Manual pCon.creator 2.21

EasternGraphics
visualize your business

Welcome to pCon.creator!

This help will guide you through the different data entry forms and functions of this application.

pCon.creator is an extensible data creation and maintenance application for OFML data based on a relational database system.

This application contains modules for data creation of :

- Commercial data
- Geometrical data
- Catalogs

pCon.creator can be embedded into complex data entry processes using external data sources like ERP systems or AutoCAD. Progressive functions for data filtering and synchronization facilitate the creation of consistent data for multiple markets.

We wish you Good Luck and Success when working with pCon.creator!

EasternGraphics
visualize your business

Manual pCon.creator

Legal disclaimer

Publisher
EasternGraphics GmbH
Albert-Einstein-Str. 1

98693 Ilmenau

GERMANY

Phone
+ 49 - (0) 36 77 - 67 82 0

Fax
+ 49 - (0) 36 77 - 67 82 50

Web
http://www.EasternGraphics.com

Email
support@EasternGraphics.com

Version 2.21.0.111

February 2025

Table of content

Part V Commercial data development 61

Part I

Overview

EasternGraphics
visualize your business

1 Overview

1.1 System requirements

The system requirements of pCon.creator are described in a separated document "System requriements". These documents can be found for download beside each new pCon.creator in pCon Solutions Download Center.

1.2 Installation

pCon.creator is distributed as Setup. The Setup based installation automatically verifies the target system and required system components. Missing system components will be automatically installed if necessary.

Further details about installation of pCon.creator and activation of licenses can be found in the additional product document "pCon.creator - Installation guide". This document is available for download for each pCon.creator version in pCon Solutions Download Center.

 If many data developers are working together in a pCon.creator based OFML project, all data developers should use the same version of pCon.creator. A new version of pCon.creator may introduce an upgrade to the workspaces. After upgrading a database of a workspace it can't be opened in older pCon.creator versions. A database upgrade is always announced in the product documents of each pCon.creator version. These documents are available for download for each pCon.creator version in pCon Solutions Download Center.

1.3 License

Application License

pCon.creator can only be used having a valid licence[31]. The application is activated by a license from pCon.login[31].

Manufacturer Activation

Manufacturer codes are activated separately. Manufacturer codes which are unknown or not activated cannot be used to edit or export OFML data in pCon.creator. The activation of manufacturer codes is part of the application license from pCon.login[31].

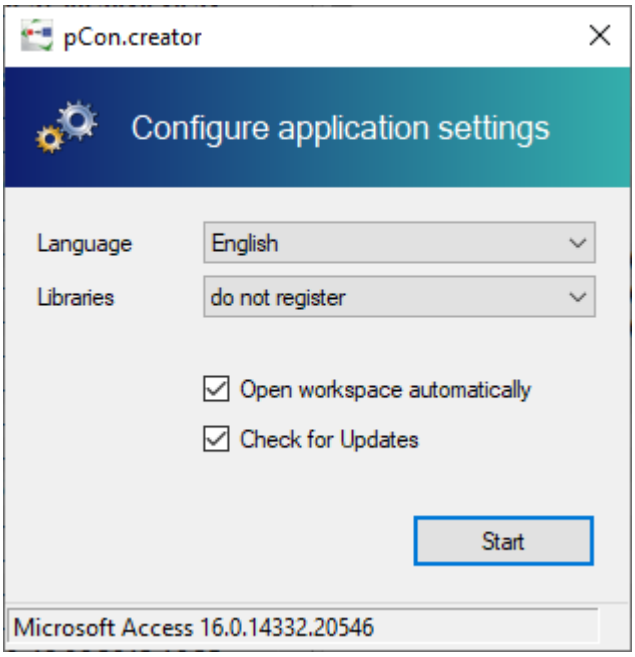
pCon.login was integrated into pCon.creator 2.20. The legacy Hardlock Dongle based licensing mechanism[31] is currently still supported.

Further notes about installation and activation of pCon.creator can be found in a separate document "pCon.creator - Installation Guide". This document can be downloaded for each pCon.creator version from pCon Solutions Download Center.

1.4 Settings

pCon.creator can be configured on application start. The configuration dialog will be shown, if it is started with command line argument config.

creator.exe /config



Select language

One of the installed languages of pCon.creator can be set in this field. This field is deactivated if only one language has been installed. The selected language remains active for every start.

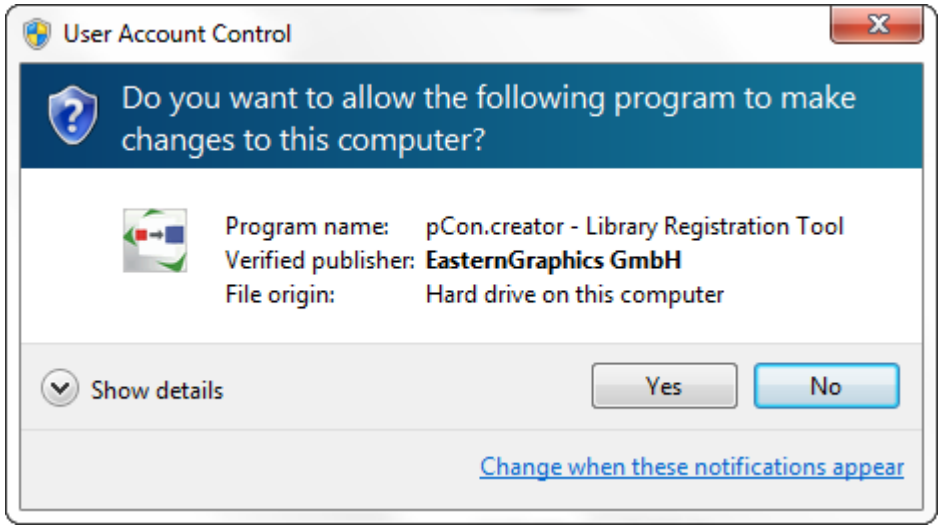
Register system libraries

pCon.creator uses several system libraries of Microsoft Windows and Microsoft Access. Normally they are registered correctly by the installers of the operation system or the Office application. Nevertheless, sometimes these libraries are not registered correctly. Using this option these libraries will be re-registered while start of pCon.creator.

One of the following options can be selected:

- do not register
- register
- register on every start

Registering system libraries requires administrator rights. Administrative rights are acquired automatically. Windows Vista / 7 or higher will prompt a user account control dialog. This dialog as to be confirmed with yes. If no administrative user rights are available errors will occur and the system libraries cannot be registered.

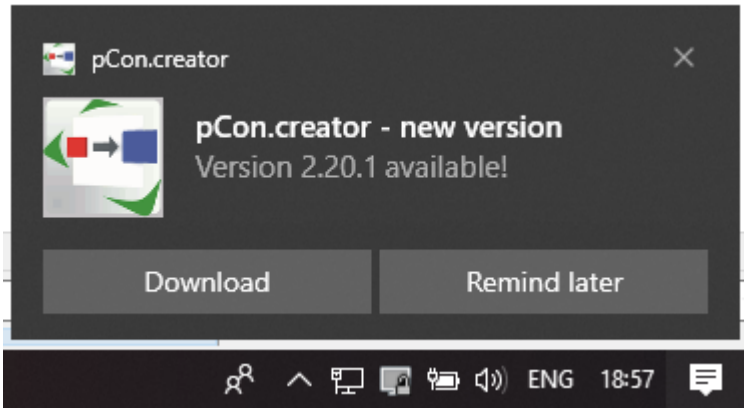


Open workspace automatically

pCon.creator remembers the last used workspace and opens it automatically on next application start. Data developers can switch off this behavior e.g. if they are working with many workspaces in different projects. In this case they have to open a workspace 39 after each application start.

Check for updates

pCon.creator checks automatically in pCon Solutions Download Center if a new application version is available for download. A notification will be shown if an update is available. The automatic check for updates on each application start can be disabled.



pCon.creator will not be automatically updated. The notification provides a "Download" command button. This command opens pCon Solutions Download Center in default web browser. The new application version can be downloaded from this website. Additional product documents can be found on this website, too. These documents contain additional information about the release, new features and changes.

The notification about an application updated can be dismissed using the "Remind later" command button. This command suppress the check for updates on application start for a certain period of time (currently 4 weeks). Within this period pCon.creator does not check for updates or prompts any notification about the new version. The notification may just appear again after this period of time if pCon.creator was not updated in the meanwhile.

If many data developers are working together in a pCon.creator based OFML project, all data developers should use the same version of pCon.creator. After upgrading a database of a workspace it can't be opened in older pCon.creator versions. A database upgrade is always announced in the product documents of each pCon.creator version. These documents are available for download for each pCon.creator version in pCon Solutions Download Center.

1.5 Directory structures in data creation projects

Different directories are required in data creation projects to save pCon.creator workspaces[29] and export the OFML data. In general it is not restricted, which directories are used. They can be chosen freely depending on the particular requirements of the data creation project.

Nevertheless it is important to use a unqiue directory structure throughout the project. Starting a new data creation project we would like to recommend to adopt the following directory structure:

```
C:\ofml_development
      \data
      \release
      \update
      \workspaces
            \workspace 1
            ..
            \workspace n
```

In this example the base folder `C:\ofml_development\` is chosen as a unique place for the data creation environment.

The applications which are used to create and test the OFML data are installed to the subfolder `\_bin\ .`

The OFML data is exported from the different data creation modules[26] to the central sub folder `\data\` . This directory contains the so called OFML-Source dataset[294].

The sub folders `\release` and `\update` are used in the data release process using the ORG-Module[26] . The source cleaned data OFML-Release datenset[299], which is used for final data tests, will be save to `\release`. OFML data installation packages, which can be distributed to end users or dealers, are saved in sub folder `\update`. These installation packages can be transferred to pCon.update.

Finally the workspaces are saved in the sub folder `\workspaces` . In this directory for each workspace a separate sub folder is created. The names of these folders should identify the workspaces simply and uniquely.

For instance following naming convention can be useful to give the workspace folders an identifying name:

```
wsp_<ManufacturerCode>[_<Suffix>]
```

The suffix is optional in case all product lines are maintained within one workspace. If different workspaces are used e.g. one workspace per product line the suffix can be the OFML code of the product line:

```
z.B. wsp_xyx_democabinets
```

An important aspect of this directory structure is a strict separation of the different data sets within a data creation project. On the one side it is the separation of the OFML-Source dataset and OFML-Release dataset, and on the other side the separation from runtime datasets which may be installed by pCon.update - DataClients is important.

