
Virtual Product

Release 15.5.0.12

CONTACT Software

Sep. 04, 2018

1	Einleitung	1
2	Stücklistenverwaltung	2
2.1	Allgemeines	2
2.2	Stücklistenausleitung aus CAD-Systemen	2
3	Sachmerkmalleisten und Sachgruppen	8
3.1	Einleitung	8
3.2	Sachgruppen / Sachgruppenhierarchie	8
3.3	Sachmerkmale	8
3.4	Generieren einer Sachmerkmalleiste	9
3.5	Löschen einer Sachmerkmalleiste	10
3.6	Integration der GUI-Elemente	10
3.7	Properties	10
3.8	Merkmalübergreifende Suche	12
4	Varianten-Editor	13
4.1	Anzeigen in CAD-Systemen	13
4.2	Embedded Modus	14
4.3	Spalten Fixierung	14
5	Variantenmatrix	15
5.1	Anpassen des Toolbars	15
6	Filtern von Maximalstücklisten	16
7	mBOM Manager	17
7.1	Registration von Operationen	17
7.2	Anzeige in der Differenzentabelle steuern	18
7.3	Abgleich von Entwicklung- und Fertigungsstücklisten	18
	Stichwortverzeichnis	21

Einleitung

CONTACT Virtual Product bietet für die Unterstützung des Work in Progress ein umfassendes Modell des aufwachsenden virtuellen Produkts. Seine aufeinander abgestimmten Bestandteile sind u.a. Produkte und Produktportfolio, Variabilitätsmodell und Produktvarianten, Komponenten und ihre logistische Ausprägung als Artikel bzw. Material, Produktstrukturen und Stücklisten sowie das Konfigurationsmanagement für die Kontrolle der Änderungen am Produkt.

Stücklistenverwaltung

2.1 Allgemeines

Ein Artikel (ein Objekt der Klasse `part`) ist gleichzeitig der Stücklistenkopf. Ihm sind Stücklistenpositionen (Objekte der Klasse `bom_item`) zugeordnet.

Beim Ermitteln der Stückliste verwendet das System in der Regel eine der Beziehungen aus dem *Beziehungsverzeichnis*, deren Name mit `CDB::Relationship::STL` beginnt. Bei diesen Beziehungen gibt es die Besonderheit, dass aus Performancegründen kein SQL-Select-Statement abgesetzt wird, wenn das Attribut `baugruppenart` der Klasse `part` leer ist. Stattdessen nimmt das System in einem solchen Fall an, dass es keine Stückliste zum Artikel gibt.

Das System selbst belegt das Attribut beim Anlegen der ersten Stücklistenposition zu einem Artikel automatisch mit dem Wert `Baugruppe`.

Bemerkung: Insbesondere bei einer Datenübernahme aus einem Fremdsystem ist darauf zu achten, dass das Datenbankattribut `teile_stamm.baugruppe` für Artikel mit Stückliste den Wert `Baugruppe` erhält - ansonsten ist die Stückliste ggf. nicht sichtbar.

2.2 Stücklistenausleitung aus CAD-Systemen

Inhalt dieses Kapitels ist die Konfiguration der serverseitigen Stücklistenausleitung, wie sie in Verbindung mit dem Workspace Manager zur Anwendung kommt. Zur genaueren Einordnung der Funktionalität wird auf den Abschnitt “Stücklistenausleitung aus CAD-Systemen” im Anwendungshandbuch *CONTACT Virtual Product* verwiesen.

2.2.1 Überblick

Es stehen folgende Konfigurationsmöglichkeiten zur Verfügung:

- Sie können unter verschiedenen Methoden zur Stücklistenausleitung wählen.
- Sie können anpassen, welche Daten der generierten Stückliste in der tabellarischen Vorschau angezeigt werden.
- Sie können Attribute auswählen, anhand derer Stücklistenpositionen zusammengefasst werden (Default: `teilennummer` und `t_index`).
- Sie können Attribute auswählen, deren Wert beim wiederholten Ausleiten automatisch aktualisiert wird (Default: `menge`).
- Sie können die verwendete Schrittweite bei der Vergabe neuer Positionsnummern anpassen (CAD-Konfigurationsschalter `STL Schrittweite`).
- Sie können unerwünschte Warnungen deaktivieren.

Falls eine Methode zum Import von CAD-Stücklisteninformationen (aus einem Stücklisteneditor) gewählt wurde, kann außerdem eine Abbildung von Attributen der CAD-Stücklisteninformationen auf Attribute konfiguriert werden.

Darüber hinaus lassen sich mit Hilfe von Objektregeln Stücklistenpositionen für bestimmte Artikel unterdrücken.

Mit Ausnahme der Schrittweite werden alle Einstellungen gruppiert verwaltet. Jede Konfigurationsgruppe ist einem CAD-System zugeordnet. Außerdem kann der Geltungsbereich einer Konfigurationsgruppe über Rollen auf verschiedene Benutzer eingegrenzt werden. Die Default-Einstellungen gelten für alle CAD-Systeme, für die keine speziellere Konfiguration verfügbar ist, sowie für alle Benutzer, die die allgemeine Rolle "Engineering" innehaben. Sollten Mehrdeutigkeiten auftreten, wird anhand der Priorität einer Konfigurationsgruppe entschieden. Mehrdeutigkeiten können auftreten, wenn es mehrere Konfigurationsgruppen für dasselbe CAD-System gibt und auch durch die Benutzerrollen keine eindeutige Auswahl getroffen werden kann. Ein *höherer* Wert für "Priorität" bedeutet, dass eine Konfigurationsgruppe Vorrang hat. Die Priorität wird automatisch vergeben, kann aber manuell angepasst werden.

Der Zugang zur Konfiguration der Stücklistenausleitung erfolgt im Navigationsbaum über *Administration/Konfiguration* → *Administration* → *Integrationen* → *CAD-Konfiguration* → *Stücklistenausleitung*.

Um unerwünschte Warnungen zu deaktivieren, fügen Sie Einträge unter *Administration/Konfiguration* → *Administration* → *Integrationen* → *CAD-Konfiguration* → *Unterdrückte Warnungen (STL)* hinzu.

Nach Änderungen der Konfiguration ist kein Neustart erforderlich.

2.2.2 Anpassen der Vorschau

In diesem Abschnitt soll beispielhaft die Anzeige der Vorschau konfiguriert werden, und zwar so, dass die neue Konfiguration für alle CAD-Systeme gültig ist.

