



Extensible Catalog Format (XCF) Specification

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

Version 2.10

August 31, 2017

Legal disclaimer

Copyright © 1999-2017 EasternGraphics GmbH. All rights reserved. This work is protected by copyright law. All rights are reserved to EasternGraphics. The translation, reproduction, or dissemination, fully or in part, is only permitted upon the prior written consent of EasternGraphics. EasternGraphics assumes no guarantee for the completeness, accuracy, currentness, continuity, and fitness of this work for the purpose intended by the user. A liability of EasternGraphics is excluded, except in cases of intent or gross negligence and personal injury. All names or labels contained in this work may be trademarks of the respective holder of rights and may be trademark-protected. The fact that a trademark is mentioned in this work should not lead to the assumption that it is free and everybody is allowed to make use of it.

Contents

1 Preliminary notes	4
2 The tables	5
2.1 Structure	5
2.2 Text	6
2.3 Articles	6
2.4 Variants	9
2.5 Resources	9
2.6 Addable articles	11
2.7 AddOn–Articles	11
3 Resource keys	13
3.1 Processing of resource keys	16
3.2 Parameter for interactive placing of articles in 3D mode	16
4 History	18

References

[asa] Article Specific View setup (ASV) Specification. EasternGraphics GmbH

[dsr] Data Structure and Registration (DSR) specification. EasternGraphics GmbH

[ebase] EBase – Database for read-only access. EasternGraphics GmbH

[glos] Libraries, product lines & Co. – comprehensive OFML-terms and definitions. EasternGraphics GmbH

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

[ofml] OFML – Standardized data description format of the office furniture industry. Industrieverband Büro und Arbeitswelt e. V. (IBA)

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

1 Preliminary notes

Physical format

All tables are text files in CSV format (extension `.csv`) within the catalog directory of given OFML library (see [dsr]).

Columns have to be separated by semicolon. Values in columns, which contain semicolons, have to be enclosed in quotes (`"`). Quotes inside columns have to be doubled (`""`).

Lines beginning with a hash mark (`#`) are treated as comments and excluded from further processing.

Explanation of used symbols

K = Key - key field

O = Optional - optional table / field may be empty

R = Required - mandatory table / field must not be empty

NU = Not Unique - same key may appear many times.

Default is „Unique“, this will not be declared.

Some concepts are marked as *obsolete*. The use of these concepts is not recommended, as they are only supported in certain applications, and even there the support is no longer guaranteed in the future.

Table overview

The tables, listed below, will be described in the following sections.

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 The tables

2.1 Structure

The entries of this central table define the structure of the catalog. A catalog consists of items of the following types:

Folder A folder can contain sub items of all types.

Information Pure informative items. The selection of such an item by the user causes the display of the stored or linked information.

Article If such an item is selected by the user an action is performed which may affect the OFML project currently processed by the application, e.g., inserting an article or performing actions on articles already contained in the project.

Subordination of catalog items under a superior folder creates a catalog hierarchy of any depth (with level 1 as the highest hierarchical level). The order of the catalog items in the table also defines the order in which they are displayed in the catalog module of the application. Note: all sub items of a folder have to be provided before the next element at the same level as the folder is provided.

Field	Parameter	Type	Field Identifier	Description
1.	K(NU)	CHAR	Item Key	see below
2.	K(NU), O	CHAR	Variant Key	see below
3.	R	NUM	Hierarchy Level	level in catalog hierarchy
4.	R	CHAR	Item Type	F = Folder A = Article resp. Action I = Information
5.	O		reserved	

Table 1: structure

Notes to the keys:

- The values in fields 1 and 2 are used as a key to access the other tables, e.g. for determining texts and resources. If field 2 is not empty, element key and variant key (optionally accompanied by the language code) together form a complex access key.
- The table can contain multiple records with the same key combination in fields 1 and 2. The corresponding catalog elements then share the same texts, resources, etc¹.
- For items of type A, which are associated with real articles from a product database, the corresponding basic article number has to be used as the key for the item.
- Items which are *not* associated with a real article from a product database can be especially identified by an @ as the first character of the item key.
Note: Items of type A, whose key begins with character @ will be excluded from the article search function of the application.

Example:

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

¹ This can be used, for example, to create a specific product version from various places in the catalog hierarchy.

2.2 Text

For each catalog item a text needs to be stored in this table for all relevant languages. This is then used by the catalog module of the application for the presentation of the catalog item. If no matching text can be found for the currently used catalog language, the presentation of the catalog item is not guaranteed!

