



Extensible Catalog Format (XCF) Spezifikation

Bernd Heinemann, Thomas Gerth
Copyright © 1999-2017 EasternGraphics GmbH

Version 2.10

31. August 2017

Rechtliche Hinweise

Copyright © 1999-2017 EasternGraphics GmbH. All rights reserved. Dieses Werk ist urheberrechtlich geschützt. Alle Rechte sind der EasternGraphics GmbH vorbehalten. Die Übersetzung, die Vervielfältigung oder die Verbreitung, im Ganzen oder in Teilen ist nur nach vorheriger schriftlicher Zustimmung der EasternGraphics GmbH gestattet. Die EasternGraphics GmbH übernimmt keine Gewähr für die Vollständigkeit, für die Fehlerfreiheit, für die Aktualität, für die Kontinuität und für die Eignung dieses Werkes zu dem von dem Verwender vorausgesetzten Zweck. Die Haftung der EasternGraphics GmbH ist, außer bei Vorsatz und grober Fahrlässigkeit sowie bei Personenschäden, ausgeschlossen. Alle in diesem Werk enthaltenen Namen oder Bezeichnungen können Marken der jeweiligen Rechteinhaber sein, die markenrechtlich geschützt sein können. Die Wiedergabe von Marken in diesem Werk berechtigen nicht zu der Annahme, dass diese frei und von jedermann verwendet werden dürfen.

Inhaltsverzeichnis

1 Vorbemerkungen	4
2 Die Tabellen	5
2.1 Struktur	5
2.2 Text	6
2.3 Artikel	6
2.4 Varianten	9
2.5 Ressourcen	9
2.6 Anfügeartikel	11
2.7 Zusatzartikel	12
3 Ressource-Schlüssel	13
3.1 Verarbeitung der Ressource-Schlüssel	16
3.2 Parameter für die interaktive Artikelplatzierung im 3D-Modus	16
4 Historie	18

Literatur

[asa] Artiklespezifische Ansichten (ASA) Spezifikation. EasternGraphics GmbH

[dsr] Datenstruktur und Registrierung (DSR) Spezifikation. EasternGraphics GmbH

[ebase] EBase - Datenbank mit ausschließlich lesendem Zugriff. EasternGraphics GmbH

[glos] Bibliotheken, Serien & Co. – Übergreifende OFML–Begriffe. EasternGraphics GmbH

[iso639-1] ISO 639-1 http://de.wikipedia.org/wiki/ISO_639

[ofml] OFML – Standardisiertes Datenbeschreibungsformat der Büromöbelindustrie.
Copyright Industrieverband Büro und Arbeitswelt e. V. (IBA)

[pec] pCon Exchange Container (PEC) Spezifikation. EasternGraphics GmbH

1 Vorbemerkungen

Physisches Format

Alle Tabellen liegen als Textdatei im CSV-Format (Endung `.csv`) innerhalb des Katalogverzeichnisses einer OFML-Bibliothek (s. [dsr]).

Die Tabellenspalten müssen durch Semikolon getrennt werden. Werden in den Tabellenspalten Semikola verwendet, müssen diese Spalten von doppelten Anführungszeichen (") eingeschlossen werden. Anführungszeichen innerhalb von Spalten müssen verdoppelt werden ("").

Zeilen, die mit einem Doppelkreuz (#) beginnen, werden als Kommentar interpretiert und von der weiteren Bearbeitung ausgeschlossen.

Bedeutung der verwendeten Symbole

K = Key - Schlüsselfeld

O = Optional - optionale Tabelle / Feld kann leer sein

R = Required - Pflichttabelle / Feld darf nicht leer sein

NU = Not Unique - Ein Schlüssel darf mehrfach angegeben werden.

Standard ist „Unique“, dies wird implizit angenommen und nicht angegeben.

Einige Konzepte sind als *veraltet* markiert. Die Verwendung dieser Konzepte wird nicht empfohlen, da sie nur in bestimmten Applikationen unterstützt werden und auch dort die Unterstützung in Zukunft nicht mehr gewährleistet ist.

Tabellen-Übersicht

Die unten aufgelisteten Tabellen werden in den folgenden Abschnitten beschreiben.

structure (R)
text (R)
article (R)
variant (O)
resource (O)
resource_ap (O)
resource_fm (O)
resource_zn (O)
addable (O)
addgroup (O)
addon (O)

2 Die Tabellen

2.1 Struktur

Über diese zentrale Tabelle wird der Aufbau des Katalogs festgelegt. Ein Katalog besteht aus Elementen folgender Typen:

Ordner Ein Ordner kann Unterelemente aller Typen enthalten.

Information Rein informative Elemente. Die Auswahl eines solchen Elements durch den Anwender bewirkt die Anzeige der hinterlegten bzw. verlinkten Information.

Artikel Bei Auswahl dieser Elemente durch den Anwender wird eine Aktion ausgeführt, die sich auf das aktuell durch die Anwendung bearbeitete OFML-Projekt auswirken kann, z.B. Einfügen von Artikeln oder Ausführung von Aktionen auf bereits im Projekt enthaltenen Artikeln.

Durch die Unterordnung von Katalogelementen unter übergeordnete Ordner entsteht eine Kataloghierarchie beliebiger Tiefe (mit Ebene 1 als oberster Hierarchie-Ebene).

