# Virtual Product (cs.vp): 15.4.0

*Release 15.5.0.12*

**CONTACT Software**

**Sep 04, 2018**

# Contents

CHAPTER 1

# Version 10.2

## 1.1 What's new in Version 10.2.0

### 1.1.1 mBOM - Manufacturing BOMs

With the version 10.2 you can build manifacturing BOMs (mBOM) from engineering BOMs.

Manufacturing BOMs can be build either manually or by copying an engineering BOM. The latter is achieved through the new function `Create Manufacturing View`. With it the assembly together with its bill of materials is copied, the new assembly is marked as an mBOM and has a reference to the originating eBOM.

Afterwards you can manually adjust the manufacturing BOM. You can also replace used engineering BOMs with the function `Replace by Manifacturing View`. Like before this will generate an mBOM and replace the corresponding BOM position.

Notice that eBOM and mBOM do not have to be in any way symmetrical. I.e. there has to be no correspondance between subassemblies. This requires special tools for the initial construction of an mBOM and for transporting changes from the eBOM to the mBOM.

The new `mBOM Manager` helps the user executing these tasks. The difference table allows you to identify differences between the eBOM and the mBOM thus securing the completeness of the mBOM.

The new mBOM functionality must be separately licensed.

### 1.1.2 Updates

#### Update v10_1_0_13.SetQualityCharacteristicsInvalid

With this Update the KPI defitions *Weight*, *Production Unit Costs* and *Material Costs* are set in the status *Invalid*.

These KPIs are only examples and should not be used in production.