Dazu kopieren Sie zunächst die Default-Konfiguration und vergeben einen eindeutigen Namen für die kopierte Konfiguration (z.B. *Angepasste Anzeige*). Beachten Sie, dass der Wert für die Priorität der neuen Konfiguration automatisch vergeben wird.

Durch Löschen der Kopie können Sie später problemlos wieder zur Default-Konfiguration zurückkehren.

Die neue Konfiguration kann nun durch Doppelklick geändert werden. Wählen Sie den Reiter *Tabellenspalten*, um die Anzeige der Vorschau anzupassen. Hier können Sie Spalten verändern, neu anlegen oder löschen. Wählen Sie die Neuanlage.

Es öffnet sich die Maske zur Neuanlage von Tabellenspalten. Sie haben hier die Möglichkeit, das anzuzeigende *Attribut* per Katalog auszuwählen. Es stehen dabei sowohl Attribute von Stücklistenpositionen zur Verfügung (auf der obersten Ebene), als auch Attribute des zugehörigen Artikels (erreichbar über den Eintrag "Reference_1: Item").

Wählen Sie beispielsweise das Attribut *mengeinheit* über die Beziehung *Item* aus.

Für spezielle Anwendungsfälle kann es notwendig sein, die Attributbezeichnung manuell einzugeben. Das ist z.B. der Fall, wenn sogenannte "Localized Fields" benutzt werden sollen. "Localized Fields" werden im Abschnitt "Multilingual attributes" des PowerScript-Handbuchs beschrieben. Ein Beispiel für den Zugriff auf ein "Localized Field" ist das Attribut *Item.designation*.

Eine andere Möglichkeit, mit mehrsprachigen Attributen umzugehen, ist die Verwendung der Umgebungsvariable "CADDOK_ISOLANG". Wenn etwa ein Attribut *name_\$* (CADDOK_ISOLANG) konfiguriert wird, wird sein Wert je nach eingestellter Sprache aus dem Datenbank-Attribut *name_de* oder *name_en* (*name_it*, *name_fr*, etc.) gelesen.

Eine Tabellenspalte benötigt außerdem eine *Beschriftung*, die im Tabellenkopf angezeigt wird. Hier kommen allgemeine Systembeschriftungen zum Einsatz. Wählen Sie beispielsweise das Label *quant_unit*.

Die *Position* einer Tabellenspalte bestimmt ihre horizontale Position im Verhältnis zu anderen Spalten. Der Wert wird automatisch hochgezählt, kann aber angepasst werden, um eine neue Spalte zwischen vorhandenen Spalten einzufügen. Sie können hier z.B. 35 eingeben, um die neue Spalte direkt hinter die existierende "Menge"-Spalte zu positionieren.

2.2.3 Aktivieren der automatischen Ausleitung von Unterbaugruppen

Standardmäßig wird die Methode *Ausleiten aus der Komponentenstruktur* benutzt. Dabei wird nur für den Artikel des Dokuments, auf dem die Operation aufgerufen wurde, eine Stückliste erzeugt.

Um automatisch Stücklisten für die Artikel von referenzierten Dokumenten zu erzeugen, muss im Datenblatt einer Konfigurationsgruppe die *Ausleitungsmethode (Rekursive) Ausleiten aus der Komponentenstruktur* gewählt werden. Die Auswahl erfolgt über einen Katalog.

2.2.4 Konfiguration des Attribute-Mappings für ein spezielles CAD-System

In diesem Abschnitt wird die Anlage einer Konfiguration speziell für ein ausgewähltes CAD-System beschrieben. Dabei wird die Annahme getroffen, dass die Daten aus dem System bereits Stücklisteninformationen enthalten und diese auch vom Workspace Manager in Form von Dateiinformatoren (.appinfo-Dateien) übertragen wurden. Es wird die dazu passende Ausleitungsmethode konfiguriert und demonstriert, wie Werte aus den Stücklisteninformationen auf Attribute abgebildet werden können.

Kopieren Sie zunächst die `Default` -Konfiguration als Ausgangsbasis für die zu erstellende CAD-spezifische Konfiguration. Geben Sie der neuen Konfiguration einen eindeutigen Namen und wählen Sie das *CAD-System* per Katalog aus. Die *Priorität* wird nach dem Kopiervorgang automatisch so festgelegt, dass die neue Konfiguration bestehende Konfigurationen für dasselbe CAD-System aufhebt.

Um die Stücklisteninformationen aus den Dateien zu übernehmen, stellen Sie die *Ausleitungsmethode* auf *Auslesen aus der Appinfo* ein.

Schließen Sie den Kopiervorgang ab und öffnen Sie anschließend die neue Konfiguration per Doppelklick. Wechseln Sie in den Reiter *Appinfo-Abbildungen*. Jeder Eintrag in dieser Tabelle bildet genau *einen* Wert auf ein Attribut ab. Erzeugen Sie per Kontextmenü einen neuen Eintrag. In der Maske wählen Sie zunächst das Zielattribut (*CDB-Attribut*) aus. Beachten Sie, dass jedes Attribut nur einmal verwendet werden kann. Der Name des Wertes der Stücklisteninformation (*Appinfo-Name*) muss manuell eingegeben werden. Konsultieren Sie dazu die Stücklistenanzeige im Workspace Manager und berücksichtigen Sie die kopplungsspezifischen Export-Settings.

Normalerweise wird der Wert unter den *Attributen* einer Stücklistenposition gesucht. Mit aktivierter Checkbox *Property* wird hingegen unter den *Properties* gesucht.

Bemerkung: Es ist darauf zu achten, dass der Wertebereich von Appinfo-Werten und Systemattributen kompatibel ist. Wenn eine Konvertierung der Werte notwendig ist, kann dies im Rahmen der Neuimplementierung einer Ausleitungsmethode realisiert werden.

Bemerkung: Es darf kein Attribute-Mapping mit dem Zielattribut `position` erzeugt werden. Das Attribut `position` wird immer fest aus dem Appinfo-Attribut `posno` befüllt. Bei fehlender `posno` werden automatisch neue Positionsnummern generiert.

2.2.5 Auswertung von Dokument-Referenzen in Stücklisteninformationen

Stücklistenpositionen, die vom Workspace Manager in Form von Dateiinformatoren übermittelt werden, können Referenzen auf die Dokumente verbauter Artikel enthalten. Bei der Ausleitungsmethode *Auslesen aus der Appinfo* werden solche Referenzen aus Performanzgründen nicht berücksichtigt, selbst dann nicht, wenn Informationen über den verbauten Artikel fehlen (leere `teilennummer`).