Die Reihenfolge der Katalogelemente in der Tabelle definiert auch die Reihenfolge bei der Anzeige im Katalogmodul der Anwendung. Dabei ist zu beachten, dass alle Unterelemente eines Ordners angegeben sind, bevor das nächste Element auf derselben Ebene wie der Ordner angegeben wird.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K(NU)	CHAR	Element-Schlüssel	siehe unten
2.	K(NU), O	CHAR	Variantenschlüssel	siehe unten
3.	R	NUM	Hierarchie-Ebene	Ebene in der Kataloghierarchie
4.	R	CHAR	Typ	F = Ordner A = Artikel bzw. Aktion I = Information
5.	O		reserviert	

Tabelle 1: structure

Anmerkungen zu den Schlüsseln:

- Die Werte in den Feldern 1 und 2 dienen als Schlüssel für den Zugriff auf die weiteren Tabellen, z.B. zum Ermitteln von Texten und Ressourcen. Ist Feld 2 nicht leer, bilden Element- und Variantenschlüssel (ggf. erweitert um das Sprachkürzel) zusammen einen komplexen Zugriffsschlüssel.
- Die Tabelle kann mehrere Einträge mit demselben kombinierten Schlüssel in den Feldern 1 und 2 enthalten. Die entsprechenden Katalogelemente teilen sich dann die gleichen Texte, Ressourcen etc¹.
- Für Elemente des Typs A, die mit realen Artikeln aus einer Produktdatenbank assoziiert sind, ist als Element-Schlüssel die entsprechende Grundartikelnummer zu verwenden.
- Elemente, die *nicht* mit realen Artikeln aus einer Produktdatenbank assoziiert sind, können durch ein @ als erstes Zeichen des Element-Schlüssels gesondert gekennzeichnet werden. Hinweis: Elemente des Typs A, deren Schlüssel mit dem Zeichen @ beginnt, werden von der Artikelsuchfunktion der Anwendung ausgeschlossen.

¹ Dies kann z.B. dazu genutzt werden, eine bestimmte Artikelvariante von verschiedenen Stellen in der Kataloghierarchie auszurufen und erzeugen zu können.

Beispiele:

```
@GOIEX; ;1;F;  
@GOIEXCB; ;2;F;  
GOIEXCB1; ;3;A;  
GOIEXCB2; ;3;A;  
GOIEXCB3;V1;3;A;  
GOIEXCB3;V2;3;A;
```

2.2 Text

Für jedes Katalogelement muss in dieser Tabelle für alle relevanten Sprachen ein Text hinterlegt werden. Dieser wird dann vom Katalog-Modul der Applikation bei der Darstellung des Katalogelements verwendet. Kann für die aktuell eingestellte Katalogsprache kein passender Text gefunden werden, ist die Darstellung des Katalogelements nicht gewährleistet!

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	siehe Tabelle <code>structure</code>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <code>structure</code>
3.	K, O	CHAR	Sprache	ISO-Sprachkürzel (s. [iso639-1])
4.	R	CHAR	Text	freier Text

Tabelle 2: `text`

Hinweise:

- Feste Zeilenumbrüche können durch `\n` im Feld `Text` erzwungen werden. Diese werden zusätzlich zu ggf. automatisch durch die Applikation erzeugten Zeilenumbrüchen angezeigt.
- Falls keiner der Tabelleneinträge für ein gegebenes Katalogelement (Felder 1 und 2) die aktuell in der Applikation eingestellte Katalogsprache unterstützt (nicht-leeres Feld 3), wird der Eintrag mit leerem Feld 3 verwendet. Existiert auch kein solcher Eintrag, ist das Verhalten der Applikation undefiniert².
Gleiches gilt für den Fall, dass mehrere passende Tabelleneinträge für ein gegebenes Katalogelement und die aktuelle Katalogsprache bzw. mit leerem Feld 3 gefunden werden.

Beispiele:

```
@GOIEX; ;de;GOI-Beispiele  
@GOIEXCB; ;de;Schraenke  
GOIEXCB1; ;de;Schrank mit Auszuegen  
GOIEXCB2; ;de;Schrank mit Schiebetueren  
GOIEXCB3;V1;de;Schrank mit Rollladen  
GOIEXCB3;V2;de;Unterschrank\n mit Rollladen
```

2.3 Artikel

In dieser Tabelle werden grundlegende Festlegungen für jedes Katalogelement getroffen, u.a. zu dessen Sichtbarkeit³. Ein Katalogelement, für das kein Eintrag in dieser Tabelle existiert, ist generell nicht sichtbar!

Die Tabelle kann auch Einträge für Artikel enthalten, die keine direkten Elemente des Katalogs sind (d.h., nicht in der Tabelle `structure` aufgeführt sind). Diese können dann durch die Artikel-Suchfunktion der Applikation gefunden und erzeugt werden.

² Im Extremfall kann dies dazu führen, dass das Katalogelement nicht angezeigt wird!

³ Der (historisch entstandene) Name der Tabelle ist etwas irreführend, da die Tabelle nicht nur für Artikel relevant ist.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	siehe Tabelle <i>structure</i>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <i>structure</i>
3.	O	NUM	Artikelkategorie	reserviert
4.	O	CHAR	Positionsnummer	reserviert
5.	R	CHAR	Einfügetyp	(Details siehe unten.) S = Standard H = (Objekt-)Hierarchie MC = Methodenaufruf FC = Funktionsaufruf PD = Teilplanung MP = MetaPlanning G = Geometrie PEC = pCon Exchange Container
6.	R	NUM	Sichtbarkeit	Bestimmt den Kontext ^a der Applikation, in dem das Katalogelement angezeigt werden darf. Die einzelnen Modi können additiv kombiniert werden. 0 = nicht sichtbar 1 = erlaubt in 3D-Umgebung 2 = erlaubt in 2D-Umgebung 4 = erlaubt in Planungsumgebung 8 = erlaubt in Konfigurationsumgebung 16 = nicht erlaubt im Warenkorb 32 = reserviert
7.	O	CHAR	Paket	Name des externen OFML-Paketes (s. [glos]), welches die Artikeldaten beinhaltet. Falls leer (Standard), dann ist dieses Paket das des Kataloges selbst.

