UPDATE* 147
- Structure area (simulation) 112
- Supervised learning 475
- Syntax element
 - ?= 226, 241, 254, 256–257
 - +, -, /, *, ** 225
 - <, LT 225
 - <=, =<, LE 225
 - <>, ><, NE 225
 - =, EQ 225
 - >, GT 225
 - >=, =>, GE 225
 - // 225
 - \$count_part 227
 - \$count_parts 256–257
 - \$del_default 226, 256
 - \$del_user_val 227, 257
 - \$del_val 227, 257
 - \$parent 223
 - \$PARENT. 223
 - \$part_of 227
 - \$root 222
 - \$ROOT. 222
 - \$self 222
 - \$SELF. 222
 - \$self.characteristic =
 - \$self.characteristic +1* 256
 - \$set_default 226, 256
 - \$set_pricing_factor 227, 257, 291, 306
- Syntax element (Cont.)
 - \$subpart_of* 227
 - \$sum_part* 226
 - \$sum_parts* 256
 - abs* 225
 - and* 227
 - AND, OR* 225
 - arcsin, arccos, arctan* 225
 - as* 243
 - assigned_vals* 228, 257
 - ceil* 225
 - EXIT* 217, 228, 255, 258
 - exp, ln, log10* 225
 - false* 245
 - floor* 225
 - frac* 225
 - FUNCTION* 64
 - IF* 225
 - if* 245
 - in* 225
 - is_a* 243
 - is_object* 243
 - LC* 225
 - max* 227, 257
 - mdata* 226, 280
 - min* 227
 - no_of_assigned_vals* 228, 258
 - NOT* 225, 227
 - or* 227
 - prefix(x, y)* 227
 - round(x, i)* 227
 - SET_PRICING_FACTOR* 248
 - set_pricing_factor* 227, 291, 306
 - sign* 225
 - sin, cos, tan* 225
 - SKEY* 119, 227
 - specified* 226
 - sqrt* 225
 - substring(x, y, z)* 227
 - suffix(x, y)* 227
 - to_int(x)* 228
 - To_String()* 66
 - to_string(x)* 228
 - trunc* 225
 - type_of* 226
 - UC* 225
 - where* 244
- System Configuration Engine (FIRE) 507

T

Table	
PLFH	278
PLFL	278
PLPO	278, 280
PLPO_CFMOD	278
STPO	278
Table constraint wizard PMEVC	106, 108
Target profile	71
Target system	407
Task list type	157
Taxes	309
Tender item	480
Test phase	75, 528
Testing in Transaction PMEVC	87, 104
Tool from the classification system	202
Trace	115, 117
Transaction	28
003	130
008	130
BD64	392, 394, 403
BD87	413
BF11	421
CA01	197, 323
CA02	327
CLO2	86, 129
CL30N	319
CLGT	278
CRWBD	404–405, 408
CS01	33, 319, 321
CS02	33
CS40	321
CSKB	193
CT04	119
CU_EXCL_CSTIC	335
CU34	342
CU41	137, 162, 166
CU42	162, 166
CU50	110, 194
CU51	34, 193
CU51E	34
CU52	34
CU52E	34
CU55	34
CU61	193
CU80	34
CUMODEL	34, 101
EXPO_TEST	411
for the transformation	74
MDO4	297
MD50	298

Transaction (Cont.)