Durch Auswahl der Ausleitungsmethode *Auslesen aus der Appinfo, mit Dokument-Referenzen* wird diese Einschränkung deaktiviert. Dokument-Referenzen werden dabei auf die gleiche Weise aufgelöst, wie dies bei der Methode *Ausleitung aus der Komponentenstruktur* der Fall ist.

2.2.6 Anpassung an CAD-Systeme mit Halbzeugen (Verwendung von Schlüsselattributen)

In der Standard-Konfiguration werden Stücklistenpositionen zusammengefasst, wenn sie sich auf denselben verbauten Artikel beziehen, also wenn `teilennummer` und `t_index` identisch sind. Dabei wird das Attribut `menge` auf die Summe der zusammengefassten Einträge gesetzt.

In manchen Situationen ist es wünschenswert, dass importierte Stücklistenpositionen nicht zusammengefasst werden, obwohl sie auf den gleichen Artikel verweisen. Das kann etwa beim Auftreten von Halbzeugen der Fall sein. Daher lässt sich die Stücklistenausleitung so konfigurieren, dass Positionen, die sich in bestimmten Attributen unterscheiden, als eigenständige Stücklistenpositionen erhalten bleiben.

Voraussetzung dafür ist, dass die fraglichen Attribute bereits auf Systemattribute abgebildet wurden (siehe *Konfiguration des Attribute-Mappings für ein spezielles CAD-System* (Seite 4)).

Ist dies der Fall, so können die entsprechenden Attribute im Reiter *Schlüsselattribute* hinzugefügt werden.

Bemerkung: Eine weitere Konfigurationsmöglichkeit besteht in der Auswahl von Attributen, die bei der wiederholten Ausleitung automatisch aktualisiert werden. In der Regel wird dies für Schlüsselattribute erwünscht sein. Fügen Sie also neue Schlüsselattribute auch unter dem Reiter *Aktualisierungsattribute* hinzu.

Bemerkung: Das Attribut `position` darf weder als Schlüsselattribut noch als Aktualisierungsattribut verwendet werden.

2.2.7 Ausschluss von Komponenten durch Objektregeln

Es existieren drei Objektregeln, mit deren Hilfe Artikel identifiziert werden, für die keine Stücklistenposition erzeugt wird. Standardmäßig enthalten diese Regeln keine Prädikate und treffen daher auf keine Artikel zu. Es handelt sich im einzelnen um folgende Regeln:

- **BOM: Ignore Component Item:** Diese Regel wird auf Artikel ("Items") angewendet. Für jede Stücklistenposition wird überprüft, ob die Regel auf den verbauten Artikel zutrifft. Wenn sie zutrifft, wird kein Stücklisteneintrag erzeugt. Es ist darauf zu achten, dass bei der Definition von Prädikaten der Typ `cs.vp.items.Item` ausgewählt wird.

Diese Regel wird von allen mitgelieferten Ausleitungsmethoden berücksichtigt.

- **BOM: Ignore Component Document:** Diese Regel wird auf Dokumente angewendet. Beim Auftreten von CAD-Referenzen wird geprüft, ob die Regel auf das referenzierte Dokument zutrifft. Bei der Definition von Prädikaten muss der Typ `cs.documents.Document` ausgewählt werden.

Diese Regel wird von den Ausleitungsmethoden (*Rekursives*) *Ausleiten aus der Komponentenstruktur* und *Auslesen aus der Appinfo, mit Dokument-Referenzen* ausgewertet.

- **BOM: Recursive Ignore Component:** Diese Regel wird auf Dokumente angewendet. Sie wird ausschließlich von der Ausleitungsmethode (*Rekursives*) *Ausleiten aus der Komponentenstruktur* verwendet. Für referenzierte Dokumente, auf die die Regel zutrifft, wird keine eigene Stückliste erzeugt. Bei der Definition von Prädikaten muss der Typ `cs.documents.Document` ausgewählt werden.

Zur Definition von Objektregeln siehe *CONTACT Elements Plattform: Administration und Konfiguration*.

2.2.8 Ignorieren von Varianten durch Objektregeln

Bei einigen CAD-System kann es erwünscht sein, dass bestimmte Varianten bei der Stücklistenausleitung ignoriert werden. Insbesondere soll dann auch keine Warnung oder Fehlermeldung angezeigt werden, wenn einer solchen Variante kein Artikel zugewiesen wurde.

Zu diesem Zweck gibt es zwei Objektregeln:

- **BOM: Ignore Variant:** CAD-Varianten, auf die diese Regel zutrifft, werden bei der Ermittlung der verbauten Komponente immer ignoriert. Stattdessen wird die Hauptbaugruppe (Artikel des CAD-Dokuments) benutzt.
- **BOM: Ignore Singleton Variant:** CAD-Varianten, auf die diese Regel zutrifft, werden ignoriert, falls es nur eine einzige Variante für das betreffende Modell gibt. In der Standardkonfiguration ist diese Regel so konfiguriert, dass solche Varianten für das CAD-Systems SolidWorks herausgefiltert werden.

Bei der Definition von Prädikaten muss der `Typ cs.vp.cad.CADVariant` ausgewählt werden. Zur Definition der Terme können neben den Attributen der Variante selbst auch die Attribute des CAD-Dokuments herangezogen werden.

Diese Regeln werden von den Ausleitungsmethoden (*Rekursives*) *Ausleiten aus der Komponentenstruktur* und *Auslesen aus der Appinfo, mit Dokument-Referenzen* ausgewertet.

2.2.9 Vergabe von Positionsnummern

Neue Stücklistenpositionen erhalten in der Regel Positionsnummern, die anhand der höchsten bisher enthaltenen Positionsnummer und des CAD-Konfigurationsschalters `STL_Schrittweite` bestimmt werden.

Falls aber beim erneuten Abgleich der Stückliste ein Bauteil lediglich durch einen höheren Indexstand ausgetauscht wurde, kommt ein Sonderfall zum Tragen: Bei eindeutiger Erkennung der Artikelnummer wird die Positionsnummer wiederverwendet. Aus technischen Gründen wird diese Situation in der Vorschau in Form von zwei Positionen dargestellt: Ein gelöschter Eintrag mit dem alten Indexstand und ein neuer Eintrag mit dem neuen Indexstand des Bauteils.

