SIEMENS

Teamcenter 10.1

Product Configurator Guide

PLM00188 • 10.1.3

Contents

Chapter 1: Getting started with Product Configurator

Overview

In a typical organization, only a very few employees define and control the overall product suite, such as the product lines and product models that are offered to the market, the features and options that are relevant for each, and the rules specifying what options should be offered or restricted in combination for an orderable product variant. These users are typically product managers or product marketing owners. Configurator experts or configurator administrators may be responsible for creating and maintaining the configurator rules base.

Configuration experts use the Product Configurator to define the variability against a nomenclature with which designers are familiar. The engineers and CAD designers must be aware of the variability their design supports. These users typically need to assert constraints only to express when a given set of options requested by product management or product marketing is not feasible for technical, financial, or productivity reasons.

An organization can maximize consistency and reuse of the variability that exists across their entire product suite by collecting all options into one or more corporate dictionaries. These dictionaries are then used as the basis for which product management and marketing declares what set of this variability is valid for a given product line or product model.

In terms of *variability*, option families can be thought of as the questions that are asked as a part of configuring a given product variant. For example, “What color do you want?”. Option values can be thought of as the allowable answers to those questions. For example, “Given the choices of blue, silver, or white colors, I choose silver”.

The configurator rules that express how these option values can coexist in a valid configuration are stated in terms of defaults, that is, suggested values that can be overridden by the user, or option value combinations that must or must not be selected in a combination.

Options and configurator rules that are valid for a particular product line are collected into a product context. All of the options and rules collected by this product context can exist and be independent of any content, such as designs, parts, processes, partitions, requirements, and documents. For example, you may define the variability of a product at the concept planning stage before you start any CAD designs. Availability of variant data at an early stage in the design process allows CAD designers to concentrate on developing solutions without needing to define variants.

Using the Product Configurator allows you to create comprehensive variant data for various Teamcenter applications in a single location. These definitions are used by other applications, for example, 4G Designer, in which you apply the defined variability to the CAD designs. You can also create items to represent variant option libraries and products in other applications (typically, in My Teamcenter) and send them to the Product Configurator.

The Product Configurator allows you to define and manage the following:

- Option families can be defined as any of the following types:
  - *Text* (lexicographic): option families where 9 is greater than 10 and 9E0 is greater than 1E1.

  - *Floating point:* option families where 9.0 is less than < 10.0 and 9.0E0 is less than 1.0E1.
  - *Integer*: option families where 9 is less than < 10 and 9E0 is less than 1E1. The scientific notation of “1.0E2” to represent 100 is a valid integer, but the scientific notation “1E-2” to represent 1% or 0.01 is not valid. The expression “Family>9 & Family <= 10” is equivalent to “Family = 10”.
  - *Date* option families using the ISO 8601 format with time zone identification. For example, option families where 2013-09-10 is less than 2013-10-09.
  - *Boolean* (logical) value that can have one of two states, for example, On or Off, True or False, and present or absent. Logical values only support operators “=” and “!=”.

    An empty value assignment such as “MyFlag = ""” is equivalent to “MyFlag != MyFlag”, and is therefore a valid assignment even if the family is mandatory.

    You cannot add or allocate values to a logical variant option. Therefore logical variant option families support neither free form or multiselect values.

- Option families may have the following behaviors specified:
  - *Discretionary* (selection is optional, even for a fully complete configuration) option families where an empty value ('') assignment means *none of* and UNSET means *any of*.
  - *Mandatory* option families where an empty value ('') assignment causes a criteria validation violation (except for logical values) with the message associated with L18N key **k_variant_criteria_outside**, and UNSET means *any of*.
  - *Free-form value* option families, in contrast to fixed enumerated value option families. You cannot add or allocate fixed values to free-form value option families. Free-form value option families cannot be multi-selection option families.
  - *Multi-selection* option families, in contrast to single selection option families whose values are mutually exclusive. *Multi-selection* value option families cannot be free-form option families.
- Option values are workspace objects with their own names, description, and owner.
- *Fixed default* and *derived default rules* specify default option values to use during configuration. Fixed and derived default rules are workspace objects with their own names, description, and owner. Derived default rules may have applicability conditions that allow you to differentiate between product models when they are applied.

  The marketing and sales organizations typically use these variant rules to specify allowed combinations of features for the product. However, such configurations are limited to those that the design engineer permitted when creating the variant data on the structure.
- *Product contexts* collect the options and configurator rules relevant for a given product line.
- *Dictionaries* of option families and values.
- *Inclusion* and *exclusion* rules (variant constraints) that express valid and invalid combinations. Constraints are workspace objects with their own names, description, and owner. The associated error messages may be localized.

- *Option family groups* are used to organize option families for view and navigation, as well as used to lead user through a guided interactive configuration.
- *Package families* allow user to define a business-relevant collection of option values that should be selected together based on selection of the package family itself.
- *Package product models* allow defining of what option values necessarily comprise the base definition of the market-facing product model; these option values automatically get selected upon selection of the product model itself.
- *Product models* representing a market-relevant product offering. You can only create a product model in a product context. You can define which options are allowed and available for a given product model.
- *Summary families* allow user to refer to any of a set of summarized values with a single statement in a configurator rule or usage expression.
- *Summary product models* are similar to summary option families. A summary product model allows efficient reference to any summarized product model in a single statement.

Delivering the product variability your customers demand

Increasing expectation of product variability

Customers expect and demand not just a product that meets their functional needs, but one that is tailored to style, color, special features and conveniences of choice. Managing this range of offers efficiently as part of your product definition is a must. You can achieve it by:

- Predicting and offering the variety that your customers demand with specific, carefully targeted product variants.
- Allowing your customers more flexibility in the features and options they select.

The challenge of choice

Managing a distinct product data for each variant of a product may be feasible if you have a small number of product variants. However, as the product evolves, you may find that you are spending an increased amount of time managing and updating many configurations.

Leveraging the power of commonality

Across all variants for a given type of product, there is typically a large amount of data common across many or all variants. Harnessing this knowledge to formally manage this variability directly within your product data allows great efficiency in defining and managing your product over its lifetime.

Product architecture drives products across platforms and generations

As products evolve and new generations of products are introduced, you can tap into and reuse your product knowledge by managing a template of product breakdowns. That template describes the functions, systems, logical elements or other breakdowns that are common across products or generations of products. This establishes a framework for reuse, product planning, and accountability for completeness of product definition relative to the expected content and variability. In addition, it enables you to formalize a corporate standard for how products are organized, planned, and delivered to support a repeatable measure of completeness and maturity as a product definition evolves.

Formally manage your product portfolio

You can introduce your product decisions by defining and marketing your product suite to your customers. These decisions flow from product planning through to engineering and production. It is important to have a formal representation of this product breakdown available to guide and account for work throughout the process. Reuse and applicability allow you to manage variability across your product portfolio and to guide what your customers are exposed to, what is expected and allowed to leverage.

- Reuse to manage a corporate set of features that can be selectively leveraged as new products and new product generations are introduced.
- Applicability to ensure that within a broad library of features that are offered across the entire product line, there is a limited set that is relevant and allowed for a given product.

Engineering on demand

Even after carefully defining and presenting the variability that you wish to offer to your customers, you may allow them to configure orders for which not all of the engineering definition is complete. In this case, typically for specialized products, you can offer a range of dimensional variability and geometric or engineering dependencies. The variability intelligence you manage includes these engineering heuristics. This allows you to *Engineer to Order* some percentage of the parts for an ordered configuration on demand after receiving the order.

Voice of the customer drives product decisions

Ultimately, product decisions are made based on what customers demand and will pay for to make the product successful and profitable. Often there is a disconnect between capturing customer demands and how these get evaluated and approved in the product planning cycle from how this gets communicated and incorporated within the engineering product definition. Product managers, product planners, and the engineers and designers must be able to seamlessly handoff and share their work, to perform impact analysis, and be accountable throughout this process.

Separating product variability from product engineering

Variability is recorded against the product content and is tied to the engineering definition of the product. That is, options and configurator constraints are defined using nodes in your product as the context. This leaves little autonomy for the product managers who are doing early product planning to reflect the voice of the customer and determining what features and products to offer. Product managers need the ability to formally consider different features, review, evaluate, whittle down, and, finally, decide what options should be offered to the market before engineering needs to get involved, and without the need to expose product management to the engineering definition of the product.

Why Product Configurator in Teamcenter?

Teamcenter is a source for the engineering definition of a product. Often the upstream market-driven decisions that drive engineering activities are very disconnected from the engineering work. Connecting these parts of the process and enabling the same level of configuration management rigor and control for product variability by utilizing the engineering product definition enables the entire product definition to be managed consistently in coordination.

Before you begin

Prerequisites

You need one of the following licenses to use the Product Configurator application.

- Advanced Product Configurator Author

  This license enables you to manage variability for your product model data.

  You can define configurator options, rules, and (if applicable) option dictionaries and product line breakdowns including controlled introduction and obsolescence of options for each product model within your product suite.

  Two levels of Solver capability allow you to manage variability from *basic* to *advanced* as follows.

- Advanced Product Configurator Level 1 Solver

  This license enables all of the Product Configurator functionality except for the ability to create product model data. An Advanced Product Configurator Level 2 Solver license is required to create product model data and specify explicit availability of option and rules.

- Advanced Product Configurator Level 2 Solver

  This license enables all of the Product Configurator functionality including the ability to create product model data. The additional functionality included with the Advanced Product Configurator Level 2 Solver license but not with the Advanced Product Configurator Level 1 Solver is as follows.

- Creating product model families
- Creating product models
- Controlling introduction and obsolescence of options
- Specifying explicit availability of options and rules per product model

Enable Product Configurator	Product Configurator does not need to be enabled before you use it. If you have trouble accessing Product Configurator functionality, see your system administrator; it may be a licensing issue. **Note** You can log on to Teamcenter only once. If you try to log on to more than one workstation at a time, you see an error message.
Configure Product Configurator	Before using the Product Configurator, ensure the preferences listed in *Administering Product Configurator* are set correctly.
Start the Product Configurator	Click **Product Configurator** in the navigation pane. You can also start the Product Configurator by right-clicking a product context item revision, variant constraint, group, value, default or saved variant rule in My Teamcenter, and then choosing **Send To** → **Product Configurator**. Teamcenter opens the Product Configurator and displays the selected object in the appropriate view.

Basic concepts for using Product Configurator

Basic concepts for using Product Configurator

The Product Configurator allows you to specify the variant data for a product at the concept stage before any detailed design is done. The variant data is independent of any application domain, for example, CAD design or part planning.

You create option families (for example, **color**) and allowed values of those option families (for example, **red** and **blue**). You then define a variant condition (for example, **only load IF option color = value red** is specified in the variant rule) on that content that is conditionally valid for a configuration based on selected options.

The variant data is collected within an item revision that represents the product line which is called a product context. For example, you can associate the product context which collects option families, values and configurator rules with the top level collaborative design. You can then define variant conditions on partitions and design elements.

To configure a particular variant of an assembly or product, select and apply an adhoc set of option selections or a saved variant rule (a group of option families and values such as **color = red**, **material = cotton**). A saved variant rule is stored in the database and retrieved later.

To suggest initial values or specify option values or combinations that are *not* allowed, you can create configurator rules (for example, an exclusive configurator rule may specify to **error if color = green AND material = cotton**).

Such exclusions and inclusions are referred to as *constraints*.

The *variability* of an item revision is defined as a complete set of option families, each with a set of allowed values (for example, **color = blue, silver, white**). You specifically associate this variability with the product. *Variance* is defined as the option combinations that are available for a scope of the product.

CAD designers then create design solutions that satisfy the variance criteria and associate them to the appropriate design assembly or module, selecting the applicable variance. You can apply different variant configurations to the design data for procedures such as digital mockup analysis or interference checking.

Objects and data you work with

You should understand the purpose of the following objects and data you may manage in Product Configurator.

100% BOM

The “as sold” product configuration, for example, the configuration of car that will be built and shipped to the dealer.

120% BOM

A partial overlay of selected variant configurations. You cannot build the product from a 120% BOM.

150% BOM

Overlays of all possible variant configurations. You cannot build the product from a 150% BOM.

Allocation

The use of option values, option families, and option family groups in a specified context. Option values may be shared across multiple product contexts or dictionaries.

- An option value is automatically allocated to the product context or dictionary selected when it is created, together with its corresponding option family and option family group.
- When an option value is selected for allocation, only the selected option value is allocated. Other option values within the same family are not allocated.
- When an option family is selected for allocation, all option values within that family are allocated. If an option family is already allocated and a user attempts to allocate the same option family again, there is no effect unless there are previously unallocated option values.
- When an option family group is selected for allocation, all option families and option values within the group are allocated. If an option family group is already allocated and a user attempts to allocate the same option family group again, there is no effect unless there are previously unallocated option values.
- It is only possible to deallocate (completely remove) an existing allocation from a product context if the value is not referenced.

Allocation source

The source of an allocation is the product context or option dictionary from which the object is selected.

Allocation target

The target of an allocation is the product context or option dictionary to which the object is allocated.

Applicability

An optional statement for any configurator rule that defines preconditions for which this rule will be relevant. If no applicability statement is specified, the given configurator rule is valid for any configuration for a given product context. If there is an applicability specified, the rule is not evaluated if the applicability statement is not met.

Configuration context

A set of recipes or filters that are applied to the product data to configure the desired variant. These recipes may include maturity, option family selections, and effectivity.

Configurator product context

An item that defines the scope of the option families and values for a product. A product context may represent the product, product line, business unit or operating unit in which variant option values are allocated and rules are authored. Configurator rules may apply to several product contexts (that is, to product models across contexts), but do not refer directly to the product context. There must be at least one product context in the system, but you may manage more product contexts to suit your business needs.

Configurator rule

A Boolean expression recognized by the Product Configurator. It represents a statement to evaluate, specify, restrict, or include allowable option combinations or model applicability.

Default option value

A value that is preset for an option family by the system before the user actively makes any configuration selections.

Derived value default

A logical condition added to a default option value so that the option only defaults under certain conditions. For example, set option 1 to A only if option 2 is set to B, but the user may still make a choice other than A for option 1. Teamcenter evaluates constraint rules before evaluating default rules.

Discretionary option

The configuration is still valid even if no selection is provided for this option. Sometimes called an *optional option*.

Exclusion rule

A variant expression restricting the combination of option values that can be selected together under specified conditions. If the constraint expression is satisfied, the rule fails.

Free form option family

An option family that defines an allowed range for the user input including one or two Boolean operators. During order string creation, the user or system may specify a value for the free form option family

Inclusion rule

A variant expression that sets one or more option values if any stated pre-conditions or post-conditions are satisfied. It provides an Add to Order string. It comprises an optional Boolean pre-condition, a Boolean inclusion statement that specifies a value to set for one or more option values, and an optional Boolean post-condition.

Mandatory

A selection must be made to have a complete, valid configuration.

Marketing option

An option that is valid for exposure to the Product Configurator user and is applicable globally. It is defined independently of product data, but may be referenced directly by product data or may derive one or more technical option values that are referenced within the product.

Note

A formal distinction between engineering and marketing options is not supported in this version of the Product Configurator.

Model Applicability

An optional statement for any configurator rule that defines one or more product models for which this rule will be relevant. If no model is specified, the given configurator rule is valid for all models within the product context. If there is a model applicability specified, the rule is not evaluated if any other product model is selected.

Multi-selection option family

An option family for which more than one option value may be selected for an order string.

Mutual exclusivity

Specifies when a set of possible selections may only have a maximum of one selection for a valid configuration. Mutual exclusivity often refers to the values of an option family, that is, only one value may be selected for a given option family. However, it may also apply more broadly, for example, if the user selects a value for one option family, the system then disallows selection from one or more other option families.

Namespace

The namespace ensures all option families and values in the system are sufficiently unique that the Product Configurator rules solver can always unambiguously refer to a namespace, option family, and option value. All option families have a namespace as a required property. Option values inherit their namespace from their option families.

Option family

Option families are a method of grouping option values with a similar purpose. For example, you can create an option family called **color** with option values of **red**, **green**, and **yellow**. An option family may be allocated to one or more dictionaries or product contexts, and family memberships may change over time. A selection from an option family may be optional (allowed but not required) or mandatory (required) for a complete, valid order string.

Option family group

Option family groups are a method of grouping option families with a similar purpose. An option family may belong to more than one group and its group memberships may change over time. All

option values within the family are included in the group, but the option family group to option family relationship is not an allocation or means of filtering option values. Option family groups are associated with at least one product context or dictionary.

Option value

An option value represents a possible value in an option family. An option value belongs to one, and only one, option family. Option values may be freeform or a member of an enumerated list.

Package Option

A combination of individual options grouped together to form a package. Member option values may come from different option families. Package values are set automatically when you select a package. Package option values are connected using a logical AND operation in the variant expression.

Product model

A designated product or subproduct, for example, a vehicle or a major assembly such as an engine or transmission. A product model may only be created in a product context. You can reuse option families, option values, and product data (for example, BVR structure, partition template, and design elements) for more than one product model. The product model may be defined by a combination of option values; in this case, the product model is referred to as a *package model*.

Product model availability

Specifies that options and all option values are designated as valid for a given product model within the product context.

Saved variant rule

An expression that enumerates a selection of option values that may represent a complete or partial configuration. A variant rule must be validated as complete and consistent with no violations to produce a buildable physical variant.

Summary model

A summary model family is a collection of models that allows you to summarize the product models in the product context. A summary model option behaves similarly to a summary option. Summary model option values are connected using a logical OR operation in the variant expression. Summary models may not be nested. Product model members must be from the same product context. Product models may reference one or more option values. A summary model family may be referenced in variant, exclusion, and inclusion rule expressions.

Summary option

A collection of option values which allows you to summarize the values from the single option family in the product context or in the dictionary. Summary option allows you to efficiently author rules by referring to the *summary option* when the rule applies to all of the option values it summarizes. A summary option may be referenced in variant, exclusion and inclusion rule expressions. For example, you can create a summary option for all V6 engines which have individual options as values. A summary option is not visible during configuration.

Technical option

(Non-marketing option) An option that is only accessed by authorized technical users and is referenced directly within the product. Where marketing options are used, the system generally derives technical option values by rules based on the values set for marketing options.

Variant

A specific configuration of the generic product that removes ambiguity and represents a physical product or subproduct that could be manufactured. A variant may not represent a configuration of the full product but, for example, may refer to a single feature combination satisfied for a partition.

Variant configuration

Allows you validate a configuration against constraints such as inclusion and exclusion rules. When you configure and assign values to option families, Teamcenter may evaluate the constraints and return a list of information, warnings, and errors on the various combinations of the families.

Variant condition

A rule that controls when an element of the product data is included in the configuration. If the rule evaluates to true, the qualified product data element is included in the configuration.

Variant option data

Variant option family groups, option families, and option values.

Variance

The permutations of option values offered for a specific design solution or generic product.

Variant expression

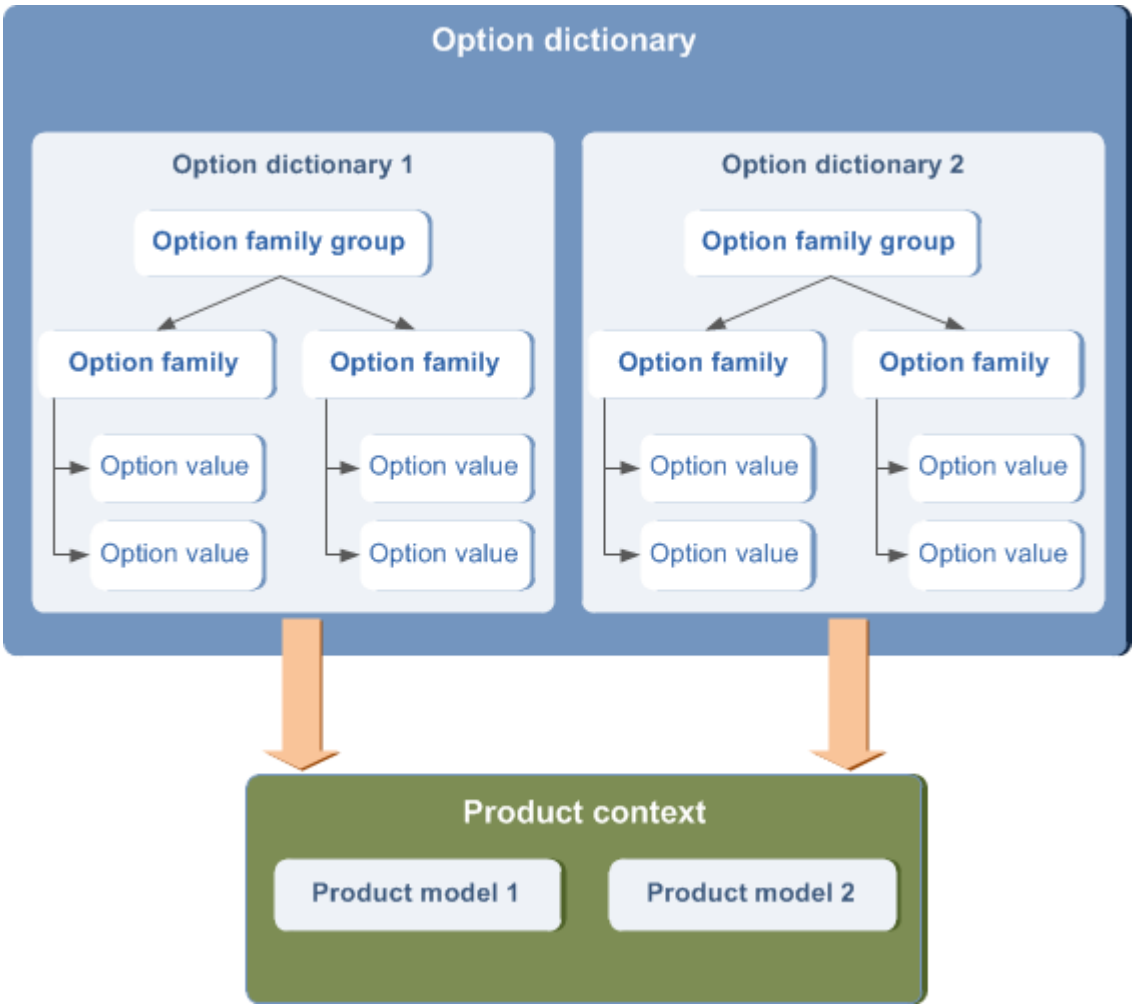
A general term that refers to any variant rule. It may refer to one or more product models, summary or package product models that span multiple product contexts.

What is an option dictionary?

An option dictionary:

- Collects option family groups, option families, and option values, making them available to share and reuse.
- Allows you to formally group or create subsets of option family groups, option families, and option values to reuse between specific products or product lines.
- Allows you to share and reuse option data. You allocate from one or more dictionaries or product contexts to formally create subsets of the option data that is relevant to one or more product contexts or other dictionaries.
- Provides a natural way to segregate data relevant to each user communities, for example, by business unit.

You can create multiple option dictionaries, but the use of option dictionaries is not compulsory.



Note

Your administrator can control who has access to create and allocate objects to the option family group, family, and values in your dictionary.

What is a product context?

A product context references option family groups, option families, and option values. In addition, it allows you to define product models and reference configurator rules. Rules you create with the Product Configurator can apply to product models across several product contexts.

Teamcenter does not *require* you to create product contexts or a dictionary. If your reuse requirements are not complex, you can create a single product context and minimize user interactions with it.

A product context:

- Defines and encapsulates one or more product models.
- Accepts the allocation of option values that you can make available to one or more product models.
- Allows any rule to refer to multiple product models within or across product context items.

- Has no limitations of scope or scalability; the scope is determined dynamically for each product model.
- Provides a natural way to segregate data relevant to each user communities, for example, by product line.
- Ensures that sharing of option data, configuration rules, and product content is limited to appropriate groups and products. Users typically work primarily in a single product context.
- Provides a boundary for enforcement and validation of business rules.
- Enables different business units to operate with a positive or negative bias for option value availability, as appropriate for their defined rules.
- Enables autonomous business units or component groups to share option and rule data when relevant.

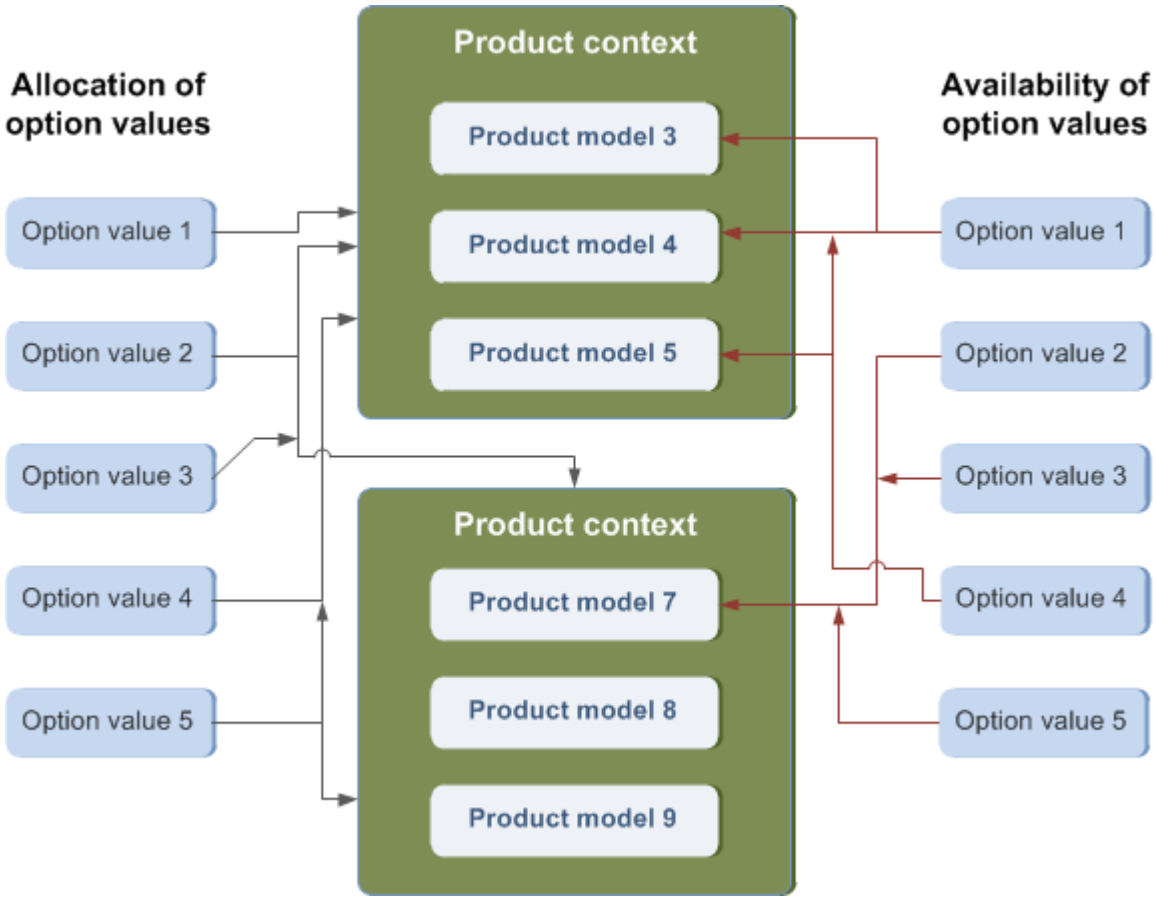
Understanding allocation and availability

The term *allocation* refers to the use of option values, option families, and option family groups in a specified product context. Option values may be shared across multiple configuration product items and dictionaries. An option value is automatically allocated to the product context or contexts selected when it is created.

- When option values are selected for allocation, the option family and option family group are also allocated.
- When an option family is selected for allocation, all option values within that family are allocated, as is the family's option family group.
- When an option family group is selected for allocation, all option families and option values within the group are allocated.

The source of an allocation action is the product context or option dictionary from which the object is selected. The target of the allocation is the product context or option dictionary to which the object is allocated.

Once allocated, you can additionally specify what set of the option families and values allocated to the product context should be available for each product model.



Constructing configurator rules

Teamcenter supports several types of configurator rules, but each type comprises three components:

- Applicability expression

  This is an optional filter that determines if the constraint expression is evaluated and applied.

- Constraint expression

  If other filters are passed, this expression is evaluated and enforced to determine what option values can, must, or must not be selected in a valid option string.

- Model applicability

  This is an optional filter that specifies to which product models this rule applies.

Understanding the configurator namespace

The solve engine requires a minimum level of option family and option value uniqueness to unambiguously run a validation or solve an order. The namespace provides the minimum level through a property on an option family. Users select a namespace from a list of values defined by the administrator.