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	see table structure
2.	K, O	CHAR	Variant Key	see table structure
3.	K, O	CHAR	Language	ISO language code (see [iso639-1])
4.	R	CHAR	Text	free text

Table 2: text

Notes:

- Fixed line breaks can be enforced by using `\n` in field *Text*. They will be shown additionally to the line breaks possibly created automatically by the application.
- If none of the table entries for a given catalog item (fields 1 and 2) matches the currently used catalog language (non-empty field 3), the entry with empty field 3 is used. If there is no such table entry, application behavior then is undefined².
The same applies to the case that multiple matching table entries are found for a given catalog item and the current catalog language resp. with empty field 3.

Example:

```
@GOIEX;;;en;GOI-Examples
@GOIEXCB;;;en;Cupboard
GOIEXCB1;;;en;Cupboard with drawer
GOIEXCB2;;;en;Sliding door cupboard
GOIEXCB3;V1;;;en;Tambour cupboards
GOIEXCB3;V2;;;en;Lower cupboard\n with roll top
```

2.3 Articles

In this table, basic characteristics for each catalog item are specified, including its visibility³. A catalog item for which there is no entry in this table, is generally not visible!

The table may also contain entries for articles that are not direct elements of the catalog (i.e., that are not listed in the table structure). These can then be found and created by the article search function of the application.

² In extreme cases this can lead to the situation that the catalog item is not displayed!

³ The (historically established) name of the table is somewhat misleading, because the table is not only relevant for articles.

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	see table structure
2.	K, O	CHAR	Variant Key	see table structure
3.	O	NUM	Article Category	reserved
4.	O	CHAR	Position Number	reserved
5.	R	CHAR	Insert Type	(details see below) S = Standard H = (Object-)Hierarchy MC = Method Call FC = Function Call PD = Partial Design MP = MetaPlanning G = Geometry PEC = pCon Exchange Container
6.	R	NUM	Visibility	Specifies the context ^a of the application in which the catalog item may be displayed. The modes can be combined additively. 0 = non-visible 1 = allowed in 3D environment 2 = allowed in 2D environment 4 = allowed in Planning environment 8 = allowed in Configuration environment 16 = not allowed in basket environment 32 = reserved
7.	O	CHAR	Package	Name of an external OFML package (see [glos]), containing the article data. If empty (default) this package is the package of the catalog itself.

^a The stated application contexts are not specified, i.e. they can be interpreted differently by the various applications.

Table 3: article

Concerning the **insert types**:

- S** This is the standard insert type.
For catalog elements of types *Folder* and *Information*, this is the only permitted insert type. For catalog items of type *Article*, it effects the insertion of the article with the base article number determined by the item key and with the variant code possibly stored in table *variant*. The insertion behaviour can be modified by means of specified resources of types AD, FM, OP and UV (see section 3).
- H** This insert type loads an OFML object hierarchy from an external file into the existing planning. The file has to be specified by means of a resource of type GF or SF⁴:
- Files of resource type GF have to be given in OGRP format. The stored therein planning element⁵ will be inserted into the existing planning (possibly in addition to already existing planning elements).
 - Files of resource type SF have to be given in FML format. The stored therein scene of OFML planning elements, depending on the application, either will be inserted into the existing planning as a complex configuration or will replace the existing planning (which can be saved prior to this by demand).

⁴ If there is none such resource specified, the behaviour of the application is undefined.

⁵ which may represent a complex configuration (group)

- MC** This insert type realizes a method call on the selected planning element. Signature and parameters of the method call have to be specified in a resource of type MC.
- FC** This insert type realizes the call of a static class method (function). Signature and parameters of the function call have to be specified in a resource of type FC.
- PD** This insert type forces the insertion of an article in the context of a partial planning (group). The presupposed partial planning has to be specified in a resource of type PD by referring to the catalog item associated with the partial planning.
 If no already existing partial planning is selected in the planning, before actually inserting the article associated with the catalog item, the presupposed partial planning will be inserted into the planning, and the article then will be inserted into this partial planning.
 If an already existing partial planning is selected in the planning, and if its type matches the type of the presupposed partial planning or is derived from it, the article will be inserted into this partial planning.
- MP** The selection of catalog item of this insert type starts a *MetaPlanning* workflow. The information required for this purpose has to be specified in a resource of type MP.
- G** This insert type imports a geometry from a file. The imported geometry file has to be specified in a resource of type ZF.
- PEC** This insert type loads data from an external container file into the edited project. The file with suffix `.pec` has to comply with the specification for pCon Exchange Container [pec] and be referenced by means of a resource of type PEC. Which data from the container can be inserted into the project and how this is done depends on the individual applications or the application mode.