^a Die angeführten Anwendungskontexte sind nicht näher spezifiziert, d.h. sie können durch die verschiedenen Applikationen unterschiedlich interpretiert werden.

Tabelle 3: *article*

Zu den **Einfügetypen**:

- S** Dies ist der Standard-Einfügetyp.
Für Katalogelemente der Typen *Ordner* und *Information* ist dies der einzig erlaubte Einfügetyp.
Für Katalogelemente des Typs *Artikel* bewirkt er das Einfügen des Artikels mit der durch den Element-Schlüssel bestimmten Grundartikelnummer und mit dem ggf. in der Tabelle *variant* hinterlegten Variantencode. Das Einfügeverhalten kann durch hinterlegte Ressourcen der Typen *AD*, *FM*, *OP* und *UV* modifiziert werden (s. Abschn. 3).
- H** Dieser Einfügetyp lädt eine OFML-Objekthierarchie aus einer externen Datei in die bestehende Planung. Die Datei muss mittels einer Ressource vom Typ *GF* oder vom Typ *SF* festgelegt werden⁴:
- Dateien des Ressourcen-Typs *GF* müssen im OGRP-Format vorliegen. Das darin gespeicherte Planungselement⁵ wird dann (ggf. zusätzlich zu schon vorhandenen Planungselementen) in die bestehende Planung eingefügt.
 - Dateien des Ressourcen-Typs *SF* müssen im FML-Format vorliegen. Die darin gespeicherte Szene von OFML-Planungselementen wird dann je nach Applikation entweder

⁴ Ist keine solche Ressource hinterlegt, ist das Verhalten der Applikation undefiniert.

⁵ welches auch eine komplexere Konfiguration darstellen kann (Gruppierung)

als komplexe Konfiguration in die bestehende Planung eingefügt oder ersetzt die bestehende Planung (welche nach Nachfrage zuvor noch gesichert werden kann).

- MC** Dieser Einfügetyp realisiert den Aufruf einer Methode auf dem selektierten Planungselement.
Signatur und Parameter des Methodenaufrufs müssen dazu in einer Ressource vom Typ MC hinterlegt werden.
- FC** Dieser Einfügetyp realisiert den Aufruf einer statischen Klassenmethode (Funktion).
Signatur und Parameter des Funktionsaufrufs müssen dazu in einer Ressource vom Typ FC hinterlegt werden.
- PD** Dieser Einfügetyp erzwingt das Einfügen eines Artikels im Kontext einer Teilplanung:
Ist keine bereits in der Planung enthaltene Teilplanung selektiert, wird vor dem eigentlichen Einfügen des mit dem Katalogelement assoziierten Artikels die Teilplanung erzeugt, die mit dem Katalogelement assoziiert ist, welches durch die Ressource vom Typ PD bestimmt ist, und der Artikel wird dann anschliessend in diese Teilplanung eingefügt.
Ist ein Planungselement selektiert und stimmt dessen Typ mit dem Typ der vorausgesetzten Teilplanung überein bzw. ist ein abgeleiteter Typ davon, wird der Artikel in diese bereits vorhandene Teilplanung eingefügt.
- MP** Die Auswahl eines Katalogelements dieses Typs startet einen *MetaPlanning*-Workflow.
Die dazu benötigten Angaben müssen in der Ressource vom Typ MP hinterlegt sein.
- G** Dieser Einfügetyp importiert eine Geometrie aus einer Datei.
Die zu importierende Geometriedatei ist in einer Ressource vom Typ ZF anzugeben.
- PEC** Dieser Einfügetyp lädt Daten aus einer externen Container-Datei in das bearbeitete Projekt. Die Datei mit Suffix `.pec` muss der Spezifikation für pCon Exchange Container genügen [pec] und mittels einer Ressource vom Typ PEC referenziert werden.
Welche Daten aus dem Container in das Projekt eingefügt werden können und in welcher Weise dies geschieht, ist abhängig von den einzelnen Applikationen bzw. vom Anwendungsmodus.

Nur der Einfügetyp `s` wird garantiert von allen Applikationen in allen Anwendungskontexten unterstützt. Insofern Applikationen die Katalogelemente mit den nicht unterstützten Einfügetypen nicht bereits selber in der Katalogansicht ausblenden, kann im Feld 6 eine solche Sichtbarkeitsbedingung angegeben werden, die die Anzeige der betreffenden Katalogelemente in den jeweiligen Anwendungskontexten unterbindet⁶.

Bezüglich des Verhältnisses der Sichtbarkeit von Ordnern und der Sichtbarkeit der Unterelemente eines Ordners gelten folgende Bestimmungen:

1. Sind alle Unterelemente eines Ordners unsichtbar, so ist der Ordner selber trotzdem sichtbar (wenn seine Sichtbarkeitsbedingung erfüllt ist).
2. Ist ein Ordner nicht sichtbar, sind seine Unterelemente generell auch nicht sichtbar (auch wenn deren Sichtbarkeitsbedingungen an sich erfüllt sind).

Beispiele:

```
GOIEXCB1;;0004;;S;15;::ofml::goiex
GOIEXCB2;;0004;;S;15;::ofml::goiex
GOIEXTA1;A4;0002;;S;15;::ofml::goiex
GOIEXCH1;B2;0003;;S;15;::ofml::goiex
@CALL;;;MC;7;
GOIEXCB3;V1;0004;;PD;15;::ofml::goiex
GOIEXCB3;V2;0004;;PD;15;::ofml::goiex
@config1;;;H;5;
@mpri1;;;MP;31;::metafactory::mt
@CALL;;;FC;7;
```

⁶ Die konkrete Verarbeitung des Sichtbarkeitsfeldes in den einzelnen Standard-Applikationen von EasternGraphics ist in einer separaten Application Note beschrieben.