The role of the namespace includes the following:

- Acts as a boundary for the uniqueness of an option family or option value.
- Is a user-selectable property on an option family that applies to all the associated option values.
- Has no impact on scoping of variability or configuration, other than controlling uniqueness.
- Has no inherent correlation or significance relative to the product context for which an option value is created.

The uniqueness of the default namespace depends on the level of uniqueness needed by the solve engine to always unambiguously understand which option values are considered for validation and solve. The solve engine must always pass an option string that unambiguously identifies each option value.

In some instances, the uniqueness provided by the namespace is insufficient for unambiguous validation: a greater threshold is required. For example, sometimes Teamcenter imposes a level of uniqueness on option family and option value data through the use of namespace that cannot unambiguously run a validation or order solve. The order strings from an external sales configurator or other external system cannot unambiguously correlate each option value in the order string to a Teamcenter option value revision. In this case, you must define a uniqueness threshold to ensure that the incoming order string information can always be correlated to individual option value revisions in the database.

If a greater threshold of uniqueness is needed, you can use multifield keys. That is, you specify a combination of properties that can be enforced as unique across all business objects. In this case, the uniqueness key for an object includes (at a minimum) the UID.

You can use multifield keys to enforce option values that are globally more unique than the solver requires. This ensures the incoming option string does not require the sales configurator to know more than the option value name to provide the solve engine with sufficient information to perform the solve. For example, you may use multifield keys to differentiate between the same option name used by different organizations in your business — **Blue + Truck Division** is not the same option as **Blue + Car Division**. In this example, **Truck Division** and **Car Division** are defined as multifield keys.

An option value name must be unique for the option family. Product model name must be unique in the product context. Additionally, the combination of an option family ID and a family namespace must be unique in the product context.

Basic tasks for using Product Configurator

Basic tasks

Use the Product Configurator to do the following basic tasks:

1. Create new option families.

   Teamcenter creates separately manageable workspace objects for each option value.

   You may create option families and option values for different purposes. For example, a marketing user may have access to create only marketing option families and values. However, a technical user may create both marketing and technical option families and values, depending on your organization's business rules.

(Optional) You may also create package families and summary families.

2. Create option values and add them to new or existing option families.

3. Create rules or constraints that determine when option families or values may or not be made, and also selections that may or may not be made in combination.

4. (Optional) Submit an option value to workflow for review and approval. Any new or revised option values can be the targets of the same workflow. The workflow may apply a status (for example, **Approved**) on completion of the workflow.

5. (Optional) Create a variant option dictionary in which you can collect option families and values. The dictionary provides a convenient method of collecting all the option families related to a particular context, for example, a product. The option dictionary is an item that manages variant option values.

6. (Optional) Create a product context to contain allocated option values and variant rules for one or more products.

   You may also create product models and product model families. You can summarize your product models as summary models and summary model families.

7. (Optional, not performed in the Product Configurator) Create an application model (for example, a 4GD collaborative design) that represents the top-level container of the product.

8. (Optional, not performed in the Product Configurator) Create application models to represent different configuration aspects of the product, for example, design and manufacturing.

9. Assign predefined variability (option families and values) to the product context. You can use option families and values from the variant option dictionary, if one exists.

   - Configure your product using the configuration wizard.

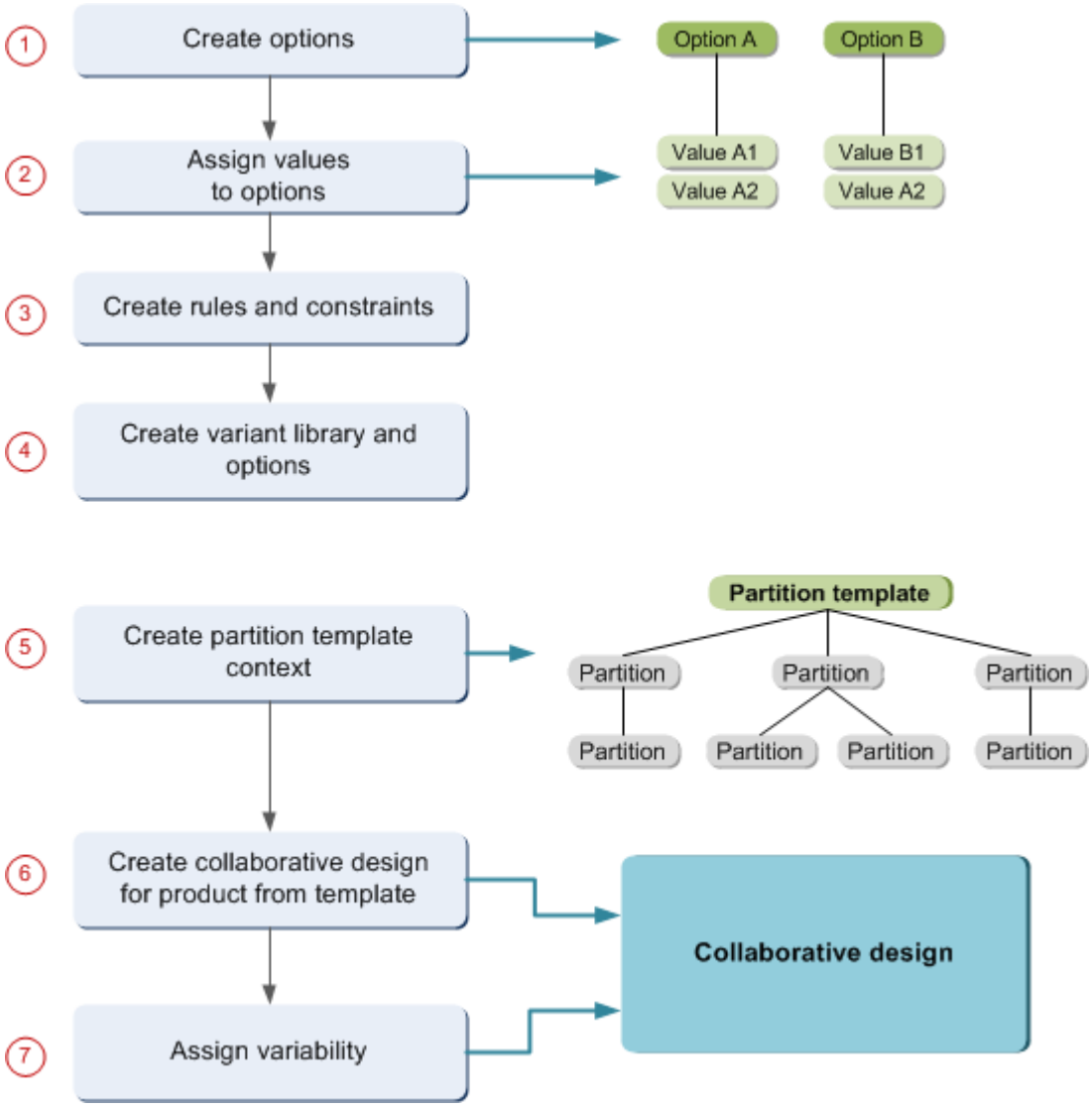
   - Configure your product manually.

10. (Optional, not performed in the Product Configurator) Assign the appropriate variability to individual design assemblies or elements, where this differs from the variability of the product.

11. Perform **Impact Analysis** to analyze the usage and minimize errors.

Example workflow for creating variant data

The following diagram shows a typical process for creating variant data and applying it to a product that is represented by a collaborative design. Optional steps in the previous procedure are omitted.



Administering Product Configurator

Before using the Product Configurator, ensure the following preferences are set correctly:

Preference	Scope	Description
Cfg0CompoundOptionValueMemberTypes	Site	Defines the type of member option values that a **Compound Option Value** can hold.
Cfg0DefaultOptionFamilyNamespace	User	Specifies the default family namespace property value for new option families created in the **Variant Options** view. This preference may contain any string.

Preference	Scope	Description
Cfg0ExpressionGridColumnProperties	User	Specifies a list of property columns that are displayed in the **Variant Expression Editor** view. This preference may contain one or more property names. The object ID is always displayed and should not be added to this preference.
Cfg0ExpressionGridSelectionCharacters	Site	Specifies the list of characters that are displayed in the **Variant Expression Editor** to represent variant expression selections. This preference must contain two characters; the first value represents a positive selection and second value represents a negative selection.
Cfg0EvaluateAvailability	Site	Enables the evaluation of availability rules defined in the product context during configuration validation using the **Variant Configuration** view in Product Configurator application and in consuming applications.
Cfg0OptionsViewColumnWidths	User	Specifies the column widths in pixels of the columns identified in the **OptionsDisplayNameColumnsShownPref** preference. This preference contains a list of integer numbers, one corresponding to each of the columns in the tree table of the **Variant Options** view.
Cfg0OptionsViewHistoryUIDs	User	Contains the history of last 20 objects opened in the **Variant Options** view. This preference is set by the system and should not be edited manually.
Cfg0OptionsViewPropertyColumns	User	Specifies the list of column properties that are displayed in the tree table of the **Variant Options** view. This preference may contain multiple property names.
Cfg0OptionsViewSortProperty	User	Specifies the property column by which the **Variant Options** view tree table is sorted. This preference may contain any property specified in the **Cfg0OptionsViewPropertyColumns** preference.

Preference	Scope	Description
PCA_enable_authoring	Site	Controls whether variant option data can be created and edited in the Product Configurator. If **FALSE**, all data in the Product Configurator data is view only.
TC_CFG_PRODUCTS_USE_VARIABILITY	Site	Enable or disable evaluation of Teamcenter configurator variability allocation statements for a given item revision. Variability allocation statements allow Teamcenter to import only some of the values of a variant option family in a product. If you leave this preference unset, Teamcenter dynamically determines this configuration setting. This existing preference is modified to allow you to view and manage variability allocation statements with the Product Configurator.
TC_Default_SVR_Relationship	User	Defines the relationship type between a saved variant rule and the item to which it refers. By default, this preference contains **IMAN_reference**.
TC_Fnd0SoaConfigFilterCriteria_AutoPublishRevisable	User	Controls whether effectivity changes made in a private edit state are automatically published. If set to **true**, effectivity and variant data changes are immediately visible to other users who have read access.
TC_VariantConfigurable_MaxCacheSize	Site	Configures the variant data cache. The Teamcenter server caches variant data for configurable objects to minimize repeated frequent database queries for the same set of configurable objects. Use this preference to limit the size of the cache. You should choose a value that optimizes the balance between memory footprint of the **tcserver** process and the performance impact of an increased database SQL query count.

Preference	Scope	Description
TC_variant_configurator_relationship	Site	Specifies the relationship between an application model (for example, collaborative design) and a related product context that connects to a variant configurator. By default, this preference contains **Mdl0HasVariantConfiguratorContext**.

You should also set the following business object constants in the Business Modeler IDE:

Business object constant	Defines	Default value
Fnd0CFilterVariantExpressionType	The type of expression objects that objects which can be configured by variants use to store their variant conditions.	**Fnd0VariantExpression**
Fnd0WhereUsed	Whether configurator objects are found by “where used” searches.	**true**
Cfg0DefaultValueType	The type of default option value used by family objects.	None
Cfg0ThreadType	The type of thread used by instantiable revision objects types (a subtype of the specified thread type may also be used). Abstract classes should not define a value for this business type constant.	Empty

Controlling access to configuration data

You can use the **ADD_CONTENT** and **REMOVE_CONTENT** privileges in Access Manager to control user access for adding or removing data from a configurator item. Teamcenter checks that the user has the required privileges on the configurator item when any of the following actions are performed.

User action	Required access
Allocate a group, family, or value to a configurator item	**ADD_CONTENT**
Unallocate a group, family, or value from a configurator item	**REMOVE_CONTENT**
Create a configurator rule	**ADD_CONTENT**
Delete a configurator rule	**REMOVE_CONTENT**
Relate a configurator rule to a configurator item	**ADD_CONTENT**

User action	Required access
Remove relation between a configurator rule to a configurator item	**REMOVE_CONTENT**

Related topics

- Getting started with Access Manager

Sharing configurator data

You can share configurator data by using:

- Multi-Site Collaboration
- **Briefcase** data transfer with the high level TC XML format to transfer objects offline, and
- TC XML low level commands.

Product context and associated objects, such as family groups, families, values, model families, product models, summary options and their members, package options and their members, constraints, defaults, and Saved Variant Rules can be synchronized to the target site. Ownership of the summary and package options must be transferred to the target site when the ownership transfer operation is performed.

You can use Multi-Site Collaboration and TC XML to share configurator data as follows:

- Replicate unconfigured configurator data between sites.
- Synchronize configurator data between sites.
- Transfer ownership of configurator data between sites.

To enable the **Briefcase** data transfer, your administrator must set the following settings in both the source and the target sites:

- In the **Organization** application, ensure the **Uses TC XML Payload** and **Is Offline** check boxes are selected for the sites.
- Set the **GMS_offline_use_TcGS** site preference to false.

Note

In this version of Teamcenter, the use of the TC XML low level features of the **data_share** and **data_sync** command utilities is restricted and requires the **SITCONS_AUTH_KEY** environment variable set to a valid key value.

Product Configurator interface

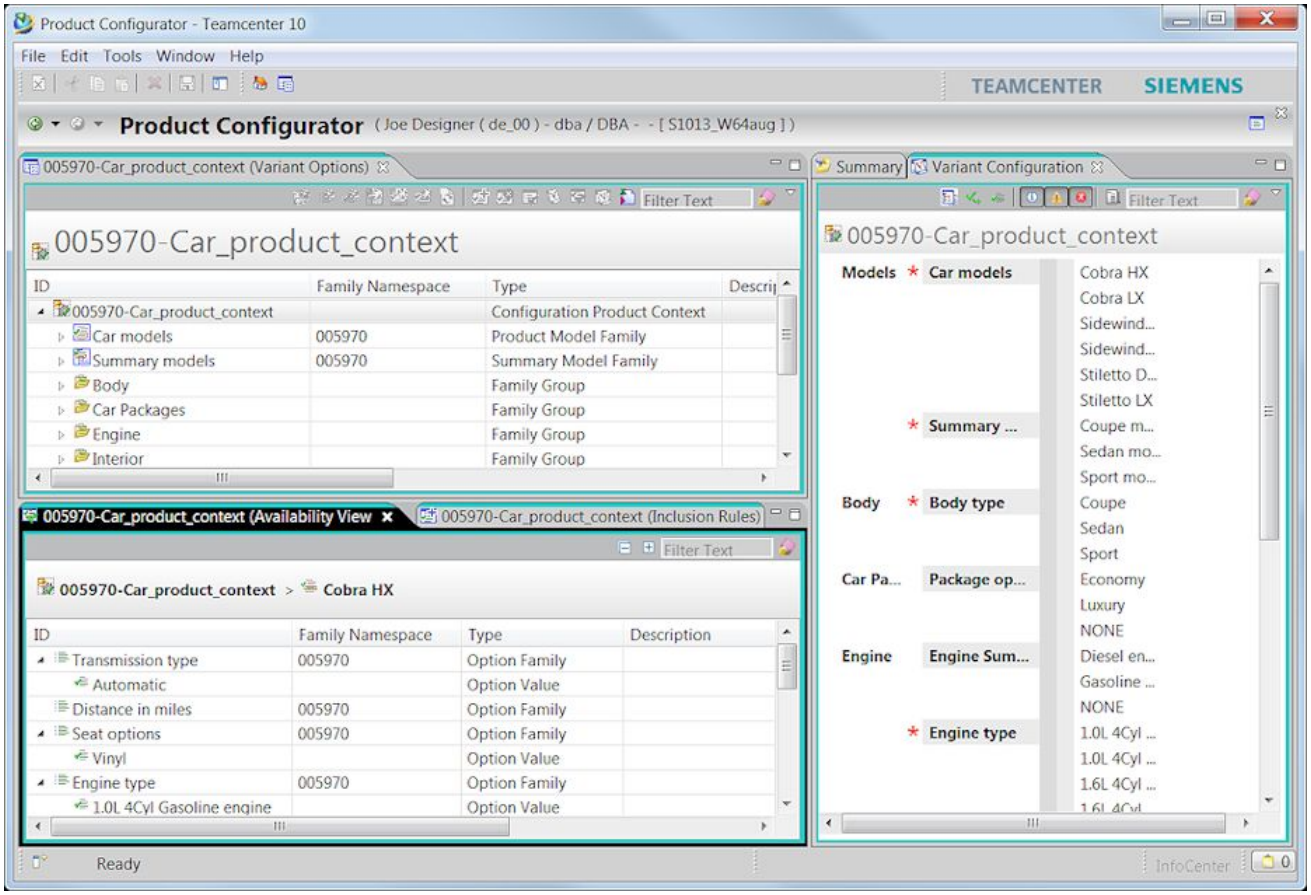
Product Configurator interface

The left-hand navigation bar of the Product Configurator application contains the Teamcenter component view:

- **Search** and **Quick Links**: Provides standard features available with all rich client applications.
- **Open Items**: Provides a list of currently open variant objects. Click an entry in the list to select and make it active.
- **History**: Provides a list of variant objects that you have recently opened. Click an entry in the list to select and make it active.

After launching the Product Configurator application, you can search for an existing variant object, or alternatively open a recently used variant object.

The main area of the Product Configurator comprises:



Multiple views	Each view allows you to manage variant data for a context item. You can also open multiple instances of the same view, one for each open item.

Menu bar	The main *menu bar* allows you to execute common tasks with menu commands.
Toolbars	Allows you to execute common tasks by clicking icon images. *Toolbars* are provided in the main Product Configurator application and in many of the views.

Note

You can modify the layout of the Product Configurator to suit your working practices and personal preferences. You can add or remove views as described in *Teamcenter rich client perspectives and views*. The previous example shows one possible configuration.

Product Configurator menu commands

The following commands are provided on the main menu bar:

Menu command	Purpose
File→New→Product context	Allows you to create a new product context. You can also create a new product context in My Teamcenter and send it to the Product Configurator.
File→New→Dictionary	Allows you to create a new option dictionary. You can also create a new dictionary in My Teamcenter and send it to the Product Configurator.
File→Save	Saves the information added or changed in the current session.
File→Close	Closes the Product Configurator application. You are prompted to save any unsaved changes.
File→Exit	Exits the rich client and all of the applications that are running in the current session.
Edit→Cut	Removes a selected data object from its current location and places it on the clipboard. You must have read privileges on the object and write privileges on its container to move or remove an object. **Note** If you perform a cut action and intend to perform a paste action to move the object, do not initiate another action before choosing **Paste**. If you initiate another action or log off, the selected object is lost.

Menu command	Purpose
Edit→Copy	Copies a selected data object to the clipboard. You must have read privileges for the selected object that you want to copy. You can also create a copy by dragging the object to another location or Teamcenter application. **Note** If you copy an option family and paste it on another option group, Teamcenter performs a move action as an option family may only exist in one group.
Edit→Copy Append	Copies the selected data object to the clipboard with the intention of appending it to the destination object.
Edit→Paste	Moves a data object from the clipboard to the current cursor location in the application. It is important to select the proper destination for the data object before choosing the **Paste** menu command. You must have read and write privileges to the destination object.
Edit→Delete	Deletes a selected data object from the database. You must have delete permission for the object to use this command. **Note** You cannot delete an object that is referenced in multiple locations from the database. Therefore, to delete an object from the database, you may need to perform a where-referenced search to locate all references to the object, and then delete those references. When there are no remaining references, you can delete an object for which you have delete permission.
Edit→User Setting	Provides access to the **User Setting** dialog box used to: • View, define, or change settings for a group, role, and/or volume for the current session. • View or change logging privileges if you are a user with **dba** privileges. • View, define, or change profile information for the current user. • Define or change logging, if you are a user with administrative privileges.
Edit→Options	Lets you set user interface, display and processing preferences for the Product Configurator and other applications.

Menu command	Purpose
Tools→Multi-Site Synchronization→Object	Allows you to synchronize objects to other sites at which they are shared.
Tools→Multi-Site Synchronization→Component	Allows you to synchronize components to other sites at which they are shared.
Tools→Multi-Site Synchronization→Assembly	Allows you to synchronize assemblies to other sites at which they are shared.
Tools→Import→From Briefcase	Allows you to import variant data into the Product Configurator from a Briefcase file.
Tools→Export→To Briefcase	Allows you to export variant data from the Product Configurator into a Briefcase file.
Window→Open Perspective	Allows you to open alternate application perspectives or collections of view panes.
Window→Show View	Allows you to open alternate views.
Window→Save Perspective As	Saves a rearranged perspective with the current name, or creates a new perspective by saving the new arrangement of views with a new name.
Window→Reset Perspective	Restores a rearranged perspective to the default view arrangement.
Window→Close Perspective	This command is not used by default. If enabled at your site, this command allows you to close an alternate perspective.
Window→Preferences	Displays the standard Eclipse **Preferences** dialog box, where you can set Relation Browser preferences, Teamcenter preferences for search and view network presentation, and Web browser preferences.
Window→Toolbar	Displays or hides the toolbar. A check mark indicating that the toolbar is displayed under the menu bar.
Window→Navigation Bar	Displays or hides the navigation pane. A check mark indicates that the navigation pane is displayed.
Window→Full Screen	Alternates between full-screen display and adjustable-window display.
Help	You can use the **Help** menu commands to access the online help library, see application-specific help, and find information about the version of Teamcenter that is currently running.

The following commands are provided on the context (right-click) menu bar. Some commands may be hidden or disabled, depending on the current selection.

Menu command	Purpose
Cut	Removes a selected data object from its current location and places it on the clipboard. You must have read privileges on the object and write privileges on its container to move or remove an object. **Note** If you perform a cut action and intend to perform a paste action to move the object, do not initiate another action before choosing **Paste**. If you initiate another action or log off, the selected object is lost.
Copy	Copies a selected data object to the clipboard. You must have read privileges for the selected object that you want to copy. You can also create a copy by dragging the object to another location or Teamcenter application.
Paste	Moves a data object from the clipboard to the current cursor location in the application. It is important to select the proper destination for the data object before choosing the **Paste** menu command. You must have read and write privileges to the destination object. **Note** If you copy an option family and paste it onto another option group, Teamcenter performs a move action as an option family may only exist in one group.
Generate Report	Creates item reports in the context of the selected objects. Item reports generate in multiple output formats and follow PLM XML standards to allow integration with third-party reporting tools.
Open with	Opens the selected object in the view you choose.
Send to	Opens the selected object in the application you choose.
Check-In/Out	Reserves exclusive access to one or more objects and/or their attachments by locking the objects in the database upon checkout. You restore access to the objects using check-in. Only your administrator is allowed to circumvent the security that these menu commands provide.
Refresh	Updates the selected object in the workspace area with any changed information in the database.
View Properties	Updates the properties of the selected object.
Access	View or update access permissions to the selected object.
Add Group	Allows you to add an option group at the current location. If the location is not valid for the addition of a group, this command is disabled.

Menu command	Purpose
Add Family	Allows you to add an option family at the current location. If the location is not valid for the addition of a family this command is disabled.
Add Value	Allows you to add an option value at the current location. If the location is not valid for the addition of a value, this command is disabled.
Add Model Family	Allows you to add a product model family at the current location. If the location is not valid for the addition of a product model, this command is disabled.
Add Model	Allows you to add a product model at the current location. If the location is not valid for the addition of a product model, this command is disabled.
Multi-Site Synchronization	Allows you to synchronize objects, components or assemblies, depending on your selection, to other sites at which they are shared.
Remove Allocation	Removes the variant data allocated to the currently selected object.

Commands are disabled (grayed out) if they are not appropriate for the current selection.

Product Configurator buttons

The following buttons are provided on the main toolbar of the Product Configurator application:

Button	Function	Description
	Soft Abort	If enabled, allows you to terminate the current operation without closing the Product Configurator or losing data.
	Cut	Cuts the selected data from the Product Configurator and places it on the clipboard. **Note** If you perform a cut action and intend to perform a paste action to move the object, do not initiate another action before choosing **Paste**. If you initiate another action or log off, the selected object is lost.
	Copy	Copies the selected data from the Product Configurator and places it on the clipboard. **Note** If you copy an option family and paste it on another option group, Teamcenter performs a move action as an option family may only exist in one group.

Button	Function	Description
	Paste	Pastes the variant data from the clipboard into the current cursor location.
	Delete	Allows you to delete one of the following selected objects: • Group Teamcenter moves any families that belonged to the group to the **Other** group. You cannot delete the **Model Designator** or **Other** groups. • Value You can select one or more values to delete simultaneously. If the value is referenced in any default rules, constraints, saved variant rules, or variant conditions, Teamcenter does not permit the deletion. • Family You can select one or more families to delete simultaneously. If any values in the family are referenced in any default rules and constraints, Teamcenter does not permit the deletion. • Rules You can delete one or more selected default rules in the **Variant Defaults** view. • Exclusion and inclusion rules
	Save	Saves changes to your data.
	Navigation Pane	Opens or hides the left-hand navigation pane.
	Open Home	Opens the Home view.
	Open the new **Variant Options** view	Opens new **Variant Options** view where you can create or edit product models, variant option groups, families, and values.

The following buttons are provided on the toolbar of the **Variant Options** view:

Button	Function	Description
	Add Group	Adds a new option group below the currently selected level. Shortcut keys: CTRL+G

Button	Function	Description
	Add Family	Adds a new option family below the currently selected level. Shortcut keys: CTRL+J
	Add Value	Adds a new option value below the currently selected level. Shortcut keys: CTRL+L
	Add summary model family	Adds a new summary model family below the currently selected level.
	Add model family	Adds a new model family below the currently selected level. Shortcut keys: CTRL+SHIFT+J
	Add model	Adds a new product model below the currently selected level. Shortcut keys: CTRL+SHIFT+P
	Refresh View	Refreshes the displayed variant data to show the impact of recent changes. Shortcut key: F5
	Open **Inclusion Rules** view	Opens the inclusion rules editor.
	Open **Exclusion Rules** view	Opens the exclusion rules editor.
	Open **Variant Defaults** view	Opens the **Variant Defaults** view.
	Open **Saved Variant Rules** view	Opens the **Saved Variant Rules** view.
	Open **Availability** view	Opens the **Availability** view.
	Open **Variant Configuration** view	Opens the **Variant Configuration** view.
	Export objects to Excel	Exports content to Microsoft Excel for viewing or editing.
Filter Text	Filter text field	Allows you to enter text to filter the displayed data.
	Clear	Clears text from the **Filter Text** field.

The following buttons are provided on the toolbar of the **Variant Defaults**, **Saved Variant Rules**, **Exclusion Rules** and **Inclusion Rules** views:

Button	Function	Description
	Open Expression Editor	Opens the **Variant Expression Editor** view.
	Disable responses to selection	The secondary view does not change when you select a different object in the primary view.
		The secondary view changes when you select a different object in the primary view.
Filter Text	Filter text field	Allows you to enter text to filter the displayed data.
	Clear	Clears text from the **Filter Text** field.

The following buttons are provided on the toolbar of the **Availability** view:

Button	Function	Description
	Expand objects to all levels	Expands the display to show all objects below the currently selected level.
	Collapse all levels	Collapses the display to hide all objects below the currently selected level.
Filter Text	Filter text field	Allows you to enter text to filter the displayed data.
	Clear	Clears text from the **Filter Text** field.

The following buttons are provided on the toolbar of the **Variant Configuration** view:

Button	Function	Description
	Active validation mode	Selects active validation mode. The validation mode always defaults to manual when you open a new session.
	Validate the configuration	Applies and validates the current configuration.
	Apply defaults	Applies defaults to the current configuration.
	Show information messages	Shows information type of violations that occur during the configuration validation.
	Show warning messages	Shows warning violation messages that occur during the configuration validation.
	Show error messages	Shows error violation messages that occur during the configuration validation.
	Load stored configuration	Allows you to load your saved product configurations.
Filter Text	Filter text field	Allows you to enter text to filter the displayed data.
	Clear	Clears text from the **Filter Text** field.

Product Configurator views

Product Configurator views

The Product Configurator contains multiple views, each responsible for managing variant data in the context of a product or dictionary. You can open multiple instances of each view, one for each open product context or dictionary. Views can be opened in other applications where you define variability for objects in a collaborative design or product structure.