Bemerkung: Existieren mehrere ausgetauschte Positionen für die Artikelnummer, ist die Reihenfolge der resultierenden Positionen allerdings nicht determiniert.

2.2.10 Verwendung einer projektspezifischen Methode

Durch Auswahl der Ausleitungsmethode *Dokumentspezifische Methode* für ein CAD-System kann die eigentliche Auswahl der Methode an das Dokument delegiert werden.

Damit in der Dokumentenmaske die Auswahl der Methode getroffen werden kann, muss das entsprechende Feld zunächst durch Umkonfiguration sichtbar gemacht werden. Suchen Sie dazu unter *Administration/Konfiguration* → *Konfiguration* → *Oberfläche* → *Maskenkonfiguration* → *Erweitert* → *Maskenelemente* nach Elementen mit dem Wert `bom_method` für das Feld `Attribut`. Bei den zwei so gefundenen Maskenelementen deaktivieren Sie die Checkbox *verborgen*. Es ist ein Client-Neustart erforderlich, damit diese Änderung wirksam wird.

Für die betroffenen Dokumente (mit der passenden Erzeuger-Applikation) kann nun im Datenblatt eine Ausleitungsmethode gewählt werden (Feld *STL-Ausleitung*).

Wenn ein Dokument keine eigene Ausleitungsmethode hat, obwohl die dokumentspezifische Methode für das CAD-System gewählt wurde, wird anhand der Prioritäten und Benutzerrollen nach einer anderen Konfiguration gesucht. Gegebenenfalls kommt die Default-Konfiguration zur Anwendung.

2.2.11 Konfiguration einer eigenen Methode zur Stücklistenausleitung

Die Implementierung von Ausleitungsmethoden im Rahmen des Customizing wird im englischen Programmierhandbuch des Pakets “Workspaces” beschrieben. Wenn eine solche Implementierung in Form einer Klasse vorliegt, ist es sehr einfach, die neue Methode in das System einzubinden.

Dazu verwenden Sie im Datenblatt einer Stücklistenkonfiguration den Neuanlage-Knopf des Feldes *Ausleitungsmethode*. In der Maske vergeben Sie einen eindeutigen, beschreibenden Namen für die neue Methode und geben den vollständigen Namen der neuen Python-Klasse an (*Voll qual. Python-Klassenname*). Wichtig ist, dass die Python-Klasse dynamisch vom Objects Framework gefunden werden kann. Die Ausleitungsmethode kann nun in Konfigurationen verwendet werden.

Sollte die Klasse nicht gefunden werden, werden Fehlermeldungen geloggt und stattdessen eine Default-Methode benutzt. Werden Code-Änderungen an der Implementierung unternommen, muss den Client neu gestartet werden, damit die Änderungen wirksam werden.

2.2.12 Berücksichtigung von Varianten

Bei der Verwendung der Ausleitungsmethoden (*Rekursives*) *Ausleiten aus der Komponentenstruktur* und *Auslesen aus der Appinfo*, mit *Dokument-Referenzen* können Varianten bei der Auflösung von Referenzen berücksichtigt werden.

Dazu ist es notwendig, dass die Option zur Übertragen von Varianteninformationen im *Workspace Manager* explizit aktiviert wird. Siehe Administrationshandbuch *Workspaces*, Kapitel "Konfiguration".

Es ist außerdem erforderlich, dass Benutzer eine Zuordnung von Artikeln zu Varianten vornehmen. Dies ist im Anwendungshandbuch *CONTACT Virtual Product* im Detail beschrieben (im Abschnitt *Verwendung von Varianten*).

Weitere Konfigurationsmöglichkeiten bestehen für diese Funktionalität zur Zeit nicht.

2.2.13 Manuell angelegte Stücklistenpositionen ersetzen

Normalerweise werden beim Abgleich einer Stückliste nur Positionen mit identischer CAD-Quelle berücksichtigt. Wenn manuelle Positionen automatisch ersetzt werden sollen, kann dies im Datenblatt einer Stücklistenkonfiguration konfiguriert werden. Dazu dient das Feld *Zu ersetzende CAD-Quelle*.

Wird hier *manuell* eingetragen, so werden manuelle Positionen, bei denen die Schlüsselattribute übereinstimmen, durch automatisch generierte Positionen ersetzt. Dabei werden die Werte für *CAD-Quelle* und *Menge* sowie für weitere konfigurierte Aktualisierungsattribute angepasst.

Wenn das Feld *Zu ersetzende CAD-Quelle* leer ist, werden Positionen mit abweichender CAD-Quelle weiterhin nicht berücksichtigt.

Sachmerkmalleisten und Sachgruppen

Dieses Kapitel beschreibt die Konfiguration und Administration besonderer Leistungsmerkmale des Systems in Bezug auf Sachmerkmalleisten und Sachgruppen.

3.1 Einleitung

Das SML-Modul enthält die Funktionen zur Administration und Nutzung von Sachgruppen, Sachmerkmalen und Sachmerkmalleisten in Anlehnung an die DIN 4000, 4001 und 4002. Im Folgenden werden ausschließlich Aspekte der Administration dokumentiert. Die Anlage, Pflege und Nutzung der Sachmerkmalleisten wird im Anwendungshandbuch *CONTACT Virtual Product* beschrieben (u.a. Klassifizierung von Artikeln, Vergabe von Sachmerkmalswerten, Recherche anhand von Sachmerkmalen etc.).

Der Konfigurationszugang ist über den Menübaum unter *Produkte* → *SML* zu erreichen.

3.2 Sachgruppen / Sachgruppenhierarchie

Aus den Einträgen der Sachgruppenhierarchie werden die Daten generiert, die im Menübaum oder im SML-Strukturauswahlkatalog die Navigation zu den Sachmerkmalleisten ermöglichen. Mit Hilfe der Properties `smfm` kann dabei der Bezeichner für Sachgruppen im Anwendungsbereich festgelegt werden.

Wichtig: Nach der Modifikation des Property `smfm` muss die Sachgruppenhierarchie erneut generiert werden, damit die Änderungen im gesamten Anwendungsbereich wirksam werden.

3.3 Sachmerkmale

Die allgemeine Konfiguration von Sachmerkmalen wird im Anwendungshandbuch *CONTACT Virtual Product* beschrieben.

