Extensible Catalog Format (XCF)

Specification, Version 2.11.0

Editors: Bernd Heinemann, Thomas Gerth

March 3, 2025

Contents

References

[article] The OFML Interfaces *Article* and *CompositeArticle* (Specification). EasternGraphics GmbH

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

[glos] Libraries, Series & Co. – fundamental OFML terms. 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] PEC – pCon Exchange Container (Specification). EasternGraphics GmbH

Except [pec][1] the specifications of EasternGraphics GmbH as well as [ofml] are available via the *pCon Download Center*

https://download-center.pcon-solutions.com

in the category *OFML Specifications*.

[1] If required, the specification [pec] can be requested from the relevant project manager at EasternGraphics.

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 - a key (consisting of 1-n fields) may be specified more than once
Default is „Unique“, this will not be declared.

Some concepts are marked as *obsolete*. The use of these concepts is not recommended, as 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)

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 arcticle 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, etc[2].
- 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;
```

[2] 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 undefined[3].
  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 visibility[4]. 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.

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

[4] 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 MC = Method Call FC = Function Call 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
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. The concrete processing of the visibility field in the individual standard applications of EasternGraphics is described in a separate application note.
Value `15` guarantees visibility in all application contexts.

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 a resource of type `OP` (see section 3).

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`.

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`.
This is an obsolete concept! The insertion type PEC should be used instead.

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 insert types `S` and `PEC` are guaranteed to be supported by all applications in all application contexts. If applications do not already hide catalog elements with unsupported insertion types in the catalog view themselves, a visibility condition can be specified in field 6 that prevents the catalog elements in question from being displayed in the respective application contexts.

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;
@config1;;;;H;5;
@mpri1;;;;MP;31;::metafactory::mt
@CALL;;;;FC;7;
```

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 itmes. 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.

3 Resource keys

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

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

File names must include the filename extension.

<table>
<tr><th>Key</th><th>Value</th><th>Description</th></tr>
<tr><td>DS</td><td>-</td><td>Description (details)[a]</td></tr>
<tr><td>FC</td><td>Package;
Function(Parameters)</td><td>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 %SELECTION% 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.</td></tr>
<tr><td>HF</td><td>File</td><td>File in HTML format</td></tr>
<tr><td>HT</td><td>-</td><td>HTML text</td></tr>
<tr><td>IF</td><td>File
see also section 3.2</td><td>Image file for high-resolution presentation
of the article in printed forms</td></tr>
<tr><td>IT</td><td>File
see also section 3.2</td><td>Image file for presentation of the item in the
catalog view of the application</td></tr>
<tr><td>MC</td><td><Class|@Interface;>
Method(Parameters)</td><td>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.</td></tr>
<tr><td>MP</td><td>Symbol,Class(,Argument)</td><td>ID of the MP work-flow, UserExit class (both fully qualified), (optional) argument[b]</td></tr>
<tr><td>MT</td><td>Content-Type;Content-File</td><td>MIME type[c]</td></tr>
<tr><td>OP</td><td>Key=Value[d]</td><td>ODB parameter (separated by comma), graphical variant[e]</td></tr>
<tr><td>PEC</td><td>File</td><td>pCon Exchange Container</td></tr>
<tr><td>TAG</td><td>-</td><td>freely usable resource for categorization of catalog items[f]</td></tr>
<tr><td>URL</td><td>-</td><td>Fully qualified internet URL with protocol prefix (RFC 3986). Will be processed by internal or system browser.</td></tr>
<tr><td>ZF</td><td>File</td><td>Geometry file in DWG format</td></tr>
</table>

[a] The description can (!) be used by the applications, e.g., instead of the name for display in the tooltip of the catalog entry.
Another possible use case is the advanced article search, where the description can be displayed in a separate column in the search results.

[b] 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.

[c] The content will be displayed either in the built-in 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.

[d] Corresponds to OAM data type *Params*.

[e] For performance reasons, the resource should only be used if the graphical variant cannot be indicated in the variant code in table `variant`, or only with great difficulty.

[f] 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.

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])"
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 Processing of resource keys

Resource Key	Insertion Type	Structure Type
DS	-	[A, F, I][a]
IF	-	A
IT	-	A, F, I
MC	MC	A
FC	FC	A
MP	MP	A
OP	S	A
PEC	PEC	A
ZF	G	A
HF, HT	-	I
MT	-	I
URL	-	I

[a]Not supported by all applications.

Table 6: 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 Specifications and recommendations for image resources

The following specifications and recommendations apply to the various image resources:

`IT` – for the presentation in the catalog view of the application:

- Image size: approx. 400 pixels for the longer side of the image.

`IF` – for high-resolution presentation in forms for printing:

- Image size: approx. 2000 pixels for the longer side of the image.
  Larger images can be provided if specific application scenarios require it.
- By default, the images required for forms are generated by the applications themselves based on the current configuration. Stored IF resources are used only for articles that are assigned to category *SALESONLY_ARTICLE* [article], or if the user explicitly has switched off image generation (provided the application allows such a setting).
- If an application wants to use the IF resource, but it is not available, it may use the IT resource instead.

The applications are responsible for correctly scaling and displaying images of different sizes in the specific usage scenarios[5].

The following recommendations apply for a visually good presentation:

- Images should not have unnecessary white space.
- Images may not have a frame/border or similar.
- The background should be transparent or white.

Formats JPEG and PNG formats are permitted for image resources, where a file in PNG format has priority. For general image format conventions see DSR specification [dsr].

[5] e.g. depending on the selected representation mode or on the resolution of the display medium

4 History

2.11.0	Declared insert type G as obsolete (use PEC instead)
	Removed no longer supported insert type H as well as the corresponding resource keys GF and SF
	Removed no longer supported insert type PD as well as the corresponding resource key PD
	Removed obsolete concept of Addon-articles (table `addon`) as well as the corresponding resource key AD
	Removed longer supported/used concept of addable articles (tables `addable` and `addgroup`)
	Supplementary explanations regarding resource keys DS, IF, IT and OP
	Removed obsolete and no longer supported resource keys: AP, FM, SA, UV, ZN
	With the removed resource keys AP, FM and ZN, also the co-resource tables are no longer required/supported
	The formats 3DS and DXF are no longer supported for geometry files in resources of type ZF
	Removed reference to article specific views
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 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“
	Removed obsolete parameter VARKONSTR of UV resource.