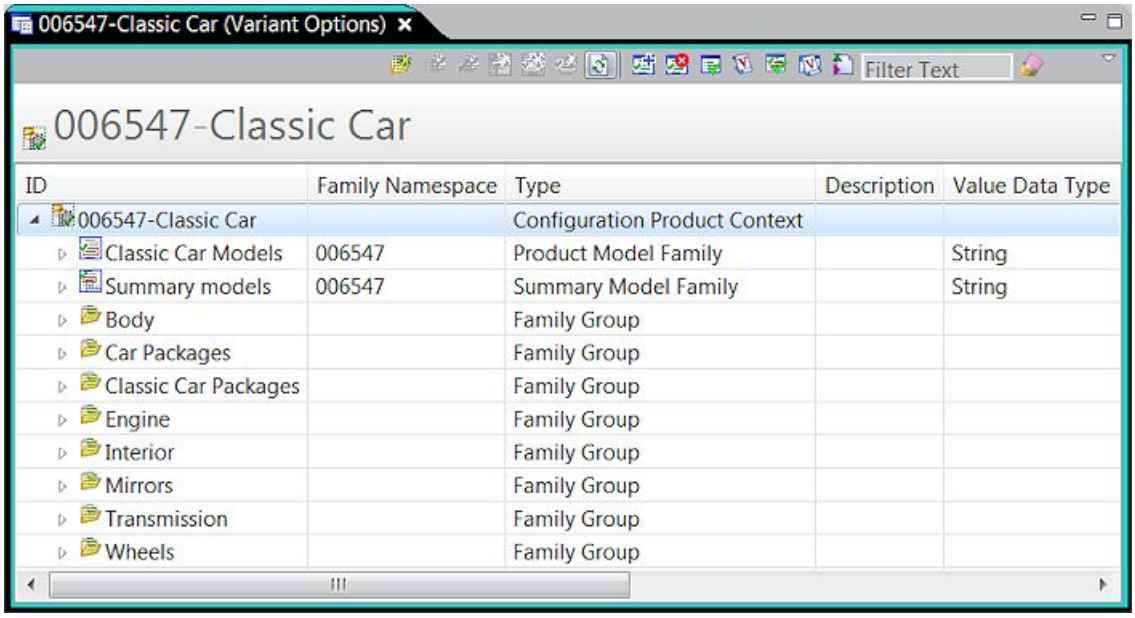
View	Description
Variant Options	Allows you to create or edit product models, variant option groups, families, and values.
Variant Defaults	Allows you to define derived default rules within the context of the currently selected product or dictionary.
Variant Expressions Editor	Allows you to view and edit the details of one or more variant expressions. It is a secondary view of any primary view that lists variant expressions,
Saved Variant Rules	Allows you to manage saved variant rules (SVRs) for the current product context or dictionary.
Exclusion Rules	Allows you to define one or more exclusion rules for the current product context or dictionary. You can define any number of exclusion rules for each product or dictionary.
Inclusion Rules	Allows you to define one or more inclusion rules for the current product context or dictionary. You can define any number of inclusion rules for each product or dictionary.
Availability View	Allows you to view and define availability of variant data for a product model.
Variant Configuration	Allows you to define and validate the variant configuration (manual validation). Allows you to automatically validate the configuration when you switch family groups (auto validation).
Summary	A standard Teamcenter *Summary view* that lets you see properties for a selected object and edit attributes such as name or description for supported items for which you have appropriate permissions.

When you initially open the Product Configurator, only an empty **Variant Options** view is displayed. You can use the quick search box to find the configuration object whose variant data you want to manage. Alternatively, you can double-click a configuration object in the most recently used (MRU) list to display its variant data or send an item revision to the Product Configurator from another application.

Variant Options view

Use the **Variant Options** view to create or edit product models, variant option groups, families, and values. The variant data is organized in a tree table, with groups at the top level of the tree. Each

group displays the option families allocated to it as children. Each family displays its list of values as children.



Note

If you are working with a dictionary, the view title displays as **Dictionary Variant Options**. The toolbar buttons that allow you to add a model family and a product model are inactive, as are the buttons to open the **Inclusion Rules**, **Exclusion Rules**, **Variant Defaults**, and **Saved Variants Rules** views. You cannot author model families, product models, inclusion rules, exclusion rules, default rules, and saved variant rules in context of a configuration dictionary.

The table includes the following columns:

Column	Purpose
ID	Shows the ID or name of the product model, variant family group, family, or value.
Family Namespace	(Family only) Shows the context item in which the family is unique. **Note** It's hidden out of the box. You administrator can enable it if you are using multiple namespaces within your organization.
Type	Shows the business object type of the product model, variant family group, family, or value.
Description	Shows the description of the group, family, or value.

Column	Purpose
Value Data Type	(Family only) Shows the data type of the values within the family. Valid values are **Text** (default), **Floating Point**, **Integer**, **Date**, and **Logical**. This field is blank for groups and values. You can use this column to create constraint rule expressions or variant conditions that compare integer families to floating point values. You cannot add or allocate floating point values to integer families. **Note** This column title is derived from the **fnd0comparison_mode** variant option family property.
Unit Of Measure	(Family only) Shows the unit of measure (UOM) for values within the family. This field is blank for groups and values. This field typically contains only numeric value types, but Teamcenter does not enforce a value type.
Optional?	(Family only) Indicates whether a family is optional for a complete configuration. The default is **No**, indicating that a value is required. This field is blank for groups and values.
Free Form?	(Family only) Indicates whether users enter free form text or pick from a list when specifying a value. If **Yes**, users enter values manually when configuring. The default is **No**. This field is blank for groups and values.
Multi Select?	(Family only) Indicates whether users can select or enter multiple values for a valid configuration. The default is **No**. This field is blank for groups and values.
Minimum Value	(Family only) Specifies the minimum value for a valid configuration. This entry is usually required for non list numeric families. This field is blank for groups and values.
Maximum Value	(Family only) Specifies the maximum value for a valid configuration. This entry is usually required for non list numeric families. This field is blank for groups and values.
Sequence	Specifies the order in which the group, family or value is presented in lists. Teamcenter applies the specified order when you save changes.
Family Object ID	(Family only) Shows the ID or name of the product model and summary model family, option and package family group.

Note

You can change or reorder the columns by right-clicking the menu bar, choosing **Column**, and then modifying the list of displayed columns in the **Column Management** dialog box.

You can click any column header to sort the list in alphanumeric order by the selected column.

When creating variant data, you can make entries at any valid location and all fields are editable. The following fields are *required* entries.

Option type	Required fields
Option group	**ID**
Option family	**ID**, **Value Data Type**, **Optional?**, **Free Form?**, **Multi Select?**
Option value	**ID**

When you have populated the required fields, you can save the new product model, group, family or value. You click **Save** on the main toolbar to save all pending changes. If you try to close the view with unsaved data, Teamcenter prompts you to save.

For existing groups, families or values, you can only edit the **Description** and **Sequence** columns. Edits of existing groups, families or values are saved when you click **Save** on the main toolbar.

All option families have a name space, which Teamcenter initializes with the value of the **Cfg0DefaultOptionFamilyNamespace** preference. Each combination of name space and family name is globally unique. Option families are prefixed with a name space or name and (if applicable) revision ID when shown outside of the context of their owning name space.

You can store reusable families or values in an *option dictionary*, and then paste or drag them into the table for the current product context. Families and their values may belong to multiple products.

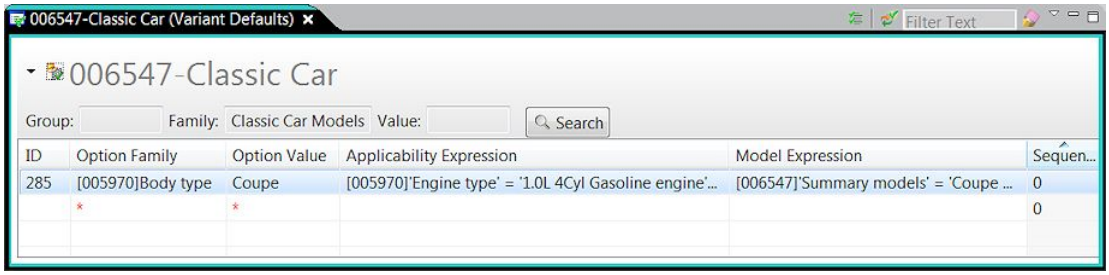
Teamcenter maintains the specific order of each group, family, and value. You can arrange the objects by editing the **Sequence** number. You can move families between groups, but you cannot move values between families. However, an option family may only exist in one group at any time.

You can sort the columns in the table to suit your needs. The default sort column is **Sequence**, but this can be changed by editing the **Cfg0OptionsViewSortProperty** preference. You can also dynamically filter the rows of the table using the **Filter Text** field in the view toolbar.

You can right-click any group, family or value in the **Variant Options** view, and then choose **View Properties** to display its properties. However, you cannot display properties of the **Model Designator** group, **Other** groups or legacy data. You can also copy any object and paste it to a different location.

Variant Defaults view

Use the **Variant Defaults** view to define derived default rules in the context of the current item revision.



As there may be many default rules, Teamcenter does not initially load all rules for the context item. Instead, it displays fields that allow you to enter the following search criteria.

- **Group**

  Display default rules that reference a value within this group.

- **Family**

  Display default rules that reference a value within this family.

- **Value**

  Display default rules that reference this value.

In each case, Teamcenter checks the default, model, and applicability expressions.

The view includes the following columns:

Column	Purpose
ID	Specifies the ID of the rule. This field may be populated automatically or you may have to enter the ID manually.
Option Family	(Required) Specifies the family that contains the default value. Enter a family ID or search for families with the required ID. If Teamcenter finds more than one family (for example, if you enter a partial ID), select the required ID from the dropdown list.
Option Value	(Required) Specifies the default value for the selected family. If the family is a list, you choose one or more values from the defined list. Otherwise, you enter a value or range of values manually. You can also enter an option ID or search for values with the required ID. If Teamcenter finds more than one value (for example, if you enter a partial ID), select the required ID from the dropdown list.
Applicability Expression	(Read only) Define the conditions under which the default value is applied. If you double-click in the field, Teamcenter displays a **Variant Expression Editor** view, allowing you to define or edit condition.
Model Expression	(Read only) Specifies an expression representing a product model that is the scope of the default rule. The expression may contain any number of product model references. If you double-click in the field, Teamcenter displays a **Variant Expression Editor** view, allowing you to define or edit the expression.
Sequence	Contains the sequence number that determines the display order of the default rule. Teamcenter assigns a number when you create the rule, but you can manually change it to resequence the display of the rules.

You can click in the **ID**, **Option Family**, **Option Value** and **Sequence** fields to edit their values.

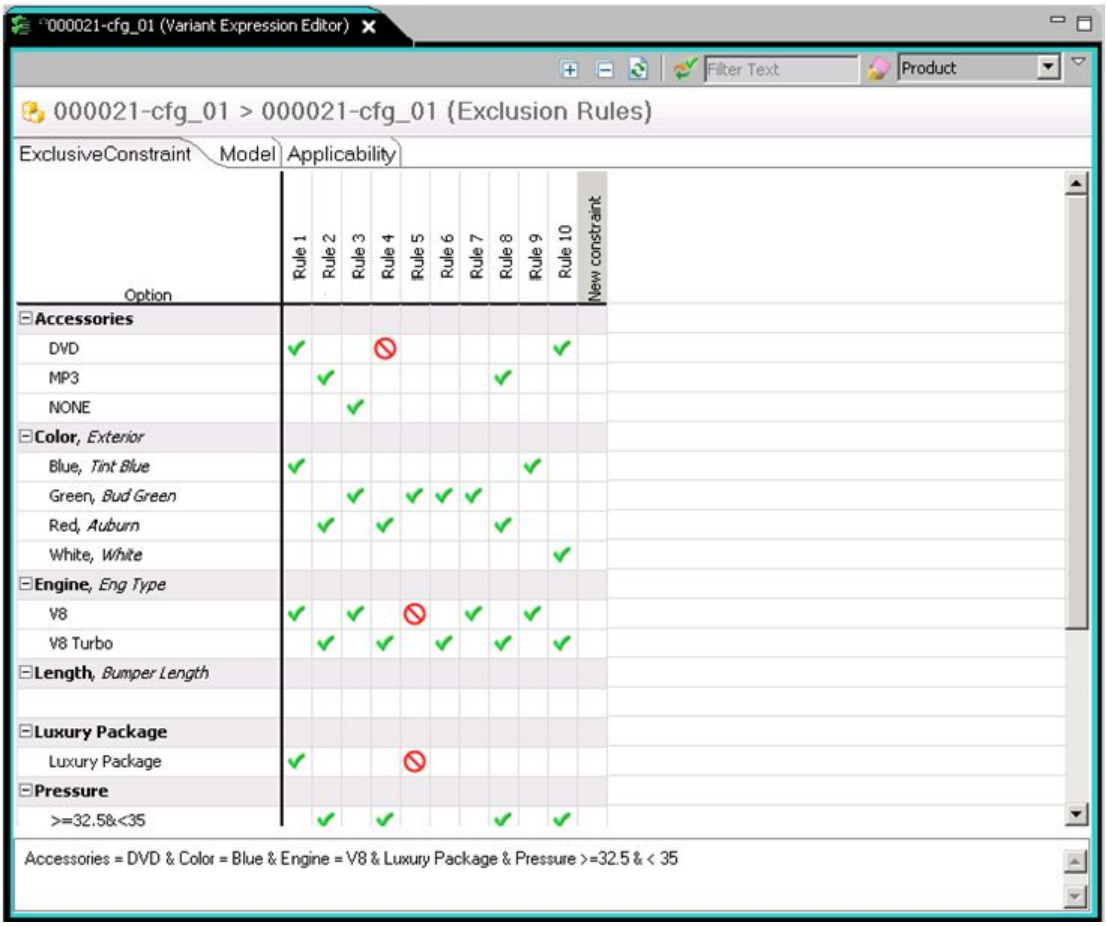
You also edit the **Model** and **Applicability** fields in the relevant panes of the **Variant Expression Editor** view. If you edit the model expression, Teamcenter includes only families marked as product model designators. When you edit a condition expression, all families available to the product context revision are included. When you select multiple default rules, each is shown in a separate column with **Family=***Value* as the column header.

You can also dynamically filter the rows of the table using the **Filter Text** field in the view toolbar.

You can add new default rules directly to the end of the list.

Variant Expression Editor view

The **Variant Expression Editor** view allows you to view and edit the details of one or more variant expressions. It is a secondary view of any other view that lists variant expressions, including the **Variant Defaults**, **Inclusion Rules**, **Exclusion Rules**, and **Variant Rules** views.



The view contains a grid that allows you to view or build variant expressions in Teamcenter Normal Form (TNF).

- Expressions display as one or more columns in the grid.
- Family names are shown in bold for easy identification.
- Option families and values are displayed as rows.
- You can click **+** next to a family name to expand its values. Likewise, you can click **–** next to a family name to collapse its values.

  You can click in the toolbar to expand all families and values. Likewise, you can click in the toolbar to collapse all families and values.
- You click cells in the grid and then select or type values to include in an expression. A green check mark indicates that its value is included in the expression. When you check a value, the

relevant column is highlighted and the expression text is updated to incorporate the change in the expression logic.

If a variant expression is authored in an external application, the Variant Expression Editor may be unable to parse the expression into the table format. In such cases, the column whose expression cannot be parsed is grayed out. You can click the relevant column header and display the expression string in the text field at the bottom of the view. You can edit such expressions by right-clicking a grayed out column and choosing **Edit**. The **Edit** command is enabled only on columns that are grayed out.

Note

A column may also be grayed if the expression is from a legacy variant constraint, default or saved variant rule. For such expressions, the **Edit** command is never enabled.

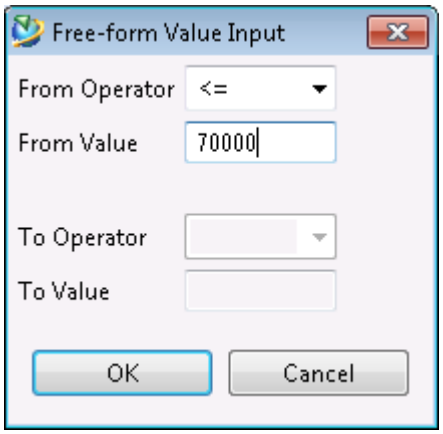
- You can view the syntax of the selected expression in string format at the bottom of the view. The string is dynamically updated when you make a different selection or edit the current expression.
- Teamcenter combines selections in the same column for the same family with an OR function. It combines the results from all such OR functions in the same column with an AND function.
- Teamcenter combines multiple selections within a family that are not mutually exclusive with an AND function.
- You represent a logical NOT by clicking the selected cell again. The cell toggles between selected, NOT, and unselected.
- You can copy an expression by right-clicking a column header and then choosing **Copy Expression**. You can then right-click the column header of another expression and choose **Paste Expression**. Any existing value on the target expression is overwritten.

  You can paste a copied expression onto multiple targets. However, you cannot copy more than one source expression.
- You can clear an expression by right-clicking a column header and then choosing **Clear Expression**. If the expression contains multiple subexpressions, Teamcenter removes all of them.
- You can choose **Filter** to show only the values in the selected column or **Clear Filter** to remove any filtering applied.
- If a field contains a red  symbol, this indicates its value is included in the expression with a NOT operator. You can select a logical NOT function by clicking the selected field again. The field toggles between selected (), NOT (), and unselected.
- Teamcenter combines multiple positive and negative selections within the same family with a logical AND function (for example, `Green AND NOT Blue`). This expression is generally redundant, but Teamcenter permits it if necessary for your business environment.
- If you switch to another primary view, Teamcenter prompts you to save changes and then clears the **Variant Expressions Editor** view. Likewise, if you close the associated primary view, Teamcenter prompts you to save changes, closes the primary view, and then clears the **Variant Expressions Editor** view.

- You can use the **Filter Text** field to limit the number of rows to those matching specified search criteria. You can filter by family name, family description, value name, value description, or both family and value names.

- You can use the shortcut menu to **Copy Selection**, **Paste Selection**, **Clear Selection** or **View Properties** for the option families and their values. The shortcut menu is displayed by right-clicking your selection.

If an option family is defined as free-form, Teamcenter displays a free text field instead of a value. You can enter any text for the value, depending on the value type, for example, a date. You can also enter ranges of values, for example, a range of dates. If appropriate, you can enter more than one free text field for a family.



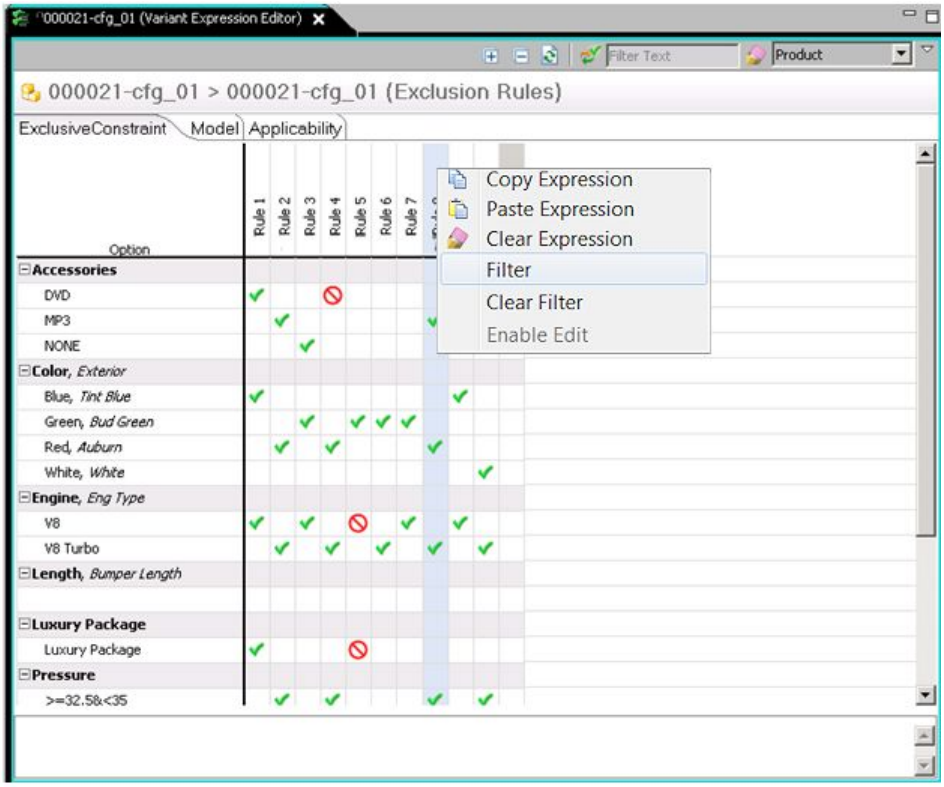
In free-form fields, you can enter both equality values (for example, **Length=10**) and inequality or range values (for example, **Length >=10 & Length <20**). The available end range operators are **<=** and **<**.

Note

In the current version of Teamcenter only certain range formats are supported. If you hover the cursor over the name of a free-form family in the Variant Expression Editor, a tooltip displays a list of supported ranges and how they are evaluated.

A new empty field appears when you enter a value in a field. You can remove (clear) or rename free text values within the grid.

You can right-click an expression and choose the relevant menu command if you want to copy, paste, or clear an expression, as follows:



You can dynamically filter the rows of the table using the **Filter Text** field and **Variability** list in the view toolbar.

Examples of how to construct TNF expressions

Examples of how to construct TNF expressions in the grid follow.

- TNF expression:

  ```
  Color = 'Red' AND Size = 'Small'
  ```

  Grid representation:

Color	**Red**	✔
	Blue	
Size	**Small**	✔
	Large	

- TNF expression:

  ```
  Color = 'Red' OR 'Blue'
  ```

  Grid representation:

Color	**Red**	✔
	Blue	✔
Size	**Small**	

	Large	

- TNF expression:

```
Accessory = 'Bell' AND Accessory = 'Whistle'
```

Note

Accessory is not mutually exclusive.

Grid representation:

Color	**Red**	
	Blue	
Size	**Small**	
	Large	
Accessory	**Bell**	✔
	Whistle	✔

- TNF expression:

```
(Color = 'Red' AND Size = 'Small') OR (Color = 'Blue' AND Size = 'Large')
```

Grid representation:

Color	**Red**	✔	
	Blue		✔
Size	**Small**	✔	
	Large		✔

- TNF expression:

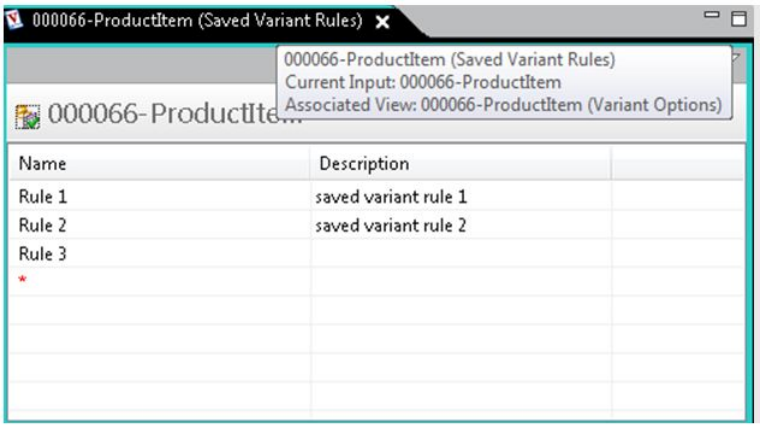
```
Color = 'Red' AND NOT Size = 'Small'
```

Grid representation:

Color	**Red**	✔
	Blue	
Size	**Small**	⊘
	Large	

Saved Variant Rules view

Use the **Saved Variant Rules** view to manage saved variant rules (SVRs).



The view lists all SVRs for the current item. If you click a field, Teamcenter displays the secondary *Variant Expressions Editor* view, allowing you to add, remove, and edit the expressions for the rules.

The grid includes the following columns:

- **Name**

  (Required) Specifies the name of the saved variant rule.

- **Description**

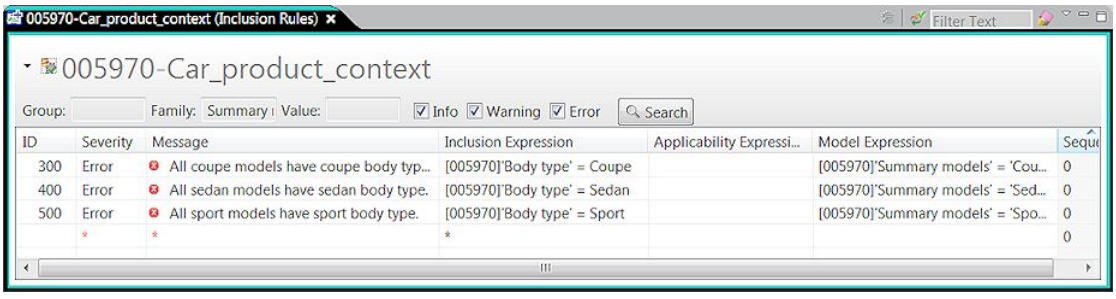
  (Optional) Specifies the description of the saved variant rule.

The table does not show the variant expressions of the SVRs, as they are typically too large to be shown in string format.

The row after the last SVR contains a red asterisk (*) in the **Name** column, indicating you can specify a new SVR here.

Inclusion Rules view

Use the **Inclusion Rules** view to define one or more inclusion rules for the current product. You can define any number of rules for each item.



You edit rule conditions in the expression editor grid. As the number of inclusion rules may be large, Teamcenter does not initially load all rules for the context item. Instead, it displays the following fields that allow you to enter the relevant filter criteria:

- **Group**

  Display inclusion rules that reference a value within this group. Teamcenter checks the model, applicability, and rule expressions.

- **Family**

  Display inclusion rules that reference a value within this family. Teamcenter checks the model, applicability, and rule expressions.

- **Value**

  Display inclusion rules that reference this value. Teamcenter checks the model, applicability, and rule expressions.

- **Severity**

  Display inclusion rules that use the selected severity.

Inclusion rules are met when the rule condition is satisfied. Teamcenter may attempt to modify an order string to ensure the inclusion rules are met, but displays a message when the rule cannot be met. For example, an inclusion rule may contain:

```
Enforce (Color = 'Red' and Size = 'Small'), else generate error message
```

The view includes the following columns for each rule:

Column	Purpose
ID	The ID of the rule. This value may be entered manually or generated by the system.
Severity	(Required) Specifies the type of message displayed if the rule is not satisfied. Click the field, and then choose **Error**, **Warning**, or **Info** from the dropdown menu. Changes are saved automatically.
Message	(Required) Specifies the message Teamcenter displays if the rule is violated. The message text may be localized. Click the field, and then add or edit the message text. Changes are saved automatically.
Inclusion Expression	(Required) Specifies the condition under which the rule is satisfied. If you double-click the field, Teamcenter displays the **Variant Expressions Editor** view, allowing you to define or edit the condition.
Applicability Condition	(Optional) Specifies an expression in the format *family=value* that would make the rule applicable (for example, a constraint may apply if `engine = 'V6'`). You click in this field and Teamcenter displays a dialog box that allows you to edit this expression by specifying a family and its value, values, or a range of values.
Model Expression	(Optional) Specifies an expression that represents one or more product model as the scope of the constraint. The expression may contain any number of product models. If you double-click the field, Teamcenter displays the **Variant Expressions Editor** view, allowing you to define or edit the expression.

Column	Purpose
Sequence	Contains the sequence number that determines the display order of the default rule. Teamcenter assigns a number when you create the rule, but you can manually change it to resequence the display of the rules.

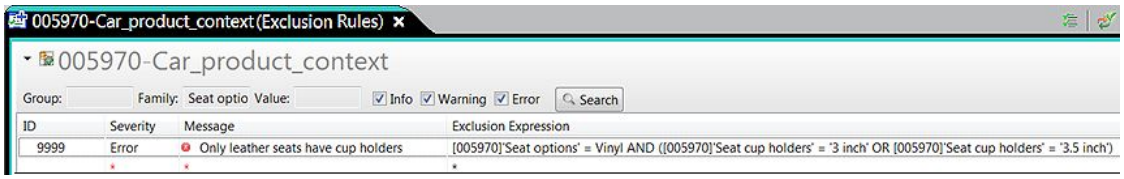
Tip

Editable expression fields are indicated by a symbol.

You can dynamically filter the rows of the table using the **Filter Text** field in the view toolbar.

Exclusion Rules view

Use the **Exclusion Rules** view to define one or more exclusion rules for the current product. You can define any number of rules for each item.



You edit rule conditions in the expression editor grid. As the number of exclusion rules may be large, Teamcenter does not initially load all rules for the product context. Instead, it displays the following fields that allow you to enter the relevant filter criteria:

- **Group**

  Display exclusion rules that reference a value within this group. Teamcenter checks the model, applicability, and rule expressions.

- **Family**

  Display exclusion rules that reference a value within this family. Teamcenter checks the model, applicability, and rule expressions.

- **Value**

  Display exclusion rules that reference this value. Teamcenter checks the model, applicability, and rule expressions.

- **Severity**

  Display exclusion rules that use the selected severity.