Part II

User interface

EasternGraphics
visualize your business

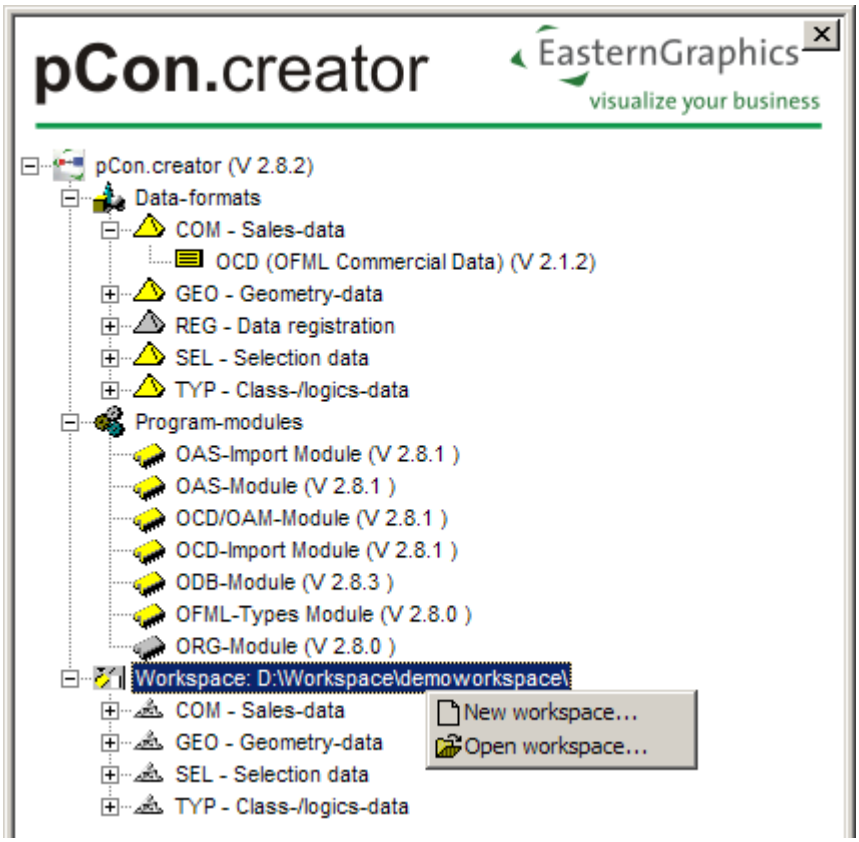
2 User interface

The main window of pCon.creator is called pCon.creator Explorer.

It contains a tree structure with the following main sections:

- Data formats 26
- Program modules 28
- Workspace 29

Each node in this tree has got a context menu, which provides several functions.



2.1 Data formats

In diesem Bereich werden die zur Datenanlage zur Verfügung stehenden Formate aufgelistet und können dem aktuellen Arbeitsbereich hinzugefügt bzw. aus diesem entfernt werden.

Die Datenformate sind nach Domänen sortiert:

- Kaufmännische Daten
- Geometrische Daten
- Auswahldaten
- Klassen- / Logikdaten

Der Zustand einer Daten-Domäne und eines Daten-Formates wird durch die Symbole gekennzeichnet.

Symbole von Datendomänen

	Die Datendomäne steht zur Verfügung, wird im aktuellen Arbeitsbereich jedoch nicht verwendet.
	Die Datendomäne wird im aktuellen Arbeitsbereich verwendet.

Symbole von Datenformaten

	Das Datenformat steht zur Verfügung, wird jedoch im aktuellen Arbeitsbereich nicht verwendet.
	Das Datenformat wird im aktuellen Arbeitsbereich verwendet.

Verfügbare Module für Datenformate

OCD/OAM-Modul	OFML-Commercial Data (OCD) zur kaufmännischen Datenanlage.
ODB-Modul	OFML-Database (ODB) zur geometrische Datenanlage
OAS-Modul	OFML-Article selection zur Katalogdatenanlage
CLS-Modul	Verwaltung programmierter OFML-Typen
ORG-Modul	Verwaltung von Registrierungsdaten und Erstellung quellenbereinigter OFML-Daten

----------OLD_TEXT----------

This section contains the available data formats which can be used for data entry in current workspace.

The data formats are grouped in specific data domains:

- Sales Data
- Geometry Data
- Selection Data
- Class / Logics Data

The status of a data domain or a data format is indicated with the following symbols.

Symbols of data domains

	The data domain is available, but not used in current workspace.
	The data domain is used in current workspace.

Symbols of data formats

	The data format is available, but not used in current workspace.
	The data format is used in current workspace.

Available modules for data formats

OCD/OAM Module 63	OFML Commercial Data (OCD) for data entry of commercial data and its mapping to geometries.
ODB Module 223	OFML Database (ODB) for data entry of geometrical data.
OAS Module 163	OFML Article selection for data entry of catalogs
CLS Module 347	Management of predefined OFML types
ORG Module 287	Management of registration data and release of source cleared OFML data

2.2 Program modules

pCon.creator is extensible using different modules. The available modules are listed in this section.

Symbols

The status of a module is shown by its symbol:

	The module is available but it is not used in the current workspace.
	The module is used in current workspace.
	The module is currently opened.

Available program modules

OCD Import Module 355	This module provides a function to import data in the OCD and XOCD format into the current workspace.
OAS Import Module 417	This module provides a function to import XCF catalogs into the current workspace.

2.3 Workspace

The workspace is quite similar to a project file. It is the starting point for data entry.

Within the workspace data of the different formats can be maintained. Each format has a separate section. The data entry dialogs of each format can be accessed using the context menus of its different data access levels.

Data Access Levels

 Manufacturer

 Distribution Region

 Product Line

 A valid license 31 is required to edit or export data of a specific data format. If this license is not available the data can only be accessed in read only mode. In this case this branch is highlighted in Grey in pCon.creator explorer.

Each manufacturer code must be activated by a license. If this manufacturer activation is missing or the manufacturer code is unknown the manufacturer is disabled. A red cross appears as overlay on the disabled manufacturer node. Without the manufacturer activation it is neither possible to edit nor export the data with this manufacturer code.

2.4 Datasheets

The data records can be edited in tables the so called data sheets. The data sheets provide the standard Microsoft Access functions in these tabular views (e.g. filtering records).

Furthermore they provide central datasheet functions 53.

Obligatory fields

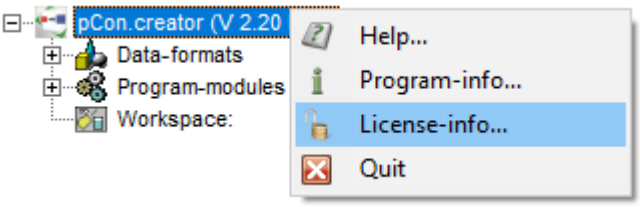
pCon.creator highlights obligatory fields in datasheets in green. A record can only be saved if all obligatory fields contain a value.

Optional fields

In pCon.creator datasheets, optional fields are highlighted in grey. To save a record, these fields can be empty.

2.5 License information

pCon.creator can only be used having a valid licence [31]. The application is activated by a license from pCon.login .



The license information dialog shows details about the currently used license.

Login

This command starts the process to login into pCon.login. The default web browser is opened. Please follow the instructions in the pCon.login web form to login using your pCon.login credentials (email address and password) and return to pCon.creator. pCon.creator will obtain a license automatically as long as any license was assigned to your pCon.login account.

pCon.creator is locked while login process is not finished, yet. It is possible to cancel this operation.

If pCon.creator is closed without logging out from pCon.login pCon.creator will remain activated. It is not essentially necessary to login again each time pCon.creator is started. Only after a longer absence you may be asked to login again.

Logout

This command logs pCon.creator out from pCon.login. The application license is returned to pCon.login.

If any unexpected problem occurs activating pCon.creator and obtaining a license, this problem may be solved by simply logout and login again.

Read Only Mode

The license information dialog appears automatically when pCon.creator is started without any valid license. It is possible to close this dialog without obtaining a license logging in to pCon.login. In this case pCon.creator runs in read only mode. It is possible to view data, but is not possible to modify, export or import OFML data.

Hardlock Dongle Mode

pCon.login was integrated into pCon.creator 2.20. Anyway the legacy Harlock Dongle based licensing mechanism is still supported. If any Hardlock Dongle is attached it will be used automatically when pCon.creator is started. In this case it is impossible to use pCon.login. Detach the Hardlock Dongle first.

Internet access

Accessing pCon.login https://login.pcon-solutions.com over internet is necessary to work with pCon.creator.

Short internet connection interrupts are tolerated, but in case of longer periods of interruption pCon.creator will switch automatically into read only mode. Nevertheless time consuming operations like imports or exports of data will be completed anyway.

 Further notes about installation and activation of pCon.creator can be found in a separate document "pCon.creator - Installation Guide". This document can be downloaded for each pCon.creator version from pCon Solutions Download Center.

 If you don't have any pCon.login account or cannot obtain any license for pCon.creator from pCon.login, please contact the administrator of your organization in pCon.login or your account manager or EasternGraphics Support .

Part

Main functions

EasternGraphics

visualize your business

3 Main functions

Der pCon.creator stellt im Explorer folgende Hauptfunktionaliäten zur Verfügung, welche in den weiteren Abschnitten näher erläutert werden:

- Erstellen eines neuen Arbeitsbereiches 37
- Öffnen eines bestehenden Arbeitsbereiches 39
- Datenformate zu einem Arbeitsbereich hinzufügen 41
- Datenformate aus einem Arbeitsbereich entfernen 42
- Einen Hersteller in einem Datenformat hinzufügen 43
- Einen Hersteller aus einem Datenformat entfernen 44
- Daten importieren 46
- Arbeitsbereich komprimieren 47
- Arbeitsbereich sichern 48
- Einstellungen 50

Darüber hinaus werden die Funktionen durch die installierten Module spezifisch erweitert.

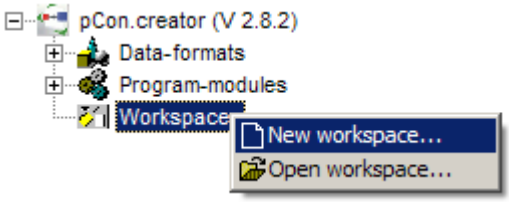
----------OLD_TEXT----------

pCon.creator provides the following main functions in the Explorer, which are explained in more detail in the further paragraphs:

- Creating a new workspace 37
- Opening an existing workspace 39
- Adding data formats to a workspace 41
- Removing data formats from a workspace 42
- Adding a manufacturer in a data format 43
- Removing a manufacturer from a data format 44
- Import data 46
- Compress current workspace 47
- Backup current workspace 48

Furthermore, the functions are specifically extended by the modules installed.

3.1 New workspace

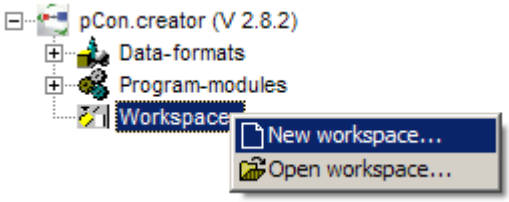


Nach Auswahl der Funktion "Arbeitsbereich neu .." des Arbeitsbereichsknotens im Explorer und der Auswahl der Codepage[38] öffnet sich ein Dialog zur Auswahl des Verzeichnisses, in welchem die Datenbanken des zu erstellenden Arbeitsbereichs gespeichert werden.

Im ausgewählten Verzeichnis wird die Datenbank pcr_workspace.mdb als zentrale Datenbank des Arbeitsbereichs erstellt.

Das Verzeichnis in dem der Arbeitsbereich gespeichert wird, dient zur Identifikation des Arbeitsbereiches. Innerhalb eines Datenanlageprojektes sollte daher ein einheitliches Schema für Arbeitsbereichsnamen verwendet werden. Dies erleichtert die Kommunikation und Identifikation der Arbeitsbereiche innerhalb eines Projektes (vgl. Sichern von Arbeitsbereichen[48])

----------OLD_TEXT----------



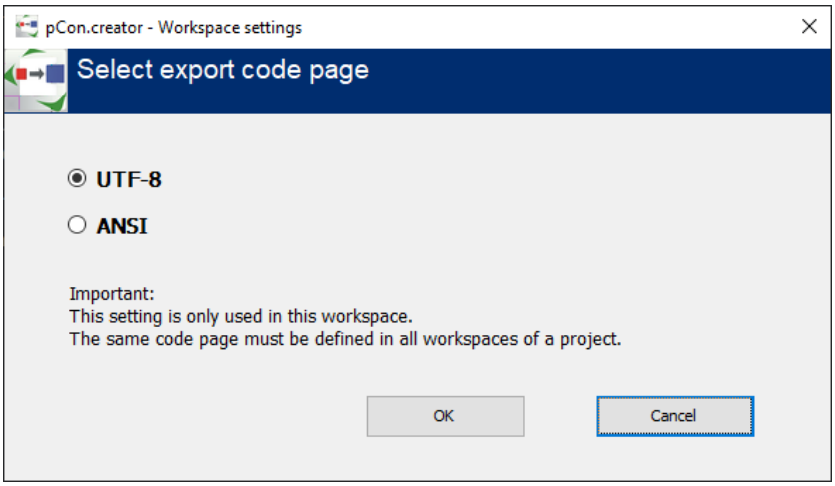
After having selected the function "New workspace .." of the workspace node in the Explorer, a dialog opens where you can select the directory to store the databases of the workspace to be created.