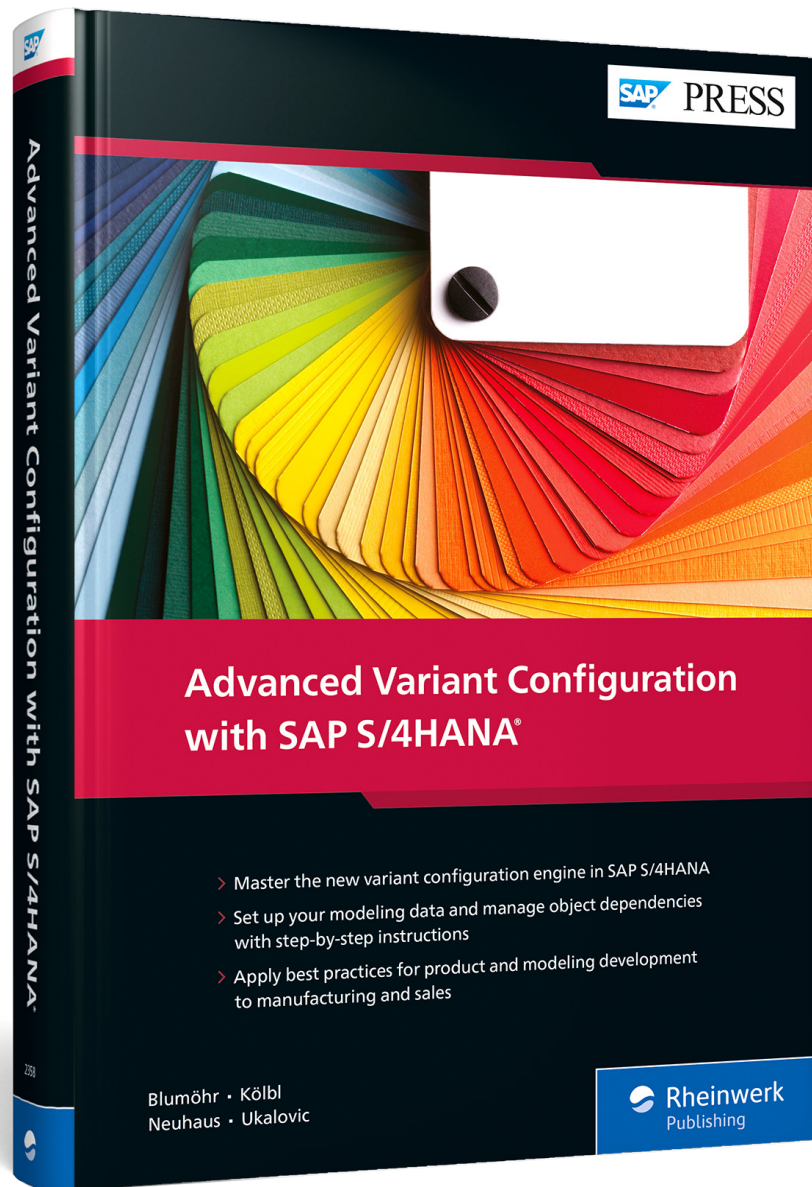
ME11	302
ME12	302
ME21N	299
MMO1	314
MMO2	84
OO03	130
OO05	119
OO08	130
OO17	124
O1CL	140
OBOM_WB_MAIN	193
obsolete	34
OPPQ	197
PMEVC	34–35, 62, 101, 119, 162, 272, 342, 530
SE80	28
SFW5	393
SLG1	415, 421
SM30	418
SM58	421
SO16	402
STO1	421
SWDD	420
SWII	420
SWPC	420
SWPR	420
SWU3	394
UPS	413
UPSRCP	421
VA01	292, 295, 297
VCH_L2A_COMPARISON	68
VCH_L2A_WORKBENCH	528–529
VCH_L2A_WORKSPACES	60, 528–529
VCHMOVCOMP	75, 528, 531
VCHMOVCOPY	528, 530
VCHMOVMMVAR	70, 528
VK11	286, 291
VK30	289
VOV7	334
VOV8	296
WE20	403
Transformation	
best practices	526
case study	519
challenges	521
constraint	66
estimation of work involved	522
from LO-VC to AVC	519
object dependencies	66
product model	57
steps	58

Transformation tools	527
Transition objects	63, 65
Transition Workbench	62, 529
Two-tier scenario	453
Type of dependency	204
constraint	207
precondition	205
procedure	206
selection condition	205
Unsupervised learning	476
UPS packet	405
Valuation	
hard	206, 254
soft	206, 254
Valuation interface	80, 197
parameters	94
settings	94
Valuation option in the variant table	249
Value restriction	133
Value-dependent pricing	283
Value-setting object dependencies	278
Variable	243
Variable size	156

Variant BOM	154
Variant class	81, 86
Variant class type	163, 168
Variant condition	97, 284, 286, 304
assigning	288, 305
creating	302
description	289
description name	289
name	288
Variant configuration	
extended	88, 91
overview	36
Variant configurators	27
Variant function	221
Variant matching	328
Variant part	271
Variant pricing	98–99
Variants, table and procedure	254
VC modeling environment	62, 86, 101
Vendor quote	383

W

WF-BATCH (workflow user)	411
Where-used list for characteristics and characteristic values	141
Where-used list for classes	142
Workflow Customizing	401



Blumöhr, Kölbl, 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.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.