Die nachträgliche Änderung einer Merkmalsdefinition und insbesondere des Datentyps und der Anzahl der Vor- und Nachkommastellen ist mit Bedacht vorzunehmen. *Modifizierung des Datentyps von Merkmalen* (Seite 9) gibt die Möglichkeiten unter dem Datenbanksystem ORACLE an.

Tab. 3.1: Modifizierung des Datentyps von Merkmalen

von	nach	ohne Daten	mit Daten
Int	Int	x	x (keine Auswirkung auf SQL-Typ, nur DIN-Eintrag)
Int	Float	x	x
Int	Text	x	–
Float	Float	x	x (keine Auswirkung auf SQL-Typ, nur DIN-Eintrag)
Float	Int	x	–
Float	Text	x	–
Text	Text	x	x (nur Erweiterung, keine Reduzierung)
Text	Int	x	–

3.4 Generieren einer Sachmerkmaliste

Nach dem Anlegen oder dem Ändern einer Sachmerkmaliste und ihrer Sachmerkmale muss diese erstmals bzw. erneut “generiert” werden, damit diese Änderungen für die Anwender sichtbar werden.

Das Generieren ist manuell durch Auswahl des Kontextmenüpunktes *Sachmerkmaliste generieren* zur entsprechenden Sachmerkmaliste anzustoßen (siehe Anwendungshandbuch *CONTACT Virtual Product*). Die Methoden zur Generierung finden sich in den entsprechenden `cdb.objects`-Klassen.

Beim Generieren der Sachmerkmaliste wird eine `DataDictionary`-Klasse (siehe *CONTACT Elements Plattform: Administration und Konfiguration*, Abschnitt *Facetten*) anhand der Eigenschaften der Sachmerkmaliste und deren Merkmale erzeugt bzw. angepasst. Für diese Klasse wird dann die Standardmethodik zur Erzeugung von Schema, Oberflächenelementen und Operationen angewendet. Diese Methodik wurde um folgende Aspekte erweitert:

Bezeichnung der Klasse Die generierte Bezeichnung der Sachgruppenklasse kann über die Fehlermeldung mit dem Meldungs-Label `cdbsml_label_desc` festgelegt werden. Innerhalb der Bezeichnung können Attribute der Relation `cdbsml_propset` verwendet werden. Die Konfiguration von `pset_id + " (" + name_de + ") "` könnte beispielsweise zur Laufzeit zur Ausgabe von *10-0280-0020-0020 (Drahtstifte)* führen.

Bezeichnung der Attribute Die Attributbezeichnung im Data Dictionary wird aus der Merkmalskennung abgeleitet. Daher können SQL-Schlüsselwörter, wie z.B. `create`, `delete`, `update`, `table`, `lower`, ... nicht verwendet werden. Der Bezeichner wird dabei ggf. angepasst, z.B. wenn er mit Zahlen oder Unterstrichen beginnt oder Großbuchstaben enthält.

Generierung der Standardmaske Enthält eine Sachmerkmaliste Sachmerkmale, werden zunächst die Felder einer Template-Maske kopiert, bevor die Attribute der Sachmerkmaliste in die Maske aufgenommen werden. Diese Template-Maske muss für die Gruppe `public` definiert sein und enthält im Standard die Felder, in denen die SML-Bilder dargestellt werden. Für die Operationen *Information* und *Ändern* wird die `sml_base_mask` als Template verwendet. Existiert die `sml_base_mask_n` wird anhand dieser Template-Maske eine Neuanlagemaske generiert. Existiert sie nicht, wird die Ändern-Maske auch zur Neuanlage verwendet. Sollen spezielle Recherchemasken generiert werden, ist `sml_base_mask_s` als Template zu verwenden.

Generierung der Maskenfelder Die Felder `teilenummer` und `t_index` werden nicht in die Maske aufgenommen.

Die Feldbeschriftung erfolge anhand des Properties `pmld`.

Generierung der Darstellungstabelle Die Felder `teilenummer` und `t_index` werden nicht in die Tabelle aufgenommen.

Die Feldbeschriftung erfolge anhand des Properties `ptld`.

Automatisch generierte Definitionen für Masken, Maskenelemente, Tabelle und Tabellenelemente können - wie andere Definitionen auch - anschließend etwa bzgl. des Layouts angepasst werden. Zu beachten ist, dass diese Anpassungen bei einer erneuten Generierung mit Property `srmt=true` ggf. überschrieben werden bzw.

bei `srmt=false` vorgenommene SML-Modifikationen (neue, geänderte oder gelöschte Merkmale) anderweitig nachgepflegt werden müssen.

3.5 Löschen einer Sachmerkmalreihe

Das Löschen einer Sachmerkmalreihe ist durch Auswahl des Menüpunktes *Löschen* zur entsprechenden Sachmerkmalsreihe anzustoßen. Die Sicherheitsabfrage ist mit *ja* zu bestätigen. Es wird der zugehörige Leisteneintrag gelöscht. Die referenzierten Merkmale, Ansichten und Hierarchieeinträge werden gelöst, das heißt es werden nur die Beziehungen gelöscht, nicht aber die Objekte bzw. Elemente selbst.

Wurde bereits die zugehörige Klasse generiert, sind klassifizierte Artikel weiter verwendbar. Ist dies nicht gewünscht, muss diese Klasse gelöscht und das Feld `sachgruppe` in den entsprechenden Artikeln geleert werden.

3.6 Integration der GUI-Elemente

Ist ein Artikel klassifiziert, wird das zugehörige SML-Maskenregister immer dann angezeigt, wenn zu der entsprechenden Operation ein Maskenverbund konfiguriert ist, in welchem ein Register mit dem Registernamen `$Facet:sachgruppe` konfiguriert ist. Existiert ein solches Register in dem Maskenverbund zur Neuanlage eines Artikels, wird dieses Register durch die Maskenkonfiguration der SML-Klasse für die Neuanlageoperation der Facette ersetzt.

In Darstellungstabellen wird die Darstellungstabelle der SML-Klasse dort eingeblendet, wo ein Feld mit dem Attribut `$Facet:sachgruppe` und dem Spaltentyp `Placeholder` konfiguriert ist.

3.7 Properties

Folgende systemweit gültige Voreinstellungen (Properties), die personen- und/oder gruppenabhängig konfiguriert werden können, stehen zur Verfügung:

[string] smfm = <\$NAME> Definiert, wie eine Sachgruppe im Anwendungsbereich benannt wird. Zulässige Werte sind:

\$ID Klassifikationskennzeichen.