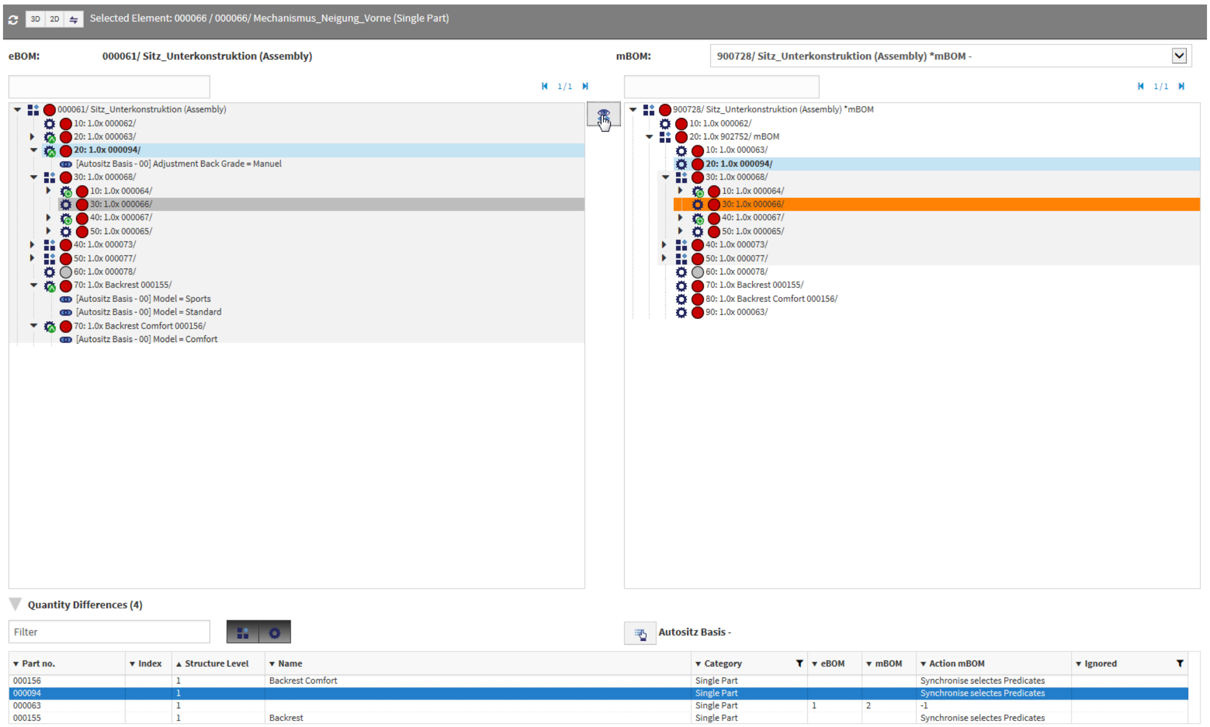
## 1.2 Update

### 1.2.1 Changes to part master data

#### New attributes

- is_mbom

  Checkbox for marking mBOM-Views. In the search mask it is configured with the search condition =0.



Selected Element: 000066 / 000066/ Mechanismus_Neigung_Vorne (Single Part)

eBOM: 000061/ Sitz_Unterkonstruktion (Assembly)

```
000061/ Sitz_Unterkonstruktion (Assembly)
    10: 1.0x 000062/
    20: 1.0x 000063/
    20: 1.0x 000094/
        [Autositz Basis - 00] Adjustment Back Grade = Manuel
    30: 1.0x 000068/
        10: 1.0x 000064/
        30: 1.0x 000066/
        40: 1.0x 000067/
        50: 1.0x 000065/
    40: 1.0x 000073/
    50: 1.0x 000077/
    60: 1.0x 000078/
    70: 1.0x Backrest 000155/
        [Autositz Basis - 00] Model = Sports
        [Autositz Basis - 00] Model = Standard
    70: 1.0x Backrest Comfort 000156/
        [Autositz Basis - 00] Model = Comfort
```

mBOM: 900728/ Sitz_Unterkonstruktion (Assembly) *mBOM -

```
900728/ Sitz_Unterkonstruktion (Assembly) *mBOM
    10: 1.0x 000062/
    20: 1.0x 902752/ mBOM
        10: 1.0x 000063/
        20: 1.0x 000094/
        30: 1.0x 000068/
            10: 1.0x 000064/
            30: 1.0x 000066/
            40: 1.0x 000067/
            50: 1.0x 000065/
        40: 1.0x 000073/
        50: 1.0x 000077/
        60: 1.0x 000078/
        70: 1.0x Backrest 000155/
        80: 1.0x Backrest Comfort 000156/
        90: 1.0x 000063/
```

Quantity Differences (4)

Filter

Autositz Basis -

| Part no. | Index | Structure Level | Name | Category | eBOM | mBOM | Action mBOM | Ignored |
|---|---|---|---|---|---|---|---|---|
| 000156 | | 1 | Backrest Comfort | Single Part | | | Synchronise selectes Predicates | |
| 000094 | | 1 | | Single Part | | | Synchronise selectes Predicates | |
| 000063 | | 1 | | Single Part | 1 | 2 | -1 | |
| 000155 | | 1 | Backrest | Single Part | | | Synchronise selectes Predicates | |

Fig. 1.1: mBOM Manager

- cdb_depends_on

  Reference to the eBOM-View. Only set for mBOM-Views.

- site_object_id

  Reference to the manufacturing site. Only set for mBOM-Views.

- bom_info

  An help attribute for the representation of mBOM-Views.

- cdb_copy_of_item_id

  Will be automatically set when copying a part. It contains the cdb_object_id of the copied part.

- materialnr_erp

  A new attribute. By default it is set equals to the part number when creating a new part or copying one. For mBOM-Views it contains the materialnr_erp of the corresponding eBOM-View.

  According to the circumstances further action may be required. Either you should deactivate this attribute, or you should integrate it in your installation as described in the following.

  If you are not planning to use the new mBOM functionality, you can deactivate the field materialnr_erp from masks and tables. You can then ignore the following points.