The database pcr_workspace.mdb is created as central database of the workspace in the selected directory.

The name of the folder which contains a workspace identifies the workspace. Within a data creation project a unique naming conventions should be used to identify the workspaces. This simplifies the communication and identification within the project. (see also Backup a workspace)[48]

3.1.1 Select code page

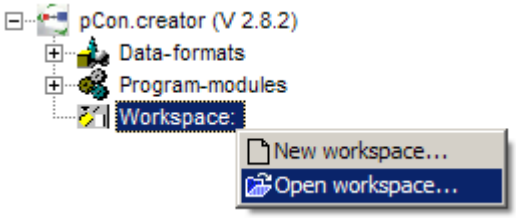
When opening a workspace for the first time with pCon.creator version 2.21 or higher, the data creator is asked to select an export code page. There is a choice between ANSI and UTF-8. As soon as a selection has been made, the data is exported in the corresponding code page.



It is important to ensure that all workspaces belonging to the project and other external data (metatypes, OAP, etc.) also have the corresponding code page. For projects in which, for example, separate workspaces are used for the catalog data or the registration files, care must be taken to ensure that all these workspaces use the same code page.

Switching the workspaces to UTF-8 as the code page must be a deliberate project decision. This conversion is particularly advantageous for data that works with characters outside the ANSI character set. This can quickly be the case if languages beyond the Western European character set are to be processed. Furthermore, the switch to UTF-8 represents a future-proof solution for processing OFML data in end applications.

3.2 Open workspace



Nach Auswahl der Funktion "Arbeitsbereich öffnen .." des Arbeitsbereichsknoten im Explorer öffnet sich ein Dialog zur Auswahl der zu öffnenden Datenbank pcr_workspace.mdb.

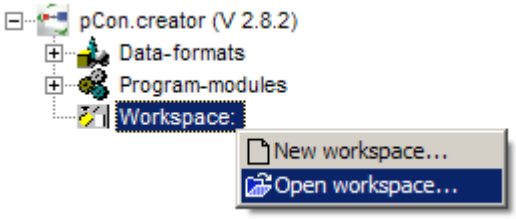
Beim Öffnen eines Arbeitsbereichs, bei dem noch keine Entscheidung über die Export Codepage getroffen wurde, öffnet sich der Dialog zur Auswahl der Codepage.

Beim Öffnen eines Arbeitsbereichs werden Konsistenzprüfungen durchgeführt. Enthält der Arbeitsbereich Datenformate, für welche kein geeignetes Modul in ihrer pCon.creator Installation zur Verfügung steht, können diese Daten nicht bearbeitet werden. Dies wird mit einem Hinweis angezeigt.

Des Weiteren wird die Version jedes Datenformats im zu öffnenden Arbeitsbereich überprüft. Liegt eine ältere Version der Datenbank im Arbeitsbereich vor, so kann ein Upgrade des Arbeitsbereichs durchgeführt werden. Es ist hierbei zu empfehlen zunächst eine Sicherungskopie des Arbeitsbereichs anzufertigen. Zur Sicherheit legt der pCon.creator eine gezippte Sicherungskopie von jeder Datenbank im Unterverzeichnis: *_dbupgrade_backup* des Arbeitsbereichs ab, bevor die Datenbank aktualisiert wird. Im Falle eines unerwarteten Fehlers beim Datenbankupgrade hilft dieses gezippte Backup im Rahmen eines Supportfalls das Problem zur prüfen und beheben. Das ZIP Archiv enthält sowohl die gesicherte Datenbank vor dem Upgrade als auch die Protokolldatei mit Fehlermeldungen des Upgrades. Nachdem der Arbeitsbereich erfolgreich aktualisiert wurde ist es nicht mehr notwendig dies Sicherung im *_dbupgrade_backup* Verzeichnis dauerhaft aufzubewahren. Diese Verzeichnis kann dann bereinigt werden. Das *_dbupgrade_backup* Verzeichnis dient nicht dazu, reguläre Backups des Arbeitsbereichs zu ersetzen.

Wenn mehrere verschiedene Datenanleger in einem pCon.creator basierten OFML Projekt zusammenarbeiten ist darauf zu achten, dass auf allen Arbeitsplätzen dieselbe pCon.creator Version verwendet wird. Nachdem ein Arbeitsbereich aktualisiert wurde, kann er nicht mehr mit einer älteren pCon.creator Version geöffnet werden. Ein Datenbankupgrade wird immer im Rahmen der Produktunterlagen mit neuen pCon.creator Versionen angekündigt. Die aktuellen Dokumente liegen jeder neuen pCon.creator Version im pCon Solutions Download Center bei.

----------OLD_TEXT----------



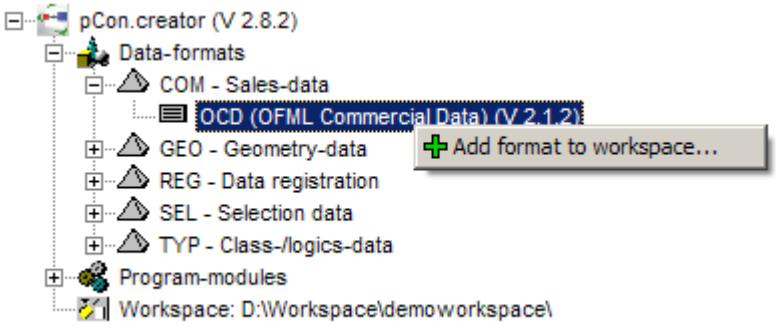
After having selected the function "Open workspace .." of the workspace node in the Explorer, a dialog opens where you can select the database pcr_workspace.mdb to be opened.

 When opening a workspace, consistency controls are performed. If the workspace contains data formats for which your pCon.creator installation does not provide an adequate module, these data cannot be processed. In this case, a note will be displayed.

 Furthermore, the version of each database in the workspace is checked opening the workspace. If the workspace contains an older data format version, you can perform an upgrade of the workspace. First of all it is recommended to have a backup-copy of the workspace before upgrading. Anyway before upgrading and modifying a database pCon.creator creates a zipped backup copy of the particular database and saves it in the workspace sub folder *_dbupgrade_backup.* If an unexpected error occurs upgrading the database, this backup ZIP file in *_dbupgrade_backup* folder will be useful for Support to investigate and solve the problem. The ZIP archive contains the database before the upgrade and the log file with error messages. After the a workspace upgrade successfully finished it is not necessary to keep the *_dbupgrade_backup* folder for a longer time. It can be cleaned up after a certain time. The *_dbupgrade_backup* folder does not intend to replace regular backups of the workspace.

 If many data developers are working together in a pCon.creator based OFML project, all data developers should use the same version of pCon.creator. After upgrading a database of a workspace it can't be opened in older pCon.creator versions. A database upgrade is always announced in the product documents of each pCon.creator version. These documents are available for download for each pCon.creator version in pCon Solutions Download Center.

3.3 Add data format

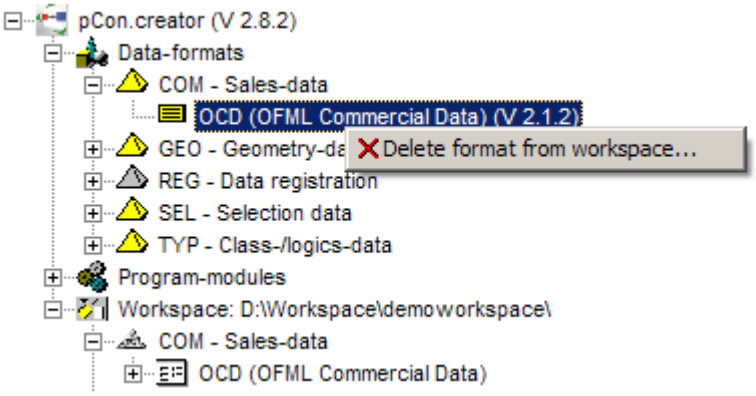


First of all you a data format has to be added to a new workspace before being able to create data therein. This function can be accessed in the area Data formats 26 in the Explorer.

This function is not available if data of this data format is already contained in the workspace.

A specific database is created in the workspace directory for each added data format.

3.4 Remove data format

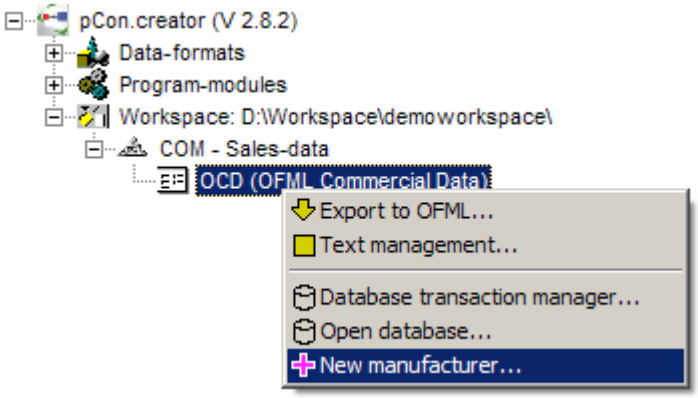


A data format can be removed from a workspace. You can access this function in the area Data formats 26 in the Explorer. Applying this function requires the confirmation of a security prompt.

This function will not be available, if the specific format is not contained in the workspace.

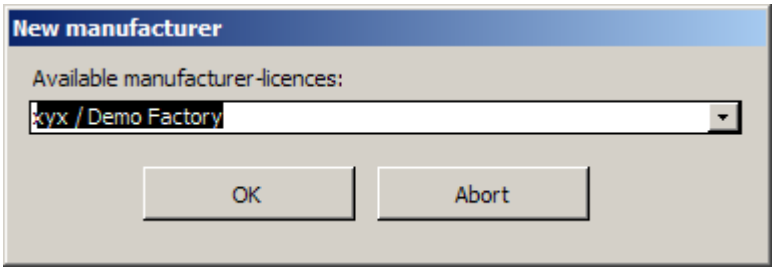
Data of this format is entirely removed from the workspace and cannot be restored.

3.5 Add manufacturer



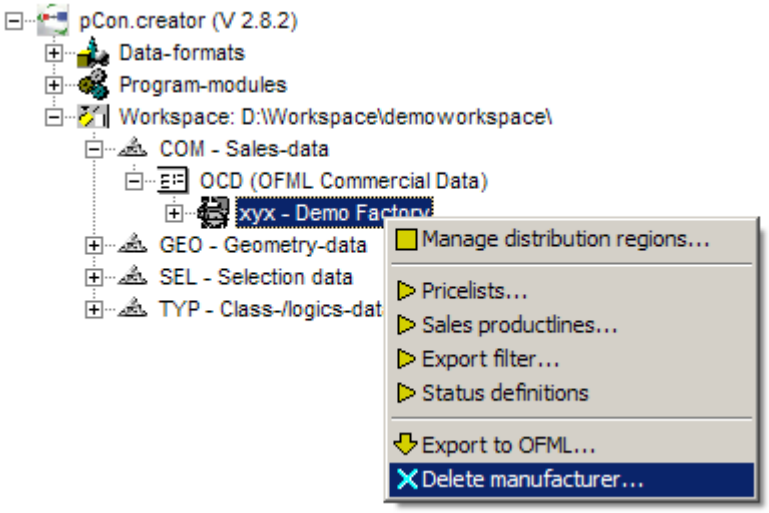
Data are stored manufacturer-specifically in the respective data format. A manufacturer is created in a data format with the function New manufacturer. You can access this function via the data format node in the workspace.