\$LONG Der Langbezeichner der Sachgruppe

\$NAME Name der Sachmerkmalreihe.

Kombinationen wie `$NAME ($ID)` sind möglich.

[string] smls = <> Legt die Sortierreihenfolge der Sachgruppen in der Sachgruppenhierarchie fest. Ist der Wert `klassifikation`, wird nach der Klassifikation (`$ID`) sortiert. Ansonsten erfolgt die Sortierung anhand des Sachgruppennamens.

[boolean] srmt = <true> Zulässige Werte sind:

true Beim Generieren von Sachmerkmalreihen werden automatisch Masken und Tabellen erzeugt, die die Sachmerkmale enthalten. Manuell vorgenommene Änderungen an den Masken und Tabellen werden überschrieben!

false Beim Generieren von Sachmerkmalreihen werden keine Masken und Tabellen erzeugt.

Unabhängig vom Wert gilt beim erstmaligen Generieren `true`.

[string] pml = <> Legt den Formatstring für die Generierung der Maskenbeschriftung der einzelnen Sachmerkmale fest. Bei leerer, nicht vorhandener oder fehlerhafter Formatstringangabe wird die Merkmalbeschriftung nach dem folgenden Standardmuster generiert.

Merkmalbezeichnung (Merkmalkennung) [Merkmaleinheit]

Bemerkung: Resultiert durch die Standardgenerierung eine Beschriftung, deren Zeichenlänge den zulässigen Wert des zugehörigen Datenbankattributes übersteigt, so wird der Anteil der Merkmalbezeichnung entsprechend gekürzt. Resultiert durch die Generierung anhand einer gültigen expliziten Formatstringangabe eine Merkmalbeschriftung, deren Zeichenlänge den zulässigen Wert des zugehörigen Datenbankattributes übersteigt, wird das Ergebnis entsprechend abgeschnitten.

Die Angabe des Formatstrings kann variable, konstante und/oder optionale Bestandteile enthalten. Für die Zusammensetzung der einzelnen Bestandteile zu einer Gesamtbeschriftung müssen alle einzelnen Angaben mit einem Pluszeichen verbunden werden.

Formatangaben

- Als variable Anteile sind Namen für die zugehörigen Attribute des jeweiligen Merkmals und seiner Verwendung (Referenzierung) zulässig. Darüber hinaus sind ergänzend bekannte Umgebungsvariablen wie CADDOK_ISOLANG zulässig. Diese werden in der Form \$(CADDOK_ISOLANG) angegeben.
- Konstante Bestandteile werden in doppelten Anführungsstrichen angegeben.
- Bestandteile können - in eckigen Klammern eingeschlossen - als optional gekennzeichnet werden. Der als Optional gekennzeichnete Bestandteil wird nur für die gesamte Beschriftung mit aufgenommen, wenn die Auswertung der einzelnen enthaltenen variablen Anteile jeweils einen nicht leeren Wert ergibt.

Beispiel: pm1d-Property

Angabe des Formatstrings zur Generierung einer Sachmerkmalmaskenbeschriftung

```
name_$(CADDOK_ISOLANG)+[" (" +prop_mk+" )"]+[" \["+prop_unit+"\""]
```

Bemerkung: Bei der Angabe einer öffnenden bzw. schließenden eckigen Klammer oder eines Pluszeichens in einem konstanten Beschriftungsbestandteil muss diese mit einem umgekehrten Schrägstrich (Backslash) maskiert werden.

[string] ptld='<>' Legt den Formatstring für die Generierung der Tabellenbeschriftung der einzelnen Sachmerkmale fest. Bei leerer, nicht vorhandener oder fehlerhafter Formatstringangabe wird die Merkmalbeschriftung in Form einer dreizeiligen Spaltenüberschrift nach dem folgenden Standardmuster generiert.

Feature name
Feature characteristic
Feature unit

Bemerkung: Resultiert durch die Generierung eine Merkmalbeschriftung, deren Zeichenlänge den zulässigen Wert des zugehörigen Datenbankattributes übersteigt, wird das Ergebnis entsprechend abgeschnitten.

Die Angabe des Formatstrings kann variable, konstante und/oder optionale Bestandteile enthalten. Für die Zusammensetzung der einzelnen Bestandteile zu einer Gesamtbeschriftung müssen alle einzelnen Angaben mit einem Pluszeichen verbunden werden.

Formatangaben

- Als variable Anteile sind Namen für die zugehörigen Attribute des jeweiligen Merkmals und seiner Verwendung (Referenzierung) zulässig. Darüber hinaus sind ergänzend bekannte Umgebungsvariablen wie CADDOK_ISOLANG zulässig. Diese werden in der Form \$(CADDOK_ISOLANG) angegeben.

ben wie CADDOK_ISOLANG zulässig. Diese werden in der Form \$ (CADDOK_ISOLANG) angegeben.

- Konstante Bestandteile werden in doppelten Anführungsstrichen angegeben. Darüber hinaus kann für die mehrzeilige Anordnung der einzelnen Bestandteile das Zeichen '\n' als Separator benutzt werden.
- Bestandteile können - in eckigen Klammern eingeschlossen - als optional gekennzeichnet werden. Der als Optional gekennzeichnete Bestandteil wird nur für die gesamte Beschriftung mit aufgenommen, wenn die Auswertung der einzelnen enthaltenen variablen Anteile jeweils einen nicht leeren Wert ergibt.

Beispiel: pt.1d -Property

Angabe des Formatstrings zur Generierung einer Sachmerkmalstabellenbeschriftung

```
name_${CADDOK_ISOLANG} + "\n" + prop_mk + "\n" + prop_unit
```

Bemerkung: Bei der Angabe einer öffnenden bzw. schließenden eckigen Klammer oder eines Pluszeichens in einem konstanten Beschriftungsbestandteil muss diese mit einem umgekehrten Schrägstrich (Backslash) maskiert werden.

3.8 Merkmalübergreifende Suche

Das System bietet die Möglichkeit, eine merkmalsübergreifende Suche durchzuführen. Um die merkmalsübergreifende Suche zu aktivieren, müssen die Merkmal-Identifikatoren und die zugeordneten Werte in die Suchmaske integriert werden. Bei der Konfiguration der entsprechenden Maskenfelder muss dabei im Feld *Attribut* der Wert `cdb::argument::sml.` vorangestellt werden. Auf diese Weise konfigurierte Felder werden nur bei Recherchen angezeigt und in allen anderen Kontexten automatisch ausgeblendet. Findet eine Suche ohnehin im Kontext einer Sachgruppe statt, werden solche Felder ebenfalls automatisch ausgeblendet.