  1. Harmonization with an eventually already existing attribute for the ERP number

     If you already have an attribute for the ERP number, you have to decide how the two attributes should be harmonized. There are three different options:

     - Synchronization of the attributes
     - Replacement of the old attribute
     - Exclusive use of the old attribute

2. Customization of the allocation of numbers for materialnr_erp

   You can customize the default allocation of numbers for the field materialnr_erp.

   It is important to observe the following specifications:

   – For mBOM-Views (is_mbom =1) the materialnr_erp should always equal the materialnr_erp of the corresponding eBOM-View. The materialnr_erp specifies the connection between the different eBOM revisions and the corresponding (site specific) mBOMs.

3. Initialization of the materialnr_erp for the existing data.

All new attributes are visible in the tab *Details* of the part masks (`tv_add_mask`, `tv_add_mask_s`).

`materialnr_erp` is also visible on the main tab of the part masks (`tv_mask`, `tv_mask_c`, `tv_mask_s`). Furthermore the default table for parts has also been changed accordingly (`tv_tab`).

The mentioned UI-Elements should be checked and eventually modified after updating. If the new attributes are not used, you can remove them from all the masks and tables.

#### New Relationships

- Manifacturing Views

  1:N Relationship for the mBOM parts of the eBOM part. In case you are not using the new mBOM functionality, you can deactivate the relationship by deleting the role allocation.

#### New Operations

- mBOM Manager

  Starts the `mBOM Manager`

- Create Manufacturing View

  Generates the manufacturing view for the selected eBOM part.

- Carry over Manufacturing View

  Carries a manufacturing view of an old eBOM revision over to the selected eBOM revision.

If you are not using the new mBOM functionality you can deactivate the operations mentioned above. If you call these operation without an mBOM license an error message is shown.

#### Changes to Object Life Cycles

The object life cycle `Part_ERP` has been extended with the status change from 300 (Released for ERP) to 180 (Obsolete). This status change is necessary so that when an mBOM view of a new eBOM revision is released, the mBOM views of the old eBOM revisions can be set to Obsolete. This is because for each site there should be at each time at most one released mBOM pro materialnr_erp.

This change will not be automatically installed when updating. If you plan to use the new mBOM functionality you should ensure that the object life cycles (status and status changes) works with the rules mentioned above.

Eventually you may need to modify the default implementation of the automatic status changes.

### 1.2.2 Changes to bill of materials

#### New Operations

- Replace by Manufacturing View

Operation to replace an engineering assembly in an mBOM with an already existing or with a new mBOM.

If you are not using the new mBOM functionality you can deactivate the operation. If you call this operation without an mBOM license an error message is shown.

### 1.2.3 Changes to properties and catalog properties

#### New attribute value type

- Value type `boolean`

  Can be used for boolean properties and catalog properties.

### 1.2.4 Changes to the Variant Editor

#### New Toolbar Button

The old button *Show Variant in CATIA* is now a dropdown button, which allows to show a variant in different authoring systems. Other packages can define new operations for this button by defining plugins. The definition of a plugin is explained in the administration manual.

## 1.3 Changes in 10.2.1

### 1.3.1 mBOM - Manufacturing BOMs

#### Changes to the revision logic of manufacturing BOMs

With the version 10.2.1 there are less restrictions on mBOM revisions: a revision of an mBOM part can be associated to any revision of the same eBOM part.

This allows to generate an mBOM for a new eBOM revision by indexing an old mBOM revision, instead of copying it. Thus the part numbers of the mBOM parts can be kept the same during the whole revision process.

You can still revision an mBOM of an older eBOM. That is the indexes of an mBOM part do not have to correspond to the indexes of the eBOM part. The only restriction is that every index of an mBOM part should be assigned to an index of the same eBOM part.

Since there is no chronological connection between different indexes of an mBOM part, it is not necessary to enforce that only the newest index has the status `Released`. In particular there are no more automatic status changes for mBOM parts. This also means that mBOM parts cannot reach the status `Revision` anymore.