Exclusion rules are violated when the rule condition is satisfied. Teamcenter displays a message when the rule is violated. For example, an exclusion rule may contain:

```
If (Color = 'Red' and Size = 'Small'), then generate error message
```

The view includes the following columns for each rule:

Column	Purpose
ID	The ID of the rule. This value may be entered manually or generated by the system.
Severity	(Required) Specifies the type of message displayed if the rule is not satisfied. Click the field, and then choose **Error**, **Warning**, or **Info** from the dropdown menu. Changes are saved automatically.
Message	(Required) Specifies the message Teamcenter displays if the rule cannot be satisfied. The message text may be localized. Click the field, and then add or edit the message text. Changes are saved automatically.
Exclusion Expression	(Required) Specifies the condition under which the rule is violated. If you double-click the field, Teamcenter displays the **Variant Expressions Editor** view, allowing you to define or edit the condition.
Applicability Condition	(Optional) Specifies an expression in the format *family=value* that would make the rule applicable (for example, a constraint may apply if `engine = 'V6'`). You click in this field and Teamcenter displays a dialog box that allows you to edit this expression by specifying a family and its value, values, or a range of values.
Model Expression	(Optional) Specifies an expression that represents one or more product model as the scope of the constraint. The expression may contain any number of product models. If you double-click the field, Teamcenter displays the **Variant Expressions Editor** view, allowing you to define or edit the expression.
Sequence	Contains the sequence number that determines the display order of the default rule. Teamcenter assigns a number when you create the rule, but you can manually change it to resequence the display of the rules.

Tip

Editable expression fields are indicated by a  symbol.

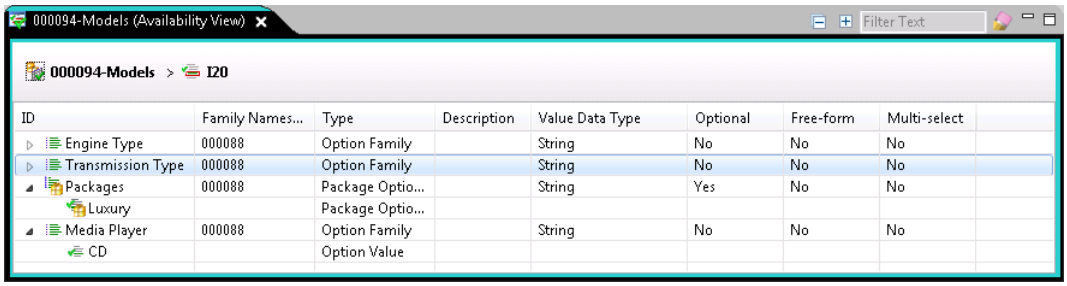
You can dynamically filter the rows of the table using the **Filter Text** field in the view toolbar.

Availability View

Use **Availability View** to define which options are allowed for a product model. You can only choose from available option values and option families and package option families and values in the product context. When you make an option value available, it also makes the family for that option value available.

Note

Summary options are not visible in **Availability View**.



You can click **Select a Model** to choose from a list of available models in your product context.

You can dynamically filter the rows of the table using the **Filter Text** field in the view toolbar. To remove filter the applied filtering, click **Clear Filter**.

The view includes the following columns for each option:

Column	Purpose
ID	The ID of the option. This value may be entered manually or generated by the system.
Family Namespace	(Family only) Shows the context item in which the family is unique.
Type	Shows the business object type of the product model, variant family group, family, or value.
Description	Shows the description of the group, family, or value.
Value Data Type	(Family only) Shows the data type of the values within the family. Valid values are **Text** (default), **Floating Point**, **Integer**, **Date**, and **Logical**. This field is blank for groups and values. **Note** This column title is derived from the **fnd0comparison_mode** variant option family property.
Optional?	(Family only) Indicates whether a family is optional for a complete configuration. This field is blank for groups and values.
Free Form?	(Family only) Indicates whether users enter free form text or pick from a list when specifying a value. If **Yes**, users enter values manually when configuring. This field is blank for groups and values.
Multi Select?	(Family only) Indicates whether users can select or enter multiple values for a valid configuration. This field is blank for groups and values.

Variant Configuration view

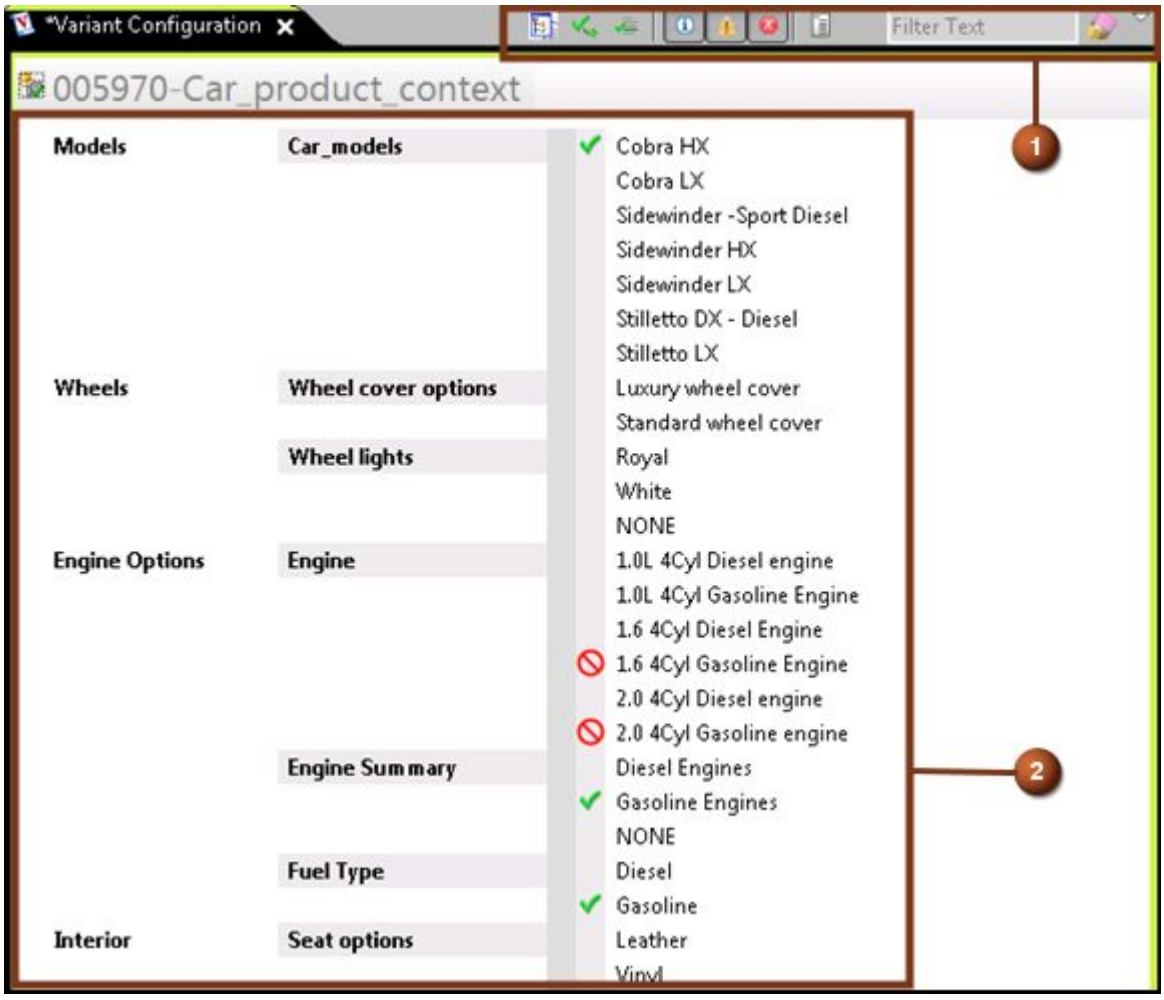
Use the **Variant Configuration** view to define and validate the variant configuration using manual or auto validation.

The **Variant Configuration** view contains the variant options that are used to define the configuration.

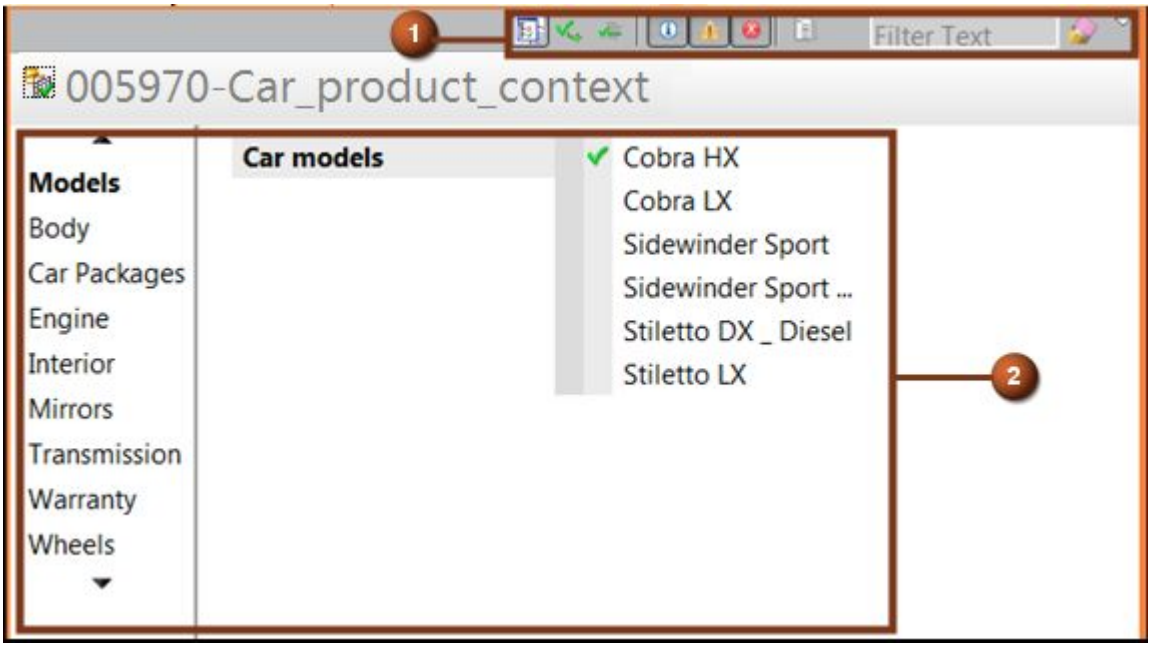
The variant option contains one of the following used for defining the configuration:

- A green check mark ✔, indicating the configuration includes the variant condition.
- A red circle backslash ⊘, indicating the configuration excludes the variant condition.
- Blank, indicating the variant condition is not currently used in the configuration.

The following figure shows configuring variants using manual validation:



The following figure shows configuring variants using auto validation:



1	Variant configuration toolbar	Allows you to set variant defaults, load a configuration, and validate the configuration.
2	Variant options	Set and view the variant options used to define the configuration

Teamcenter rich client perspectives and views

Within the Teamcenter rich client user interface, functionality is provided in *perspectives* and *views*. Some applications use perspectives and views to rearrange how the functionality is presented. Other applications use a single perspective and view to present information.

- Perspectives

  Are containers for a set of views and editors that exist within the perspective.

  - A perspective exists in a window along with any number of other perspectives, but only one perspective can be displayed at a time.
  - In applications that use multiple views, you can add and rearrange views to display multiple sets of information simultaneously within a perspective.
  - You can save a rearranged perspective with the current name, or create a new perspective by saving the new arrangement of views with a new name.

- Views and view networks

  In some Teamcenter applications, using rich client views and view networks, you can navigate to a hierarchy of information, display information about selected objects, open an editor, or display properties.

  - Views that work with related information typically react to selection changes in other views.

- Changes to data made in a view can be saved immediately.
- Any view can be opened in any perspective, and any combination of views can be saved in a current perspective or in a new perspective.
- A view network consists of a primary view and one or more secondary views that are associated. View networks can be arranged in a single view folder or in multiple view folders.
- Objects selected in a view may provide context for a shortcut menu. The shortcut menu is usually displayed by right-clicking.

  For more information about using the shortcut menu, see the *My Teamcenter Guide*.

Note

If your site has online help installed, you can access application and view help from the rich client **Help** menu or by pressing F1. Some views, such as **Communication Monitor**, **Print Object**, and **Performance Monitor**, are auxiliary views that may be used for debugging and that may not be displayed automatically by any particular perspective.

For more information about auxiliary views, see the *Client Customization Programmer's Guide*.

For more information about perspectives and views and changing the layout of your rich client window, see the *Rich Client Interface Guide*.

Getting ready to adopt Product Configurator for existing classic variant data

If you already use classic variants, you can start planning your existing variability data to prepare for using Product Configurator in a future release. Review the following list to ascertain if you can benefit from any of the listed Product Configurator functionality.

- **Saved variant rules (SVR)**

  An equivalent SVR is provided with the new Product Configurator to store variant configuration criteria and optional validation records. They are attached to the product context item or the application model (collaborative design) by GRM relationships.

- **Multi-select option families**

  This is new functionality for existing users. They can now combine the families that were explicitly created to allow you to create conjunction of option values. For example, you can create a multi-select family for accessories and keep all these options in one family and still configure structure with multiple options from the same accessory family.

- **Default rule**

  Existing option defaults (with or without preconditions) can be represented using this rule type.

- **Inclusion rule**

  This is new functionality for existing users. An inclusion rule defines a condition for which a specified selection of one or more option values implies the setting of one or more other option

values. For example, set option 1 to A if option 2 is set to B. In this example, option value 1 must be set to A if option value 2 is set to B. If the user wants to make a different choice for option value 1, the selection for option value 2 must be changed.

Inclusion rules express *Do* or *Must Do* conditions. They contrast with exclusion rules that express *Don't* conditions. If an inclusion rule says "If Exterior=Black → Interior=Anthracite", the system forces anthracite interiors for black exteriors. Conversely, if an exclusion rule says "If Exterior=Black → Interior=Anthracite", the system disallows anthracite interiors for black exteriors.

Existing option defaults (with or without preconditions) can be mapped to the inclusion rules with messages and severity.

The evaluation of an inclusion rule changes the product configuration or it fails, while the evaluation of an exclusion rule leaves the product configuration unchanged or it fails. That is, exclusion rules only validate conditions, while inclusion rules validate *and* modify conditions.

- **Exclusion rule**

  Existing rule checks can be represented using exclusion rules. An exclusion rule is a convenient way to represent exclusions. If an exclusion rule says "If Exterior=Black → Interior=Anthracite", the system disallows anthracite interiors for black exteriors.

- **Feasibility Rule**

  Existing rule checks (must do's) can also be represented using feasibility rules. A feasibility rule is a convenient way to represent must or must not conditions. If a feasibility rule says "If Exterior=Black → Interior=Anthracite", the system require anthracite interiors for black exteriors.

- **Free form family**

  This is new functionality for classic variant users. Create free form variant families and assign values during authoring or configuration. You can define optional boundaries, rather than just lists of predefined values.

- **Nondiscretionary family**

  This is new functionality for existing users. You can classify a family to be discretionary or nondiscretionary (required) variant option families. System will enforce option values from nondiscretionary family during configuration.

- **Selective rule type evaluation**

  This is new functionality for existing users. You can turn off certain rules type during configuration. For example, turn off feasibility check when working with 120% variant configuration.

Following are the best practices to adopt Product Configurator in a future release:

- Make all options global.
  - Do not define any variability within the product structure (in general).
  - Attach all options to one or more global option items. Create global option items to correspond to your projected future product context needs.
    - For a given product, collect all variability in a single Global Option Item (GOI).

- Conflicting scenarios may occur due to rules or constraints when variability for all products is added to single GOI.
  - In such scenario, the variability should be defined within the product structure.
  - The top item could then be a future product context.

  For all other scenarios, the variability could be defined in a single GOI for all products that belong to same product line.
- Position options for new Product Configurator from the start. Think about:
  - Numeric option values you are out of necessity storing as strings.
  - Any other option values you are storing as strings but are really intended to be dates, Booleans, etc.
  - Be aware of any options you are creating that represent product models and partitions.
- Do not revise Global Option Items.

Chapter 2: Defining product configurations

Creating variable products

Creating variable products

Managing a discrete product variant for every possible option family combination is inefficient and error prone. Instead, you can choose to leverage the commonality across the product and build the variability into the product data. You manage the variable product definition and derive all valid product variants from it.

Once you have identified the commonality in the generic product, you define:

- The valid feature combinations for that generic product.
- The variant rules and constraints needed to ensure that the valid combinations are allowed but no others.
- The design or part solutions required to satisfy any variant within the range of valid variants.

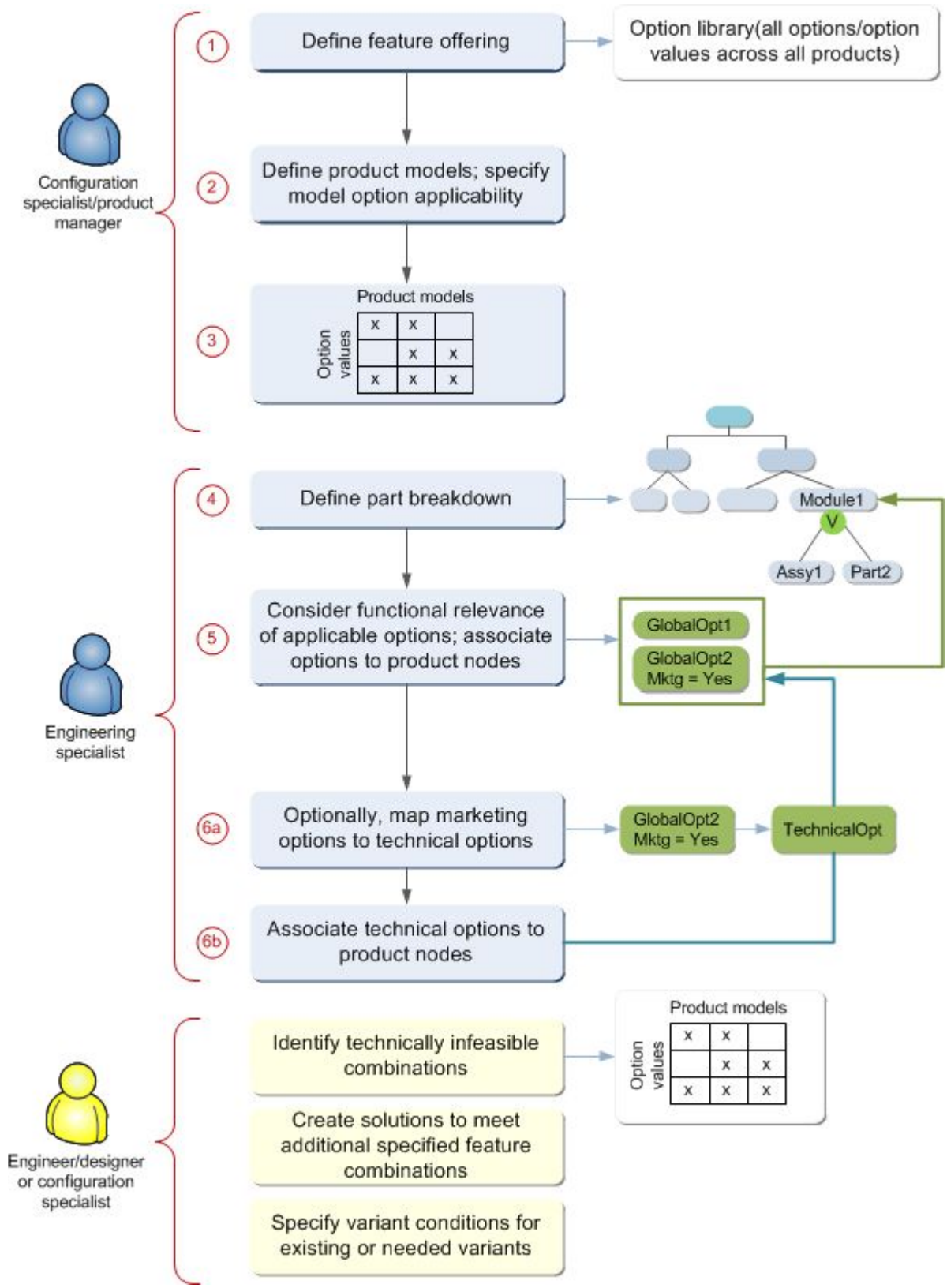
You perform these steps when creating the variable product definition, regardless of whether you manage a conventional product decomposition or specify variability relative to a 4GD partition breakdown.

You may not always perform these steps in the same order each time. You can always see the valid feature combinations specified for a generic product, and also the design solutions specified for the generic product. This is possible even if the solution or the feature combination is not specified, or if they are specified but not yet linked.

Note

If you define variability against 4GD partitions, it may not be possible to specify a solution except when it is linked to a particular set of features.

The following diagram shows the task flow to create a variable product.


Configuration specialist/product manager
1
Define feature offering
Option library(all options/option values across all products)
2
Define product models; specify model option applicability
3
Product models
Option values
Engineering specialist
4
Define part breakdown
Module1
V
Assy1
Part2
5
Consider functional relevance of applicable options; associate options to product nodes
GlobalOpt1
GlobalOpt2
Mktg = Yes
6a
Optionally, map marketing options to technical options
GlobalOpt2
Mktg = Yes
TechnicalOpt
6b
Associate technical options to product nodes
Engineer/designer or configuration specialist
Identify technically infeasible combinations
Product models
Option values
Create solutions to meet additional specified feature combinations
Specify variant conditions for existing or needed variants

Managing variant data for products and dictionaries

You can create product contexts or *dictionaries*, and then associate variant option families and their values with them. You can create constraints (including fixed or derived defaults, inclusive or exclusive rules) and associate them with the product context. You can use Teamcenter to manage the life cycle of the product context or dictionary, for example, by using workflows to approve additions to them.

Note

If you create a new product context or dictionary by copying an existing product context or dictionary (that is, using the **Save As** command), variant data associated with the existing product context or dictionary is not carried forward.

Updating and managing variable product data

Once you have created a variable product definition, it may not remain static for long. Any of the pieces of product data within the variable product definition may evolve, including saved variant rules, default rules, and variant rules for compatibility. The changes in the product data must be folded into the variable product definition. To do this, you must implement configuration control of the prechange configuration to allow a controlled introduction of the changes.

You may encounter this situation if product or marketing managers update the features offered for a product model. They may define option values that will now become available, will become unavailable (obsolete), or will be available only in a combination not previously allowed.

If you defined allowed variability directly against a product, you follow these steps to reflect the changes in the product data:

1. Engineering evaluates option availability changes and determines if new solutions are required to satisfy the updated features or if existing solutions become obsolete as a result of the change.

2. Engineering determines if any new option families or derived default rules that set option values should change.

3. If the product impacted by the change is currently released or uneditable, Engineering typically revises the product and makes necessary adjustments to the revision. This is achieved by specifying effectivity on the revision to match the effectivity of the option family changes. All variant conditions are effective based on the effectivity of the parent on which they are qualified.

4. Users ensure that saved variant rules, derived rules and variant constraints are effective consistent with the product data update.

5. Option family and option value updates are reviewed and released.

6. New and updated product data is reviewed and released.

If you defined allowed variability against a 4GD partition breakdown, follow these steps to reflect the changes in the product data:

1. A configuration specialist updates feature combinations that are relevant for a particular partition. Depending on your business process, you may revise the partition item to introduce the updated feature combinations and corresponding engineering solutions.

2. The partition is reviewed and released.

3. Engineering reviews the updated feature combinations and may certify that an existing solution is valid for an updated feature combination. Alternatively, they may identify and link a new solution that satisfies the feature conditions.

The change may be driven by a necessary design change that is not related to any change in option family availability. If you defined the allowed variability directly against the product data, you follow these steps to reflect the changes in the product data:

1. Engineering makes a change to a design. If the design is not in an editable state, you revise the impacted design to make the change. You then update the engineering content of the design, for example, the CAD file.

2. Engineering evaluates all variant conditions applied against any updated node. Any variant conditions that have become invalid as a result of the module update must be updated and made valid. For any new nodes, a variant condition must be applied.

3. Engineering performs a solution replacement (replaces the design in the product) for any updated product data.

4. Engineering ensures all specified feature combinations have valid solutions identified.

   Note

   This operation should not result in any new option families or values relevant to the selected product node. It should not require updates to derived option rules or compatibility rules.

Working with variant option families and values

Working with variant option families and values

Variant option families and values are the basic objects of any variant scheme. You create option families using terminology that users understand, for example, **color**. You then assign allowed values to those option families, for example, **red** and **blue**. You associate option families and their values with an item revision that represents the product library.

Configuration management of variant data

As variant data is introduced and changed, Teamcenter ensures the correct versions of the data are considered and offered when users select option family values and configure the product data.

It is important that your business processes ensure the appropriate product data, variant option families, and variant rules are provided in a coordinated way. Teamcenter helps the users understand what changes they have already accounted for in the variable product definition and what changes have yet to be considered. That is, they should understand if they have introduced new option values that are not referenced elsewhere and also if option values they are obsoleting are referenced in active rules.

Careful configuration management is important if you are performing any of the following tasks:

- Introducing option families, option values and applicability

As marketing specialists or product managers introduce new option families and option values, you may need to define business rules that determine what level of maturity or approval is needed before these option families and option values can be accessed by users.

- Validated for applicability for a product model (that is, a model-option mapping can be created for them).
- Eligible to have rules written to control their selection
- Referenced directly by product data.
- Exposed to a user for configuration.

Typically, you enforce these business rules with Access Manager rules on the relevant objects to allow only authorized users to view and manage the data.

You may define additional business rules to determine eligibility to have existing rules exercised during a configuration process. For example, when a user first creates and tests variant rules, they do not impact the existing variant rules and product data until they are validated, mature, and effective.

In addition to defining and enforcing access rules on the variant expressions, you should control when new variant option families, values, model-option applicability, and variant rules created by a user should become available, applicable or enforced. One method of doing this is with a release process.

For example, if you introduce a new color paint for one of the product models, you would make it available for testing and validation before making the new color option value a selection that is visible to the user.

- Creating and managing variable product data

  The creation and updating of the product data and the variant rules and option families that are referenced directly by the product data are straightforward to manage. Typically, your company already has an established business process for the management of product data that requires the revising of the product data when changes significant to the business occur. The same process can normally be used for variable product data.

  You manage changes that are significant to the business (for example, adding or removing referenced variant option families or updating variant rules) by making these changes only to an editable version of the product data. A specific revision of the product data references an option value and (as that same option value evolves), the latest version that meets the defined business rules applies. This allows the user to make form, fit, and function compatibility decisions when making changes to an option value. If its interface definition does not change, the product data and rules that reference it should not change. If the interface definition changes, you should introduce a new option value or revise the product data that references it.

Defining option families, option values, and option rules

When users specify the variability to offer on products, they work with the following objects:

- Variant option families

  Users perceive these as the questions asked during the configuration process.

- Option values

  Users perceive these as the possible answers to the questions asked during the configuration process.

- Saved variant rules or variant constraints (inclusion rules or exclusion rules)

  Users perceive these as expressions of the valid or invalid combinations of allowed answers.

When defining variant data, users must be able to:

- Identify variant option families that are offered for their product line. Option family or feature choices can be developed by deciding the question that the user is asked at each step of the process. For example:
  - Color: What color car would you like?
  - Sunroof: Would you like a sunroof?
- Define allowable values for each variant option family. Option values are all the possible answers to the questions the user is asked. For example:
  - Color: cayenne red, dusty blue, charcoal grey
  - Sunroof: yes, no

In the previous examples, there is one expected answer for each question. That is, these questions are mandatory for a complete, valid configuration and the answers are also mutually exclusive. Consequently, only one answer may be given.

There are also situations where no answer or multiple answers to the same question are valid for a certain configuration. For example:

- CD changer: 1 disc, 6 disc, 10 disc.

  This is an optional (discretionary) option family. The user need not make any selection to have a complete, valid configuration. If the user does make a selection, only one selection is allowed. Consequently, these option values are mutually exclusive as well as discretionary.

- Cold weather accessories: snow chains, engine block heater, heated seats, remote starter.

  This is also an optional (discretionary) option family, but in this case the user may select more than one value. These option values are not mutually exclusive.

You can make all of the option families mandatory and all option values mutually exclusive using a constraint rule. For example:

```
CD changer: 1 disc, 6 disc, 10 disc, none
Snow chains: yes, no
Engine block heater: yes, no
Heated seats: Driver only, driver and passenger, none
Remote starter: yes, no
```

You can also make the different option families themselves mutually exclusive. For example, if the user selects cold weather accessories, they cannot select warm climate accessories. Likewise, if the user selects split rear seats (60/40, 40/20/40), they cannot select a rear cargo barrier, aluminum lattice, elastic mesh, or vertical bars.