Für die IDs wird im Feld *Attribut* der Wert `cdb::argument::sml.propid_nr` konfiguriert, wobei `nr` Werte von 01 bis 99 annehmen kann, also `cdb::argument::sml.propid_01`, `cdb::argument::sml.propid_02`, ..., `cdb::argument::sml.propid_99`. Bei den zugeordneten Werten wird im Feld *Attribut* der Wert `cdb::argument::sml.propval_<nr>` angegeben.

Werden bei einer Suche Werte in die Felder der merkmalsübergreifenden Suche eingegeben, wird die Recherche auf allen Sachgruppen ausgeführt, welche alle angegebenen Merkmale enthalten. Die ausgewählten Merkmale werden automatisch am Anfang der zugeordneten Darstellungstabelle angezeigt. Eine Sortierung der Trefferlisten nach mehreren Kriterien ist nicht möglich.

Eine Sachgruppe sollte ein Merkmal nie mehrfach enthalten. Ist dies dennoch der Fall, muss das eingegebene Suchkriterium für alle Werte des Merkmals zutreffen. In der Darstellungstabelle wird nur ein Wert angezeigt.

Varianten-Editor

4.1 Anzeigen in CAD-Systemen

Der Varianten-Editor bietet die Möglichkeit eine Variante in einem CAD-System anzuzeigen. Sie können diese Funktionalität erweitern, um weitere Autorensysteme zu unterstützen. Dafür müssen Sie ein Plugin schreiben und dieses im Varianten-Editor registrieren. Die Definition des Plugins kann dabei in jedem beliebigen Modul erfolgen.

Zur Registrierung von Plugins wird die Methode `register_plugin` aus dem Modul `cs.vp.variants.apps.generatorui` benötigt. Sie wird folgendermaßen importiert:

```
from cs.vp.variants.apps.generatorui import register_plugin
```

Die Methode `register_plugin` erwartet als Parameter ein Python-Dictionary, in dem folgende Schlüssel/Wert-Paare vorhanden sein müssen:

- `icon`: Name des Icons für den Button
- `label`: Name des Labels für den Tooltip
- `json`: eine JSON-Methode
- `json_name`: der öffentliche Name der JSON-Methode

Diese Parameter **müssen** mit Werten belegt sein, um ein neues CAD-System erfolgreich dem Dropdown-Menü zuordnen zu können.

Außerdem gibt es noch die folgenden optionalen Parameter:

- `position`: Gibt die Position in dem Dropdown-Menü an.
- `open_new_window`: Bestimmt, ob die Operation ein neues Browser-Fenster öffnen soll. Der Defaultwert ist `True`.

Die CAD-Systeme werden in absteigender Reihenfolge anhand der Positionsnummern im Dropdown-Menü der Toolbar angezeigt.

Wird der Parameter `position` nicht explizit mit einem Wert belegt, so erhält er den Default-Wert 0. D.h. dass alle registrierten Plugins ohne explizite Wertbelegung des Attributes `position` an das Ende des Dropdown-Menüs angehängt werden. Alle Plugins, die hingegen mit einer expliziten Wertbelegung des Attributes `position > 0` registriert wurden, nehmen im Umkehrschluss eine höhere Position im Dropdown-Menü ein.

Ein beispielhafter Aufruf der `register_plugin` Methode sähe folgendermaßen aus:

```
register_plugin({"icon": "cdbvp_show_in_catia",
               "label": "cdbvp_show_in_catia",
               "json_name": "show_in_catia",
               "json": show_in_catia,
               "position": 10})
```

Es wird die ID eines konfigurierten Icons sowie die ID eines konfigurierten Labels übergeben. Des Weiteren wird eine JSON-Methode sowie der Name der JSON-Methode als Label übergeben. Zuletzt wird der optional Parameter `position` gesetzt, um die Reihenfolge des Dropdown-Menüs zu beeinflussen.

Die JSON-Methode soll ein Python-Dictionary zurückgeben, das sowohl den Schlüssel `url` als auch einen Wert, den der Client verarbeiten kann, enthält. Dies kann zum Beispiel die URL einer Operation sein. Ein Beispiel einer solchen JSON-Methode finden Sie in `cs.vp.variants.apps.generatorui.show_in_catia`.

4.2 Embedded Modus

Um den Varianten-Editor als Panel im Client oder in einer externen Anwendung einbetten zu können, bietet der Varianten-Editor einen *Embedded Modus*. Wenn dieser Modus eingeschaltet ist, wird die Titelleiste nicht angezeigt und der Bereich *Produktdetails* wird per default zusammengeklappt.

Um der *Embedded Modus* einzuschalten, rufen Sie den Varianteneditor mit dem URL-Parameter `embedded=1` auf.

4.3 Spalten Fixierung

Um im Varianten-Editor bei einer größeren Anzahl von Spalten die beschreibenden ersten vier Spalten stets sehen zu können, bietet der Varianten-Editor die Möglichkeit, die ersten vier Spalten zu fixieren. Sie können mittels der Property `vecf` einstellen, ob diese Funktion standardmäßig aktiv ist. Dazu muss diese auf den Wert `true` gesetzt werden. Sie finden die Property unter *Administration/Konfiguration -> Konfiguration -> Properties*.

Der Anwender hat zusätzlich im Varianten-Editor die Möglichkeit mittels eines Buttons die Fixierung ein- bzw. abzuschalten.

Variantenmatrix

5.1 Anpassen des Toolbars

In der Variantenmatrix wird für die ausgewählte Variante oben rechts ein Toolbar mit diversen Operationen angezeigt.

Sie können im Operationskontext `VariantMatrixVariant` festlegen welche Operationen in diesem Toolbar angezeigt werden. Die Operationen müssen im Menu sichtbar sein und auf ein `SingleObject` vom Typ `cdbvp_variant` anwendbar sein.

Filtern von Maximalstücklisten

Für die Entwicklung von Schnittstellen besteht die Möglichkeit, Maximalstücklisten anhand von Python-Ausdrücken zu filtern.

Somit entstehen weniger Objekte als durch die Anlage herkömmlicher Auswahlprädikate.