2.4 Varianten

Für Katalogelemente des Typs *Artikel* mit Einfügetyp *s* kann in dieser Tabelle ein produktdatenabhängiger Variantencode hinterlegt werden. Dieser wird dann beim Einfügen des Artikels verarbeitet und führt zu einer von der Grundauführung abweichenden Artikelvariante.

Der Variantencode muss die Artikelvariante nicht vollständig beschreiben, sondern kann an beliebiger Stelle abgebrochen werden, wenn die bis dahin codierten Merkmale die gewünschte Variante des Artikels ausreichend definieren.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	Artikelnummer, s.a. Tabelle <i>structure</i>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <i>structure</i>
3.	R	CHAR	Variantencode	produktdatenabhängiger Variantencode

Tabelle 4: *variant*

Beispiele:

GOIEXCB3;V1;V1KL
 GOIEXCB3;V2;V2KL
 GOIEXTA1;A4;A4KF A4TP
 GOIEXCH1;B2;B2BS B2BL

2.5 Ressourcen

In dieser Tabelle können den Katalogelementen Ressourcen zugewiesen werden. Der Verwendungszweck und das Format der möglichen Ressourcen-Typen sind in Abschn. 3 näher beschrieben.

Ressourcen können sprachspezifisch angelegt werden (s. Feld 3). Für die Verwendung gelten dabei dieselben Regeln wie für die Beschreibungen in der Tabelle *text*, d.h.:

Falls keiner der Tabelleneinträge für ein gegebenes Katalogelement (Felder 1 und 2) die aktuell in der Applikation eingestellte Katalogsprache unterstützt (nicht-leeres Feld 3), wird der Eintrag mit leerem Feld 3 verwendet. Existiert auch kein solcher Eintrag, ist das Verhalten der Applikation undefiniert. Gleiches gilt für den Fall, dass mehrere passende Tabelleneinträge für ein gegebenes Katalogelement und die aktuelle Katalogsprache bzw. mit leerem Feld 3 gefunden werden.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	siehe Tabelle <i>structure</i>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <i>structure</i>
3.	K, O	CHAR	Sprache	ISO-Sprachkürzel
4.	K	CHAR	Typ	Typ (Schlüssel) der Ressource (siehe Abschn. 3)
5.	R	CHAR	Ressource	Inhalt/Beschreibung der Ressource (siehe Abschn. 3)

Tabelle 5: *resource*

Beispiele siehe Abschn. 3.

Co-Ressourcen

Ressourcen bestimmter Typen können aus der Tabelle `resource` in spezifische Tabellen ausgelagert werden – die sogenannten Co-Ressourcen-Tabellen. Beweggründe dafür können sein:

- Auslagerung von Ressourcen, die von einem externen Tool generiert werden (z.B. AP).
- "Verschlankung" der Tabelle `resource` und damit verbundene bessere Performanz (da die ausgelagerten Ressourcen erst gelesen werden müssen, wenn sie benötigt werden, und nicht schon beim Katalog-Aufbau).

Der Aufbau der Co-Ressourcen-Tabellen ist identisch zu `resource`, mit der Ausnahme, dass die Spalte `Typ` entfällt. Der Typ aller Ressourcen in einer Co-Ressourcen-Tabelle ist einheitlich und wird über die Erweiterung des Namens angegeben.

Werden die Ressourcen eines gegebenen Typs benötigt, führt die Applikation die entsprechenden Einträge aus der Tabelle `resource` mit denen der ggf. vorhandenen zugehörigen Co-Ressourcen-Tabelle zusammen. Mehrfache Einträge für einen kombinierten Schlüssel aus den Feldern 1-3 sind dabei nicht zulässig (bzw. führen zu einem undefinierten Verhalten der Applikation).

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	siehe Tabelle <code>structure</code>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <code>structure</code>
3.	K, O	CHAR	Sprache	ISO-Sprachkürzel
4.	R	CHAR	Ressource	Inhalt/Beschreibung der Ressource

Tabelle 6: `resource_<RESTYPE>`

Aktuell ist die Verwendung eine Co-Ressourcen-Tabelle nur für folgende Typen erlaubt:

Key	File name
AP	<code>resource_ap</code>
FM	<code>resource_fm</code>
ZN	<code>resource_zn</code>

2.6 Anfügeartikel

Anfügeartikel sind Vorschlags-elemente, die als Erweiterung an ein Basiselement (Katalogelement vom Typ *Artikel*) angefügt werden können.

Ob und wie die möglichen Anfügeartikel dem Anwender präsentiert werden, ist applikationsabhängig.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Element-Schlüssel	Artikelnummer, s.a. Tabelle <i>structure</i>
2.	K, O	CHAR	Variantenschlüssel	siehe Tabelle <i>structure</i>
3.	R	CHAR	Anfügeartikel	Eine Liste von Artikeln oder Artikelgruppen (s.u.), getrennt durch Semikolon. Artikelgruppen in eckigen Klammern. Anfügeartikel ^a werden anhand von Element- und ggf. Variantenschlüssel ^b des entsprechenden Eintrags in der Tabelle <i>article</i> spezifiziert, getrennt durch "::".

^a Anfügeartikel müssen nicht in der Tabelle der Katalogeinträge *structure* enthalten sein.

^b Die Angabe des Variantenschlüssels entfällt, wenn das entsprechende Feld in der Tabelle *article* leer ist.