You want to establish other types of rules or constraints among the different option families or values. Also, you may want to create more sophisticated logic between the different option families or values. For example:

- Rules defining what selections may not be made in combination. For example:

```
"if you select Body Color = Green you cannot select Interior Color = Blue"

"(((if Body Color = Green AND Wheel Type = Chrome)
  OR (if Body Color = Black AND Wheel Type Matte Grey))
  AND seat type = Leather) do not allow Interior Color = Blue)"
```

- Default values. You can define a default option value as a starting point or the commonly expected choice. A basic default may be defined independently of the values chosen for any other option family. That is, it does not matter what other option value selections the user makes, any option family for which a default is defined initially defaults to the defined value. The user may make a selection other than the default.

- Derived option values. In some situations, the user may want to define a rule where the setting or default value of one or more option values is a direct result of other option value selections. These might be fixed or overrideable. For example, if you select sport suspension, you must also select the performance tires. This could be expressed by exclusion rules or error checks, for example:

```
('(if Suspension = Sport ErrorIf Tires = All Season)
OR (if Suspension = Sport ErrorIf Tires = RunFlat)
OR (if Suspension = Sport ErrorIf Tires = SnowTires)')
```

  However, it is more efficient to express this dependency as `if Suspension=Sport SET Tires=Performance`. You can express the same type of dependency but allow the user to override it. For example:

```
((Heated Seats = Driver
 OR Heated Seats = Driver and Passenger)
 SET HeatedSteeringWheel = Yes)
```

  In this example, the heated steering wheel is a default or suggestion that is presented to the user only if they have made other complementary choices. The user may still decline the heated steering wheel and to change the default to `No`.

Option values can move over time to different option families. That is, the option family with which an option value is associated is not static and can evolve.

Related topics

- Create variant groups and families for a product context or dictionary
- Writing variant expressions and solve criteria

Writing variant expressions and solve criteria

If product data does not have a variant condition, some businesses consider an error has occurred. Even if a component is always configured, your business process may require an explicit statement to always configure this component. In this scenario, it is important to find objects that do not have a variant condition, otherwise you cannot ascertain if an object configures for all variant criteria because it was forgotten or because it is intended always to configure.

The following table illustrates possible variant configuration scenarios, the configurable objects and their variant condition. Each row represents a variant solve configuration using the criteria shown in the criteria column. Columns A through E represent objects configured with the variant conditions shown in the condition row. For example, a cross (X) in cell A,1 indicates that configurable object A configures for the criteria in row 1.

Criteria →		**A**	**B**	**C**	**D**	**E**
Condition →		**ENG!=V8**	**ENG=V8**	**ENG=V6 | ENG!=V6 (TRUE)**	**ENG=V6 & ENG!=V6 (FALSE)**	**None**
1	Formula “ENG=V6”	X		X		X
2	Formula “ENG!=V6 | ENG!=V8”	X	X	X		X
3	Formula “ENG=V6 & ENG=V8”					X
4	State “TRUE”			X		
5	State “FALSE”				X	
6	State “NULL”					X
7	State “NOT_NULL”	X	X	X	X	
8	State “NON-CONTSNT CONDITION”	X	X			

The following table shows the configuration conditions in the previous example, with the corresponding solve types.

Configuration condition (cell reference)	**Solve type**
A1; B1	Configure objects with a variant condition against variant solve criteria.
C1	Configure objects with a variant condition that is equivalent to the Boolean constant TRUE against variant solve criteria.
D1	Configure objects with a variant condition that is equivalent to the Boolean constant FALSE against variant solve criteria.
E1	Configure objects without variant conditions against variant solve criteria.
A2; B2	Configure objects with a variant condition against variant solve criteria that are equivalent to the Boolean constant TRUE.
A3; B3	Configure objects with a variant condition against variant solve criteria that are equivalent to the Boolean constant FALSE.

Configuration condition (cell reference)	Solve type
C2	Configure objects with a variant condition that is equivalent to the Boolean constant TRUE against variant solve criteria that are equivalent to the Boolean constant TRUE.
D2	Configure objects with a variant condition that is equivalent to the Boolean constant FALSE against variant solve criteria that are equivalent to the Boolean constant TRUE.
C3	Configure objects with a variant condition that is equivalent to the Boolean constant TRUE against variant solve criteria that are equivalent to the Boolean constant FALSE.
D3	Configure objects with a variant condition that is equivalent to the Boolean constant FALSE against variant solve criteria that are equivalent to the Boolean constant FALSE.
E2	Configure objects with no variant condition against variant solve criteria which are equivalent to the Boolean constant TRUE.
E3	Configure objects with no variant condition against variant solve criteria that are equivalent to the Boolean constant FALSE.
C4	Configure objects with a variant condition that is equivalent to the Boolean constant TRUE against variant criteria selecting objects with a variant condition that is equivalent to the Boolean constant TRUE.
D5	Configure objects with a variant condition that is equivalent to the Boolean constant FALSE against variant criteria selecting objects with a variant condition that is equivalent to the Boolean constant FALSE.
E6	Configure objects with no variant condition against variant criteria selecting objects without a variant condition.
A7	Configure objects with a variant condition against variant criteria selecting objects with a non-NULL variant condition.
B7	Configure objects with a variant condition against variant criteria selecting objects with a non-NULL variant condition.
C7	Configure objects with a variant condition that is equivalent to the Boolean constant TRUE against variant criteria selecting objects with a non-NULL variant condition.
D7	Configure objects with a variant condition that is equivalent to the Boolean constant FALSE against variant criteria selecting objects with a non-NULL variant condition.
B8	Configure objects with a variant condition against variant criteria selecting objects with a non-constant variant condition.

Configuration condition (cell reference)	Solve type
A8	Configure objects with a variant condition against variant criteria selecting objects with a non-constant variant condition.

Defining summary options

Summary family option is a collection of option values which allows you to summarize the values from the same option family. Summary option values are connected using a logical OR operation in the variant expression. Option values may belong to more than one summary option.

A summary model family is a collection of models that allows you to summarize the product models in the product context. A summary model behaves similarly to a summary option.

Summary option allows you to efficiently author rules by referring to the *summary option* when the rule applies to all of the option values it summarizes. A summary option may be referenced in variant, exclusion, and inclusion rule expressions.

You can only summarize options within the same mutually exclusive family. You cannot summarize options from the following:

- Across mutually exclusive families
- In a multi-select family
- From a Boolean option family
- From a free form family

Note

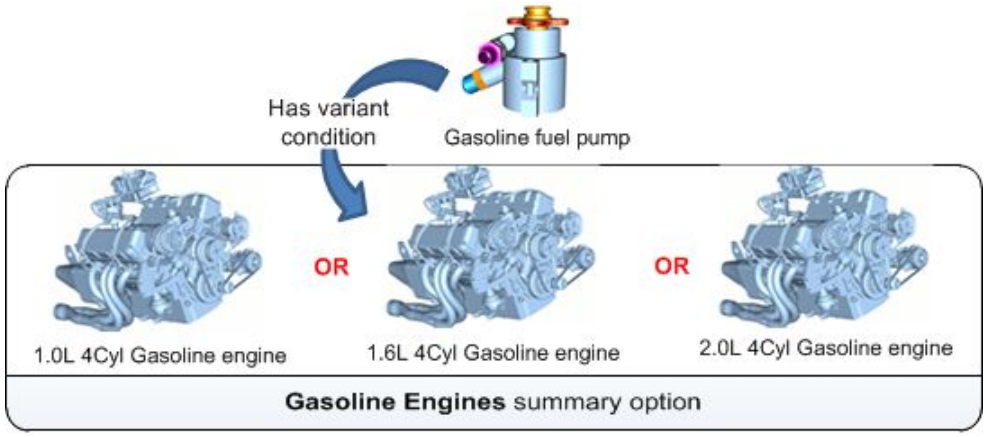
You cannot use the same summary in more than one clause in the **Model Applicability** section of the inclusion rule and inclusive constraint, and exclusion and default rules.

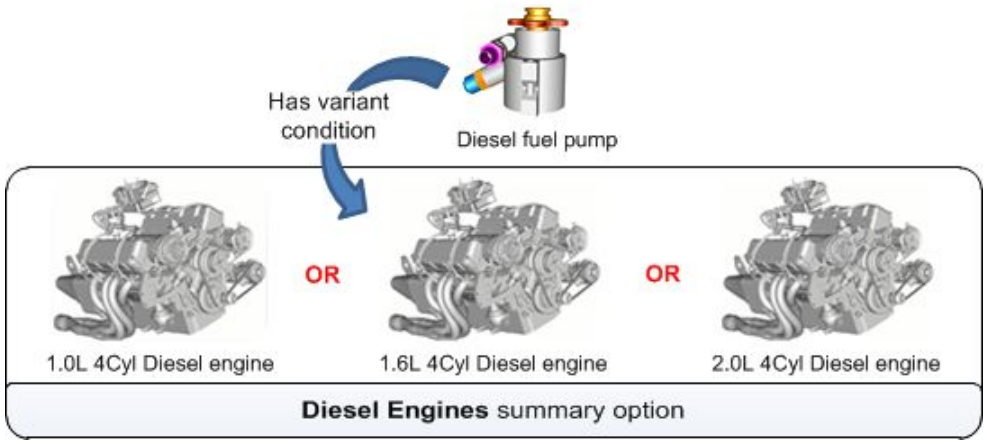
Summary options are typically used to represent sets of alternative choices. For example, **V6 Engines** summary option can contain all V6 engines.

Summary options cannot be ordered by a customer, thus they are only available when configuring with a custom variant configuration using *manual validation* and *not* available when configuring using *active validation*. After selecting a summary option, all the options that are valid members are selected when you validate the configuration using manual validation.

Example

A generic car engine uses the same gasoline fuel pump for all gasoline model engines and the same diesel fuel pump for all diesel engines. The configurator developer, using Product Configurator, creates the **Gasoline engine** summary option that contains all gasoline engine models and the **Diesel engine** summary option that contains all diesel engine models.





You can determine where the summary option and its members are used by performing an Impact analysis. Right–click the summary option or its member, and choose **Open with→Impact Analysis**.

For example, you can perform a where-used query on a summary option and choose the **Product Context** scope to find:

- All product contexts where that summary option is allocated to.
- All rules that use the product contexts with the selected summary option.
- A family and family group for the summary option.
- Option values that it summarizes and summary options, product contexts, rules and a family for these member option values.

You can perform a where-used query on a summary option with the **Dictionary** scope to find:

- All dictionaries where that summary option is allocated to.

- All summary members and summary options and all dictionaries for these members.

Creating packages to group option families

You can create groups of option families or *packages* that you would like customers to order together in a bundle. *Package* options contain option values within or across option families. Existing option values are assigned as members of a package option and are connected using a logical **AND** operation in the variant expression.

Package options are typically used to represent sales or marketing packages, such as the luxury or economic package for a car. You set the variant condition to include the package option for the design content that is to be part of the package.

Typically, a product manager or similar user identifies one or more of the following reasons to group option families in a bundle.

- You want to offer a wide range of variability to your customers, but recognize there are efficiencies and cost savings in more commonality.

- You recognize that customers want to be able to order and receive their products quickly. Defining set configurations allows you to produce common product variants as needed and keep stock on hand. Offering feature packages instead of complete configured variants allows you to maintain a balance between permitting the customer to select features they want dynamically without forcing them to choose only from fully specified configurations.

- There are features that are easier or harder to accommodate late in the process. For example, a laptop computer is available in different colors, but installation of the colored cover is the last step before it is boxed and shipped to the purchaser. You may have several configurations built and stocked that are only waiting for the cover color matching incoming orders to facilitate the final assembly of those products.

- You want to encourage the customer to order more accessories or premium features. By grouping them in an easy to order package, you may achieve this objective. You may also offer a discount over ordering each option individually.

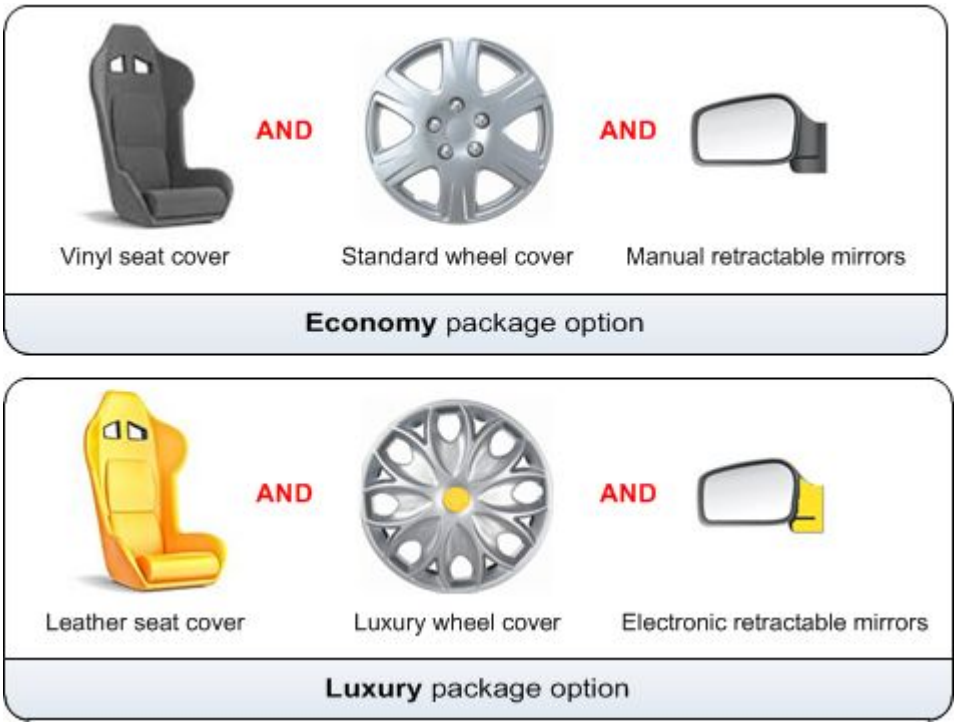
- By creating repeatable configurations or partial configurations, you reduce the likelihood of issues with a particular combination of features.

If you create packages to bundle sets of option values, you should also optimize your other business processes around the use of feature packages. That is, considerations such as engineering feasibility and the creation of engineering solutions should be partially determined from the context of the feature package, rather than individual features separately.

Feature packages are exposed during the configuration process. If you select a package, Teamcenter responds by selecting all of the corresponding option values collected in that feature package when you validate the configuration using manual validation or active validation.

Example

The economic option package for a generic car includes a vinyl seat cover, standard wheel cover and manual retractable mirrors. The luxury option package for a generic car includes a leather seat cover, luxury wheel cover and electronic retractable mirrors. The configurator developer, using Product Configurator, creates the luxury and economic package options that contains the options for each package.



Defining applicability of variant option families and rules

You can control the reuse of variant option families and option values by designating certain option families or subsets of option values that are allowed for a particular product or product model.

As you introduce new option families, you can reuse them across several product models. However, you can restrict the product model applicability of option values to only the desired product models. As a result, if you introduce a new product model, the user can identify the option values to offer for that product model.

You can use the Product Configurator to view and manage all product models and option values in a single matrix. Alternatively, you can build a table with only the product models and option values of active interest.

You can identify product model-option value pairs to authorize. When you author saved variant rules or configure selected option values in the context of a product model, only those option values authorized for the selected product model are shown.

Related topics

- Associate a product context with an application model

- Define variant defaults

Defining availability of options and families

You can define which option values, families and package options are available for the individual product models within a product context. You can make options, families, and values available to one or more product models. When you make an option available to the selected product model, the family of that option also becomes available.

You cannot define availability for the summary options or summary models.

Using **Availability view**, you can designate available options, values, and families for the specifically selected model in your product context. You can identify the option values to be available for the product model and drop them into the **Availability view** for that model. When you make the option value available, its family also becomes available. Family can be mandatory or discretionary, as identified by the property of the family for that product context.

Note

You cannot drop the family group into the **Availability view**.

Distinguishing between marketing and technical options

There are two categories of options or features to consider when defining and managing product variability:

- Marketing options and values. These options and values are expressed in terms easily understood by the end user or customer. They describe the feature or benefit obtained by selecting that option value.
- Technical options and values. Those options and values describe some characteristic or variability that is required for technical reasons. They are normally not directly relevant to, exposed to, or understood by the customer.

Note

You cannot currently define separate marketing and technical options in Teamcenter. You must create generic options that address both purposes.

A marketing representative or product manager typically determines the product variability to offer to the marketplace. Often, but not always, this decision is presented in terms of the specific product you are developing or planning. The terms used to describe these features are ones that resonate with a sales engineer or customer, and describe what capability or benefit is obtained by choosing that feature. Usually, the process of deciding the features to offer to the marketplace occurs separately from development of the actual parts and product data that will support the physical delivery of those features.

Only marketing features and values are available to an end user, not any technical features and values, and not product or BOM data. You can make a complete, valid set of feature selections that result in a valid 100% BOM from only the marketing option families and values.

The process of evaluating and deciding the features to offer may include technical or feasibility input from Engineering. However, at this stage of product planning, there are often no design solutions designed to support the features suggested.

Once the features to offer and any rules that restrict or mandate particular combinations are defined, Engineering typically completes the design work necessary to support the features. That is, technical personnel identify, certify, or develop the design solutions that become the physical parts delivering the specified feature.

There may be technical parameters or features implied by the marketing features exposed to the end user. For example, `Operating Market = United States` may have implications for the voltage characteristics of the electrical components in the product. This relationship or dependency is expressed by creating derived value rules from the marketing option family to the technical option families it implies. In this example, the technical option families are referenced directly from the product data, and included in the variant conditions.

These marketing and customer facing features and their characteristics may be captured in a data management system, or they may be largely or partially captured in a separate external configurator. For example, basic option pricing may be captured in a data management system, but complex pricing or discounting, including geographic dependencies, are typically managed in an external sales configurator.

For any marketing or customer feature information captured in Teamcenter, the marketing specialist or product manager has significant additional flexibility to involve product specialists as they consider the technical feasibility of proposed offerings. Both the marketing users and the engineers participate in the review and approval process.

Sales configurators generally lack the configuration management capabilities of Teamcenter. Teamcenter supports controlled evolution, history, and release of new option families, option values and rules where the sales configurator does not.

In addition, the marketing features may be directly referenced by the product data. That is, the marketing feature value is directly referenced by the product data and also directly referenced in a variant condition that controls product data applicability.

Working with variant option families

Working with variant option families

Variant option families are used to organize option values by common characteristics. They are unique in the context of the item in which they are created. The combination of variant family name and parent item ID is enforced to be unique. Variant value names must be unique in the context of their family.

You can create variant option families with the following *value data types*:

- String
- Integer
- Floating point
- Logical
- Date

You can define the list of allowed values as an enumerated list of fixed values, or as free form values within optional boundaries.

You can set the value data type and specify the list of allowed values independently. For example, you can create a **Displacement** variant family with a value data type of floating point, for which the Product Configurator holds a defined fixed list of valid values.

Related topics

- Create variant groups and families for a product context or dictionary
- Defining option families, option values, and option rules

Creating mutually exclusive variant option family values

You can define whether variant option families have mutually exclusive values or values where multiple selections can be applied at the same time. For example, a variant option family called `Accessories` may contain values that are not mutually exclusive. This allows you to select multiple accessories for a variant configuration solve. If the variant option values are defined as mutually exclusive, you can select only one value at a time.

Typically, most variant families and their values are mutually exclusive. For example, for a variant option family called `Engine`, the values are typically mutually exclusive and only one value is valid for a configured product.

Distinguishing between mandatory and optional variant option families

An optional (discretionary) variant option family contains valid product configurations that do not select a value for the family. For example, you may have a variant option family with values for optional equipment but valid product configurations exist with no optional equipment.

Conversely, when you define mandatory variant option families, all valid and complete product configurations must select at least one value for each mandatory variant option family. Leaving a mandatory variant option family unset is equivalent to making an *irrespective of* selection. Assigning an empty value to an optional family (except logical families) is equivalent to making a *none of* selection.

Alternatively, if you create optional variant option families, there may be valid and complete product configurations that do not select any value. Leaving an optional variant option family unset indicates a *none of* selection.

You can validate variant configuration criteria against constraint rules managed in the Product Configurator. The Product Configurator provides feedback as to which variant option families are not yet set sufficiently.

Creating optional variant option families

Teamcenter allows you to create optional (discretionary) variant option families. Configuration criteria that assign an empty value for optional variant option families do not select product data that references this optional variant option family. The following example shows configuration criteria that reference the discretionary variant option family called Accessories:

Configuration criteria	DesignElement DE1 Model = M1 & Accessories = A1	DesignElement DE2 Model = M1 & Accessories != A1	DesignElement DE3 Model = M1	DesignElement DE4 Model = M1 & Accessories = “”
Model = M1 & Accessories = A1	Yes	No	Yes	No
Model = M1 & Accessories != A1	No	Yes	Yes	Yes
Model = M1	Yes	Yes	Yes	Yes
Model = M1 & Accessories = “”	No	Yes	Yes	Yes

Note

In previous Teamcenter versions, you could only create mandatory variant option families for which it was not possible to define configuration criteria that explicitly do not select any value in this family.

The following table shows the differences between mandatory and optional variant option families:

Scenario	Meaning for mandatory variant option families	Meaning for optional variant option families
Family is left unset	Irrespective of	Irrespective of
Use of empty values in variant criteria	Invalid use	None of
Use of empty values in variant conditions	Invalid use	Valid if the configuration criteria do not require any value to be set.

Defining multiple-selection option families

You can create multiple-selection variant option families. They comprise configuration criteria that combine more than one value to select product data with variant conditions. They may combine values with an AND condition and an OR condition. For example, you may have a variant option family called `Accessories` and a valid product configuration exists where multiple accessory values are selected.

Previous Teamcenter versions supported only single-selection variant option families. Configuration criteria that combine more than one value for the same single-selection variant option family are interpreted in the same way as the Boolean constant FALSE. Therefore, they select only product data with a variant condition that is also equivalent to the Boolean constant FALSE, irrespective of their use of this variant option family. These criteria do not select product data with variant conditions that combine both these values with an OR condition.

The following example shows configuration criteria referencing the multiple selection variant option family `Accessories`:

Configuration criteria	Description	DesignElement DE1 Accessories = A1 & Accessories = A2	DesignElement DE2 Model = M1 & Model = M2	DesignElement DE3 Accessories != A1 | Accessories != A2	DesignElement DE4 Model != M1 & Model != M2
`Accessories = A1 & Accessories = A2`		Yes	No	No	Yes
`Model = M1 & Model = M2`	Criteria are equivalent to FALSE	No	No	No	No
`Accessories != A1 | Accessories != A2`		No	No	Yes	Yes
`Model != M1 & Model != M2`	Criteria are equivalent to TRUE	Yes	No	Yes	Yes

In this example, `Model` is not a multiple selection and the default solve type of 529 is used.

Caution

You may experience slow responses if you create multiple-selection variant option families with a large number of members.

Grouping option families

You can group option families to make it easier for the user to create variant configuration criteria. Presenting families for which the user must assign values in manageable sets reduces the complexity of working with the variant model. This approach also reduces the complexity of the techniques that can be employed to compute available variability. The effort to filter the list of available values based on all constraints grows exponentially with the number of families and the number of constraints. Response times may be acceptable for manageable sets of variant option families, but may lead to unacceptable response times if you have a large numbers of families or constraints.

After allocating variability for an option family to a product context revision, the user can group these variant option families. An option family may be assigned to one or more groups, but it is not compulsory to assign any family to a particular group. You manage the grouping at the product context revision level; the same option family may be assigned to different groups in different product revisions.

You can use Product Configurator to browse and edit the list of groups and their option families.

Related topics

- Create variant groups and families for a product context or dictionary

Working with variant rules

Working with saved variant rules

Saved variant rules define the option values which users have selected and saved for a given configuration. To configure a particular variant of an assembly or product, you set a variant rule. The variant rule is a group of option families and values such as **color = red**, **material = cotton**.

Creating fixed default and derived default variant rules

You can use the Product Configurator to enter fixed default and derived default variant rules. Users of other applications can request Teamcenter to evaluate the default rule for their variant criteria.

Teamcenter allows you to evaluate default rules in the sequence in which they were stored. You can use the Product Configurator to view and change the sequence of existing default rules.

During automatic variant criteria completion, Teamcenter gives dynamic derived default rules higher precedence than (static) fixed default rules. Teamcenter evaluates derived default rules first.

Teamcenter skips a default rule if applying it would cause a constraint rule violation. Likewise, a derived default rule is skipped if its applicability condition does not satisfy the variant configuration criteria as defined prior to applying the default rule.

Teamcenter only applies a derived default rule to the portion of the variant configuration criteria that satisfies its applicability condition. For example a derived default rule of `Seat=Flexible if Model=Adventure` changes variant criteria of `Model=Adventure | Model=Sport` to `(Model=Adventure & Seat=Flexible) | Model=Sport`.

The derived default of `Seat=Flexible` is only applied to the `Model=Adventure` portion of the criteria, while the remainder (`Model=Sport`) is not defaulted.

Default rules allow you to assign a range of values to an option family; you can create default rules for a 100% BOM product configuration and also for a 120% BOM product configuration.

You can qualify default rules for a specific set of product families. An example of a qualified default rule follows:

```
WHILE Model=Adventure : SET Seat=Luxury IF Seatbelt=3-point
```

Where:

- `Model=Adventure` is the qualifying condition that references the `Model` product model designator variant option family.
- `Seat=Luxury` is the default value assigned.
- `Seatbelt=3-point` is the applicability condition that must be met (in addition to the qualifying condition) for Teamcenter to apply the default.

The order of fixed and derived default rules impacts the evaluation order and outcome. For example, consider the following three variant option families:

Variant option family	Variant option values		
Model	Adventure	Sport	Luxury
Seat	Flexible	Firm	Comfort

Variant option family	Variant option values		
Seat belt	3+2 point	4 point	3 point

Sequence	Set default	IF condition
1	Model = Adventure	
2	Seat = Flexible	Model = Adventure
3	Seatbelt = 3+2 point	Seat = Flexible
4	Seat = Luxury	Seatbelt = 3 point
5	Seatbelt = 3 point	Model = Adventure

By default, this set of variant rules configures the Adventure model with flexible seats and a 3+2 point seat belt. However, if you move rule 5 to the top, the default configuration specifies the Adventure model with flexible seats and a 3 point seat belt.

Related topics

- Define variant defaults

Retrieving and saving variant rules

Variant rule objects store variant configuration criteria, and also optional configurator validation records. They may exist at the product level, where they are managed by the Product Configurator; they may also be managed by the product model (for example, the collaborative design). In both cases, the variant rules are attached to the managing object with relations that define their scope.

Allocating and deallocating option data

You can allocate each option family group, family, and value to one or more configurator items. The objects are shared across configurator items.

- When you allocate an option value to a configurator item, Teamcenter:
  - Ensures the value's family is allocated to the target product context.
  - Ensures the family’s group is allocated to the target product context.
- When you allocate an option family to a product context, Teamcenter:
  - Ensures the family’s values are allocated to the target product context. Every value of the family is allocated, regardless of the allocations in the source item.
  - Ensures the family’s group is allocated to the target product context.
- When you allocate an option family group to a product context, Teamcenter:
  - Ensures the group's families are allocated to a target product context item. Every family of the value is allocated, regardless of the allocations in the source item.

  - Ensures each family's values are allocated to the target product context item.

- When you allocate a summary option family to a product context:
  - Teamcenter ensures all family's values are allocated to a target product context item.
  - If you allocate a summary option value whose summary option is not allocated to the target product context, then only the selected summary option value and its summary option family are allocated to the target product context.
  - If new values are added to the source summary option family, these new values are not automatically allocated to the target product context.
- When you allocate a package option from a source dictionary to a target dictionary or a product context:
  - Teamcenter allocates a new package option to a target dictionary or a product context.
- When you deallocate an option value from a dictionary or product context, Teamcenter validates that the value is not referenced in any variant expressions for rules associated with the product context.
- When you deallocate an option or a summary family from a dictionary or a product context:
  - Teamcenter validates that the option or summary family has no allocated values in the dictionary or the product context.
  - If the option family is free form, validates that no free form values are referenced in any variant expressions for rules associated with the dictionary or the product context.
- When you deallocate an option family group or a package from a dictionary or a product context, Teamcenter validates that the group has no allocated families in the dictionary or the product context.

Teamcenter does not support updating of an allocation. For example, if you allocate a group from a source item to a target product context and then add a new option family to the group in the source, Teamcenter does automatically update the family allocations. You must manually allocate the new family to the target product context.

Deleting variant data

You can delete option family groups, families, and values only if they do not reference any configurator items and are not referenced by any configurator rules or variant expressions. You must have delete access to the relevant objects.

You cannot delete an option family if it contains any values.

You can delete configurator rules only if you have delete access to the rules.