Um mit Python-Ausdrücken zu filtern, instantieren Sie pro zu filternder Stücklistenposition ein Objekt der Klasse `cdbvp_bom_string_predicate`. Dieses ist über die REST-API möglich. Der Ausdruck wird im Attribut `expression` gespeichert.

Die Ausdrücke können Python-Operatoren und Konstanten enthalten. Variablen werden wie folgt ausgewertet:

1. Anhand des Variablennamens wird das passende Merkmal über das Attribut *ERP Code* gesucht
2. Die Variable wird durch das Attribut `ERP Code` des Merkmalwertes in der Variante ersetzt.

mBOM Manager

7.1 Registration von Operationen

Der mBOM Manager bietet bereits eine Reihe von Standardoperationen an. Diese Standardoperationen können mit einem Registrationsmechanismus um beliebige weitere Operationen erweitert werden. Dafür müssen Sie eine Operation in Form eines Python-Dictionaries schreiben und diese im mBOM Manager registrieren. Die Definition der Operation kann dabei in jedem beliebigen Modul erfolgen.

Zur Registrierung von Operationen wird die Methode `register_operation` aus dem Modul `cs.vp.bom.diffutil.pages` benötigt. Sie wird folgendermaßen importiert:

```
from cs.vp.bom.diffutil.pages import register_operation
```

Die Methode `register_operation` erwartet als Parameter ein Python-Dictionary, in dem folgende Schlüssel/Wert-Paare vorhanden sein müssen:

- `name`: Name der konfigurierten Operation
- `is_ebom_op`: Flag für die Zuordnung der Operation zu der eBOM oder mBOM
- `is_item_op`: Flag für die Zuordnung der Operation zu einem Artikel oder einer Stücklistenposition

Diese Parameter **müssen** mit Werten belegt sein, um eine neue Operation erfolgreich dem jeweiligen Dropdown-Menü des mBOM Managers zuordnen zu können.

Außerdem gibt es noch drei optionale Parameter:

- `needsReload`: Flag für das Neuladen nach Ausführung der Operation, Defaultwert `True`
- `unselect`: Flag für das Deselektieren des Baumknotens nach Ausführung der Operation, Defaultwert `False`
- `multi_select_op`: Legt fest ob die Operation auf mehreren Artikeln aufgerufen werden kann, Defaultwert `False`

Die Operationen werden in der Reihenfolge, in der sie im mBOM Manager registriert wurden, angezeigt.

Ein beispielhafter Aufruf der `register_operation` Funktion sähe folgendermaßen aus:

```
register_operation({"name": "CDB_Workflow",  
                  "is_ebom_op": False,  
                  "is_item_op": True,  
                  "needsReload": True,  
                  "unselect": False,  
                  "multi_select_op": False})
```

Es wird der Name einer konfigurierten Operation übergeben. Des Weiteren wird ein Flag übergeben, welches festlegt, dass die Operation nur für die mBOM angeboten werden soll und ein weiteres Flag legt fest, dass diese Operation nur auf Artikeln ausführbar sein soll. Außerdem werden die beiden optionalen Parameter gesetzt, um festzulegen, dass nach der Ausführung der Operation die Seite erneut geladen werden soll und dass der Artikel

nach Ausführung der Operation nicht deselektiert werden soll. Des Weiteren wird mit dem letzten Parameter festgelegt, dass die zu registrierende Operation nur auf einem Artikel aufrufbar ist.

7.2 Anzeige in der Differenzentabelle steuern

Anhand der Objektregel `mBOM Manager: Ignore differences` können Sie konfigurieren, welche Differenzen in der Mengendifferenzentabelle **nicht** angezeigt werden sollen. Die Objektregel prüft gegen Artikel. Standardmäßig filtert diese Objektregel alle mBOM-Artikel, die keinen Bezug zu einer eBOM haben.

7.3 Abgleich von Entwicklung- und Fertigungsstücklisten

Der Abgleich von Entwicklung- und Fertigungsstücklisten im mBOM-Manager setzt die korrekte Pflege des Attributs `mbom_mapping_tag` der Klasse `bom_item` voraus.

Das Attribut enthält ein Mapping zwischen den Stücklistenpositionen eines eBOM-Artikels und denen eines mBOM-Artikels. Entsprechende Stücklistenpositionen müssen den gleichen Wert für das Attribut `mbom_mapping_tag` haben.

Wenn die Fertigungsstückliste mit den Standardoperationen erzeugt wird, wird dieses Attribut automatisch korrekt gesetzt. Wird hingegen die Fertigungsstückliste über eine Schnittstelle oder ein Drittsystem erzeugt, müssen Sie sich vergewissern, dass das Attribut korrekt gesetzt wird.

3.1	Modifizierung des Datentyps von Merkmalen	9
-----	---	---

C

CADDOK_ISOLANG, 11, 12
cdbvp_bom_string_predicate, 16
cdbvp_variant, 15
cs.vp.bom.diffutil.pages, 17
cs.vp.variants.apps.generatorui, 13
cs.vp.variants.apps.generatorui.show_in_catia, 14

E

ERP Code, 16
expression, 16

I

icon, 13
is_ebom_op, 17
is_item_op, 17

J

json, 13
json_name, 13

L

label, 13

M

mBOM Manager: Ignore differences, 18
multi_select_op, 17

N

name, 17
needsReload, 17

O

open_new_window, 13

P

position, 13, 14
Property
 pmld, 10
 ptld, 10
 smfm, 8, 10
 smls, 10
 srmt, 9, 10

R

register_operation, 17
register_plugin, 13

S

SingleObject, 15

U

Umgebungsvariable

 CADDOK_ISOLANG, 11, 12
 cdbvp_bom_string_predicate, 16
 cdbvp_variant, 15
 cs.vp.bom.diffutil.pages, 17
 cs.vp.variants.apps.generatorui, 13
 cs.vp.variants.apps.generatorui.show_in_catia, 14
 ERP Code, 16
 expression, 16
 icon, 13
 is_ebom_op, 17
 is_item_op, 17
 json, 13
 json_name, 13
 label, 13
 mBOM Manager: Ignore differences, 18
 multi_select_op, 17
 name, 17
 needsReload, 17
 open_new_window, 13
 position, 13, 14
 register_operation, 17
 register_plugin, 13
 SingleObject, 15
 unselect, 17
 url, 14
 VariantMatrixVariant, 15
unselect, 17
url, 14

V

VariantMatrixVariant, 15