Tabelle 7: *addable*

Beispiele:

```
GOIEXC01;C1;"[@GOIEXDR];GOIEXDR4"
GOIEXC02;A4;"[@GOIEXDR];[@GOIEXSB];GOIEXDR4;GOIEXDR6"
GOIEXSB1;B2;"GOIEXHB1;GOIEXSMIB1::A1;GOIEXTEDR"
GOIEXSB2;B2;"GOIEXHB1;GOIEXTEDR;GOIEXSMIB1"
```

Anfügegruppen

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K	CHAR	Anfügegruppe	freier Schlüssel
2.	R	CHAR	Artikel der Gruppe	Eine List von Artikeln, getrennt durch Semikolon. Artikelangabe wie oben in Tabelle <i>addable</i> .

Tabelle 8: *addgroup*

Beispiele:

```
@GOIEXDR;"GOIEXDR1;GOIEXDR2;GOIEXDR3"
@GOIEXSB;"GOIEXHB1;GOIEXHB2;GOIEXHB5"
```

2.7 Zusatzartikel

(Dies ist ein veraltetes Konzept.)

Zusatzartikel werden automatisch mit einem Basiselement erzeugt.

Mögliche Sets von (gemeinsam zu erzeugenden) Zusatzartikeln werden in dieser Tabelle unter einem sogenannten *AD*-Schlüssel angelegt. Dieser Schlüssel kann dann bei den relevanten Basisartikeln in der Ressource vom Typ *AD* hinterlegt werden.

Die Zusatzartikel werden anhand von Element- und Variantenschlüssel des entsprechenden Eintrags in der Tabelle *article* spezifiziert⁷.

Die Zusatzartikel werden als Geschwister des Basiselements eingefügt. Das Verhalten bzgl. Platzierung ist applikationsabhängig. Im Normalfall werden die Zusatzartikel neben dem Basiselement eingefügt. Das Verhalten kann u.U. mittels der Ressource vom Typ *UV* beeinflusst werden.

Feld	Parameter	Typ	Feldbezeichner	Beschreibung
1.	K(NU)	CHAR	„AD“-Schlüssel	siehe Abschn. 3
2.	R	CHAR	Element-Schlüssel	Artikelnummer, siehe Tabelle <i>article</i>
3.	O	CHAR	Variantenschlüssel	siehe Tabelle <i>article</i>

Tabelle 9: *addon*

Beispiele:

A01;GOIEXDR4;
A02;GOIEXDR6;

⁷ Zusatzartikel müssen nicht in der Tabelle der Katalogeinträge *structure* enthalten sein.

3 Ressource-Schlüssel

Die folgende Tabelle beschreibt mögliche Ressourcen-Typen (Schlüssel) und Werte.

Für Information über Ablageorte von Ressourcen und unterstützte Formate von Bilddateien siehe DSR-Spezifikation [dsr].

Dateinamen sind inklusive Suffix anzugeben.

Schlüssel	Format	Beschreibung
AD	<i>Key</i>	Addon-Schlüssel; siehe Tabelle <code>addable</code>
AP	siehe Abschn. 3.2	Parameter für die interaktive Artikelplatzierung im 3D Modus (s.u.: FM=[FDB3D,FDB3DP])
DS	-	Beschreibung (Langtext)
FC	<i>Package;</i> <i>Function(Parameters)</i>	OFML-Paketname und Methodenaufruf. Das Paket wird vor dem Aufruf der statische Klassenmethode geladen (s. [ofml]). Der optionale Platzhalter <code>%SELECTION%</code> wird vor dem Aufruf durch eine Liste der selektierten Element ersetzt (z.B. [t.e1, t.e2]). Ist kein Element selektiert wird eine leere Liste übergeben.
FM	<FOS,FDB3D,FDB3DP> FOS FDB3D FDB3DP FOS,[FDB3D FDB3DP]	Modi für interaktive Artikelplatzierung ^a (veraltet): interaktive, freie Platzierung im 2D-Modus interaktive, freie Platzierung im 3D-Modus programmierbare interaktive 3D-Platzierung ^b 2D- und 3D-Parameter sind komma-separiert kombinierbar.
GF	<i>File</i>	Datei im OGRP-Format
HF	<i>File</i>	Datei im HTML-Format
HT	-	HTML Text
IF	<i>File</i>	Bilddatei für die hochauflösende Präsentation des Katalogelements (z.B. in Formularen zum Druck) ^c
IT	<i>File</i>	Bilddatei für die Präsentation des Elements in der Katalogansicht der Applikation

Fortsetzung auf der nächsten Seite ...

^a Diese Modi sind nur für Applikationen relevant, die nicht schon per se eine interaktive Artikelplatzierung vornehmen. Falls die interaktive Artikelplatzierung nicht aktiviert oder keine Ressource vom Typ FM hinterlegt ist, verwendet die Applikation beim Einfügen eines Artikels zur Bestimmung von dessen Position dann eigene Planungslogiken, ggf. unter Berücksichtigung von Planungslogiken der bereits in der Planung befindlichen Elemente.

^b In diesem Modus kann die Platzierung im 3D-Modus durch die OFML-Daten des (selektierten) Referenzobjektes beeinflusst und aufgrund von Kollisionsprüfungen eingeschränkt werden.

^c Ob eine hinterlegte IF-Ressource tatsächlich in Formularen zur Anwendung kommt oder stattdessen nicht doch ein Bild der aktuellen Konfiguration generiert wird (vgl. [asa]), hängt von den Einstellungen in den jeweiligen Applikationen ab und ggf. von der Kategorie der betreffenden Artikelinstanz (siehe z.B. Kategorie SALESONLY_ARTICLE). Wenn eine Applikation die IF-Ressource nutzen möchte, diese aber nicht hinterlegt ist, kann sie stattdessen die IT-Ressource verwenden.