Only support of insert type S is guaranteed by all applications in all application contexts. If an application not already by itself hides the catalog items with the unsupported insert types in its catalog view, in field 6 such a visibility condition can be specified, which disables the display of the relevant catalog items in the respective application contexts⁶.

Concerning the relationship of the visibility of folders and the visibility of the items in a folder, the following provisions apply:

1. If all sub-elements of a folder are not visible, the folder itself is still visible (if its visibility condition is satisfied).
2. If a folder is not visible, its sub items are generally not visible (even if their visibility conditions are satisfied themselves).

Example:

```
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 The concrete processing of the visibility field in the individual standard applications of EasternGraphics is described in a separate application note.

2.4 Variants

In this table a product data-dependent variant code can be stored for catalog items of type *Article* with insert type *S*. This code is then processed during insertion of the article and leads to a product version deviating from the basic version.

It is not required that the variant code completely describes the product version. The code can be cut short at any point, if the characteristics coded up to this point already define the required version of the product.

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	article number, see also table structure
2.	K, O	CHAR	Variant Key	see table structure
3.	R	CHAR	Variant Code	product data dependent variant code

Table 4: variant

Example:

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

2.5 Resources

In this table, resources can be allocated to catalog items. The purpose and format of the possible resource types are described in more detail in section 3.

Resources can be language-specific (see field 3). They will be used complying with the same rules as specified for descriptions in table *text*, i.e.:

If none of the table entries for a given catalog item (fields 1 and 2) matches the currently used catalog language (non-empty field 3), the entry with empty field 3 is used. If there is no such table entry, application behavior then is undefined. The same applies to the case that multiple matching table entries are found for a given catalog item and the current catalog language resp. with empty field 3.

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	see table structure
2.	K, O	CHAR	Variant Key	see table structure
3.	K, O	CHAR	Language	ISO language code
4.	K	CHAR	Type	type of the resource (see section 3)
5.	R	CHAR	Resource	content of the resource (see section 3)

Table 5: resource

Examples see section 3.

Co-Resources

Resources of certain types can be outsourced from table `resource` to specific tables – the so-called co-resource tables. Motives for this may be:

- Outsourcing of resources that are generated by an external tool (e.g., AP).
- "Slimming down" table `resource`, accompanied with improved performance (since the outsourced resources only need to be read when they are required, and not already when building up the catalog view).

The structure of the co-resources tables is nearly identical to `resource`, except that field `Type` is dropped. The type of all resources within a co-resources is the same and is defined by the file name suffix.

If the resources of a given type are required, the application merges the appropriate entries from table `resource` and the entries from the possibly existing associated co-resources table. Multiple entries for a combined key comprising fields 1-3 are not allowed (resp. lead to a undefined behavior of the application).

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	see table structure
2.	K, O	CHAR	Variant Key	see table structure
3.	K, O	CHAR	Language	ISO language code
4.	R	CHAR	Resource	content of the resource

Table 6: `resource_<RESTYPE>`

Currently the usage of co-resources tables is only allowed for the following resource types:

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

2.6 Addable articles

Addable articles are suggested elements which can be attached to a base element (catalog item of type *Article*).

Whether and how the possible addable articles are presented to the user, depends on the application.

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Item Key	article number, see also table structure
2.	K, O	CHAR	Variant Key	see Table structure
3.	R	CHAR	Addables	A list of addable articles or article groups, separated by semicolon. Article groups in square brackets. Addable articles ^a are specified by means of the item key and possibly the variant key ^b of corresponding entry in table <i>article</i> , separated by "::".

^a Addable articles need not be included in the table of catalog items *structure*.

^b The variant key is omitted if the corresponding field in table *article* is empty.

Table 7: addable

Example:

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

Addable Group

Field	Parameter	Type	Field Identifier	Description
1.	K	CHAR	Addable Group	free key
2.	R	CHAR	Articles	A list of articles separated by semicolon. Specification of articles as above in table <i>addable</i> .

Table 8: addgroup

Example:

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

2.7 AddOn–Articles

(This concept is obsolete.)

AddOn–Articles (add-ons) are created automatically with their base element.

Possible sets of add-ons (to be created together) are specified in this table employing a so-called *AD–Key*. This key can then be used in the resource of type *AD* for the relevant base articles.

Add-ons are specified by means of the item key and the variant key of corresponding entry in table `article`⁷.

The add-ons are created and inserted as siblings of the base element. The behavior regarding placement is application-specific. In the normal case, add-ons are inserted beside the base element. The behavior can be influenced by circumstances (application-specific) using the resource of type `UV`.

Field	Parameter	Type	Field Identifier	Description
1.	K(NU)	CHAR	„AD“-Key	see section 3
2.	R	CHAR	Item Key	article number, see <code>article</code>
3.	O	CHAR	Variant Key	see table <code>article</code>

Table 9: `addon`

Example:

```
A01;GOIEXDR4;
A02;GOIEXDR6;
```

⁷ Add-ons need not be included in the table of catalog items structure.

3 Resource keys

The following table describes possible resource types (keys) and values.

For information about storage places of resources and allowed formats of image files see DSR specification [dsr].

File names must include a suffix.

Key	Value	Description
AD	<i>Key</i>	Addon key; see table <code>addable</code>
AP	see section 3.2	Parameter for interactive placing of articles in 3D mode (see below: FM=[FDB3D FDB3DP])
DS	-	Description (long text)
FC	<i>Package;</i> <i>Function(Parameters)</i>	OFML package name and method call. The package will be loaded before the call of the specified static class method (see [ofml]). The optional place holder <code>%SELECTION%</code> will be replaced with the list of currently selected elements (e.g. [t.e1, t.e2]) before executing. If no element is selected an empty list will be passed.
FM	<FOS,FDB3D,FDB3DP> FOS FDB3D FDB3DP FOS,[FDB3D FDB3DP]	Modes for interactive placing of articles ^a (obsolete): interactive free placing of article in 2D mode interactive free placing of article in 3D mode programmable interactive placing of article in 3D mode ^b Parameters for 2D and 3D can be combined; separated by comma.
GF	<i>File</i>	File in OGRP format
HF	<i>File</i>	File in HTML format
HT	-	HTML text
IF	<i>File</i>	Image file for high resolution presentation of the catalog item (e.g. in printed forms) ^c
IT	<i>File</i>	Image file for presentation of the item in the catalog view of the application
MC	<Class/@Interface;> <i>Method(Parameters)</i>	OFML class name/interface and method call. Calls method (see [ofml]) of selected object, if it inherits from given class or supports specified interface as well as specified method. Specification of class and interface is optional.

Continued on next page ...

^a These modes are only relevant for applications that do not per se perform an interactive placement of articles. If the interactive placement is not enabled or if no resource of type FM is specified, the application then will use its own planning logics when inserting an article in order to determine the position of the article, possibly taking into account planning logics of elements already contained in the planning.

^b In this mode, the placement can be influenced by the OFML data of the (selected) reference object and can be limited due to collision tests.

^c Whether a stored IF resource actually is applied in forms or, instead, a generated picture of the current configuration is used (see [asa]), depends on the settings in the respective applications and, possibly, on the category of the article instance (consider, e.g., category SALESONLY_ARTICLE). If an application wants to use the IF resource, but it is not available, it may use the IT resource instead.

Key	Value	Description
MP	<i>Symbol,Class(,Argument)</i>	ID of the MP work-flow, UserExit class (both fully qualified), (optional) argument ^a
MT	<i>Content-Type;Content-File</i>	MIME type ^b
OP	<i>Key=Value^c</i>	ODB parameter (separated by comma), graphical variant
PEC	<i>File</i>	pCon Exchange Container
PD	<i>Key;key</i>	item key and variant key of part design catalog item
SA	<COM MAT GEO> COM MAT GEO	Parameter for conversion into a special article ^d . Multiple parameters can be specified, separated by comma. Permits changing commercial properties Permits changing materials Permits translation, scaling and removing of components
SF	<i>File</i>	OFML scene file name
TAG	-	freely usable resource for categorization of catalog items ^e
URL	-	Fully qualified internet URL with protocol prefix (RFC 3986). Will be processed by internal or system browser.
UV	<i>Params</i> U,x V,x W,x W,REF W,PUN WKL,x WKL,MANU BAS,1	Parameters for article creation in 2D mode ^f (obsolete). Multiple parameters can be specified, separated by comma ^g . Article will be moved in X direction by x m Article will be moved in Z direction by x m Article will be moved in Y direction by x m Article will be placed at the height of the selected article Placement at the height 0.0 will be suppressed, instead the pick-point is used Article will be rotated by x degrees around Y axis Force manual (interactive) angle selection Element will be inserted at the base point of the selected element.
ZF	<i>File</i>	Geometry file in 3DS, DXF or DWG format ^h
ZN	<i>OFML identifier</i>	Fully qualified or local geometry name ⁱ

^a The optional argument is that argument which, possibly, is required by the *initialize()* function of the specific UserExit class, derived from base class *::ofml::go::GoMetaPlanning*, in addition to the standard arguments of the inherited function.

^b The content will be displayed either in the built-in Internet browser of the application or in the system browser, optionally with the aid of separate viewer applications. Thus, the extent of the processable content types depends on the user's system. The content has to be given in a separate file, i.e. may *not* be specified as a byte stream.

^c Corresponds to OAM data type *Params*.

^d This resource has no influence on the insertion of the article. It is evaluated only then, if the user has actually enabled the conversion into a special article.

^e This resource is not evaluated in the applications of EasternGraphics themselves, but is passed for processing to clients in the context of the Online Configurator. The automated transformation (migration) of TAG-resources into standardized mechanisms for the filtering of articles, as specified in the upcoming new OFML catalog format, is not guaranteed.

^f Under certain circumstances (application-specific) individual parameters can be also employed in the placement of AddOn-Articles (see 2.7).

^g First, the object is moved to the calculated position, then, if necessary, it is rotated around the vertical axis (Y).

^h Beyond these standard formats, certain applications may support other formats, too.

ⁱ Geometry names are specified without a file extension and path, but can be qualified by a prefixed namespace. For a local geometry name (without a prefixed namespace) the associated geometry file has to be located in the data directory of the active OFML package. If a namespace is prefixed, the associated geometry file has to be located in the appropriate directory relative to the root data directory of the application. The names of the geometry files have to be lowercase and the file extension has to match the geometry format. Currently, the only allowed format is 3DS (.3ds).

Table 10: Resource keys and values

Examples:

```
@GOIEXCB;;en;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;;en;HT;<html><h2>Foobar/h2></html>
@FOOBAR;;fr;URL;http://www.homepage.com/page.cgi?lang=fr#about
@INTRO;;en;MT;"application/octet-stream;start.exe"
@VIDEO;;de;MT;"video/x-msvideo;info.avi"
@INFO;;en;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, @Value])"
GOIEXCB3;V1;;PD;"@GOIEXPD;"
GOIEXCB3;V2;;PD;"@GOIEXPD;"
A04711AF;;;FM;FOS
@config1;;;GF;config_1.ogrp
@example1;;;SF;example_1.fml
"ART10/O      XXXXXX";SE;;FM;FOS
"ART10/O      XXXXXX";SE;;UV;W,REF
"ART10/O      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
@mpr11;;;MP;"::metafactory::mt::@ri1,::metafactory::mt::MfMetaPlanning"
@CALL_FC;;;FC;"::man::prog;Class::staticClassMethod(%SELECTION%)"
855;V1;;OP;"ALIGN=@L"
```

3.1 Processing of resource keys

Resource Key	Insertion Type	Structure Type	combined Resource Key
AD	S	A	-
AP	S	A	FM = [FDB3D FDB3DP] ^a
DS	-	[A, F, I] ^b	-
FM	S	A	ZF or 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] (see also above)
ZF	G	A	
HF, HT	-	I	-
MT	-	I	-
URL	-	I	-

^a Resources of type AP will be evaluated only if there is a resource of type FM specified with value FDB3D or FDB3DP.

^b Not supported by all applications.

^c The geometry specified by the resource of type ZF or ZN will be used as feedback during interactive placing of articles in 3D mode.

^d Some applications (e.g. pCon.planner 5) evaluate resources of type OP only if there is a resource of type FM specified with value FOS.

Table 11: Combinations of resource types

For the structure type Information (I) the resources are evaluated in the following order: HF, HT, MT, URL. Usually only the first matching resource is used.

3.2 Parameter for interactive placing of articles in 3D mode

Resources of type AP are used to specify the parameters required for realizing the feedback during interactive placing of articles in 3D mode. The format of this resource is a String notation of an OFML vector containing the following elements:

1. base article number (String)
2. categories related to planning logics (Symbol[])
3. local delimiting volume (Float[2][3])
4. degrees of freedom for translation according to *getTrAxis()* (Int)
5. degrees of freedom for rotation according to *getRtAxis()* (Int)
6. information about attach points (see below)

Sixth element is a list containing [key, value] pairs, where key is a Symbol specifying the type of information provided in value element. (The element can be NULL or be omitted if there are no attach points defined.) Typically, the information type corresponds to a method of the attach point interface. The information types and their corresponding values are:

- @StdAttPts:
 - value is the list of keys of standard attach points
 - generated/used by *getStdAttPts()*
 - may be omitted if the complete set of standard attach points is used
- @AddAttPts:
 - value is the list of keys of additional attach points
 - generated/used by *getAddAttPts()*
 - may be omitted if there are no additional attach points defined
- @AttPtsOrder:
 - value is the list of [key, dir] pairs defining the processing order of attach points if no attach point is preselected
 - if the list contains an entry [@AllStdAttPts, NULL], it will be replaced by a set of entries for each standard attach point
 - generated/used by *getAttPtsOrder()*
- @AddAttPtDefs:
 - value is a list of [key, definition] pairs, where *definition* is a vector corresponding to return value of method *getAddAttPtDef()*, i.e. containing the following elements:
 - * local coordinates (Float[3])
 - * attach direction (Symbol)
 - * PY rotation (Float)
 - * mode (Symbol: @Sibling or @Child)
 - generated/used by *getAddAttPtDef()*
 - may be omitted if there are no additional attach points defined
- @Opposites:
 - value is a list of [key_dir, opp_keys] pairs, where *key_dir* is a vector [key, dir] specifying an accepted set of parameters for method *oppositeAttPts(pKey,pDir)* (*dir* may be NULL if direction doesn't matter), and where *opp_keys* specifies the list of fitting opposite attach points.
 - generated/used by *oppositeAttPts(pKey,pDir)*.
 - must be provided only for additional attach points

Examples:

```
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 History

- 2.10.0 New insert type PEC together with according resource key
Removed obsolete insert type LS
- 2.9.1 More precise specification regarding resource keys IF, MP and MT
- 2.9.0 Removed XCF2 section
Removed unused resp. not supported insert types P and Y as well as corresponding resource keys YF and YN
Removed obsolete insert types B and C as well as corresponding resource keys BX, CB and GR
Marked insert type LS as obsolete
Removed unused tables `category` and `cattext`
Marked visibility flag 32 as *reserved*
Addable articles can no longer be qualified with a OFML package
New resource key: TAG
Removed obsolete resource key BB
Marked resource keys FM and UV as obsolete
Removed not supported parameter POS,MANU for resource UV
Added parameter BAS,1 for resource UV
Slight re-organization of document structure
Added explanations regarding various aspects
Minor corrections and enhancements
- 2.8.0 New section: Generation of 3D feedback data
File formats for YF and ZF extended
Insert types B and C and resource keys BB, BX and CB marked as obsolete
New resource key: URL
Additional remarks to resource processing
Separate format version for XCF and XCF2 added, for reference purposes.
- 2.7.1 New resource key: OP
- 2.7.0 New insert type: FC
New resource key: FC
Renamed CSV-XCF to XCF
Renamed EBASE-XCF to XCF2
- 2.6.2 New insert type: LS
New resource key: LS
"Insertion mode" renamed to "Visibility"
- 2.6.1 Correction of MP resource specification
- 2.6.0 New insert modes: B, C, G, MP
New resource keys: BB, BX, CB, MP
New field parameter notation.
Specification of multiple FM insertion modes allowed.
Corrections in „Combination of resource keys“^{3.1}
Description of fixed line breaks `Text` table.
New section: „Generation of 3D feedback data“^{3.2}
Removed obsolete parameter VARKONSTR of UV resource.
- 2.5.2 Removed mode EXT for SA resources
- 2.5.1 New tables: `resource_fm`, `resource_zn`
- 2.5.0 New table: `resource_ap`
- 2.4.2 Added new interaction mode to resource key FM: FDB3DP
Extension to resource key MC
- 2.4.1 Revised comments to `article` and `structure`
- 2.4.0 New section: Combinations of resource keys
„PD“ allowed for insert type „H“ now
- 2.3.0 Added new resource types: ZF, ZN, AP
Added new resource FM mode: FDB3D
Added resource key for special articles: SA
- 2.2.0 Revision; German version created
- 2.1.1 Detailed description of resource types
- 2.1.0 Added support for ADDON
EBASE XCF `resource` examples extended
- 2.0.0 Added XCF 2