Defining the solve type for variant configuration criteria

You can define criteria to filter product data based on variant configuration criteria to use the following solve types:

Type	Value	Result
MISMATCH	1	Select objects with variant conditions that do not match the solve criteria. It is usually combined with INVERT.
EXPLICIT	2	Select only objects with variant conditions that explicitly satisfy the solve criteria (positive solve).
COPRIME[1]	4	Select objects potentially satisfying the solve criteria.
TRUE	8	Select objects with a variant condition equivalent to the Boolean constant TRUE.
FALSE	16	Select objects with a variant condition equivalent to the Boolean constant FALSE.
CONDITION	32	Select objects with a nonconstant variant condition.
ERRORCHECK	64	Select objects with variant conditions returning an error when solved.
NOCONDITION	128	Select objects with configurable behavior having no variant condition.
NOCONFIGBEHAVIOR	256	Select objects without configurable behavior.
INVERT	512	Invert the filter results, that is, remove what otherwise passes, and pass what would otherwise have been taken out.

Authoring variant constraints and validating configurations

Working with configurator constraints

You create variant constraints to restrict or warn users of unsupported combinations when they perform a configuration activity. When the user tries to select one of the option value combinations restricted by the configurator constraint, Teamcenter displays a message of the appropriate severity. Constraints are sometimes called *rules*.

You create constraints in the product context. Constraints reference product models through a model expression. The model expression is a variant expression that consists of a logical combination of product models, for example, `ModelA OR ModelB`. Product models behave as option values in these expressions.

1. Two integral numbers are coprime if their greatest common divisor is 1. Two Boolean expressions are coprime if their greatest common Boolean divisor is TRUE.

Constraints may be defined as:

- Inclusive

  Inclusion rules are met when the condition is satisfied. Teamcenter modifies the configuration expression to enforce the constraint.

  ```
  WHENEVER modelCondition { IF applicabilityCondition THEN ENFORCE constraintCondition
  ELSE ENFORCE message WITH severity }
  ```

  In this example, `constraintCondition` is a valid combination and enforces `message` if the valid combination cannot be set.

- Exclusive

  Exclusion rules are met when the condition is not satisfied. Teamcenter does not attempt to modify the configuration expression to satisfy the exclusion rule. It does display a message when the exclusion rule is violated.

  ```
  WHENEVER modelCondition { IF applicabilityCondition AND constraintCondition
  THEN ENFORCE message WITH severity }
  ```

  In this example, `constraintCondition` is an invalid combination and enforces `message` if the invalid combination is applied.

In both examples, `modelCondition` and `applicabilityCondition` may be NULL (always TRUE).

Understanding configurator constraints

You can use the Product Configurator to write both exclusive constraints and inclusive constraints. Users of other applications can request Teamcenter to validate their variant criteria against these constraints. You can qualify constraints for a specific set of option families.

- Exclusive constraints describe a disallowed condition. The following exclusive constraint disallows the 3+2 point seat belt for the `Luxury` model.

  ```
  WHILE Model=Luxury : IF Seatbelt=3+2-point
  ENFORCE message WITH SEVERITY ERROR "Incompatible selection";
  ```

- Inclusive constraints describe a required condition. The following example of an inclusive constraint forces the 3+2 point seat belt for the `Adventure` model.

  ```
  WHILE Model=Adventure : UNLESS Seatbelt=3+2-point
  ENFORCE message WITH SEVERITY ERROR "Incompatible selection";
  ```

For each constraint, you can set one of three severity values—information, warning, or error. Teamcenter displays a message with the appropriate severity value if a constraint condition violation occurs. A message occurs only after Teamcenter considers every possible way of satisfying the variant configuration criteria and determines whether the remaining variant configuration criteria allow you to avoid a constraint condition violation.

When requesting available variability of a family or set of families, Teamcenter only considers constraints with error severity when filtering available values. If constraints exist that reduce the available variability of the family to a single value, requests for available variability only return this value. However, automatic variant criteria completion does not automatically assign the remaining value, even if the family is marked as mandatory.

Authoring a variant constraint

You create constraints in the context of an item revision that acts as an option library, dictionary, or product. However, in the same way as option families, constraints are independent of the specific product or structure.

When defining a constraint, you provide a variant expression that acts as the constraint condition. The expression is a logical combination of option value selections.

Constraints may be defined as inclusive or exclusive. Inclusive constraints are met when the condition is satisfied; exclusive constraints are met when the condition is not satisfied. Teamcenter displays a message if a constraint is violated.

Note

Inclusive constraints express *Do* or *Must Do* conditions. They contrast with exclusive rules that express *Don't* conditions". If an inclusive constraint says "If Exterior=Black → Interior=Anthracite", the system forces anthracite interiors for black exteriors. Conversely, if an exclusive constraint says "If Exterior=Black → Interior=Anthracite", the system disallows anthracite interiors for black exteriors.

An example of an inclusive constraint:

```
WHILE Model=Deluxe IF Exterior=Black → Interior=Anthracite
ENFORCE Color=Black WITH SEVERITY ERROR
"You must choose an Anthracite interior";
```

In this example, the constraint is a valid combination and displays an error message if the valid combination is not observed.

An example of an exclusive constraint:

```
WHILE Model=Deluxe IF Exterior=Black → Interior=Anthracite
DISALLOW Color!=Anthracite WITH SEVERITY ERROR
"You cannot choose an Anthracite interior with a black exterior";
```

In this example, the constraint is an invalid combination and displays an error message if the invalid combination is applied.

Constraint messages may be localized, so that they are displayed in the user's current locale.

Checking variant constraints

Variant constraints (including derived option value rules and compatibility constraints) form a complex network of logic that may potentially include circular references, rules that can never be satisfied, and rules that never result in a selected option value for a valid set of option families. Teamcenter detects these conditions when they exist, allowing you to ensure that your rules result in complete and consistent results for all combinations.

Teamcenter provides feedback if any of the following situations exist:

- A rule (independent of any other rules) can never be satisfied.
- A rule (independent of any other rules) is always true.
- For a product model, a set of rules when executed together can never be satisfied.

- For a product model, a set of rules when executed together result in an endless chain of logic — a circular reference.

Teamcenter provides feedback as you work if your current selections would not provide any valid combination of selections for the remaining mandatory option families. It also provides feedback on the remaining number of valid possible configurations based on the selections made so far in the configuration process.

Validating a configuration

How do you validate a configuration?

You validate a configuration against constraints such as inclusion and exclusion rules using the *Variant Configuration view*. When you configure and assign values to option families and ask Teamcenter to validate the configuration, the system evaluates the constraints and returns a list of information, warnings, and errors on the various combinations of the families.

You can use the following methods to validate the variant configuration:

- Configure with custom variant configuration using manual validation.

  In manual validation mode, no wizards are used. Variant option defaults and configuration validation is not automatically applied. You can optionally apply defaults and validate the configuration manually.

- Configure with custom variant configuration using active validation.

  In active validation mode, a wizard-like format is used to display variant option families of one group at a time. Variant option defaults and configuration validation is automatically applied whenever you navigate to a different group.

  When you select a different group, Teamcenter does the following:

  - Validates the current selections and displays any violations. If any of the violations are errors, you cannot apply the configuration.

  - Applies default rules to the families in the next group.

Validating a configuration against variant constraints

When a user configures (assigns values to) option families and asks Teamcenter to validate the configuration, the system evaluates the constraints and returns a list of information, warnings, and errors on the various combinations of the families.

Example

You have an option family with the variability shown in the following table and constraints of `ERROR IF L=1 & W=20` and `ERROR IF L=2 & W=10`.

Family	Values	Required
L	`{1; 2}`	Yes
W	`{10;20}`	Yes
O	`{Y}`	No

The validation results are as follows.

Validation result	Description	Example Configuration
Valid = true Complete = true	A valid product configuration for a product that can be built, that is, one buildable product. It is sometimes called a 100% BOM. Configurations that assign a discrete value to all families are complete. Empty values are considered to be discrete in this context. Only the EQUAL operator is considered to be an assignment.	L=1 & W=10 & O="" L=1 & W=10 & O=Y
Valid = true Complete = false	A valid overlay of product configurations for multiple products that can be produced, that is, many buildable products. It is sometimes called a 120% BOM. Configurations that do not violate any constraint are valid.	L=1 & O="" L=1 & W=10 & O!=Y L=1 & W=10 & O!="" L=1 & W=10
Valid = false Complete = false Valid = false Complete = true	An invalid overlay of product configurations that does not match any product that can be produced, that is, no buildable products. It is sometimes called a 150% BOM.	L=1 & W=20 L=2 & W=10 & O!="" L=1 & W=20 & O="" L=2 & W=10 & O=Y

You can ignore the validation check constraint messages and continue to create the variant constraints. This action may be appropriate if you are a product manager or a sales engineer who requires a new variant design based on a market need.

For example, you have the following constraint that has an option family `[Production]Country` with two values, `India` and `US`:

```
IF [Car]DriverSeatPosition = Left AND
[Production]Country = India THEN error
"You are choosing wrong Driver Seat position
for a Car in India"
```

If you select `[Car]DriverSeatPosition = Left AND [Production]Country = India`, the constraint returns an error. However, Teamcenter does not correct or clear the incorrect value of `India`.

Validating a configuration against variant constraints with package options

Creating package options allows you to group a set of option values for convenience of selection within configuration order strings. You assign existing option values are assigned as members of a package option and are connected using a logical AND operation in the variant expression.

When validating a configuration with package options, the AND combination of all the package members of selected package options is considered when processing the constraints and availability rules. If a validation rule is violated based on the package option, package option is flagged with a violation.

Example

You create an option family with the following values.

Family	Values	Required
L	`{1; 2}`	Yes
W	`{10;20}`	Yes
H	`{a;b;c}`	Yes
O	`{Y}`	No

You create a package family with the following values.

Package family	Values	Members
P	`{P1; P2}`	P1: `{L=1;W=20}` P2: `{L=2;W=10}`

You create the following constraints:

- `ERROR IF H=a & P=P1`
- `ERROR IF H=b & P=P1`

The validation results are as follows.

Validation result	Description	Example Configuration
Valid = true Complete = true	A valid product configuration for a product that can be built, that is, one buildable product. It is sometimes called a 100%	H=c & P=P2 & O="" H=c & P=P1 & O=Y

	BOM. Configurations that assign a discrete value to all families are complete. Empty values are considered to be discrete in this context. Only the EQUAL operator is considered to be an assignment.	
Valid = true Complete = false	A valid overlay of product configurations for multiple products that can be produced, that is, many buildable products. It is sometimes called a 120% BOM. Configurations that do not violate any constraint are valid.	L=1 & O="" H=c & W=10 & O!=Y P=P2 & O!="" P=P1
Valid = false Complete = false	An invalid overlay of product configurations that does not match any product that can be produced, that is, no buildable products. It is sometimes called a 150% BOM.	H=a & P=P1 H=b & P=P2 & O!=""
Valid = false Complete = true		H=a & P=P1 & O="" H=b & P=P2 & O=Y

Validating a configuration against variant constraints with summary options

Creating summary options allows you to create an option that summarizes multiple option values within the same option family. Existing option values are assigned as members of a summary option and are connected using a logical OR operation in the variant expression.

Summary options are only available when configuring with a custom variant configuration using manual validation and *not* available using active validation. An OR combination of all members of selected summary options is considered when processing constraints and availability rules. If a validation rule is violated based on a summary option, the summary option is flagged with the violation.

Example

You create an option family with the following values.

Family	Values	Required
L	`{1; 2}`	Yes
W	`{10;20}`	Yes
H	`{a;b;c}`	Yes
O	`{Y}`	No

You create a summary option family with the following values.

Package family	Values	Members
S	`{S1; S2}`	S1: `{L=1;L=2}` S2: `{W=10;W=20}`

You create the following constraints:

- `ERROR IF H=a & S=S1`

- `ERROR IF H=b & S=S2`

The validation results are as follows.

Validation result	Description	Example Configuration
Valid = true Complete = true	A valid product configuration for a product that can be built, that is, one buildable product. It is sometimes called a 100% BOM. Configurations that assign a discrete value to all families are complete. Empty values are considered to be discrete in this context. Only the EQUAL operator is considered to be an assignment.	H=c & L=1 & W=10 & O="" H=a & L=2 & W=20 & O=Y
Valid = true Complete = false	A valid overlay of product configurations for multiple products that can be produced, that is, many buildable products. It is sometimes called a 120% BOM. Configurations that do not violate any constraint are valid.	H=a & S=S2 & O="" H=b & S=S1 & O!=Y L=1 & O!="" S=S1
Valid = false Complete = false Valid = false Complete = true	An invalid overlay of product configurations that does not match any product that can be produced, that is, no buildable products. It is sometimes called a 150% BOM.	H=a & S=S1 H=b & S=S2 & O!="" H=a & S=S1 & O="" H=b & S=S2 & O=Y

Validating a configuration with availability rules

Using *availability rules*, you can define which option values, families and package options are available for the individual product models within a product context. When validating a configuration using availability rules, a violation of type error is displayed if a selected option in the configuration is not available for *all* the selected product models. No violation messages are displayed if a selection option is available for at least one selected product model.

Note

You can turn off the evaluation of availability rules by setting the **Cfg0EvaluateAvailability** site preference to **false**.

Example

You create an option family with the following values.

Family	Values	Required
L	{1; 2}	Yes
W	{10;20}	Yes
H	{a;b;c}	Yes
O	{Y}	No

You create a model family with the following values.

Model family	Product Model	Available option values
MF	PM1	{L=1; W=20; H=a; O=Y}
MF	PM2	{L=2; W=10; H=b; O=Y}
MF	PM3	{W=10; H=c}

The validation results are as follows.

Validation result	Description	Example Configuration
Valid = true Complete = true	A valid product configuration for a product that can be built, that is, one buildable product and sometimes called a 100% BOM. Configurations that assign a discrete value to all families are complete. Empty values are considered to be discrete in this context. Only the EQUAL operator is considered to be an assignment.	MF=PM1 & L=1 & W=20 & H=a & O="" MF=PM2 & L=2 & W=10 & H=b & O=Y
Valid = true Complete = false	A valid overlay of product configurations for multiple products that can be produced, that is, many buildable products. It is sometimes called a 120% BOM. Configurations that do not violate any constraint are valid.	MF=PM1 & L=1 & O="" MF=PM2 & L=2 & W=10 & O!=Y W=10 & O!=""
Valid = false Complete = false	An invalid overlay of product configurations that does not match any product that can be produced, that is, no buildable products. It is sometimes called a 150% BOM.	MF=PM1 & L=1 & W=20 & H=b The **b** option value is not available for **PM1**. MF=PM2 & L=2 & W=20 & O!="" The **20** option value is not available for **PM2** .
Valid = false		MF=PM1 & L=2 & W=10 & H=b & O=" The **2**, **10**, and **b** option values are not

Complete = true	available for **PM1**. MF=PM3 & L=2 & W=10 H=b & O=Y The **2**, **b**, and **Y** option values are not available for **PM3**.

Working with variants in 4GD

Associating a product context with a 4GD collaborative design

Special considerations apply if you are defining variants to use in a 4GD environment.

You associate a product context with a 4GD collaborative design by attaching it to the collaborative design with the GRM relation type specified in the **TC_variant_configurator_relationship** preference. The default value of this preference is **Mdl0HasConfiguratorContext**. The collaborative design is the primary object in this relation, and the configurator item is the secondary object.

Alternatively, you can associate item revisions that reference legacy variant data with the collaborative design using the same relationship.

Rolling down variant conditions

Variant conditions roll down from 4GD design control elements to design features, and also from design elements to their subordinate design elements, in the same way as effectivity conditions. Variant conditions are stored in two properties:

- **mdl0allowed_var_formula**

  Contains the variant condition that the user explicitly assigned to this element.

- **mdl0variant_formula**

  Contains the computed variant condition that takes roll down into consideration.

The computed variant condition in the **mdl0variant_formula** property is always owned locally. It is never exported and always computed from locally available roll down contexts, even if the corresponding 4GD object is a remote replica. Saving a variant condition for a locally owned design control element that controls a remote replica design feature causes Teamcenter to compute a new **mdl0variant_formula** property for the remote replica design feature. Conversely, the **mdl0allowed_var_formula** property of a remote replica design feature is updated only as a result of a Multi-Site Collaboration synchronization.

Overlaying multiple configurations for impact analysis

For some tasks, end users should only configure valid variants. If a selection violates a rule, the user is prevented from proceeding until the error condition is resolved. In other tasks, the user may be permitted to make invalid selections if they validate the selections before saving or applying the selected option string and resolve any invalid selections. That is, you can make invalid choices during

the configuration process but cannot configure an invalid variant of the structure. In some scenarios, a user may overlay multiple configurations to make comparisons or perform impact analysis.

Teamcenter allows you to:

- Overlay multiple valid variants on top of each another. All configured parts are valid for at least one single discrete valid variant, even though there is more than one variant represented in the configuration. For example, variant rule 1 OR variant rule 2 OR variant rule 3, where each variant rule represents a valid 100% BOM product variant.

- See an unbuildable or invalid variant that includes multiple selections for an option family whose values are mutually exclusive or multiple selections that are otherwise invalid for a 100% configuration. For example, `variant rule 1 = (Transmission = Manual, Engine = V6 or V8, Engine = Gasoline OR Diesel)` may configure content if `Gasoline` and `V8` are selected together, even if this is not an allowed configuration.

- Select a configuration or partial configuration, and then request to see all parts that are valid in *any* configuration together with the selected option families. This configuration is often referred to as valid overlays only (VOO).

To work with overlays, you define multiple saved variant rules that can be applied simultaneously, resulting in a 120% BOM. Each time you apply a new saved variant rule or remove an applied rule, Teamcenter reconfigures the BOM.

Some of the configuration types listed are only necessary for certain user roles and tasks. You can use access control to configure whether a user can produce and save an invalid, overlayed partial, or 120% BOM variant rule, or only a complete, validated option string.

Related topics

- Define variant rules

Writing variant expressions in Teamcenter Normal Form (TNF)

You build variant expressions in Teamcenter Normal Form (TNF) and the Product Configurator displays them in **Variant Expression Editor** grid, as shown in the following examples. They must adhere to the following rules:

- A variant expression consists of one or more subexpressions. Each subexpression is joined by an OR.

- A subexpression consists of one or more expression groups. Each group is joined by an AND.

- An expression group consists of one or more terms. Each term is joined by an OR.

- A term can represent one of the following:

  - A discrete selection in the form of [*namespace*]*family* <*operator*> value, where *operator* can be one of = or !=.

  - One or two range expressions. Each range expression is joined by an AND.

- A range expression is of the form <*operator*><*valueText*>, where *operator* can be one of >, >=, <, <=.

When you build multilevel expressions:

- The outermost level of an expression represents a separate column in the grid.
- The next level represents selections in different families and selections within the same family if they are not mutually exclusive.
- The next level represents selections within the same mutually exclusive family. This is the family level where only values belonging to the same family are nested, with the exception of non-mutually exclusive selections.
- The next level represents ranges and equalities.
- The innermost level is the literal, indicating the selection value itself.

Note

Use a forward slash character `/` to represent `AND/OR`, that is, `EITHER ONE, OR BOTH`. Use a vertical bar character `|` to represent `ONE OF`.

Binary operators require two operands.

AND and OR are commutative. Teamcenter may not necessarily return the expression in the same order you entered it, providing the expression remains logically correct. For example, you enter `Color=Red & Engine=V8 & Accessories=Bells`; Teamcenter may return `Engine=V8 & Accessories=Bells & Color=Red`.

The following examples show how common TNF expressions are represented in the grid:

- TNF expression:

```
Color = Red
```

Grid representation:

Color	**Red**	✔

- TNF expression:

```
Width >= 100 & Width <= 200
```

Grid representation:

Width	**>=100 & <=200**	✔

- TNF expression:

```
Width <= 100
```

Grid representation:

Width	**<=100**	✔

- TNF expression:

```
GPS != GPS
```

Note

`GPS` is a logical type.

Grid representation:

GPS		⊘

- TNF expression:

```
(Model = 0PC08 | Model = 0PC35 ) &
 (Length = 2000 | (Length = 3000)
```

Grid representation:

Model	**OPC08**	✔	
	OPC35	✔	
Length	**2000**		✔
	3000		✔

- TNF expression:

```
Width >= 1000 & Width <= 1500 &
 ( Length >= 2000 & Length <= 2500 ) |
 ( Length >= 3000 & Length <= 3500)
```

Grid representation:

Width	**>=1000 & <=1500**	✔
Length	**>=2000 & <=2500**	✔
	>=3000 & <=3500	✔

- TNF expression:

```
( ( Model = 0PC08 | Model = XYZ) &
( Color != Red ) ) | ( ( Color = Red ) &
( Width >= 1000 & Width <= 1500 ) & NOT
( Length >= 2000 & Length <= 2500 ) )
```

Grid representation:

Model	**OPC08**	✔

	XYZ	✔	
Color	**Red**	⊘	✔
Width	**>=1000 & <=1500**		✔
Length	**>=2000 & <=2500**		⊘

The TNF expression is a normalized form of the input expression.

- If the expression cannot be satisfied, it always resolves to false and you get an empty TNF vector. For example, `'Engine = Diesel' AND 'Engine = Petrol' and ‘Engine’` is not a multiple selection family.
- If the expression can always be satisfied, it always resolves to true and you get a TNF vector with only one element (a null variant expression). For example, `'ACRequired = ACRequired | ACRequired != ACRequired'`.

Managing how variant expressions display

The displayed format of variant expressions is configured with the following preferences:

- **TC_show_option_namespace_prefix**

  Enables the display of the family namespace within the expression string.

- **TC_show_option_family_prefix**

  Enables the display of the option families within the expression string.

Use these preferences to configure the following display formats:

TC_show_option_ namespace_prefix	TC_show_option_ family_prefix	Expression format
true	**true**	[*namespace*]*family* = *value*
false	**true**	*family* = *value*
true	**false**	[*namespace*]*value*
false	**false**	*value*

Character representations:

- AND is represented by **&**

  ```
  [NS]A = a1 & [NS]B = b1
  ```

- OR is represented by **|**

  ```
  [NS]A = a1 | [NS]A = a2
  ```

- NOT is represented by **!**

  ```
  [NS]A != a1
  ```

- Boolean values are always displayed without a family, that is, as if the **TC_show_option_family_prefix** preference is false.

```
[NS]BOOL
![NS]BOOL
```

- Date values are displayed in the standard Teamcenter locale-specific date format.

- Range expressions are displayed using inequality operators

```
[NS]A = a1 & [NS]LENGTH <= 50
[NS]A = a1 & [NS]LENGTH <= 50
[NS]A = a1 & [NS]LENGTH >= 20
```

- Multiple column expressions are grouped in parenthesis

```
([NS]A = a1 & [NS]B = b1) | ([NS]A = a2 & [NS]C = c1)
```

- Multiple selections within a family are grouped in parenthesis

```
([NS]A = a1 | [NS]A = a2) & [NS]B = b1
```

- Optional families support **NONE** or **ANY**

```
[NS]A = a1 & [NS]D = NONE (same as [NS]D = '')
[NS]A = a1 & [NS]D = ANY (same as [NS]D != '')
```

Chapter 3: Using the Product Configurator

Create variant groups and families for a product context or dictionary

You can create variant option groups, families, and values for a product context or dictionary. Variant data can be reused in multiple product structures and application models (collaborative designs or partition templates).

1. If necessary, create the product context or dictionary by choosing **File→New→Product context** or **New→Dictionary**, respectively.

   Teamcenter displays the **New Business Object** wizard, allowing you to create the necessary product context or dictionary.

   Alternatively, you can search for and open an existing product context or dictionary.

2. Teamcenter opens the product context or dictionary in the **Variant Options** view, with any existing variant data shown. Any existing variant groups, families, and their values are displayed in the tree table.

   Note

   You can send more than one item revision to the Product Configurator and each one opens a new **Variant Options** view.

3. Select the product context or dictionary and click **Add Group** in the view toolbar.

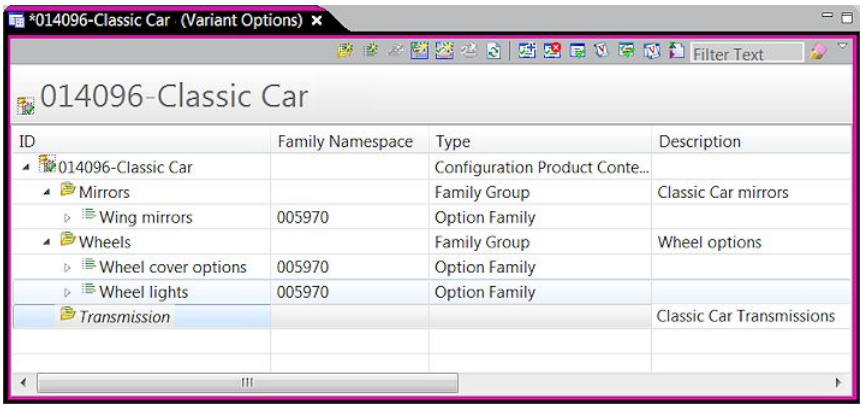
   Teamcenter adds an empty row for the new group.

4. Click in the **ID** field of the empty row and enter the ID of the group. The chosen ID must be unique for the current product context.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

5. Click in the **Family Namespace** field of the new row and enter the family namespace of the group.

6. Click in the **Description** field of the new row and enter the description of the group.



7. Click **Save** to save the new group.

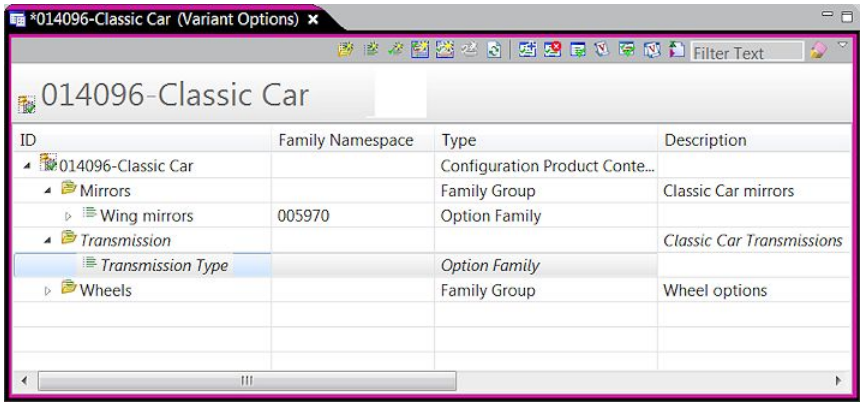
   Teamcenter displays the name and description in nonitalic text, indicating the group is saved.

8. Enter data for all ungrayed fields in the group by clicking in each column in turn, and then selecting or typing the required value. You can also edit data of an existing group.

9. Define variant data for the new group by doing one or more of the following:

   - Add a subgroup to the new group by right-clicking the new group, choosing **Add Group**, and following steps 2 through 7 previously.

   - Click **Add Family** to add a new family to the selected group. A new empty row appears as the last child of the selected group, and the group is expanded if necessary. When you enter the ID of the family, the chosen ID must be unique for the current product context or dictionary.



   - Click **Save** to save the new family.

     Teamcenter displays the name and description in nonitalic text, indicating the family is saved.

   - Enter data for all fields in a family by clicking in each column in turn, and then selecting or typing the required value. You can also edit data of an existing family.

     Note

     If you create a unit of measure for the family, it is read-only and you cannot subsequently modify it.

- Remove a selected family by clicking **Cut** .
- Change the values in the **Sequence** column to reorder the values in the family.

  Note

  You can also paste or drag existing families and their values from a product context or dictionary. However, if you try to paste or drag legacy data, you are not able to save any changes.

10. Click **Save** on the main toolbar to save all changes.

    Note

    If you subsequently revise the underlying product context or dictionary, Teamcenter carries the associated variant data forward to the new revision. However, if you copy the underlying product context or dictionary by saving it under a new name (that is, by selecting the **Save As** command), Teamcenter does not carry forward the variant data.

Define variant option values

You can create values for variant option families in the context of the product context or dictionary. Variant option values can be reused in multiple product structures and application models (collaborative designs or partition templates).

1. Open the product context or dictionary in the Product Configurator and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Select the family for which you want to add an option value and click **Add Value** .

   Teamcenter adds a new empty row as the child of the selected family and expands the family if necessary.

3. Click in the **ID** field of the empty row and enter the ID of the option value. The chosen ID must be unique for the current family; however, the same ID may be used in different families for the same product context or dictionary.