Schlüssel	Format	Beschreibung
MC	<Class @Interface;> Method(Parameters)	OFML-Klassenname/Interface und Methodenaufruf. Ruft die Methode (s. [ofml]) auf dem selektierten Element auf, wenn dieses von der voll qualifizierten gegebenen Klasse abstammt oder das Interface sowie die angegebene Methode unterstützt. Die Angabe von Klasse oder Interface ist optional.
MP	Symbol,Class(,Argument)	MP-Workflow-ID, UserExit-Klasse (jeweils voll qualifiziert), (optionales) Argument ^a .
MT	Content-Type;Content-Datei	MIME-Typ ^b
OP	Key=Value ^c	ODB-Parameter (komma-separiert), grafische Variante
PEC	File	pCon Exchange Container
PD	Key;Key	Element-Schlüssel und Variantenschlüssel des die Teilplanung spezifizierenden Katalogelements
SA	<COM MAT GEO> COM MAT GEO	Parameter für die Umwandlung in einen Sonderartikel ^d . Mehrere Parameter werden mit Komma separiert. Erlaubt die Änderung der kfm. Eigenschaften Erlaubt die Änderung der Materialien Erlaubt Schieben, Skalieren und Löschen von Komponenten
SF	File	OFML Szenendatei
TAG	-	frei verwendbare Ressource zur Kategorisierung des Katalogelements ^e
URL	-	Vollqualifizierter Internet-URL mit Protokoll-Präfix (RFC 3986). Wird durch internen oder System-Browser verarbeitet.

Fortsetzung auf der nächsten Seite ...

^a Bei dem optionalen Argument handelt es sich um ein Argument, welches die *initialize()*-Funktion der von der Basis-Klasse *ofml::go::GoMetaPlanning* abgeleiteten spezifischen UserExit-Klasse ggf. zusätzlich zu den Standard-Argumenten der geerbten Funktion benötigt.

^b Die Anzeige erfolgt entweder im integrierten Internet-Browser der Applikation oder im System-Browser ggf. unter Zuhilfenahme von separaten Viewer-Programmen. Der Umfang der verarbeitbaren Content-Typen hängt somit vom System des Anwenders ab. Der Content muss in einer separaten Datei vorliegen, d.h. darf *nicht* als Byte-Stream angegeben werden.

^c Entspricht dem OAM-Datentyp *Params*.

^d Diese Ressource hat keinen Einfluß auf den Einfügevorgang des Artikels. Sie wird erst und nur dann ausgewertet, wenn der Nutzer tatsächlich die Umwandlung in einen Sonderartikel aktiviert hat.

^e Diese Ressource wird in den Applikationen von EasternGraphics selber nicht ausgewertet, im Kontext des Online-Konfigurators aber zur Verarbeitung an Clienten weitergereicht. Die automatisierte Abbildung (Migration) von TAG-Ressourcen auf standardisierte Mechanismen zur Artikel-Filterung des zukünftigen, in der OFML-Normung befindlichen neuen Katalog-Formats ist nicht gewährleistet.

Schlüssel	Format	Beschreibung
UV	<i>Params</i>	Parameter für die Artikelerzeugung im 2D-Modus ^a (veraltet). Mehrere Parameter (Key,Value) werden mit Komma separiert ^b .
	U,x	Artikel wird in X-Richtung um x m verschoben
	V,x	Artikel wird in negative Z-Richtung um x m verschoben
	W,x	Artikel wird in Y-Richtung um x m verschoben
	W,REF	Artikel wird auf der Höhe eines selektierten Artikels platziert
	W,PUN	Platzierung auf Höhe 0.0 wird unterdrückt, sondern der Pick-Punkt verwendet
	WKL,x	Artikel wird um x Grad im Uhrzeigersinn um die Y-Achse rotiert
	WKL,MANU	manuelle interaktive Winkelbestimmung
	BAS,1	Element wird auf dem Basispunkt des selektierten Elements eingefügt
ZF	<i>File</i>	Geometriedatei im 3DS-, DXF- oder DWG-Format ^c
ZN	<i>OFML-Bezeichner</i>	vollqualifizierter oder lokaler Geometriename ^d

^a Einzelne Parameter können u.U. (applikationsabhängig) auch bei der Platzierung von Zusatzartikeln (s. 2.7) herangezogen werden.

^b Zuerst wird das Objekt auf die berechnete Position verschoben, dann ggf. um die Hochachse (Y) rotiert.

^c Über diese Standardformate hinaus können bestimmte Anwendungen weitere Formate unterstützen.

^d Geometriennamen werden ohne Dateiendung und Pfad angegeben, können aber durch einen vorangestellten Namensraum qualifiziert werden. Bei einem lokalen Geometriennamen (ohne vorangestellten Namensraum) muss sich die zugehörige Geometriedatei im Datenverzeichnis des aktiven OFML-Pakets befinden. Ist ein Namensraum vorangestellt, muss sich die zugehörige Geometriedatei in dem entsprechenden Verzeichnis relativ zum Wurzel-Datenverzeichnis der Applikation befinden. Die Namen der Geometriedateien müssen dabei kleingeschrieben sein und die Dateiendung muss mit dem Geometrieformat übereinstimmen. Erlaubt ist aktuell nur das Formate 3DS (.3ds).