The registered manufacturer has to be selected in the following dialog.



Please contact our Support, if your company is already registered but does not appear in the manufacturers list,

3.6 Remove manufacturer



The respective function entirely removes data of a manufacturer from the data format of the workspace. This function can be accessed via the respective manufacturer node in the workspace.

3.7 Maintain and export data

Functions to enter and export data to the different formats are enabled by separate modules 26.

Commercial data

- OCD - OFML commercial data

Mapping data

- OAM - OFML article mapping

Graphical data

- ODB - OFML database

Catalogs

- OAS - OFML article selection

Registrierungsdaten

- ORG - OFML Registration

3.8 Import data

Function to import data are enabled by specific program modules 28

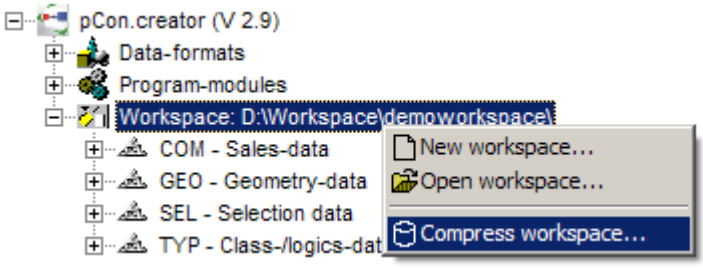
Commercial data

- XOCD - Extended OFML Commercial data 355
- OCD - OFML Commercial Data 355

Catalogs

- XCF- Extensible catalog format 417

3.9 Compress workspace

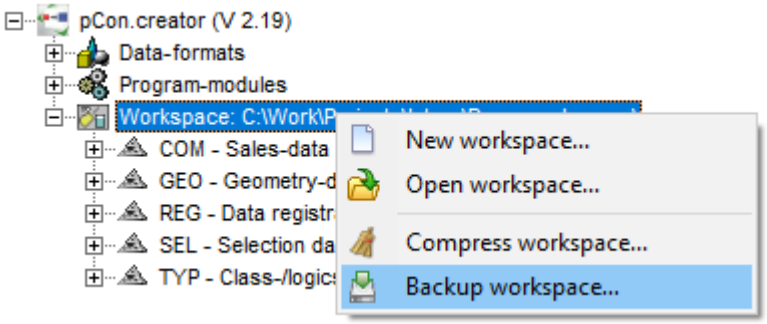


The function to compress the current workspace can be found in context menu of the workspace node in pCon.creator - Explorer. This command compresses each database of the workspace. Furthermore temporary import databases are deleted automatically.

To improve the data access performance internal statistics are always updated in the the opened databases. Furthermore deleting records does not automatically shrink the size of database files. For this reason the database size increases with the time and the saved data is fragmenting over time. This effect could have negative effects on the data access. To prevent this situation it is recommend to compress the workspaces regularly.

It is necessary to close any module and database to compress the workspace successfully. This is performed automatically after confirming this command. Additionally please verify, that any other application, which is accessing the databases, is already closed. You should cancel this command if there is still any long term process running, which accesses the data e.g. while a data import or transaction is still active. Compressing a workspace requires additional temporary space on the harddrive. Please ensure to have at least as much free space left as the largest database of the workspace requires to be saved. The progress of the compression command is written to log file p-cr.log. If errors occur for any reason please verify the compressed data and contact our Support.

3.10 Backup workspace



A backup should be created regularly for each workspace. pCon.creator Explorer contains a function to create a function to backup the current workspace. Using this function a dialog will appear to configure the following settings which are applied to the backup.

Fields

`Backup-Directory`	The folder where the backup is saved.
`File name suffix`	An optional suffix which is added to the backup file name. An additional suffix can be used to tag a backup for a specific purpose. Generally this field may be left empty.
`User name`	The user name which is saved in the change history for this backup. The default value is the user name of the user which is currently logged into the system.
`Comment`	An optional comment to describe the changes which have been made in the workspace. This field should be used in data projects to be able to trace changes.
`Filename`	The value in this field cannot be changed. It just shows the file name under which the backup will be saved.

Command buttons

`Cancel`	Cancels the backup and closes the dialog.
`Backup`	Starts the backup. In case it is finished successfully the dialog is closed automatically.

Backup file name

The backup of a workspace is saved as an ZIP archive. It can be restored easily with different ZIP packing applications.

The file name of the backup is build automatically depending on the name of the workspace, an optional suffix and a time stamp. The naming scheme of the backup file name is:

```
<WorkspaceName>[_<Suffix>]_<yyyymmdd>_<hhMMss>.zip
```

The name of the workspace is equal to the name of the folder where it is saved. The names of workspace folders should follow one unique convention throughout a data creation project to identify the workspaces easily.

Change history

While creating the backup a change history document is automatically saved. The data developer can specify his user name and a comment, to document the last changes. This information including the time of the backup and the current pCon.creator version a saved automatically in the change history document. This file can be viewed in any text editor.

This document is always saved in the workspace's resource folder:

```
\resource\history.txt
```

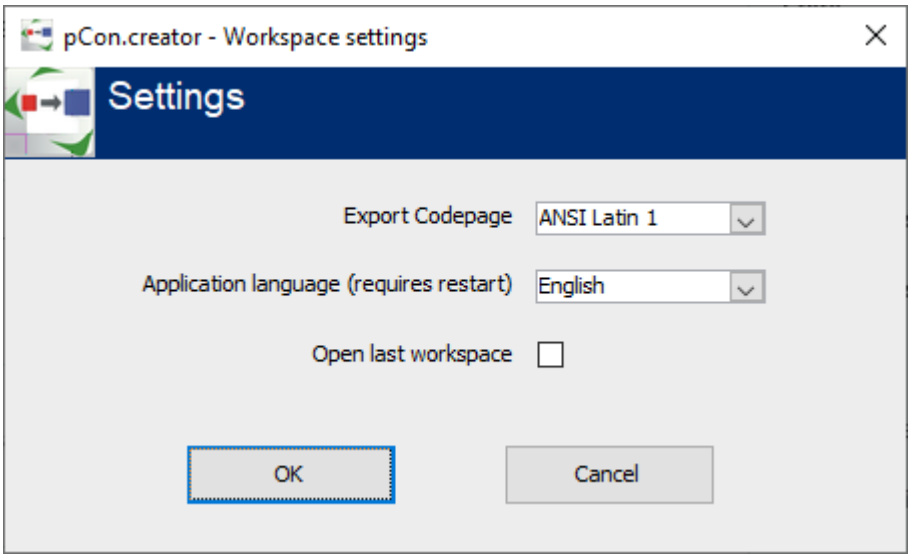
 All data formats and modules must be closed to create a backup. If any of them is still open a question dialog will appear to close them automatically before the backup starts. Users should ensure that there is no file of the workspace opened in any other application (e.g. in AutoCAD).

 For each workspace a lock is established while backup. This lock ensures that one and the some workspace can only be saved to backup directly by one user at the same time. If concurrent backups are started by two different users for one and the same workspace, the second users will receive an error message.

 The Backup-Directory and user name settings are saved for each user in the workspaces. They will be automatically reloaded when the next backup is started.

 If a backup file exceeds the files size of 2 gigabyte it is automatically split. This prevents possible problems on file systems which don't support such large files. Split backup files have the same file name. Besides the backup file with file extension ZIP additional files with the same name but an enumerated file extension are saved. All of these parts have to be available to restore the backup. Please also note split ZIP archives possibly cannot be extracted with all ZIP packing applications.

3.11 Settings



After opening a workspace, it is possible to make central settings for the workspace or the application via the context menu of the node.

The export code page 38 can be selected and always refers to the currently opened workspace. All other options in this dialog refer to the central settings of pCon.creator. Changing the application language requires pCon.creator to be restarted. To automatically open the last selected workspace when pCon.creator is started, the last option in this dialog must be checked.

Part IV

Datasheet functions

EasternGraphics
visualize your business

4 Datasheet functions

Enhanced Functions in pCon.creator

Data is displayed on many entry masks in so-called datasheets in form of a table providing a clear overview.

pCon.creator provides the following enhanced functions to efficiently work in datasheets:

- Looking up records 54
- Cloning records 55
- Reorganisation of sorted records 56
- Displaying additional information 57
- Saving and loading individual layouts 59
- Using user-defined filters 60

Standard Access Function

Furthermore, the Microsoft Access default functions can be used as well such as:

- Sorting
- Filtering datasets
- Copying and pasting datasets
- Copying and pasting data in individual columns

More detailed information on these functions can be found in the Microsoft Access documentation.

4.1 Look up records

Different fields can refer to data in other - so called foreign - tables. The data, allowed to be entered, can normally selected from a selection list.

E.g. the field "Article short text" of an article in OCD references a record defined on the "Text-Block" dialog

Opening a referenced record

A double-click into a field referencing another record automatically opens the form which allows you to edit this foreign record. The referenced record is selected to be edited further. If the option field was empty, the focus is on a new record.

The selection list of the original field is updated as soon as you switch back to it.

One record can be referenced from different locations. Modifications of this record influence all the references.

E.g. after modifying a language or the name a specific text-block for article short texts. This modification is automatically updated for all articles already using this text-block

4.2 Clone records

Cloning a record creates a deep copy of it. Records in other tables referenced by the cloned record are copied to the target, too.

Keyboard shortcuts

F8	The selected records are copied from a datasheet to the clipboard for cloning.
F9	Inserts records which have been copied to clipboard for cloning (F8) into a datasheet. This function creates deep clones of these records including the referenced records.
CTRL+SHIFT+C	see F8
CTRL+SHIFT+V	see F9

Example: Clone an article in OCD

Two articles *XY1* and *XY2* are selected in the article root of the OCD data format of product line *A* (source context) and taken over to the clipboard pressing F8. These records are pasted in the article master of product line *B* (target context) using F9.

The article *XY1* in series *A* refers to the property class *CLASS_1*. This class is already contained in series *B*. Cloning makes sure that the reference of article *XY1* in product line *B* refers on property class *CLASS_1* located there.

The article *XY2* refers to property class *CLASS_2*, which is not contained in product line *B*. Cloning also copies *CLASS_2* into product line *B* and the reference of article *XY2* to this one is reproduced in product line *B*.

Cloning within the same context

If source and target context of a record to be cloned are identical, the newly generated record will be renamed. The value in the field identifying it obtains a suffix. This is, for instance, the case, when a relation object is cloned within one OCD product line.

Cloning between different contexts

If source and target context are different, you will be prompted to decide how to handle identical datasets to be cloned. You have the options of

- only adding new records or
- overwriting existing records in the target context with the records from source

This is, for instance, the case when this operation is used to copy an article from one OCD product line into another. If the record already exists and has to be overwritten only this record is affected. Records which are related to this one are not updated. This means child records of the existing record are not appended or updated.

4.3 Reorganization of sorted records

Records for which a specific order has been defined contain a field specifying its relative position within the order.

A double-click on this field effects an automatic reorganization observing the relative order. New position numbers are being allocated to the records. The numbers start with an offset value and are incremented in certain interval. The interval is usually bigger than one. The resulting gaps between the position numbers allocated allow a fast resorting.