If you implemented any custom automatic status changes you should check if they also make sense for mBOM parts and eventually modify them.

mBOM parts which are currently in the status `Revision` must be set to another status via script. mBOM parts cannot leave this status manually.

In order to be able to generate new indexes of mBOM parts some changes to the rights configuration are necessary. These are done in the new access control domain `mBOM: all`. You should check if this configuration is compatible with your own customized rights configuration.

The attribute *Copy of* (`cdb_copy_of_item_id`) is now also filled when creating new indexes.

## 1.4 Service Levels

### 1.4.1 Changes in 10.2.1 Service Level 4

The update task `v10_2_0_1.GenerateQualityCharacteristics` has been removed. In `cs.metrics` you have the option, starting from the version 10.1.1 SL 17, to execute the same action manually.

If you want to use the KPI definition of this module and you are updating from a version older than 10.2.0, you have to execute the update tool `cs.metrics.tools.migration` with the following parameters

```
powerscript -m cs.metrics.tools.migration create-missing -im cs.vp.items
powerscript -m cs.metrics.tools.migration create-missing -im cs.vp.products
```

Since this action may take a long time if you have a huge quantity of data, you should set all the KPI definitions that you don't plan to use to the status `invalid`. This way they will be excluded from the data migration.

### 1.4.2 Changes in 10.2.0 Service Level 9

#### Misspelling in the class `part`

In the class `part` the following attributes were misspelled:

- cs_bennenung
- ja_bennenung
- ko_bennenung
- pl_bennenung
- pt_bennenung
- tr_bennenung
- zh_bennenung

These have been renamed as follows:

- cs_benennung
- ja_benennung
- ko_benennung
- pl_benennung
- pt_benennung
- tr_benennung
- zh_benennung

The existing data is migrated with an update task. If you configured any of these attributes in a mask or in a table, you should modify your configuration.

CHAPTER 2

# Version 15.3

## 2.1 What's New in cs.vp 15.3.0?

### 2.1.1 Internationalization

CONTACT Elements 15 and its applications contain many newly translated and updated terms. The following paragraphs list all affected elements in this application, while details concerning the actual conversion (called "internationalization" for simplicity) can be found in each product's release notes.

#### Translated Relationship Access Profiles

The following Relationship Access Profiles have been renamed:

**`Stücklistenpositionen`** Renamed to `BOM Items`

**`Dokument der Variante`** Renamed to `The variant's document`

Please rename any usages in custom relationships accordingly.

#### Translated Catalogs

The following catalogs previously not supporting multiple languages (and the classes using them) are updated as detailed in the product release notes:

- Part Category (`cdb_part_categ`)
- Condition (`cdb_part_usab`)
- Surface (`cdb_part_surfac`)
- Material (`werkstoffe`)
- Purpose (`cdbsml_doc_purpose`)
- Characteristics Group Type (`cdbsml_cg_types`)
- Responsibles (`cdbvp_resp_brows`)
- Quantity Unit (`cdb_units`)

> **Note:** The updated catalog configuration is not automatically applied. If you did not customize these catalogs and their usages, you can manually apply the update patches.
>
> The update scripts `cs.vp.items.updates.v15_3_0.MigrateI18nAttributes` and `cs.vp.classification.updates.v15_3_0.MigrateI18nAttributes` will copy the old key values into the german language attribute. Other languages will have to be migrated by hand.

The catalog class `cdb_part_protyp` has been removed.