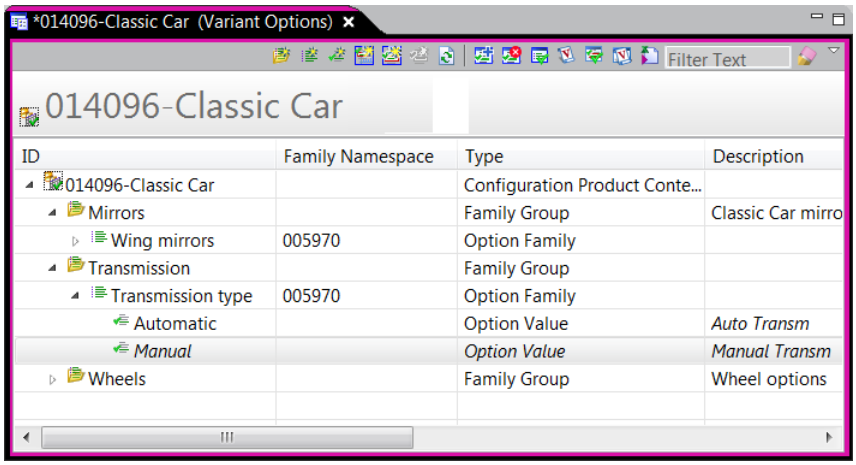
   Teamcenter adds the ID in italics, indicating the entry is not yet saved.

4. Click **Save** to save the new value.

   Teamcenter displays the ID in nonitalic text, indicating the value is saved.

5. Click in the **Description** field of the empty row and enter the description of the option value.

   Teamcenter adds the description in italics, indicating the entry is not yet saved.



6. Click **Save** on the main toolbar to save all changes.

Note

You can change or edit an existing option value by clicking the field corresponding to the value, then typing in the required new value.

You can copy option values on an existing family associated with a product context or dictionary in one **Variant Options** view, and then paste them on to a family associated with another product context or dictionary in another **Variant Options** view. You can also drag and drop option values to achieve the same results. However, you cannot save legacy variant option values copied in this way.

Define variant rules

You can author variant rules for selecting option values in the context of a product context or dictionary. These selection rules include fixed defaults, derived defaults, and derived values.

The product context or dictionary maintains a set of variant options and their rules.

1. Open the product context or dictionary and then make the **Variant Options** view active.

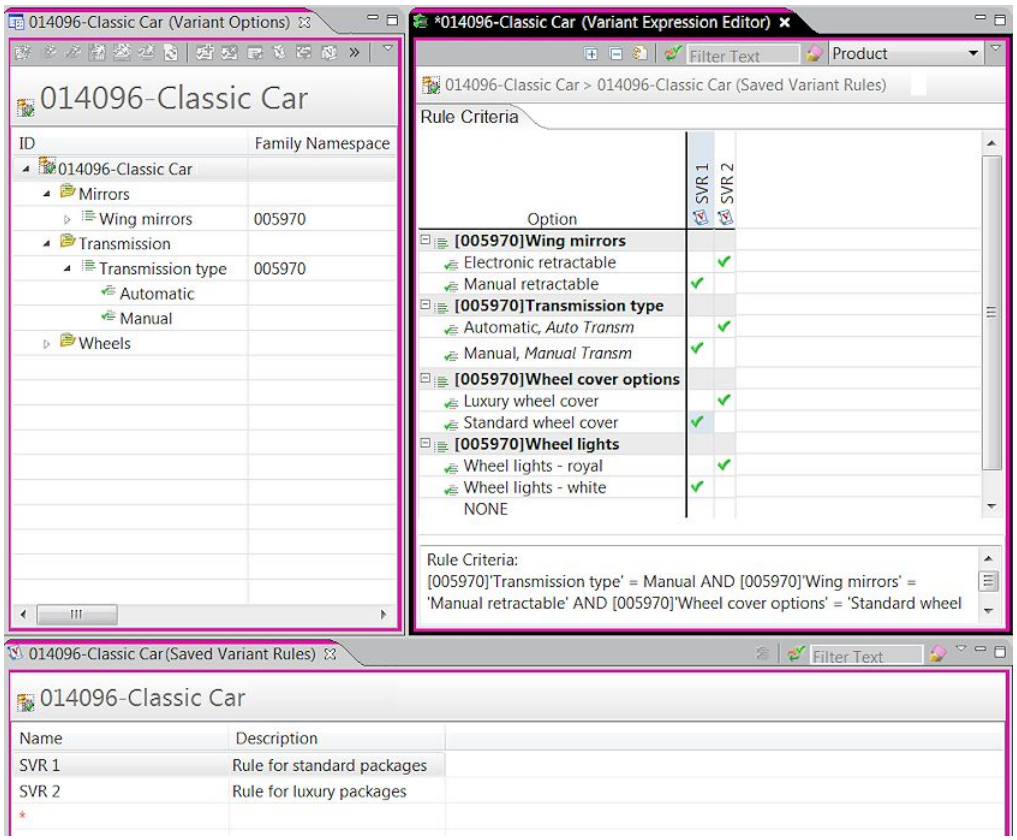
2. Click to add a new saved variant rule (SVR).

   Teamcenter displays the **Saved Variant Rules** view.

3. Enter a name and description of the rule, and then click .

   Teamcenter displays the **Variant Expressions Editor** view.

4. Construct a variant expression for the selected rule in Teamcenter Normal Form (TNF), as shown in the following example:



In Teamcenter Normal Form:

- Option values are represented by columns, organized by their families. You specify an expression by clicking the cells below the required values one or more times.
- Selections are indicated by a check mark or crossed circle in the cell.
- Selections in the same row are combined (AND function), except for selections in the same family, which are alternatives (OR function). You can add multiple rows for the same expression.
- Multiple subexpressions are combined (OR function).

You can also edit an existing rule by changing any of the fields. Updated rules are displayed but not yet saved.

5. Repeat the previous steps as necessary for other rules.
6. Click **Save** on the main toolbar to save all additions, deletions, and edits to the database.

Define variant defaults

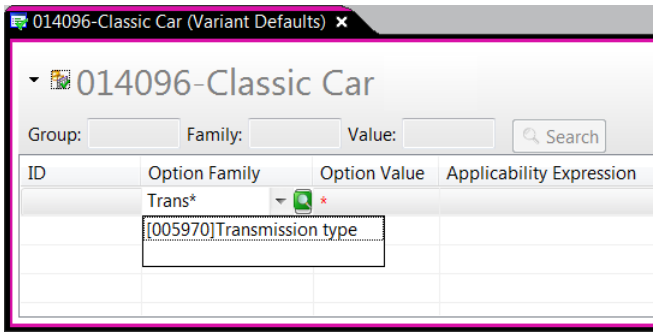
1. Open the product context or dictionary in the **Variant Options** view, and then click the **Variant Defaults** button.

   Teamcenter displays the **Variant Defaults** view.

2. Specify values in the **Group**, **Family** and **Value** fields to limit the number of displayed variant defaults. You can also click **Search** to refresh the list.

3. Click in the **ID** field of the empty row and enter the ID of the variant default. The chosen ID must be unique for the current group, family and value; however, the same ID may be used in different families for the same product context or dictionary.

4. Type or search for **Option Family** in the first row of the table.

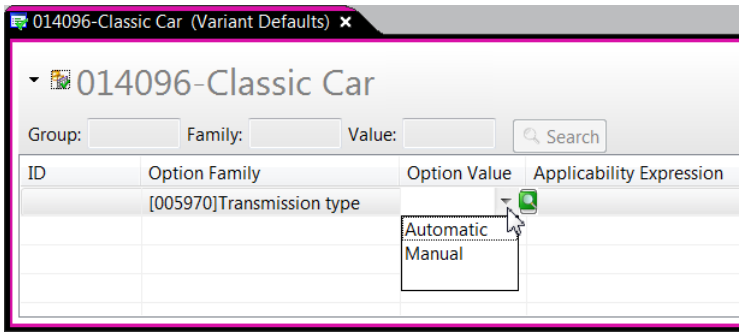


   Teamcenter displays a list of the names of all available families.

5. Choose the required family from the list.

6. Click **Option Value** of the same row.

   Teamcenter displays a list of values for the selected family. Alternatively, if the family is not a list, it displays a free text field.



7. Choose the required value from the list or key its value.

8. Click the **Applicability Expression** field of the same row and then click .

   Teamcenter displays the **Variant Expression Editor** view with the **Applicability** tab selected.

9. Define a TNF expression for the applicability.

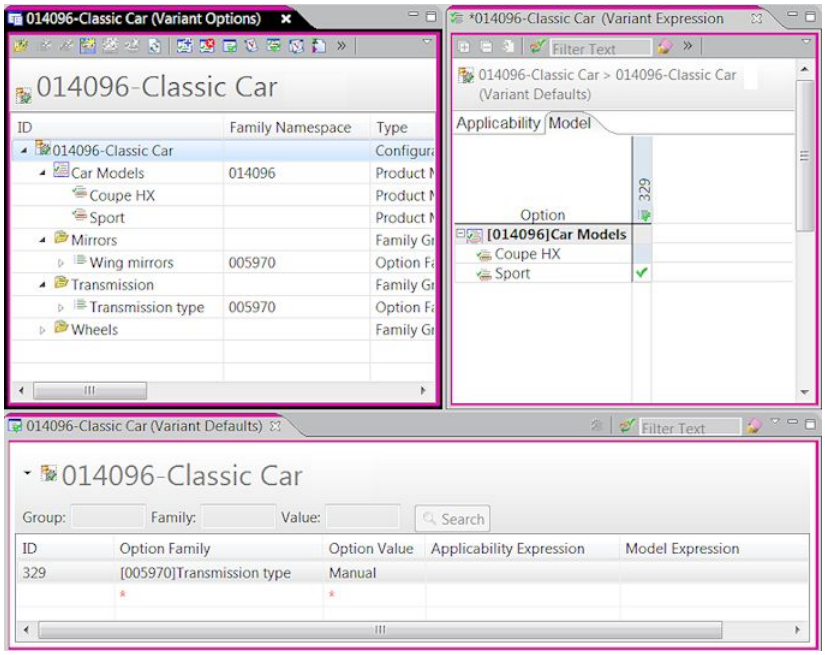
   Teamcenter populates the applicability field with the expression you defined.

10. Click the **Model Expression** field of the same row and click .

    Teamcenter displays the **Variant Expression Editor** view with the **Model** tab selected.

11. Define a TNF expression for the model.

Teamcenter populates the model field with the expression you defined. For example:



In Teamcenter Normal Form:

- Selections are indicated by a check mark ✔ or crossed circle ⃠ in the cell.
- Selections in the same row are combined (AND function), except for selections in the same family, which are alternatives (OR function). You can add multiple rows for the same expression.
- Multiple subexpressions are combined (OR function).

12. Repeat steps 3 to 11 to create additional variant default rules, as required.
13. (Optional) Change the values in the **Sequence** column to reorder the sequence in which variant default rules are displayed.
14. Click **Save** on the main toolbar to save all new and changed data.

Related topics

- Creating fixed default and derived default variant rules

Define configurator constraints

You can author variant constraints for a set of variant option values in a product context or dictionary. Variant constraints (sometimes called *rules*) define errors, warnings, and informative messages that Teamcenter generates if an invalid selection of variant option values is made. Teamcenter supports two types of variant constraint: inclusion rules and exclusion rules. Separate editors are provided for each type of rule but the procedure for their use is very similar.

1. Open the product context or dictionary in the **Variant Options** view, and then click to open the **Inclusion Rules** editor or to open the **Exclusion Rules** editor.

   Teamcenter displays the selected editor.

2. In the editor, specify values in the **Group**, **Family** and **Value** fields to limit the number of displayed variant constraints. You can click **Search** to refresh the list. You can also select one or more of the **Info**, **Warning**, and **Error**, check boxes to additionally filter the list of displayed constraints.

3. Click in the **ID** field of the empty row and enter the ID of the rule. The chosen ID must be unique for the current group, family and value; however, the same ID may be used in different groups, families, or rules in the same product context or dictionary.

4. Click the **Severity** field in the same row.

   Teamcenter displays a list of all severity types.

5. Select a severity (**Info**, **Warning** or **Error**).

6. Click the **Message** field of the same row and key the text of the message.

7. Click the **Inclusion Expression** or **Exclusion Expression** field of the same row and then click .

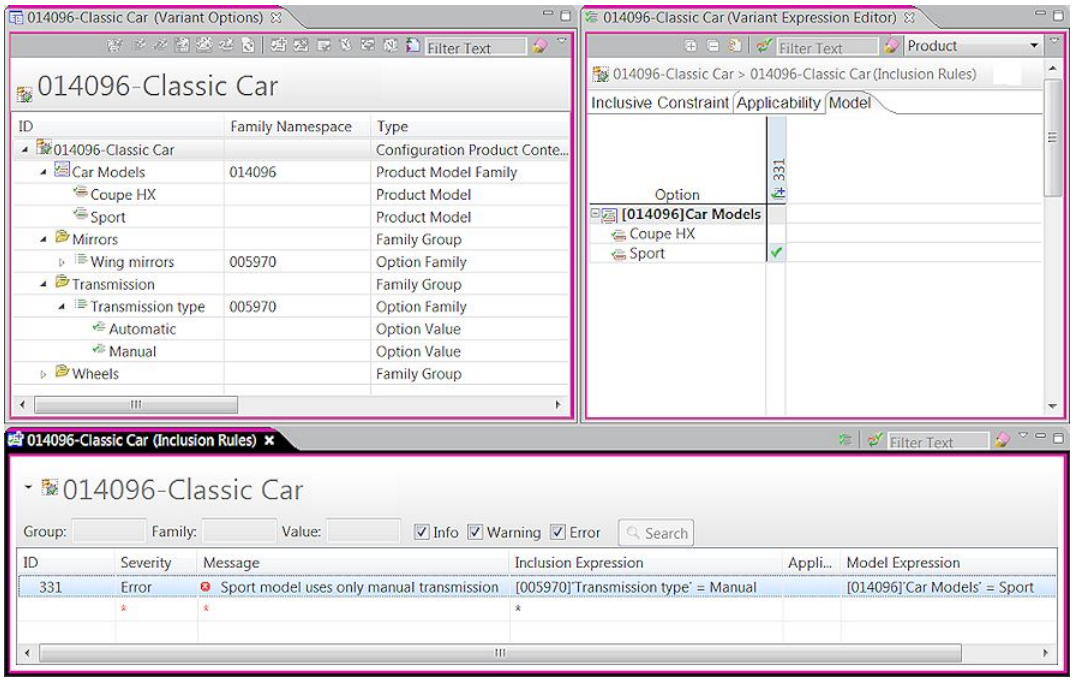
   Teamcenter displays the **Inclusion Rules** editor or the **Exclusion Rules** editor.

8. Define a TNF expression for the constraint.

   Teamcenter populates the constraint field with the expression you defined.

9. Repeat steps 7 and 8 for the **Applicability Expression** and **Model Expression** fields.

   Teamcenter populates the fields with the expressions you defined.



Define a TNF expression for the model.

When you define expressions in Teamcenter Normal Form:

- Selections are indicated by a check mark or crossed circle in the cell.
- Selections in the same row are combined (AND function), except for selections in the same family, which are alternatives (OR function). You can add multiple rows for the same expression.
- Multiple subexpressions are combined (OR function).

10. Repeat steps 3 to 10 to create additional variant constraints, as required.
11. Click **Save** on the main toolbar to save all new and changed data.

Create a configuration dictionary

You can create a configuration dictionary in which you can store option families and values. The dictionary provides a convenient method of collecting all the variant data related to a particular context, for example, a product. You can allocate variant option families and values directly from a dictionary item, from other products, or from library revisions

1. Choose **File→New→Dictionary**.

   Teamcenter displays the **New Business Object** dialog box.

2. Select the **Configuration Dictionary** type, enter a name and description, and then click **OK**.

   Teamcenter creates the dictionary.

The dictionary manages variant option families. Variant option family objects have a family name, a description, and a reference to the dictionary item with the **parent_item** attribute. The dictionary defines a namespace in which variant option families have a defined meaning, for example, the variant option family **CAPACITY** may exist for both refrigerators and washing machines. The two **CAPACITY** families should exist in the context of different dictionary items as follows:

- The variant option family **CAPACITY** when measured in liters references an option dictionary item representing refrigerators.
- The variant option family **CAPACITY** when measured in kilograms references an option dictionary item representing washing machines.

The dictionary may define characteristics and boundaries that apply to all values in this family across all possible uses. For example, you can define the **COUNT** variant family as an INTEGER type variant option family, whose value must always be greater than or equal to zero.

Create a summary option family in a product context or dictionary

You can create summary option families for a product context or dictionary.

1. Search for and open an existing product context or dictionary in Product Configurator and ensure the **Variant Options** view is active.

Teamcenter displays any existing values for the families in the tree table.

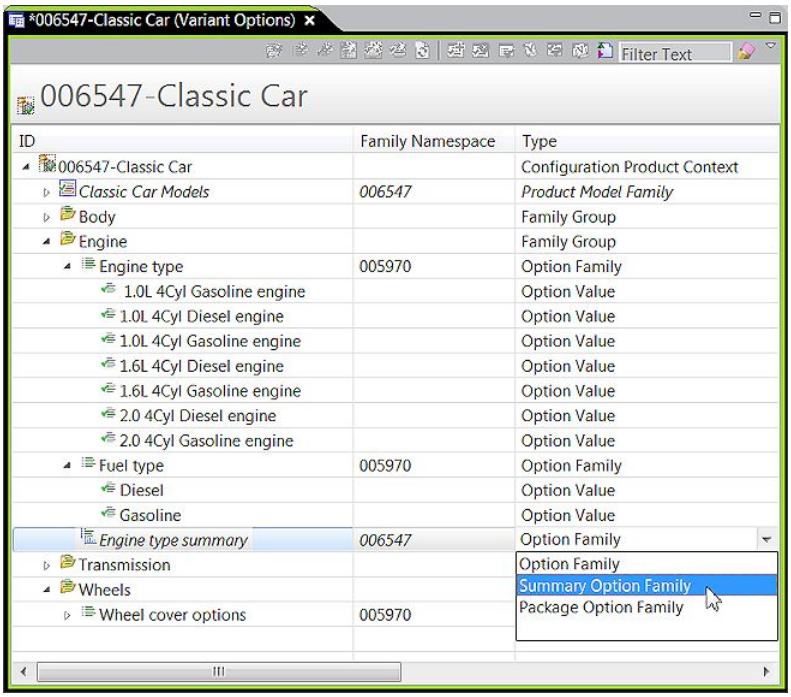
2. Select the option family for which you want to add a summary option family, and click **Add Family** .

   Teamcenter adds a new empty row as the child of the selected family and expands the family if necessary.

3. Click in the **ID** field of the empty row, and enter the ID of the summary family option. The chosen ID must be unique for the current product context.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

4. Click **Type** for the new row, and select **Summary Option Family** from the list of available types.



5. (Optional) Click **Description** for the new row, and enter the description.

   The **Value Data Type**, **Optional**, **Free-form**, **Minimum Value** and **Maximum Value** values are not modifiable.

6. (Optional) You can change the **Multi-select** field or set the **Sequence** value for the newly created summary option.

7. Click **Save** .

   Teamcenter displays the summary option family icon and name and description in nonitalic text, indicating the group is saved.

   The summary option family is created.

8. To remove a selected summary option family, right-click and choose **Remove Allocation**.

Note

Deallocate the dependent objects prior to deallocating the summary option family.

Define summary option values

You can create summary values for summary option families in the product context.

1. Search for and open an existing product context or dictionary in Product Configurator, and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Select the summary option family for which you want to add a summary option value, and click **Add Value** .

   Teamcenter adds a new empty row as the child of the selected family and expands the family if necessary.

3. Click in the **ID** field of the empty row, and enter the ID of the summary option. The chosen ID must be unique for the current product context; however, the same ID may be used in different families for the same product context or dictionary.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

4. Click **Save** .

   Teamcenter displays the summary option value icon and name and description in nonitalic text, indicating the value is saved.

   The summary option family value is created.

5. To remove a selected summary option value, right-click and choose **Remove Allocation**.

6. Define summary option values for the new option by using doing one or more of the following:

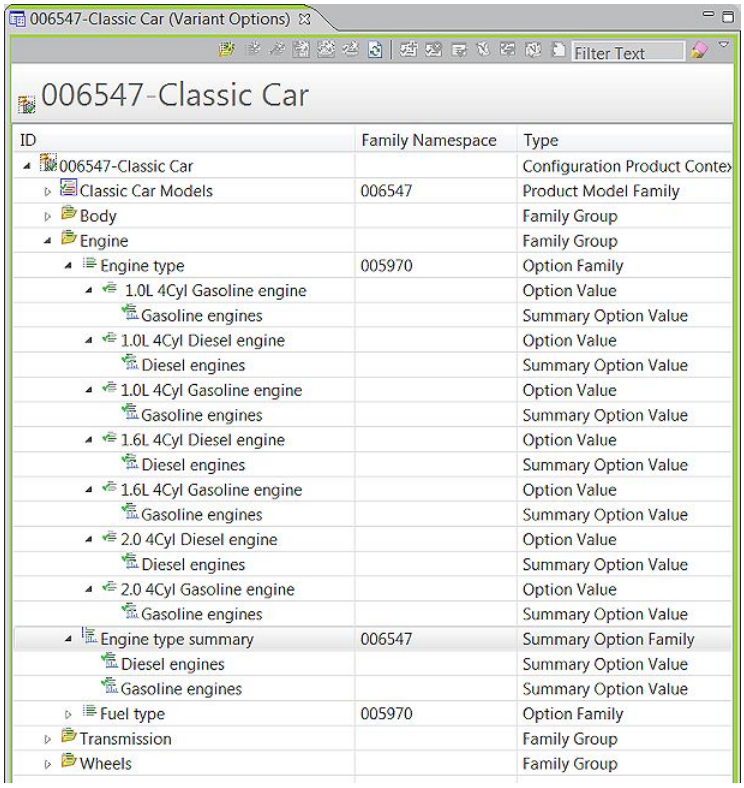
   - Copy option values and paste them onto the summary option. You can copy and paste multiple option values on the summary option.
   - Drag an existing option value onto the summary option.
   - Remove a selected option value by right-clicking and choosing **Remove Membership**. If the summary option is used in rules, an error message is displayed.

     Note

     You must have write access to the summary option family to add or remove members. Option values may belong to more than one summary option. Option values may also be from multiple multi-select or mutually exclusive families.

   For example, **1.0L 4Cyl Gasoline engine**, **1.6L 4Cyl Gasoline engine** and **2.0L 4Cyl Gasoline engine** are summarized by the **Gasoline Engines** summary option family. Teamcenter displays **Gasoline Engines** as the child element of **1.0L 4Cyl Gasoline engine**, **1.6L 4Cyl Gasoline**

engine, and **2.0L 4Cyl Gasoline engine** option values. The **Diesel Engines** summary option contains the **1.0L 4Cyl Diesel engine**, **1.6L 4Cyl Diesel engine**, and **2.0L 4Cyl Diesel engine** options.



Note

You can copy summary option values or a summary option on an existing family associated with a product context or dictionary in one **Variant Options** view and then paste them onto a family associated with another product context or dictionary in another **Variant Options** view. You can also drag and drop summary option values or a summary option to achieve the same results. You must have write access to the members of that target dictionary or product context.

If option values in the dictionary were initially summarized in the product context, you can only view those summary options that are already allocated to the dictionary and summarize the option value.

Create package option families and values for a product context or dictionary

You can create package option families and assign values for a product context or dictionary. Variant data can be reused in multiple product structures and application models (collaborative designs or partition templates).

1. Open the product context or dictionary in the **Variant Options** view.

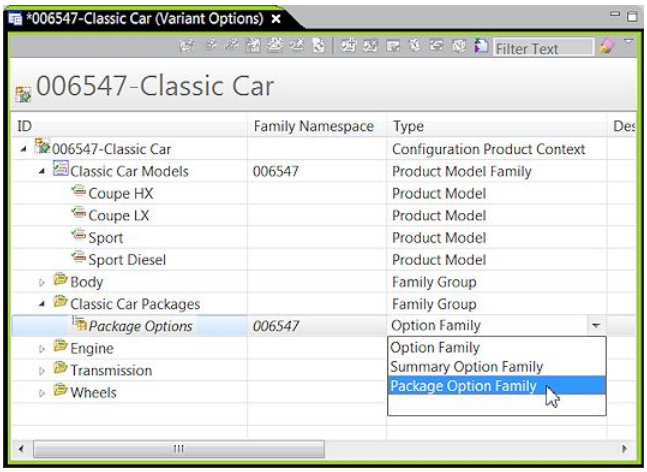
   Teamcenter displays any existing values for the families in the tree table.

2. Select an option family group. The selected family will contain the package options.

3. Click **Add Family** to add a new family to the selected group.

   A new empty row appears as the last child of the selected group, and the group is expanded if necessary. When you enter the ID of the family, the chosen ID must be unique for the current product context or dictionary.

4. Click **Type** for the new row, and select **Package Option Family** from the list of available types.



Note

If your administrator created your customer-specific types of package options, you can specify the custom type.

5. Select the package option for which you want to add an option value, and click **Add Value**.

   Teamcenter adds a new empty row as the child of the selected package and expands it if necessary.

6. Click in the **ID** field of the empty row, and enter the ID of the package option value. The chosen ID must be unique for the current family; however, the same ID may be used in different families for the same product context or dictionary.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved.

7. Click **Save** to save the new value.

   Teamcenter displays the package option family icon and name and description in nonitalic text, indicating the group is saved.

   The package option family is created. The package option values are also saved, and displayed with the icon .

8. To remove a selected package option, right-click and choose **Remove Allocation**.

Note

Deallocate the dependent objects prior to deallocating the package option family.

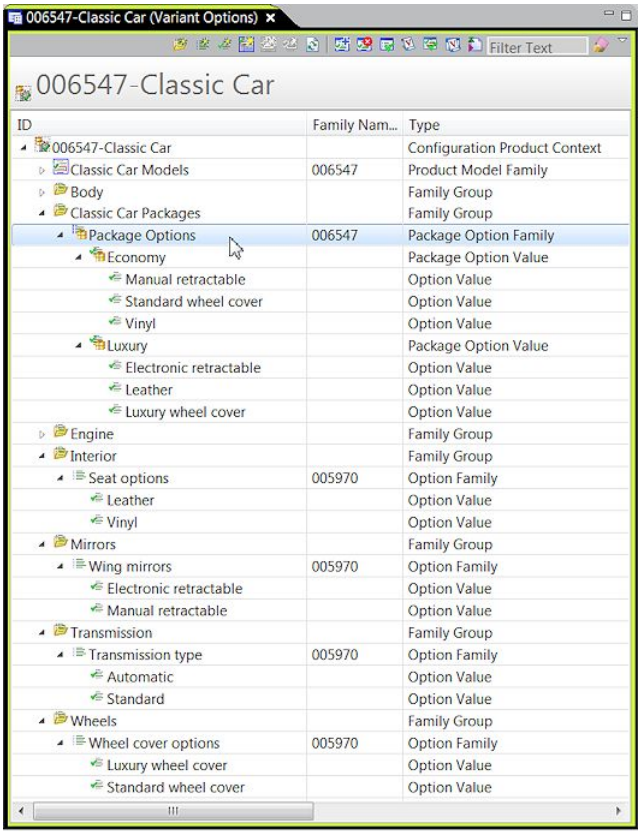
9. Define package option values for the new option by doing one or more of the following:

   - Copy option values and paste them onto the package option.
   - Drag an existing option value onto the package option.
   - Remove a selected option value by right-clicking and choosing **Remove Membership**.

   Package options can only be created in the package option family. Summary options or another package options cannot be added to a package option. You must have write access to the package option to add and remove members.

   You can assign values to a package option from multiple mutually exclusive and multiple multi-select families. Multiple values *cannot* be copied from the single select family. Multiple values *can* be copied from one mutually exclusive family. If you added a new package option in the source dictionary, that newly added option is allocated to the package in the target dictionary only if that option is already allocated to it.

   For example, the **Economy** package contains the **Manual retractable** mirrors, **Standard wheel cover**, and **Vinyl** seats. The **Luxury** package contains the **Electronic retractable** mirrors, **Luxury wheel cover**, and **Leather** seats.



Note

You can copy package option values or a package option on an existing family associated with a product context or dictionary in one **Variant Options** view, and then paste them onto a family associated with another product context or dictionary in another **Variant Options** view. You can also drag and drop package option values or a package option to achieve the same results. However, you cannot save legacy variant option values copied in this way.

10. Click **Save** on the main toolbar to save all changes.

Create a product model family

You can create product model families in a product context.

1. Search for and open an existing product context in the Product Configurator, and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Select the product context, and click **Add model family**.

   Teamcenter adds a new empty row as the child of the selected family and expands the family if necessary.

3. Click in the **ID** field of the empty row, and enter the ID of the product model family. The chosen ID must be unique for the current product context.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

4. (Optional) Click **Type** for the new row, and select **Product Model Family** from the list of available types.

   Note

   If your administrator created your customer-specific types of model family options, you can specify the custom type.

5. (Optional) Click **Description** for the new row, and enter the description of the group.