Example: Properties in the OCD data format

Position	***Name***
100	HEIGHT
110	DEPTH
120	**WIDTH**

The property is supposed to appear on the second place instead on the third place. Therefore, its relative position is adapted:

Position	***Name***
100	HEIGHT
110	DEPTH
105	**WIDTH**

A double-click on the field "Position" immediately restores the sorting.

Position	***Name***
100	HEIGHT
110	**WIDTH**
120	DEPTH

4.4 Display additional information

In datasheets you can show or hide columns with additional information. Switching the visibility of these columns affects all datasheets of the open module. The following keyboard shortcuts allow you to display or hide the additional information.

Keyboard shortcuts

F10	Toggle the visibility of the database record id (#ID) in all datasheets.
SHIFT+F10	Toggle the visibility of the source or temporary record id (#SrcID in all datasheets after a transaction.
F11	Toggle the visibility of the internal transaction status (#TAG) in all datasheets.
F12	Toggle the visibility of the user defined status in all datasheets.

Database ID of records

The column #ID shows the database ID of the records. The database IDs are unique database internal keys to identify the records. These IDs are internally used for references between records.

Source or temporary ID after database transactions

The column #SrcID shows the source or temporary ID after database transactions. This information allows you to follow up and verify database transactions. The content of this field is not static and vary with different actions.

Transaction tag

The column #TAG shows information about the dataset, which is stored as temporary state, e.g. through database transactions. This information allows you to follow up and verify database transactions. The content of this field is not static and can vary with different actions.

User-defined status

The status column allows you to allocate a user-defined status information for a dataset.

In data formats supporting user-defined status information, the entry "status definition" can be found in the context menu of the respective manufacturer. The appearing entry mask allows you to enter any status definitions and to allocate one of the colours red, yellow, or green to them in order to distinguish them by colour.

Linking a status definition with a transaction code allows you to permanently map transaction states after a synchronization with the help of the transaction manager. While the fields #TAG and #SrcID are dynamic, the field "status" allows you to permanently display the final state of a comparison of data or a synchronization by means of a database transaction.

User-defined status definitions are mainly used in the OCD data format.

4.5 Individual layouts

It is possible to individually adjust visibility, width, and order of the individual columns. The current settings of a datasheet's view can be saved and restored. A stored layout is thus being preserved when pCon.creator is launched again.

Keyboard shortcuts

F7	Save the current layout of a datasheet.
SHIFT+F7	Delete a saved datasheet layout.
F6	Restore a previously saved datasheet layout.
CTRL+S	see F7
CTRL+SHIFT+R	see SHIFT+F7
CTRL+R	see F6

4.6 User-defined filters

Different entry masks allow you to create user-defined filters for datasheets. A double-click on the selection field above the datasheet opens an entry mask to edit the filter definitions.

Selecting one of the filter definitions created applies it on the datasheet. An active filter is highlighted by a red filter symbol.

Fields

Inverse	The filter condition is negated when applying it.
Condition	A pattern according to which the filtering is effected. You have the possibility of using wildcards: ? - any character * - any number of characters
Name	A unique filter name according to which it can be selected

The filter definitions for the current form are context-specifically stored in the workspace. They can be re-used, in case the form is opened again at a later point in time.

Examle: Article master in OCD

Artikel

Y ConferenceTables

	Status	Artikelnummer	Kfm. S	Typ
		A1000	S2	Konfigurierbar
		A100R	S2	Konfigurierbar
▶		A2010	S2	Konfigurierbar
*			S2	Konfigurierbar

In this example, a user-defined filter with the name `ConferenceTables` was applied to the article master in the OCD data format.

The filter condition is the following:

```
A?0*
```

and includes all articles whose article number starts with A and contains a 0 at the third position.

Part V

Commercial data development

EasternGraphics
visualize your business

5 Commercial data development

5.1 Overview

The OCD module enables data entry of sales data in the OFML Commercial Data (OCD) format. Furthermore, commercial articles can be assigned to geometric representations using the data format OFML Article Mapping (OAM) provided by this module as well. This module supports the creation of consistent data in the OCD 4.0 format by using different verification routines. Export filters and derived distribution regions provide a powerful mechanism to efficiently generate data for different markets.

5.1.1 System requirements

The general system requirements of pCon.creators have to be fulfilled.

Microsoft Office

Microsoft Excel must be installed to be able to maintain value combination tables 139.

5.2 Main functions

5.2.1 Management of commercial data

The OCD module provides functions and dialogs to enter and maintain product data in data format OFML commercial data OCD.

Before entering the first article[86] in a new workspace[37] a new product line can be created. Following steps have to be performed to do this:

- Add a new manufacturer[43]
- Configure the used languages[67]
- Create a price list[70]
- Create a distribution region[79]
- Create the sales groups[78]
- Create the product line[80]

Afterwards the dialogs to maintain product data can be accessed from the context menu of the product line from pCon.creator - Explorer.

To use the data in an OFML runtime application (e.g. pCon.configurator/planner) it has to be exported[151].

5.2.2 Management of mapping data

The OCD module provides functions and dialogs to enter and maintain mapping data in data format OFML Article mapping (OAM).

Mappingdata[154] realizes the connection of sales articles with parameterized graphics (ODB) and the catalog (OAS).

Export of mapping data is done automatically with export of commercial data[151].

In early stages of data creation projects the maintenance of mapping data may be skipped, as long graphics are not available yet. If not maintained a standard mapping is exported automatically. In this case articles are visualized just for configuration purposes as a grey cube in OFML runtime applications.

5.2.3 Generation of market specific data

The OCD module provides functionalty to maintain one central dataset from which different OFML datasets with specific pricelists for different markets or user groups can be generated.

For this purpose the data is maintained in one central master distribution region 79 (e.g. ANY). For each market additional distribution regions are created which derive 79 from this master distribution region. A specific active pricelist can be assigned to each distribution region. Additionally export filters 148 can be assigned, which are applied while export to OFML:

Following items can be filtered separately:

- Product lines 80
- Articles 86
- Property values 105
- Article properties 90

In derived distribution region it is not allowed to maintain product lines. Furthermore it is not possible to derive a distribution region from another as long as it contains product lines.

Please note when structuring the data creation projects, that one product line 80 of only one distribution region can be loaded into an OFML application at runtime at the same time.

5.2.4 Update of price lists

The pricelist management dialog 70 provides additional functions to simplify price list updates.

Using the price list generation 71 function it is possible to create a new price list from an existing one by application additional rules how prices are calculated based on the old price list.

Additionally it is possible to export and import new price lists 77.

Afterwards a new price list was generated or imported it just has to be set active for one or more distribution regions 79.

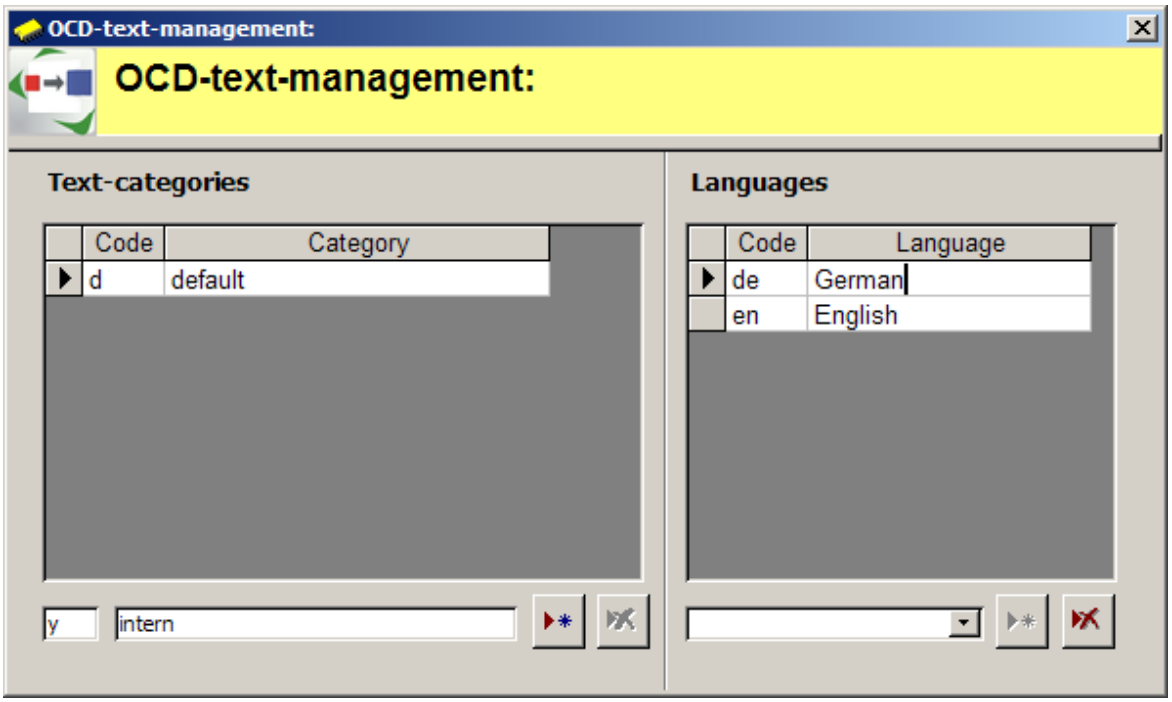
5.2.5 Export and import of texts

OCD data contains texts in different languages. The data creation can be embedded into external translation processes. For this purpose the text block dialog[95] provides functions to export[96] and reimport[97] the text blocks containing the language specific texts.

Currently just a specific excel data format[98] is supported.

5.3 OCD user interface

5.3.1 Dialog Text management



This data entry form manages the text categories [68] and languages [69] in the current workspace.

Several text categories can be created, which can contain several different languages. The categories are entered in the left area, while the languages of a selected category can be edited in the right area of the form.

It is necessary to create at least one text category with at least one language.

It is important to use the same text categories in all workspaces in a distributed data created project. Therewith it is possible to use several functions for data transfer and synchronization easier.

5.3.1.1 Text categories

Text categories are used to distinguish the use of texts in distribution regions. It is thus possible, for instance, to create texts describing the catalog entries in the specialized trade as well as texts intended for an internal use only. When defining derived distribution regions, the text categories therein valid can be determined.

Fields

Code	A unique one-digit code to identify the category is indicated.
Category	The name of the respective category is entered here.

Buttons

	Add the new text category
	Delete the selected text category

A text category can only be deleted when it does not contain any language.

Text categories, which do not contain languages, are automatically removed when exiting the text management. A message will be displayed.

5.3.1.2 Languages

A text category can contain several different languages. They can be entered by means of the two-digit ISO-639 language abbreviation or selected from a selection list.

Fields

Code	A unique two-digit code according to the standard ISO 639 is indicated to identify the language.
Language	This field shows the name of the language. The language cannot be edited but directly depends on the language code.

Buttons

	Add the new language to the selected text category.
	Delete the selected language

5.3.2 Dialog Price lists

This dialog centrally manages the price lists of a manufacturer. The prices themselves are captured within the respective product line.

Fields

Label	Name of the price list
Currency	This field determines the currency in which the prices in this price list are indicated. It contains a selection list of valid currencies (e.g. EUR, CHF, USD, RUB, …)
Type	The type of the price list is determined here. A price list can be of the following type.
Valid from	This field defines the date of the first day from which on the price list is valid.
Valid to	This field defines the date of the last day until which the price list is still valid.
Multiple Scopes	If this field is activated it is possible to have multiple currency / dates inside the price list. If this field is deactivated the price list is restricted to one currency / date. This simplify the handling in context of multiple price lists.