Tabelle 10: Ressourcentypen und Werte

Beispiele:

```
@GOIEXCB;;de;HF;cupboards.html
GOIEXCB1;;de;HF;de/gOiExCupboard1.html
GOIEXCB1;;en;DS;Welcome
GOIEXCB1;;;YF;gOiExCupboard1.fig
GOIEXCB1;;;IF;gOiExCupboard1.jpg
GOIEXCB1;;;IT;gOiExCupboard_t_1.jpg
@FOOBAR;;de;HT;<html><h2>Foobar/h2</html>
@FOOBAR;;fr;URL;http://www.homepage.com/page.cgi?lang=fr#about
@INTRO;;de;MT;"application/octet-stream;start.exe"
@VIDEO;;de;MT;"video/x-msvideo;info.avi"
@INFO;;de;MT;"application/ms-excel;info.xls"
@CALL_CLS;;;MC; "::ofml::goiex::gOiExChair1;setPropValue(@S_SITZ, @gOiExBlue)"
@CALL_INT;;;MC;"@IF_GOIEX_MC_INT;setSpecial([@Key, @Val])"
GOIEXCB3;V1;;PD;"@GOIEXPD;"
GOIEXCB3;V2;;PD;"@GOIEXPD;"
A04711AF;;;FM;FOS
@config1;;;GF;config_1.ogrp
@Beispiel1;;;SF;Beispiel_1.fml
"ART10/0 XXXXXX";SE;;FM;FOS
"ART10/0 XXXXXX";SE;;UV;W,REF
"ART10/0 XXXXXX";SE;;AD;A01
GOIEXHB2;;;AP;["GOIEXHB2", @(), [[0.0, 0.0, 0.0], [0.8, 0.025, 0.5]], 2, 0, @()]
GOIEXHB2;;;FM;FDB3D
```

```

GOIEXHB2;;;ZN;;;ofml::goiex::etc::GOIEXAE2
GOIEXGEO;;;ZF;geo/goiexgeo.3ds
@mpri1;;;MP;;;metafactory::mt::@ri1,::metafactory::mt::MfMetaPlanning
@CALL_FC;;;FC;;;man::prog;Class::staticClassMethod(%SELECTION%) "
855;V1;;;OP;"ALIGN=@L"

```

3.1 Verarbeitung der Ressource-Schlüssel

Ressource-Schlüssel	Einfügetyp	Element-Typ	kombinierter Ressource-Schlüssel
AD	S	A	-
AP	S	A	FM = [FDB3D FDB3DP] ^a
DS	-	[A, F, I] ^b	-
FM	S	A	ZF oder ZN ^c
GF	H	A	-
IF	-	A	-
IT	-	A, F, I	-
MC	MC	A	-
FC	FC	A	-
MP	MP	A	-
OP	S	A	[FM = FOS] ^d
PEC	PEC	A	-
PD	PD	A	-
SA	S	A	-
SF	H	A	-
UV	S	A	-
ZF,ZN	S	A	FM = [FDB3D FDB3DP] (s.a. bei FM)
ZF	G	A	
HF, HT	-	I	-
MT	-	I	-
URL	-	I	-

^a Die AP-Ressource wird nur ausgewertet, wenn für das Katalogelement auch eine FM-Ressource mit dem Wert FDB3D bzw. FDB3DP hinterlegt ist.

^b Wird nicht von allen Anwendungen unterstützt.

^c Die durch die ZF- bzw. ZN-Ressource bestimmte Geometrie wird bei der interaktiven Artikelplatzierung im 3D-Modus als Feedback verwendet.

^d Einige Applikationen (z.B. pCon.planner 5) werten die OP-Ressource nur aus, wenn eine FM-Ressource mit dem Wert FOS hinterlegt ist.

Tabelle 11: Kombinationen von Ressource-Typen

Für den Element-Typ *Information* (I) werden die Ressourcen in folgender Reihenfolge ausgewertet: HF, HT, MT, URL. Es wird i.d.R. nur die erste gefundene Ressource verwendet.

3.2 Parameter für die interaktive Artikelplatzierung im 3D-Modus

In Ressourcen des Typs AP werden die Parameter hinterlegt, die zur Realisierung des Feedbacks während der interaktiven Artikelplatzierung im 3D-Modus benötigt werden. Das Format der Ressource ist die String-Notation eines OFML-Vektors, der folgende Elemente enthält:

1. Grundartikelnummer (String)
2. Kategorien bzgl. Planungslogiken(Symbol[])
3. lokale Volumenausdehnung (Float[2][3])
4. Freiheitsgrade der Verschiebung bzgl. *getTrAxis()* (Int)
5. Freiheitsgrade der Rotation bzgl. *getRtAxis()* (Int)
6. Information über Anfügepunkte (s. unten)

Das sechste Element ist eine Liste die [Schlüssel, Wert] Paare beinhaltet. Wobei Schlüssel ein Symbol ist, welches den Typ der in Wert spezifizierten Information angibt. (Das Element kann NULL sein oder weggelassen werden, wenn keine Anfügepunkte definiert sind.) Im Regelfall korrespondiert der Informationstyp zu einer Methode der Anfügepunkt-Schnittstelle. Die Informationstypen und deren Werte sind:

- @StdAttPts:
 - Der Wert ist eine Liste der Schlüssel der Standard-Anfügepunkte.
 - erzeugt/verwendet von *getStdAttPts()*
 - Darf weggelassen werden, wenn das komplette Set der Standard-Anfügepunkte verwendet wird.
- @AddAttPts:
 - Der Wert ist eine Liste der Schlüssel zusätzlichen Anfügepunkte.
 - erzeugt/verwendet von *getAddAttPts()*
 - Darf weggelassen werden, wenn keine zusätzlichen Anfügepunkte definiert sind.
- @AttPtsOrder:
 - Der Wert ist eine Liste von [key, dir] Paaren, welche die Anwendungsreihenfolge der Anfügepunkte festlegt, wenn kein Anfügepunkte vorselektiert ist.
 - Wenn die Liste einen Eintrag [@AllStdAttPts, NULL] enthält, wird dieser ersetzt durch eine Set von Einträgen für jeden Standard-Anfügepunkt.
 - erzeugt/verwendet von *getAttPtsOrder()*
- @AddAttPtDefs:
 - Der Wert ist eine Liste von [key, definition] Paaren. Wobei *definition* ein Vektor ist, korrespondierend zum Rückgabewert der Methode *getAddAttPtDef()*, der beinhaltet z.B. folgende Elemente:
 - * lokale Koordinates (Float[3])
 - * Anfügerichtung (Symbol)
 - * PY-Rotation (Float)
 - * Modus (Symbol: @Sibling oder @Child)
 - erzeugt/verwendet von *getAddAttPtDef()*
 - Darf weggelassen werden, wenn keine zusätzlichen Anfügepunkte definiert sind.
- @Opposites:
 - Der Wert ist eine Liste von [key_dir, opp_keys] Paaren. Wobei *key_dir* ein Vektor [key, dir] ist, *dir* darf bei bel. Richtung NULL sein, der ein Set von akzeptierten Parametern der Methode *oppositeAttPts(pKey,pDir)* darstellt. *opp_keys* gibt die Liste der passenden entgegengesetzten Anfügepunkte an.
 - erzeugt/verwendet von *oppositeAttPts(pKey,pDir)*.
 - Darf nur für zusätzliche Anfügepunkte angegeben werden.

Beispiele:

```
GOIEXHB2;;;AP;["GOIEXHB2", @(), [[0.0, 0.0, 0.0], [0.8, 0.025, 0.5]], 2, 0, @()
GOIEXAE2;;;AP;["GOIEXAE2", @(), [[0.0, 0.0, 0.0], [1.107, 0.72, 1.27]], 5, 2,
@([@AddAttPts, @(@GOIEXVL)], [@AddAttPtDefs, @([@GOIEXVL, [[0.0, 0.0, 0.4], @L, 0,
@Sibling]])]), [@Opposites, @([@GOIEXTL, @L], [@GOIEXVR], [[@GOIEXTR, @R],
[@GOIEXVL]])])])]
```

4 Historie

- 2.10.0 Neuer Einfügetyp PEC nebst zugehörigem neuen Ressource-Key
Veralteter Einfügetyp LS entfernt
- 2.9.1 Präzisierungen zu den Ressource-Schlüsseln IF, MP und MT
- 2.9.0 Abschnitt zu XCF2 entfernt
Nicht benutzte bzw. nicht unterstützte Einfügetypen P u. Y sowie zugehörige Ressource-Keys YF u. YN entfernt
Veraltete Einfügetypen B und C sowie zugehörige Ressource-Keys BX, CB und GR entfernt
Einfügetyp LS als veraltet markiert
Nicht benutzte Tabellen `category` und `cattext` entfernt
Sichtbarkeitsflag 32 als *reserviert* markiert
Anfügeartikel können nicht mehr mit einem OFML-Paket qualifiziert werden
Neuer Ressource-Key: TAG
Veralteter Ressource-Key BB entfernt
Ressource-Keys FM und UV als veraltet markiert
nicht unterstützter Parameter POS,MANU in der Ressource UV entfernt
Parameter BAS,1 in der Ressource UV hinzugefügt
Leichte Re-Organisation der Dokumentstruktur
Ergänzende Erläuterungen zu diversen Aspekten
Kleinere Korrekturen und Verbesserungen
- 2.8.0 Neuer Abschnitt: Erzeugung der 3D-Feedback Daten^{3.2}
Dateiformate für YF und ZF erweitert
Einfügetypen B und C und die Ressource-Keys BB, BX und CB als veraltet markiert
Neuer Ressource-Schlüssel: URL
Ergänzende Hinweise zur Verarbeitung der Ressourcen
Separate Format-Version für XCF und XCF2 hinzugefügt, zu Referenzzwecken.
- 2.7.1 Neuer Ressource-Schlüssel: OP
- 2.7.0 Neue Einfügetyp: FC
Neuer Ressource-Schlüssel: FC
CSV-XCF umbenannt in XCF
EBASE-XCF umbenannt in XCF2
- 2.6.2 Neue Einfügetypen: LS
Neuer Ressource-Schlüssel: LS
"Einfügemode" umbenannt in "Sichtbarkeit"
- 2.6.1 Korrektur der MP-Ressourceangabe
- 2.6.0 Neue Einfügetypen: B, C, G, MP
Neue Ressource-Schlüssel: BB, BX, CB, MP
Neue Notierung der Feldparameter.
Angabe mehrerer FM Einfügemodi möglich.
Korrekturen in „Kombinationen der Ressource-Schlüssel“^{3.1}
Beschreibung fester Zeilenumbrüche in der Text-Tabelle.
- 2.5.2 Modus EXT für SA-Ressource entfernt
- 2.5.1 Neue Tabellen: `resource_fm`, `resource_zn`
- 2.5.0 Neue Tabelle: `resource_ap`
- 2.4.2 Neuer Interaktions-Modus zum Ressource-Schlüssel FM: FDB3DP
Erweiterung zu Ressource-Schlüssel MC
- 2.4.1 Kommentare zu `article` und `structure` überarbeitet
- 2.4.0 Neuer Abschnitt: Kombinationen der Ressource-Schlüssel
„PD“ ist jetzt auch für Einfügetyp „H“ erlaubt
- 2.3.0 Neue Ressourcen-Schlüssel hinzugefügt: ZF, ZN, AP
Neuer Interaktions-Modus: FDB3D
Neue Ressourcen-Schlüssel für Sonderartikel: SA
- 2.2.0 Überarbeitung; Deutsche Version erstellt
- 2.1.1 Detaillierte Beschreibung der Ressourcen-Typen
- 2.1.0 Unterstützung der ADDON-Tabelle
XCF 2.0 `resource` Beispiele erweitert
- 2.0.0 XCF 2.0 hinzugefügt