### Internationalized Catalogs

These catalog classes were internationalized along with classes using them:

- Part Name (`woerter`)
- Names (`cdbsml_pset_names`)
- Constraint (`cdbvp_constraint`)
- Enum (`cdbvp_enums`)
- Catalog Property (`cdbvp_catalogue_property`)
- Catalog Property Value (`cdbvp_catalogue_enum_def`)
- Catalog Folder (`cdbvp_catalogue_folder`)
- Property (`cdbvp_property`)
- Characteristic (`cdbsml_property`)
- Characteristics Category (`cdbsml_propcat`)
- Property Folder (`cdbvp_property_folder`)
- Product (`cdbvp_product`)
- Predicate (`cdbvp_predicate`)
- Maturity Level (`cdbvp_maturity`)
- BOM Predicate (`cdbvp_bom_predicate`)
- Characteristics Group (`cdbsml_cgroup`)
- Class List of Characteristics (`cdbsml_propset`)
- Variant (`cdbvp_variant`)
- Material (`werkstoffe`)
- Value Folder (`cdbvp_catalogue_value_folder`)
- BOM Generation Method (`bom_method`)

### Translated Catalog Values

The following catalog's values have been newly translated:

- BOM Generation Method (`bom_method`)
- Part Category (`cdb_part_categ`)
- Surface (`cdb_part_surfac`)
- Condition (`cdb_part_usab`)
- Quantity Unit (`cdb_units`)
- Characteristics Group Type (`cdbsml_cg_types`)
- Characteristics Group (`cdbsml_cgroup`)
- Purpose (`cdbsml_doc_purpose`)
- Characteristics Category (`cdbsml_propcat`)
- Catalog Folder (`cdbvp_catalogue_folder`)
- Enum (`cdbvp_enums`)

- Maturity Level (cdbvp_maturity)
- CLC Characteristic Status (sml_mm_state)

Keys remain unchanged.

The new german and english translations will be automatically updated.

**Note:** This does only apply to catalog values not changed by any custom module.

#### Translated Status Names

Status names have been internationalized for the classes:

- Product (cdbvp_product)
- Part (part)
- Catalog Property (cdbvp_catalogue_property)
- Corresponding protocol classes

### 2.1.2 New Module cs.vp.classification

The module sml.py has been relocated from platform to cs.vp.classification.

The attribute ClassificationCategory.name is no longer a LocalizedField but a MultiLangAttributeDescriptor.

### 2.1.3 Miscellaneous

#### Removed Elements

The following elements have been removed:

- The table verw_tab. Custom usages can be changed to use stl_usage_tab or a new custom table instead.
- Dialogs sml_xase_mask*.
- The browsers SMLBrowser_ID and SMLBrowser_SG have been removed.

#### BOM Comparison

The User Exit cdbstlcompare has been relocated to cs.vp.bom. The file CADDOK_BASE\lib\cdbstlcompare.ue is not in use anymore and can be deleted.

## 2.2 Service Levels

### 2.2.1 Changes in 15.3.0 Service Level 9

#### Changes in Object Framework API: LocalizedField in cs.vp.classification

Since version 15.3.0 the fields cs.vp.classification.PropertySet.name and cs.vp.classification.Property.name are multilingual fields. These can be used as described in the *Powerscript API* manual.

The old LocalizedFields have been replaced by cs.vp.classification.PropertySet.Name and cs.vp.classification.Property.Name.

## 2.3 What's New in cs.vp 15.3.1?

### 2.3.1 Variants Management

The minor release 15.3.1 comes with an extension of the variant management. This includes following features:

- Persistent instantiation of Maximum BOMs with the instantiation wizard (see user manual).
- Variant Matrix as alternative UI for managing variants (see user manual).
- Property hierarchy inside a product (see user manual).
- Possibility to filter a Maximum BOM using python expressions (see administration manual).

All these changes are backward compatible. Only the last one requires data migration.

### 2.3.2 Reengineering MBOM Manager

In this release there has been a reengineering of the algorithm for computing the differences between EBOMs and MBOMs. The new algorithm provides a better performance and more reliable results.

**Important:** The new algorithm bases on the new attribute mbom_mapping_tag of the class bom_item. Every EBOM-position and its correspondent MBOM-position must have the same value in this attribute.

If the MBOM is constructed using the standard operations the value of the attribute mbom_mapping_tag is automatically set. If the MBOM is imported from a third system or constructed in any other way, the interface is responsible for setting the attribute correctly.