Price types

The prices are distinguished into the following types.

sales	Sales prices
purchase	Purchase prices

Commands

	This command button provides a context menu containing additional commands to update price lists: - The generation of a new price list 71 on the basis of an already existing price list is launched. - Export and Import 77 of price lists

The functions to update price lists are disabled, if no price lists have yet been created.

It is recommend to not restrict the temporal validity of the prices within the data. It should be better organized by distribution of the data.

The price list's options (currency, type or validity date) are just default values which are applied while entering new price components. Modifications to these fields value are not applied automatically to all of the price lists pricing components. The price list generation command[71] can be used to create new price lists instead.

Export

The section export is used to define into which distribution regions the selected price list is going to be exported. It is possible to define in which distribution regions the price list is the default price list. The default price list becomes preselected when a dialog to maintain prices is opened in the according distribution region (see dialogs Article master[87], Global prices)[142].

In Dialog Distribution regions[79] it is also possible to setup, which active price lists are exported.

5.3.2.1 Price list generation

A new price list[73] can be generated by applying a price list generation profile to an existing price list[72].

The profile contains calculation rules[74] to generate the new prices. These rules are separately determined for article prices[87] and global prices[142]. Each of these rules can be equipped with a filter to restrict the prices it is applied to.

Please note that only one calculation rule at most is executed to generate a price. Furthermore, only absolute prices are calculated. Relative upcharges or discounts are taken over from the source price list in an unmodified form.

The profile containing the calculation rules can be used in different workspaces. You therefore have the option to store it in an external file.

Buttons - Profile

	Loading a price list generation profile
	Saving the current price list generation profile
	Saving the current price list generation profile under a new name.
	Creating a new profile.

Settings: Source price list

An existing price list has to be selected as source price list. Prices in this source price will be used to created the new price list applying specific calculation rules[74] on them.

It is allowed in OCD to use prices in different scopes within one price list. Therefore it is recommended to use only one scope defined by the price list itself. This scope has to be activated for price list generation.

Fields

Name	The name of the source price list.

Fields - Scope

Active	Use prices in this scope for price list generation.
Currency	The currency of the new price list.
valid from	This field defines the date of the first day from which on the new price list is valid.
valid to	This field defines the date of the last day until which the new price list is still valid.

Any price of the activated scopes will be copied to the target price list. This operation applies the scope of the target price list to them.

It has to be considered that if multiple scopes of the source price list are activated and one price with a specific variant condition exists in more than one of these different scopes the price will occur multiple times in the target price list.

Settings: Target price list

Fields

Name	The name of the new price list.

Fields - Scope

Currency	The currency of the new price list.
valid from	This field defines the date of the first day from which on the new price list is valid.
valid to	This field defines the date of the last day until which the new price list is still valid.

Calculation rules

The calculation rules for price updates can be defined in the following fields. These rules are separately determined for article prices[87] and global prices[142]:

Fields - Price calculation

The price is calculated with the help of the following formula:

```
NewPrice = Round[(OldPrice + FirstSurcharge) * Multiplier] +
SecondSurcharge
```

This formula is usually parameterized with the following fields:

Prio.	This field determines the order of execution of the calculation rules.
1. Surcharge	Determines the first surcharge. This value is added to the old price prior to the rounding.
Multiplier	A factor with which the price is multiplied after having added the first surcharge.
Rounding	This field determines the rounding rule[75].
Precision	This value determines the precision used for the rounding[77].
2. Surcharge	A second surcharge added to the price subsequent to the rounding.

Fields - Filtering

A filter with different criteria to select the prices, to which the rule is applied, can be defined for each price calculation rule. The calculation is only applied to those prices which comply with all criteria selected. If a criterion has not been selected, it is not evaluated for the filtering.

OFML package	This criterion restricts the OFML packages, in which the rule is applied to prices. This value is a filter expression for the product line abbreviations.
COM group	A filter expression for the code of the sales product line of the articles.
Article code	A filter expression for the articles.
Price type	The type of the prices the rule is applied to. Valid values are: • sales • purchase
Prices from	Old prices, whose values are greater than or equal to the value defined in this field, are included in the calculation.
Prices to	Old prices, whose values are less than or equal to the value defined in this field, are included in the calculation.
related to result	The criteria "Prices from" / "Prices to" define the limits of the calculated unrounded new prices. If this criterion has not been selected, the values of these fields refer to the old price.
Variant condition	A filter expression for the variant condition of the prices to be calculated.

The placeholders ? (any character) and * (any number of characters) can be used in filter expressions.

Rounding rules

up	The values are rounded up. Negative values always get greater. Example 2.5 3 1.5 2 -1.5 -1 -2,5 -2
down	The values are rounded down. Negative values always get smaller. Example 2.5 2 1.5 1 -1.5 -2 -2.5 -3
commercial	Amounts from including .5 are always rounded up. Amounts until .5 are rounded down. Example 2.7 3 2.5 3 2.2 2 1.5 2 -1.5 -2 -2.2 -2 -2.5 -3 -2.7 -3
round to even	The amount of the value is rounded up from .5 .It is rounded down until .5 . The amount at .5 is rounded to the next even number. This rounding type is also known as "bankers rounding" Example 2.7 3 2.5 2 2.2 2 1.5 2

Examples: Rounding precision settings

A factor sets the rounding precision. The following examples illustrate the function of this factor.

1	Rounding to whole values (to whole euros)
0.1	Rounding to one place after the decimal point
0.01	Rounding to two places after the decimal point (exactly to the cent)
0.5	Rounding to half values (accurate to 50 cents)
0.25	Rounding to fourth values (accurate to 25 Cents)
0.05	Rounding to twentieth values (Rappen-rounding)
10	Rounding to the second place before the decimal point (to ten euros)
100	Rounding to the third place before the decimal point (to hundred euros)

5.3.2.2 Price list export and import

Price lists can be exported from pCon.creator to update them in external processes.

The prices are exported into a comma separated text file (CSV); one for each price list. The changed prices can be reimported from these text files in same format afterwards.

Using this function in price list update processes and the specification of the CSV data transfer format is described in detail on an additional document **Application Note AN-2021-01 pCon.creator – Price updates .**

Please ask your Account Manager at EasternGraphics or support if you do not have access to this document.

5.3.3 Dialog Sales product lines

The commercial product lines used in the workspace are managed in this entry mask.

 To distinguish commercial product lines and OCD product lines80, further explained below, the following has to be taken into account. An OCD product line is represented by an OFML-Package. It can include several commercial product lines. Usually, there is a 1:1 relationship among these two. However, to efficiently store data, it may be practical in individual particular cases to combine several commercial series in one OCD product line (OFML-Package).

Fields

Code	This field contains a unique two-digit abbreviation of the commercial product line.
Label	In this field, a label has to be indicated for the commercial product line.

5.3.4 Dialog Distribution regions

Distribution regions are a logical compilation of catalogs with specific validity. They usually map different markets.

Fields

OFML code	A unique code to identify the distribution region. It can be selected from a selection list of predefined codes for regions. Furthermore, the OFML code is used for registering the packages in the OFML application system.
Label	A name for the distribution region, which is used in pCon.creator for purposes of presentation, e.g. in the Explorer window.
Derived from	The distribution region from which the present one is derived to create market specific data 65.
Active price lists	The price lists 70 which are exported with this distribution region. One price list can be marked as default for the distribution region. The default price list becomes preselected when a dialog to maintain prices is opened (see dialogs Article master 87, Global prices) 142
Text category	The text category 67 valid in this distribution region is indicated here.
Default language	This value defines the default language for the distribution region.
Product line filter	Defines the filter 148 for product lines 80. The filter is evaluated during the export 151.
Article filter	Defines the filter 148 for articles 86. The filter is evaluated during the export 151.
Value filter	Defines the filter 148 for property values 105. The filter is evaluated during the export 151.
Article property filter	Defines the filter 148 for article properties 90. The filter is evaluated during the export 151.

The selected default language is used to preview texts in the other dialogs within the respective distribution region.

5.3.5 Dialog Product lines

5.3.5.1 Settings

Fields

OFML code	A unique product line abbreviation to identify the product line. The value in this field must correspond to the rules of a valid OFML identifier and has to be in small letters. Furthermore, the abbreviation must not contain an underscore.
Label	This field contains a label of the product line.
Sales code	This field determines the commercial product line 78, which is mapped by this OCD product line. This is the suggestion value for the articles to be created in this product line.
Version-Major	Shows the major version number of the product line. It starts with 1. Incrementing the major number means significant modifications such as adding or deleting complete product lines.
Version-Minor	Shows the minor version number of the product line. It starts with 0. Incrementing the minor number is used e.g. in case of removing or adding articles, properties, property values etc. A regular price list update requires a minor increment as well.
Version-Build	Shows the build number of the product line. It starts with 0. Incrementing the build number means a minor modification in terms of error eliminations (graphic, price etc.).
Release date	Date of the release of the product line of this product line version.
Remarks	Remarks about the product line can be stored here.
OCD format version	The version of the OCD format into which the export is performed.
Language for relation coding	The field determines the relation coding language, in which the code of the relation knowledge is stored.
Date valid from	This field defines the date of the first day from which on the product line is valid.
Date valid to	This field defines the date of the last day until which the product line is still valid.
Export filter	A character string (filter marking) which is evaluated when applying the product line filter of the distribution region to this product line.
OAM format version	The version of the OAM format, into which the article mapping data are exported.
Wildcard comparisons	Using this option the OCD 4.0 function for wildcard comparisons with the IN operator 111 can be enabled.
Variant condition variable	The optional value of this field defines a special variable to activate pricing components on OCD relations 125. The standard variable `$VARCOND` will be used if this field is empty. The variable name must be a unique language-independent OFML identifier. The name must only consist of alphanumeric characters and the underscore. The first character must not be a numeric character. Special characters and blank characters are not allowed. In this field the variable is specified without prefix $ but it in relations the prefix $ must be used. Using SAP relation code the prefix `$self.` must be omitted in this field accordingly. If specified this variable is also used in packaging relations 126 as variant condition variable.

OCD format versions

The stored data can be exported into one of the following OCD versions.

- 1.0.0
- 2.0.0
- 2.1.0
- 4.0.0
- 4.1.0
- 4.2.0
- 4.3.0 (Default)

When creating new OCD data, it is recommended to use 4.2.0, which is already preset. In case you use one of the previous versions, please note the limited range of functions of the respective version when creating the data. OCD 4.3.0 is only supported in OFML runtime applications (e.g. pCon.planner) since releases in autumn 2020.

Changing the OCD Format version also affects the DSR Registration file a product line. For this reason it is necessary to apply this modification in product line dialog of the ORG Module, too.

OAM format versions

- OAM 0.1.0
- OAM 1.0.0

When creating new data, it is recommended to use OAM 1.0.0. This is the preset value. The version OAM 0.1.0 is supported regarding data already created.

Relation coding languages

- OCD 1.0
- OCD 2.0
- OCD 4.0
- SAP (LO-VC)

OCD 4.0 is the default relation coding language.

Since OCD 4.3 the previously available SAP relational language sets ABAP SAP 3.1 and 4.6 are replaced by just SAP (LO-VC) (see OCD specification). The supported feature set of the SAP relational language set is still the same. So in pCon.creator only SAP (LO-VC) can be chosen. But when exporting to OCD 4.2 or older ABAP SAP (4.6) is exported to version information table ocd_version.csv . So the data will remain compatible for OFML runtime applications despite this pCon.creator internal difference.

The diligent incrementation of the fields of the version and the release date of the product line are absolutely necessary prior to each publication after having updated the data. On the basis of this information an installation routine for OFML libraries is able to decide whether a library needs to be updated.