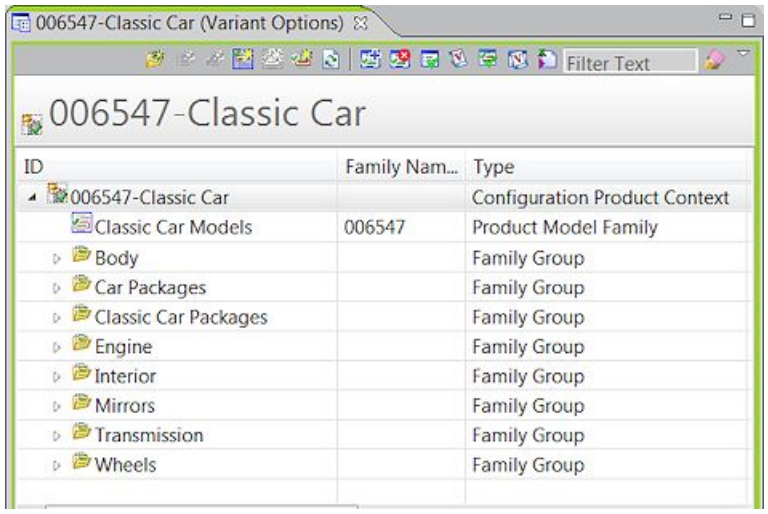
   The **Value Data Type**, **Optional**, **Free-form**, **Multi-select**, **Minimum Value**, and **Maximum Value** values are not modifiable.

6. (Optional) You can set the **Sequence** value for the newly created model family option.

7. Click **Save**.

   Teamcenter displays the product model family icon and name and description in nonitalic text, indicating the group is saved.

   The product model family is created.



Create product models

You can create product models for product model families in the product context.

1. Search for and open an existing product context in Product Configurator, and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Expand the existing product model family.

3. Select the product model family for which you want to add a product model, and click **Add Model** .

   Teamcenter adds a new empty row as the child of the selected product model family and expands the family if necessary.

4. Click in the **ID** field of the empty row, and enter the ID of the model. The chosen ID must be unique for the current product context; however, the same ID may be used in different product contexts.

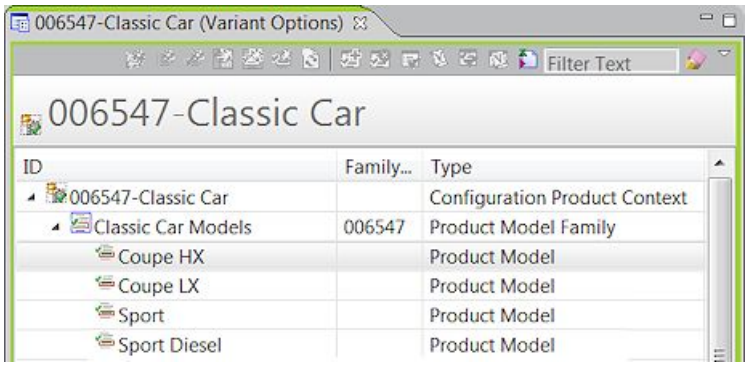
   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

5. Click **Save** .

   Teamcenter displays the product model icon and name and description in nonitalic text, indicating the value is saved.

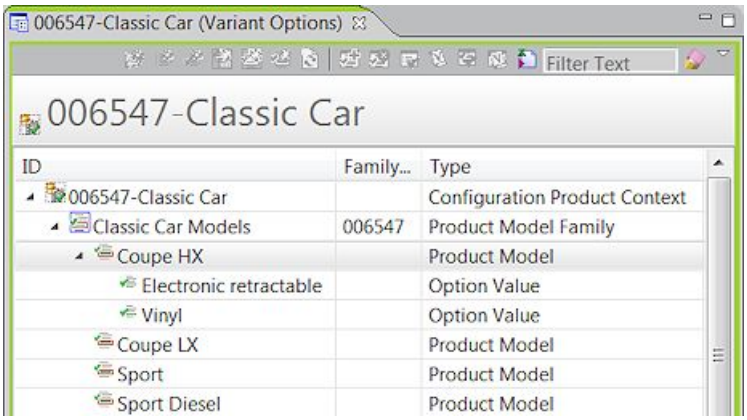
   The product model is created.

   For example, the **Classic Car Models** family contains the **Cobra HX**, **Cobra LX**, **Sports**, and **Sport Diesel** models.



Note

You can also copy option values from existing families that are made available to that product model within the product context, and then paste them onto a product model to create a model package. You can also drag and drop these values to achieve the same result.



Create a summary model family

You can create variant summary model families for product models in a product context.

1. Search for and open an existing product context in the Product Configurator, and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Select the product model family for which you want to add a summary model, and click **Add summary model family** .

   Teamcenter adds a new empty row as the child of the selected family and expands the family if necessary.

3. Click in the **ID** field of the empty row, and enter the ID of the summary model. The chosen ID must be unique for the current product context.

   Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

4. (Optional) Click **Type** for the new row, and select **Summary Model Family** from the list of available types.

   Note

   If your administrator created your customer-specific types of summary model options, you can specify the custom type.

5. (Optional) Click **Description** for the new row, and enter the description of the group.

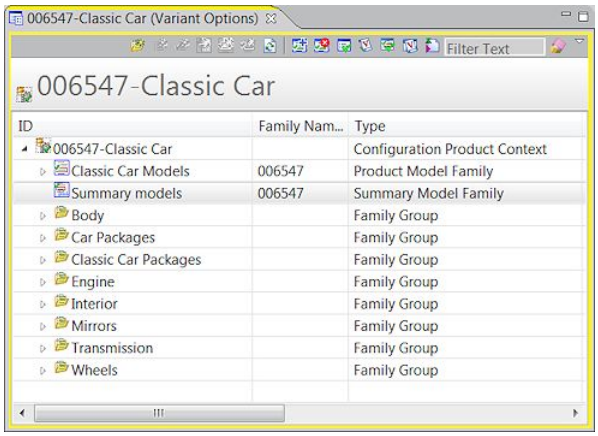
   The **Value Data Type**, **Optional**, **Free-form**, **Multi-select**, **Minimum Value**, and **Maximum Value** values are not modifiable.

6. (Optional) You can set the **Sequence** value for the newly created model summary option.

7. Click **Save** .

   Teamcenter displays the summary model family icon , and name and description in nonitalic text, indicating the group is saved.

   The summary model family is created.



Create summary models

You can create summary models for summary option families in the product context.

1. Search for and open an existing product context in Product Configurator, and ensure the **Variant Options** view is active.

   Teamcenter displays any existing values for the families in the tree table.

2. Select the summary model family for which you want to add a summary model, and click **Add Model** .

   Teamcenter adds a new empty row as the child of the selected summary model family and expands the family if necessary.

3. Click in the **ID** field of the empty row, and enter the ID of the model. The chosen ID must be unique for the current product context; however, the same ID may be used in different product contexts.

Teamcenter adds the ID in italics, indicating the entry is not yet saved. An asterisk (*) next to the name of the view also indicates there is unsaved data.

4. Click **Save**.

   Teamcenter displays the summary model icon and name and description in nonitalic text, indicating the value is saved.

   The summary option family value is created.

5. Summarize existing product models with summary model by doing one or more of the following:

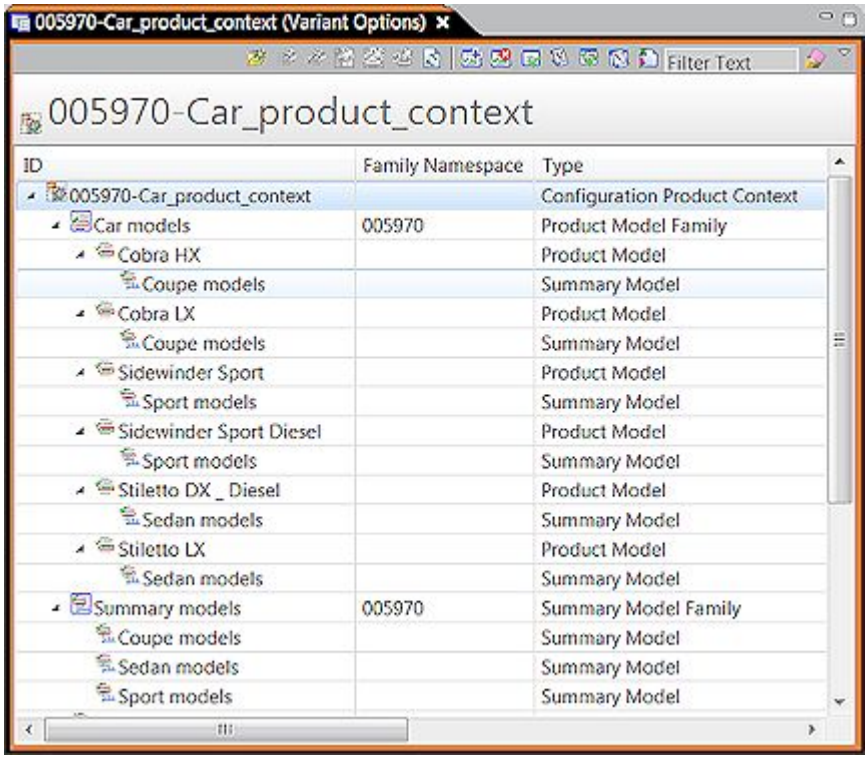
   - Copy models and paste them onto the summary model.
   - Drag a product model onto the summary model.
   - Remove a selected summary model by right-clicking and choosing **Remove Membership**.

   You can summarize multiple summary models with the same product model.

   Note

   You cannot allocate option values or families to the summary model.

   For example, **Cobra HX** and **Cobra LX** are summarized by the **Coupe models** summary model. Teamcenter displays **Coupe models** as the child element of **Cobra HX**, **Cobra LX** models.



Associate a product context with an application model

You can create different application model containers, such as collaborative designs or partition templates, to hold data that models specific aspects of a product, for example, the engineering BOM may be managed in a different application model to the manufacturing BOM. Even if they are modelled in different application models, they are linked in the same product configuration context. Linked application models in the same product context share the following:

- A list of available values for an option family.
- A list of default variant rules.
- A list of variant constraints, that is, compatibility inclusion and exclusion rules.

You can associate a product context item created with Product Configurator to a 4GD collaborative design by attaching it to the collaborative design. The collaborative design is the primary object in this relation, and the configurator item is the secondary object. Alternatively, you can associate item revisions that reference legacy variant data with the collaborative design using the same relationship.

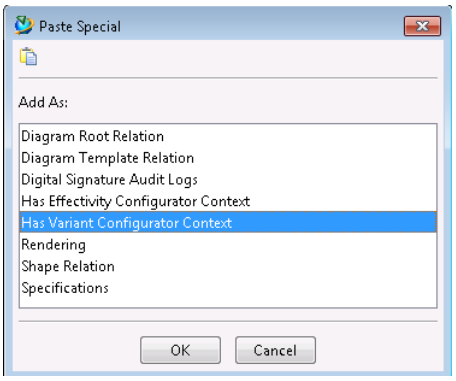
Note

The GRM relation type is specified in the **TC_variant_configurator_relationship** preference. The default value of this preference is **Mdl0HasVarConfiguratorCxt**.

Only one item can be related at a time to the collaborative design using the **Mdl0HasVarConfiguratorCxt** relationship.

To associate a product context item or an item revision that contains reference legacy variant data with an application model:

1. In Product Configurator, create the product context by choosing **File→New→Product context**.
2. In My Teamcenter, copy the product context item or the item revision that contains reference legacy variant data, and then paste it under the application model (collaborative design or partition template) using the **Edit→Paste Special...** command. Then, choose **Has Variant Configurator Context**.



   Teamcenter sets the product name and product namespace from the attributes of the product context item revision.

3. In 4G Designer, specify the variant configuration of the model in one of the following ways:
   - Based on one or more saved variant rules.

This allows you to configure the partition or collaborative design to a standard configuration.

- Based on selected elements.

  This allows you configure the partition or collaborative design to show all product variants that include the selected elements.

- Based on a custom expression with active validation.

  This allows you to configure the partition template or collaborative design to a configuration that has not been included in the set of SVRs for the product. Teamcenter actively validates your selections as you build the expression.

- Based on a custom expression with manual validation.

  This allows you to configure the partition template or collaborative design to a configuration that has not been included in the set of SVRs for the product. Teamcenter does not actively validate your selections as you build the expression and you must manually validate the expression when you have finished. This approach may be particularly suitable if you are creating a new expression or are making extensive changes to an expression, and want to avoid frequent validation messages.

Note

Linked application models are associated with an **Mdl0HasConfiguratorCxt** relationship.

Configure with custom variant configuration (manual validation)

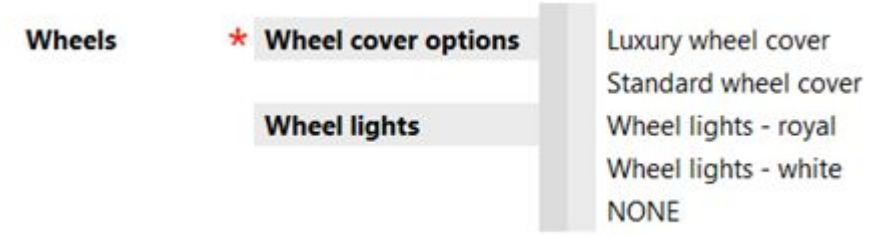
1. In Product Configurator, click **Open Variant Configuration view** on the toolbar to configure variants for the product context.

2. Select manual validation mode by clicking **Automatically validate the current configuration** off. (The validation mode always defaults to manual when you open a new session.)

   Teamcenter displays the **Variant Configuration** view with all the available variant options including package, summary, and summary model options.

   A red asterisk * next to the option indicates the variant option is a required option.

   All variant options that are optional also display value equal to **NONE** in addition to the option values.



3. Click **Applies defaults to the configuration** if you want to set the variant options to their default values.

A is displayed next to the default options.

4. Click the cell to select the variant condition you want to use to define the configuration:

   - Click one time to display a green check mark to include the variant condition when defining the configuration.
   - Click two times to display a red circle backslash to exclude the variant condition when defining the configuration.
   - Click three times to display a blank cell to indicate the variant condition is not used when defining the configuration.

   Multiple variant conditions are connected using a logical **AND** operation in the variant expression that defines the configuration.

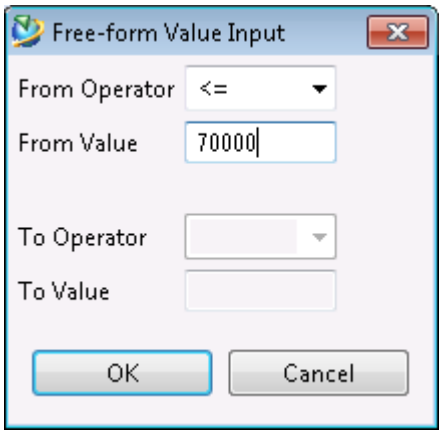
5. If the option is free form text, perform the following steps to set the option:

   a. Click to the right of the cell to display the edit pencil symbol.

   b. Click to enter the values.

   c. In the **Free-form Value Input** dialog box, type the values to set the variant condition.

   d. Click **OK**.

   If the entered value contains any validation or formatting issues, the free form value is displayed in red in the option value text field and flagged with an error message.

6. To remove all variant expressions, click **View menu** and choose **Clear Expression**.

7. To load an existing configuration for viewing or editing, perform the following steps:

a. Click **Replace or overlay existing selections**

Teamcenter displays the **Load selections from saved configurations** dialog box, listing all saved configurations with a check box next to each.

b. Select each product configuration you want to load and click **OK** to replace the existing selections.

Alternatively, you can select **Add values to existing configuration** and Teamcenter loads each in a separate column, preserving the existing selections. Teamcenter applies the configuration to the current content and logically overlays multiple columns, resulting in a 120% BOM configuration or an imprecise configuration.

8. To remove all variant expressions, click **View menu** and choose **Clear Expression**.

9. Click **Validate the Current Configuration**.

Teamcenter applies the relevant constraints and defaults. It highlights any violations with symbols and also displays a message indicating the number of violations of each type. It also lists required families that have no values, mutually exclusive families that have more than one value, and free form values that lie outside their minimum or maximum values.

a. Click **Set Info level violations to be fetched while applying validations** to switch between display or hide all informational messages.

b. Click the **Set Warning level violations to be fetched while applying validations** to switch between display or hide all warning messages.

Error level violations are always displayed.

10. Click to save the custom rule.

If you started from a blank view, Teamcenter displays the **Create Saved Variant Rule** dialog box; otherwise it saves the changes to the existing SVR.

If you exit the **Variant Configurator** view without saving, you are prompted to save the rule.

11. To save a new configuration, in the **Create Saved Variant Rule** dialog box, enter a name and description, and click **OK**.

Teamcenter creates a custom rule with the specified name and variant configuration. It makes this rule available in the **Variant Configuration** list in the Content Explorer header of the application model.

Configure with custom variant configuration (active validation)

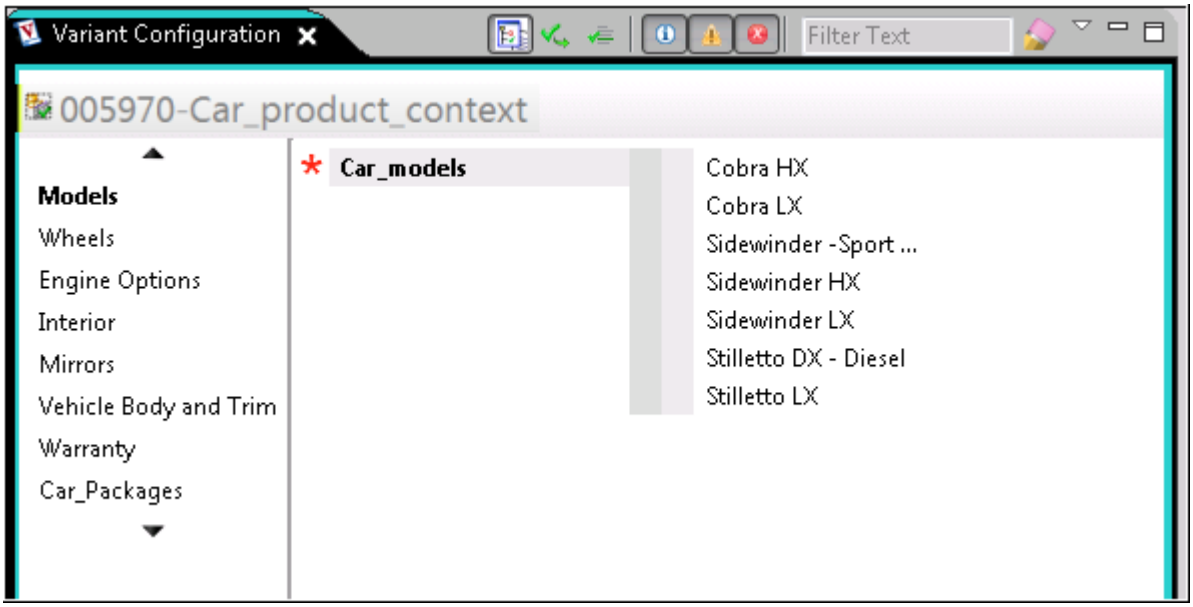
1. In Product Configurator, click **Open Variant Configuration view** on the toolbar to configure variants for the product context.

2. Click to use active validation mode by clicking **Automatically validate the current configuration** on. (The validation mode always defaults to manual when you open a new session.)

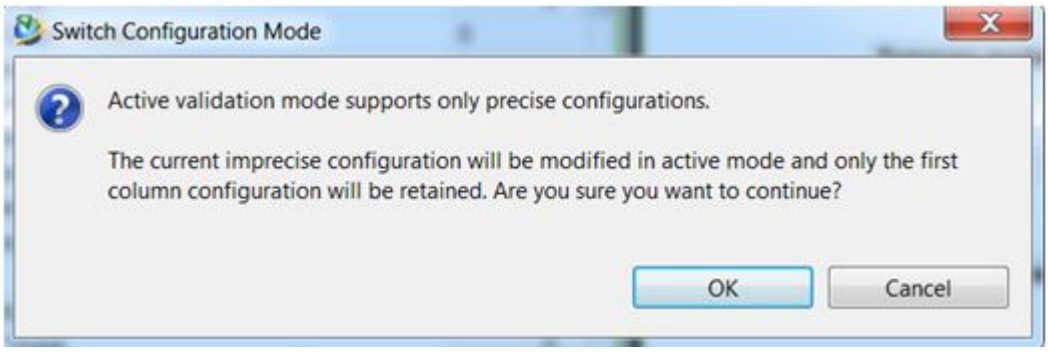
Teamcenter shows a list of groups with the first in bold and arrows above and below the list. Any values that violate constraints are indicated. The families and options of the group are displayed in the right pane.

A red asterisk * next to the option indicates the variant option is a required option.

All variant options that are optional also display value equal to **NONE** in addition to the option values.



If you have an imprecise configuration displayed using manual validation and you switch to active validation, the following dialog appears:



Click **OK** to continue using active validation.

Note

Summary, summary model and package options are not displayed using active validation.

3. Click **Applies defaults to the configuration** if you want to set the variant options to their default values.

   A is displayed next to the default options.

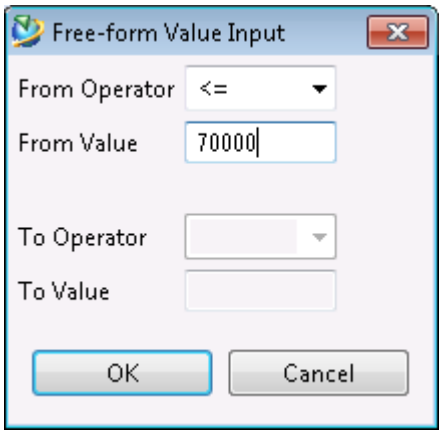
4. Click the cell to select the variant condition you want to use to define the configuration:

- Click one time to display a green check mark to include the variant condition when defining the configuration.
- Click two times to display a red circle backslash to exclude the variant condition when defining the configuration.
- Click three times to display a blank cell to indicate the variant condition is not used when defining the configuration.

Multiple variant conditions are connected using a logical **AND** operation in the variant expression that defines the configuration.

5. If the option is free form text, perform the following steps to set the option:
   a. Click to the right of the cell to display the edit pencil symbol.

   b. Click to enter the values.
   c. In the **Free-form Value Input** dialog box, type the values to set the variant condition.



   d. Click **OK**.

      If the entered value contains any validation or formatting issues, the free form value is displayed in red in the option value text field and flagged with an error message.

6. To remove all variant expressions, click **View menu** and choose **Clear Expression**.
7. Click **Set Info level violations to be fetched while applying validations** to switch between display or hide all informational messages.
8. Click the **Set Warning level violations to be fetched while applying validations** to switch between display or hide all warning messages.

Error level violations are always displayed.

9. Select variant option families, one group at a time, making modifications and resolving violations.

   Click ▲ or ▼ to switch groups to select variant option families, one group at a time.

   Because configuration validation is automatically done when switching groups, you make modification to resolve the violations.

   Teamcenter applies defaults and constraints to each group according to selections you made in previous groups. Each time you select a different group, the following occurs:

   - Validates the current selections and identifies violations with appropriate symbols. If any of the violations are errors, you cannot apply the configuration.
   - Determines valid selections in the next group and applies constraints.

     If there are no valid selections for a family in the next group, that family is not included. If the family is required, it displays an error message.
   - Applies fixed and derived defaults to the families in the next group.

10. Click to save the custom rule.

    If you started from a blank view, Teamcenter displays the **Create Saved Variant Rule** dialog box; otherwise it saves the changes to the existing SVR.

    If you exit the **Variant Configurator** view without saving, you are prompted to save the rule.

11. To save a new configuration, in the **Create Saved Variant Rule** dialog box, enter a name and description, and click **OK**.

    Teamcenter creates a custom rule with the specified name and variant configuration. It makes this rule available in the **Variant Configuration** list in the Content Explorer header of the application model.

Configure from saved variant rule

Saved variant rules (SVRs) are created with the Product Configurator application by saving the configuration.

- In Product Configurator, click **Open Saved Variant Rules view** on the main toolbar.
  - o In the saved variant rules view, right-click the SVR and choose **Open with→Variant Configuration**.

    Teamcenter reconfigures the selected elements with the applied SVR.
- In My Teamcenter, search for the product context containing the SVRs.
  1. Expand the product context to display the saved variant rules.

     000100–Product_context
     ▷ **Variant_rule1**

2. Right-click the SVR and choose **Open with→Variant Configuration**.

   Teamcenter reconfigures the selected elements with the applied SVR.

Related topics

- Saved Variant Rules view

Appendix A: Glossary

4

4GD
See *4th Generation Design*.

4th Generation Design
Teamcenter application that allows designers to work together on the design of a large product such as a ship.

A

Access Manager (AM)
Teamcenter application that enables the system administrator to grant users access to Teamcenter objects.

AM
See *Access Manager (AM)*.

B

BOM
Bill of materials.

- 100% BOM

  The *as sold* product configuration, for example, the configuration of a car to be built and shipped to the dealer.

- 120% BOM

  Partial overlay of selected variant conditions. You cannot build the product from a 120% BOM.

- 150% BOM

  Overlays of all possible variant configurations. You cannot build the product from a 150% BOM.

See also *design bill of materials* and *manufacturing bill of materials*.

BOM view
Teamcenter object used to manage product structure information for an item.

BOM view revision (BVR)
Workspace object that stores the single-level assembly structure of an item revision. Access can be controlled on the structure (BOM view revision) independently of other data. BOM view revisions are meaningful only in the context of the item revisions for which they are created.

BVR
See *BOM view revision (BVR)*.

C

collaborative design
Design model of a product, typically a large complex structure such as a ship or automobile. It comprises many design elements, possibly more than 1 million.

Configurator rule
Condition that specifies the option values or combinations of values not allowed. A configurator rule check is attached to an item revision.

D

design bill of materials
List of components and subassemblies used to define an assembly structure, and the representation of the assembly structure. Compare with *manufacturing bill of materials*.

design element
Occurrence of a design item within a collaborative design.

E

effectivity
Identification of the valid use of an aspect of product data tracked by unit, date, or intent. You can specify a start definition, end definition, or both for a particular effectivity. There are three types of effectivities:

- *Unit effectivity* specifies the range of item units or serial numbers.
- *Date effectivity* specifies the range of dates. This is also known as an incorporation point.
- *Intent effectivity* specifies a purpose, target, or milestone, for example, **Production**, **Prototype**, or **Carryover**.

I

item
Workspace object generally used to represent a product, part, or component. Items can contain other workspace objects including other items and object folders.

item revision
Workspace object generally used to manage revisions to items.

M

manufacturing bill of materials
Defines how the product is manufactured, rather than how it is designed. Compare with *design bill of materials*.

O

option family
Attribute of a product item revision with a set of allowed values (for example, family = engine: 1200, 1600). Option families are used when specifying variant data to configure a variant of the product. Option names are unique within the product but not within the database.

P

partition (4th Generation Design)
Element in a collaborative design that organizes and groups other elements as part of a hierarchical breakdown.

partition breakdown
Predefined hierarchical organization of partition members in the context of a collaborative design. A collaborative design typically has multiple partition breakdowns.

partition item
Item that maintains data about a partition. A partition item can be instanced into multiple breakdowns in multiple collaborative designs. Customers may create types of partition items for particular purposes, for example, Zone, System, Functional, or Manufacturing.

partition scheme
Taxonomy or hierarchical view that categorizes the contents of a collaborative design. A collaborative design may include several schemes. A scheme references a particular breakdown and maintains common business rules and properties for the breakdown.

partition template
Represents a context-independent architecture for a specific class of product. Any partition architecture can serve as a partition template for any other partition architecture of the same type, depending on business rules.

perspective
Container in the rich client user interface for a set of views and editors collected to accomplish specified tasks. See also *view*.

S

Structure Manager
Teamcenter application that enables creation of generic product structures that can be configured to show the product structure that is in production, effective on a certain date, used by a particular customer, and so forth. Structure Manager enables creation and modification of a product structure and its associated occurrence data, display of a product structure in a multilevel indented format, and viewing graphics tightly coupled to the structure for easy identification of a component by location in the structure or in the embedded viewer.

V

variability
Complete set of option values referenced by a set of named variant condition templates.

variant BOM
BOM configured by applying a variant rule.

variant condition

- Rules applicable to one component in a product structure.
- Condition set on an occurrence to specify the option values required to configure that occurrence (for example, Load IF engine = 1200).

variant rule
Collection of option values used in determining the variant of the BOM to be configured (for example, car type = GLS, engine = 1200, gearbox = manual).

view
Software element in a rich client user interface perspective. It provides the ability to navigate hierarchies of information, display information about selected objects, open an editor, or display properties. See also *perspective*.

Index

T

V

Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.