#### Update tool

If you already have MBOMs in your system, you need to update your data to set the attribute mbom_mapping_tag.

We provide an update tool which sets the attribute according to an heuristic. Since the heuristic is not always correct, the update tool **does not run automatically**.

To run the update tool and have more information use the following powerscript command:

```
powerscript -m cs.vp.bom.tools.make_mbom_mapping_tag -h
```

### 2.3.3 Powerscript API for BOMs

With this release there is a powerscript API for efficiently querying BOM structures.

The Powerscript module cs.vp.bom.bomqueries provide methods for querying an entire product structure.

The module cs.vp.bom.diffutil.differences provide methods for comparing EBOMs and MBOMs.

The module cs.vp.bom.mapping provide methods to map EBOM-positions with MBOM-positions.

The module cs.vp.bom.usages provide methods for querying the usages of one or more parts inside one or more product structures.

For more information see the Powerscript manual of *cs.vp*.

## 2.3.4 Update Tasks

### cs.vp.variants.updates.v15_3_1_0.Init

This task will perform the necessary data migration related to the new feature "Possibility to filter a Maximum BOM using python expressions".

### cs.vp.variants.updates.v15_3_1_0.DropObsoleteVariantAttribute

Starting from this release the attribute `cdbvp_variant_info_txt` of the table `cdbvp_variant` is a multilanguage attribute and must not exist in the database. This task will remove the column from the database table.

## 2.3.5 Changes in Service Level 15

### Bugfix E038693: Instantiated assemblies have BOM predicates

With this bugfix BOM predicates and BOM terms are no longer copied when instantiating an a maxBOM for a variant. This leads to some performance optimization of the MBOM Manager.

If you are using the MBOM Manager, it is important that you clean up your data to remove any redundant bom predicate, which was created before the bugfix. Use the following tool:

```
powerscript -m cs.vp.bom.tools.delete_redundant_predicates --help
```

## 2.3.6 Changes in Service Level 16

### Bugfix E047328: Wrong type substitution when evaluating string predicates

Prior to this bugfix, string predicates would always try to convert property values to a numeric type, regardless of the property type.

Example:

> If you had an alphanumeric property named `COLOR` with the string value `"300"` and the BOM predicate `COLOR == "300"`, then this predicate would never match, as the engine would have substituted the variable COLOR with the value `300` (as a number).

Now this behavior is fixed and the engine will substitute the variable `COLOR` with the correct value `"300"` (as a unicode string).

# Version 15.4

## 3.1 What's New in cs.vp 15.4.0?

### 3.1.1 Service Level 0

#### WebUI

With this release there is a new WebUI which supports searching and viewing of parts.

#### cdb_cad_assembly_component

The class `cdb_cad_assembly_component` together with its masks, tables and operations has been removed.

It has been replaced by the class `threed_cad_assembly_component` in the package `cs.threed`.

### 3.1.2 Service Level 8

#### Object Rule `3D Models`

Starting with this service level, there is now an object rule named `3D Models`. It is used to determine the objects which hold 3D data that can be used for the preview. It will replace the rule `3D Viewer: Viewable Models` from the package `cs.threed`.

**Important:** When you have customized the old rule, make sure to apply the same modifications to the new rule. This won't happen automatically.

CHAPTER 4

# Version 15.5

## 4.1 What's New in cs.vp 15.5.0?

### 4.1.1 Changes in Service Level 10

#### Bugfix E038693: Instantiated assemblies have BOM predicates

With this bugfix BOM predicates and BOM terms are no longer copied when instantiating an a maxBOM for a variant. This leads to some performance optimization of the MBOM Manager.

If you are using the MBOM Manager, it is important that you clean up your data to remove any redundant bom predicate, which was created before the bugfix. Use the following tool:

```
powerscript -m cs.vp.bom.tools.delete_redundant_predicates --help
```

# List of Figures

# List of Tables

# Index