5.3.5.2 Miscellaneous

Fields - Appearance in article lists

Article code	This field controls either to print a final article number or the base article number in article lists. Optionally the maximum length of the printed article code can be restricted. If this option is not selected the complete article code will be printed.
Display each pricing component	This field controls the appearance of individual pricing components. If it is active each pricing component is printed separately otherwise just the total sum of an article's active pricing components is shown.
Summarize identical articles	While creating the article list of a planned scene identical articles are summarized into one order position.

These central settings are part of the OFML Proginfo Table. They will not be exported if the export of the OFML proginfo table is deactivated. As default this export is active (see below).

Fields - Material package

In OAM material mapping it can be defined which properties and property values are graphical represented by materials. The following fields can be used to define a central OFML package which contains the material definitions.

Manufacturer	The OFML code of the manufacturer.
Product line	The OFML code of the package.

If the material package fields are left empty the material definitions are loaded from the current package which contains the commercial data. In general it is recommended to use one central base library e.g. "basics" to distribute material definitions.

This central setting is part of the OFML Proginfo Table. It will not be exported if the export of the OFML proginfo table is deactivated. As default this export is active (see below).

Data control tables

Export OFML Proginfo table	This option is active as default and ensures that the OFML proginfo table (proginfo.csv) will be exported. The button  opens another dialog to maintain additional generic settings of the OFML proginfo table (proginfo.csv). A detailed description of this table and the allowed settings is described in the application note "AN-2006-01 Data control tables" The proginfo table may be used to configure special settings to assign preview images in property value lists.
Export OFML EpdfProductDb table	Activates / Deactivates export of the optional data control table epdfproductdb.csv. This optional table may contain settings to modify the default-behavior of the product data . Usually data developers do not need to take care about this table. Except the data project requires and defines some specific settings to be modified. The different settings in this table are described in the additional application note document "AN-2006-01 Data control tables". The button  opens another dialog to maintain these settings. Creating new product lines this option is active as default containing the default setting @SafePropertyNames=1. This setting notifies OFML runtime applications, that the product data only contains valid property names, as defined in OCD specification an ensured in pCon.creator OCD module. For historic reasons and backward compatibility older data may use the setting @SafePropertyNames=0 e.g after upgrading with pCon.creator 2.16.
Export OFML PlElement table	This option activates the export of the data control table plelement.csv. It cannot be deactivated because, this table has to be created with each OCD/OAM export. Normally pCon.creator uses default settings to export this table. Usually it is not necessary to maintain this table manually. A detailed description of this table and the allowed settings is described in the application note "AN-2006-01 Data control tables". Using the button  a dialog is opened where these settings can be maintained.

5.3.6 Dialog Article master

The dialog to enter the article master data is divided into two areas. The article record is edited in the left area, while other data - concerning the article being processed - e.g. prices and property classes, are maintained in the right area.

Both areas are separated from each other by a splitter control, which you can move with the mouse to enlarge the respective area of visualization.

Fields

Article code	This field contains the base article number.
Commercial product line	In this field, the article is assigned to a commercial product line. The commercial product line assigned to the OCD product line is preset as default.
Type	The article type is determined here.
Order unit	The unit in which you can order the article.
Discountable	Determine if the article can have discounts on it's purchase price.
Short-text-block	This field refers to the text block 95 containing the short text which is to be displayed for the article.
Long-text-block	This field refers to the text block 95 containing the long text which is to be displayed for the article.
Relation object	This refers to a relation object 108 assigned to the article.
Code scheme	This field indicates a code scheme 129 for the generation of the final article number.
Export filter	A character string (filter marking) which is evaluated when applying the article filter of the distribution region of this product line.
OFML type	The OFML class 157 representing the article is determined here.
ODB manufacturer	The manufacturer code of the OFML package of which an ODB object is used to graphically visualize the article.
ODB product line	The serial code of the OFML package of which an ODB object is used to graphically visualize the article.
ODB name	The name of an ODB object graphically representing the article can be determined here.
ODB parameters	Additional ODB parameters to generate an ODB object are entered in this field.

Furthermore, additional ODB parameter columns are available, which are set by means of the mapping column assignments 156.

Article types

Configurable	The article has configurable properties.
Primitive	The article cannot be configured.

Change "Discountable" setting of multiple articles

Creating new articles the setting "Discountable" is active as default. This field provides a context menu which contains commands to change the discountable value for many or all articles of one product line easily in one step (→ right mouse button click).

5.3.6.1 Property classes

This tab allows you to assign different property classes to an article.

Fields

Position	This field allows you to determine the order of the property classes assigned to the article for visualization in the property editor. It contains the respectively relative position of the class.
Class	The name of a property class 101, which is assigned to the article. A property class can only be assigned once to an article.
Relation object	By indicating an optional relation object 108, relationships of the 'action' type can be bound to this property class.
Text block	The text of the text block 95 optionally assigned here is displayed in the property editor to group the properties of this property class at runtime.

5.3.6.2 Prices

The tab Pricing centrally contains the different prices of an article. The price list to be currently edited is set in the selection list. The default price list is always the active price list of the distribution region.

Prices are divided into the following price levels:

- Base-prices
- Upcharges
- Discounts

Fields

Variant condition	Indication of the variant condition under which the price is valid. The valid variant conditions are determined at runtime by evaluating price relations 125.
Price	This field contains the value of the price in the currency indicated.
Currency	This field determines the currency, in which the price is indicated. The suggestion value of this field corresponds to the currency of the price list the price is assigned to. The value "%" is allowed in this field to generate a relative price.
Type	The type 70 of the pricing component is determined here.
Text block	A text block can be assigned to a price module in this field. The text block 95 will be used by the runtime environment for visualizing the order list
Rounding	Using this field a optional rounding rule 144 can be assigned to each pricing component.
Valid from	This field defines the date of the first day from which on the price is valid.
Valid to	This field defines the date of the last day until which the price is still valid.
Rule	Indicating a calculation rule modifies the manner of the price calculation.

Pricing rules

The following calculation rules are allowed for relative prices of the discount level:

1	The price is calculated in relation to the base price
2	The price is calculated in relation to the price accumulated during the price calculation.

No calculation rules are supported for base prices and upcharges. For discounts a rule has to be specified.

Index prices

An OCD price list normally contains real prices which consist of a discrete value and currency. Alternatively so called index prices can be used. Index prices do not have any discrete value and currency. Instead an abstract price index is assigned to them which can be used to look up in an external price table for a real value.

A field besides the price list selector can be used to switch between real and index prices.

The price index must be a positive integer value.

 Currently it is not recommended to use index prices in standard data creation and distribution processes using pCon.creator ORG-Module and pCon.update. Index prices and the mentioned external tables are not part of the OCD specification. Furthermore the processes of creation, distribution and processing of index prices are not standardized. Index prices are just a special extension to realize special requirements of the German living furniture market.

5.3.6.3 Article properties

By defining article properties the properties stored in property classes are article-specifically restricted to concrete characteristics. They describe fixed properties of an article and have priority over possible suggestion values of the property value list of the property.

Fields

Class	This field contains the name of the property class.
Property	The name of the property is determined in this field.
Value	A property value which is valid for this article is stored in this field.
Export filter	A character string (filter marking) which is evaluated when applying the article property filter 79 of the product line's distribution region.

The fields in the table are selection fields, which allow you to look up property classes, properties, or property values already assigned to the respective article.

Article properties cannot be used to define additional article specific properties. The property 102 has to be defined in a property class 101 which is already assigned to the article 86. The article property is only be used to define it as a fixed attribute with a constant value or to restrict its property value list 105.

If the property is just an internal invisible property 102, which is not configurable, a constant value would be assigned as fixed attribute. The property value list is not required for properties with this scope. If the property does not contain a property value list 105 the specified value will be just assigned to it.

Configurable properties are restricted by article properties. A maximum property value list 105 is required for properties in this scope 102. This list will be restricted to the sub range of value which is defined in article properties.

If the property has got a property value list, it is not possible to add article specific values to this property value list using article properties.

The value of an article property is an internal language independent property value. If the property is visible and a language specific property value text is required, the property has to have a maximum property value list which also contains this value. The definition of the property value in this list can also specify a translated text 95 as property value text 105.

5.3.6.4 Property groups

On tab "Property groups" the property groups[107] of each article are organized. The groups define the appearance of the properties in property editors at runtime.

Property groups are optional. If property groups are not used, the properties are ordered and grouped according to their definition in property classes[87].

Fields

Pos.	This field allows you to determine the order of the property group assigned to the article for visualization in the property editor. It contains the respectively relative position of the group.
Property group	The name of a property group[107] which is assigned to the article. Each property group can only be assigned once to an article.
Text-Block	The text of the text block[95] optionally assigned here is displayed as title of this group in the property editor at runtime.

Articles may have visible properties which are not defined in any of its property groups. Finally such properties are automatically grouped as "Other" in property editor at runtime. This behavior will only be applied if the article has got any property group.

5.3.6.5 Taxes

This tabsheet of the article master can be used to assign a taxation scheme[146] to each article. Taxation schemes are optional.

Taxation schemes are used to define tax categories for ECO-Tax calculation in the French office furniture market. Currently OFML runtime applications use these data tables only for this purpose. Currently the calculation of Value-Added Tax (VAT) cannot be controlled using taxation schemes. OFML runtime applications use always the standard rate for value-added tax calculation. (State of 2013-05-30)

Fields

Taxation scheme	This field is used to select the taxation scheme[146], which is used for the article.
Valid from	An optional effective date from which the taxation scheme is used.
Valid to	An optional effective data until which the taxation scheme is used.

The standard VAT rate is used for all articles which don't have got a taxation scheme which defines the VAT taxation category.

If more than one taxation scheme is assigned to an article, the data of validities from and to have to be defined for each taxation scheme of this article. These validity dates have to be defined in intervals which do not overlap.

5.3.6.6 Packaging

This tabsheet on article master to maintain packaging information of the products. Packaging information can be dependent from the configuration of an article. The configuration dependent logic is expressed in packaging relations 126.

In OCD packaging data is optional. Normally this information is not needed at the point of sales.

Packaging data is used to save product weights for the ECO-Tax calculation in French office furniture market. Currently OFML runtime applications use this data only for this purpose. Although any packaging information which is specified in OCD can be added to the OCD data, it will be ignored by current OFML runtime applications. (State of 2013-05-30)

Fields - General

Variant-condition	Defines the variant condition under which the packaging entry is valid. This variant condition is used by packaging relations 125 to activate this entry. The value itself is just a unique character string, which identifies the entry. This field may be left empty. An empty string variant condition is always active. It is not required to activate it in relations.

Each article may have multiple package data entries using different variant conditions. The values of the following fields are always defined as a difference to base values, which are defined with an entry without variant condition. The values can be negative. For all active entries all values of fields, which are not empty, are summed up at runtime.

Fields - Dimensions of the packaging unit

Width	The width of the packaging unit
Height	The height of the packaging unit
Depth	The depth of the packaging unit
Unit	The unit of measurement of the packaging dimensions

Dimensions of packaging units are optional. But if any dimension is defined, its unit of measurement has to be defined, too.

Fields - Volume of the packaging unit

Volume	The volume of the packaging unit
Unit	The unit of measurement of the packaging unit volume.

The volume of the packaging unit is optional. But if the volume is defined, its unit of measurement has to be defined, too.

Fields - Weights

Article	The weight of the individual article
Packaging	The tare weight of the packaging unit
Unit	The unit of measurement of the weights

The weights are optional. But if any weight is defined, its unit of measurement has to be defined, too.

Fields - Count

Packaging units	The optional count of packaging units
Articles per unit	The optional count of articles in each packaging unit

5.3.6.7 Classification

On tab "Classification" it is possible classify each article using a different classification systems.

The field "Category" is a synonym, for the ID of the class which is used in the particular classification system.

This tab is only visible after at least one classification system was defined to be used on Dialog Classification systems 147. It remains invisible as long no classification system is defined there.

5.3.7 Dialog Text blocks

In this dialog translatable and reusable texts of a product line are defined as so called text blocks. They are separated according to their respective use:

- Article short text 86
- Article long text 86
- Property class text 87
- Property text 102
- Property value text 105
- Price text
- User defind messages 113

Fields

Name	This field contains a symbolic (language-independent) name for the text block.
Text fields	A separate field is available for each text category and language defined in the text management 67 . The respectively internationalized name is entered into this field.

 The left area of the form lists the texts of the individual languages and text categories in form of a table. The right area allows you to oppose the respective texts in two languages and to edit them. The respective languages have therefore to be set in the selection fields.

Article long texts, value texts and user defined messages can contain multiple lines. OFML applications will render each line as paragraph adding automatic line breaks.

Buttons

	Pushing this button a context menu appears which provides access to functions for export 96 and import 97 of the text blocks to/from external data formats.

5.3.7.1 Export text blocks

Translatable texts can be exported to translate them externally. Currently just a specific excel data format 98 is supported. Any text block type is exported. It is not required to export e.g. article short and long texts separately.

Settings

Text catagory	In this field the text category has to be selected of which the translatable texts are exported.
Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
Source language	This field configures the source language. The texts of this language are export as reference for the translators.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
all packages	Normally the text blocks of the current product line are exported. With this option the text blocks of all product lines in the current distribution region are exported in one step.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Export	Start the export after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

5.3.7.2 Import text blocks

After translation of exported texts[95] the updated texts can be reimported to pCon.creator. The used data format is the same like the exported excel data format[98].

Settings

Text catagory	In this field the text category has to be selected of which the translatable texts are exported.
Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
all packages	Normally the text blocks of the current product line are exported. With this option the text blocks of all product lines in the current distribution region are exported in one step.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Import	Start the import after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

The excel file format contains additional information to identify the text blocks correctly. For this reason only excel files which have been exported[95] can be reimported. For this reason it can be possible that excel files cannot be reimported if e.g. these files have been created newly while the external translation process. Please contact support in such cases.

The excel files have to be closed. Otherwise they cannot be imported in any case.

The import automatically tags the modified text records with a status definition[149] "Updated". Maybe the field with the user defined status has to be shown explicitly.

5.3.7.3 Excel text data format

The following section describes the supported Excel data format for export and import of translatable texts.

Filenames

A specific excel workbook is created for each language. The file name of these workbooks identifies the text category and language. It is build in the following scheme:

```
pcr_text_com_ocd_<category>_<language>.xls
```

where <category> matches the single-digit code of the text category and <language> matches the two-digit target language code (ISO-639).

Workbook

A workbook contains all translatable texts of multiple product lines and text types. Different text types are saved in different worksheets.

Worksheet fields

A	***Product line***	The OFML code of the product line.
B	***Name***	The name of the text block.
D	***Source***	The translatable text in the source language.
E	***Target***	The translated or updated text in target language.

Using the Excel workbook to transfer translations please note:

- Translations have to be entered in column E. Only values in this column should be changed. Values in other columns must not be changed.
- Additional Values must not be added in additional columns.
- It is not recommended to add or delete records. Data developers must add translatable records in workspaces first. The purpose of the Excel workbook is just to transfer translations.
- A simple worksheet protection is enabled to ensure these recommendations.
- If users do not comply with these recommendations, problems and errors may occur re-importing the translations to pCon.creator.
- The content of Excel workbooks with translations should not be copied into new empty workbooks. These new Excel workbooks can't be re-imported to pCon.creator directly. Since pCon.creator expects additional essential information in the workbooks to be re-imported (see below for further details).

Custom document properties

The excel workbook contains additional custom document properties to identify the data. Following properties are used:

pcr_appid	An identifier of the application system. The value "pcr" is fixed for this property.
pcr_dbtype	The data format type. The value "text" is fixed.
pcr_FormatDomain	The code of the pCon.creator data format domain. The value "COM" is fixed.
pcr_FormatName	The name of the pCon.creator data format. The value "OCD" is fixed.
pcr_FormatVersion	The version of the excel data format. The value is "1.0.0"
pcr_TextCategoryCode	The single-digit code of the text category where the text blocks belong to.
pcr_LanguageCode	The two-digit target language code (ISO-639)
pcr_LocItemTable_<texttype>	For each text type these property define the name of the excel worksheet where the according text blocks are saved. See below for supported <texttype> values.

Supported text types

artlong	Article long texts
artshort	Article short texts
price	Price texts
propclass	Property class texts
property	Property texts
propvalue	Property value texts
usermessage	User defined messages

5.3.8 Dialog Property classes

Properties to configure articles are defined in property classes under logical aspects. A property class can contain several properties in a defined order. The property classes are assigned to the articles in the article master 87.

Fields

Name	This field defines the unique language independent name of the class. The value has to be a valid OFML identifier. The name must only consist of alphanumeric characters and the underscore. The first character must not be a numeric character. Special characters and blank characters are not allowed.

If properties are changed after the data has been released, articles in existing offers or planned drawings could become unconfigurable anymore. This can happen for instance in case when properties or property values have been renamed or deleted. OFML runtime applications detect if the product data is compatible to the loaded offer. Order positions with a configuration which is incompatible to the installed product data become frozen automatically.

5.3.8.1 Properties

Fields

Position	This field indicates the relative position in the order of the properties of this class.
Name	The property name is a unique language-independent OFML identifier. The name must only consist of alphanumeric characters and the underscore. The first character must not be a numeric character. Special characters and blank characters are not allowed.
Type	This field determines the data type of the property.
Usage	The scope is set for the property. See further explanations below.
Relation object	This field offers you the option to attach a relation object 108 to the property.
Text block	A text block 95 is assigned to this field. The text block contains the labeling of the property in different languages.
Print settings	This field controls the text generation for the property. See further explanations below.
#tot.	This field determines the maximum length of a property value. It includes the separator for floating point numbers.
#post	Count of decimal digits of the property value of type number or length. Property values 105 with decimal digits have to be entered with a decimal point.
EA position	Defines the digit of the final article number where the value of this property is contained in the article number coding scheme EAPos 134.
Obligatory	This field determines whether the property is an obligatory property, to which the user has to assign a property value.
Add values	Here it is defined whether the user can configure additional values to those that have been created in the property value list. If the property contains a range value the option "Add values" does not have to be activated, to allow entering of values between the lower and upper bound of the range.
Restrict.	If the property value list of a property is restricted by a constraint, it has to be activated in this field. Restrictable properties are only considered as evaluated when they have been restricted to exactly one value or a value has been selected by the user.
Multivalued	Usually only one value can be selected or assigned to a property at the same time. This option can be activated for properties to allow selection of a set of discrete current property values or assign a assign them in relations. Currently only internal properties with scope "Relation" can be multivalued. Configurable multivalued properties are currently not supported in OFML runtime applications.

Using the field "Add values" should be avoided in most scenarios. It is not suitable to express special articles. OFML applications provide functionalities to convert standard articles to special articles and give the users the possibilities to describe the special modifications. Such special articles are tagged accordingly and can be detected in order data exchange processes more easily.

The field "Obligatory" is activated for all properties as default. Please note: OCD does not support optional numeric properties. Properties which are of numeric type are always mandatory at runtime regardless if the field "Obligatory" is true or false. If an optional numeric property has to be used in the OCD data model it is necessary to use an additional pseudo property values which represents the optional "not selected" state..

Property types

character	The values represent character strings.
number	The values are real or whole numbers.
length	The property values are real or whole numbers and are indicated in meter in the OFML measuring unit. A conversion to the measuring unit set by the user is performed in the property editor at runtime.
text	Users can enter multi line text freely into this property. The maximum length of text which can be entered is limited by the value of field ***#ges.*** (Maximum length of a property value) If the maximum property value length is set to 0 the user will not be limited to a maximum length. Properties of this type do not have a discrete property value list 105. Using this type the property must be configurable.

Before using properties of type 'text' in data models the scenarios where they are used should be verified carefully. It should be considered that freely entered texts have to be evaluated in order exchange processes. Properties with free texts which have been entered by users may increase the complexity and efforts there.

Before using this property type 'text' it should be taken into account that most OFML runtime applications provide functions to change standard articles to special articles (or manual articles) (e.g pCon.planner / pCon.basket). These special articles provide text properties where the user can describe his special desires. It is not necessary to consider this special scenario using additional properties of type 'text' in the data model of standard products. Furthermore OFML runtime applications to structure and calculate orders provide functions to modify texts and add additional texts to line items which represent standard products (e.g. pCon.basket). Using this feature in these applications it is not necessary to provide additional 'text' properties in the standard product data model.

Property scopes

Configuration	The property is visible and configurable.
Relation	The property is invisible, but can be used within OCD relations.
Graphic	The property is invisible, but can be used within OCD relations. Furthermore, these properties are graphic-relevant and they can be accessed from the ODB.
Display	The property is visible, but disabled and cannot be configured by the user.

Print settings

The values in the field Print control trigger the following behavior when generating the order list:

Standard	The first line of the property description consists of the property name as well as the first line of the text of the current property value. The remaining lines of the property value text succeed.
L1: <Value>	The property description only corresponds to the property value text.
L1: <Property>	The first line of the property description only contains the property name. The lines 2..n of the text of the current property value succeed. The first line of the property value text will not be printed.
L1: suppress	The property description only consists of the lines 2..n of the text of the current property value.
no text	No description of the property will be printed on the order form.

5.3.8.2 Property values

The property value list of the selected property is defined in the lower right area of the form. The type set on the property as well as the maximum length and number of decimal places have to be observed.

You have the option of defining individual values as well as intervals. However, only one interval is allowed to be visible on the property at runtime. In case of assigning several intervals, this has to be assured by appropriate preconditions excluding each other.

Fields

Position	This field indicates the relative position in the order of the properties belonging to this class.
Default	Selecting this field defines whether this property value is a suggestion value or not.
Op. from	The operator for the bottom limit of an interval, which is defined by this value. If no interval but an individual value is defined, the value = is valid, which is already selected as the suggestion value.
Value from	This field contains the string representation of a numeric or symbolic (language-independent) identifier. It represents the individual value or the value of the bottom limits when defining an interval.
Op. to	This optional field determines the operator for the upper limit when defining an interval.
Value to	This optional field contains the value of the upper limit when defining an interval.
Raster	For an interval in this optional field a step range can be defined.
Relation object	This optional field attaches a relation object 108 to this property value.
Text block	This field represents the linking to a text block 95 containing the language-specific texts of the property value description.
no text	If this field is selected, no property description is printed, when generating the order list and this property value has been selected.
Export filter	A character string (filter marking) which is evaluated when applying the property value filter of the distribution region of this product line.
Valid from	An optional date from which the property value is valid and can be selected by users (requires OCD 4.2).
Valid to	An optional data until which the property values is valid. After this date users cannot select it (requires OCD 4.2).

The value*VOID* is reserved and thus not permitted. Values of properties with data type "character" 102 are not allowed to contain space characters.

If no value of the list is defined as suggestion value, the first value of the list is automatically selected in case of obligatory properties. In case of optional properties, a virtual value "not specified" is automatically displayed at runtime.

Validity period of property values

An optional period of validity can be defined using the fields "Valid from" / "Valid to". Using a period of validity requires that OCD export format version 4.2.0 is active (see product line management 80)

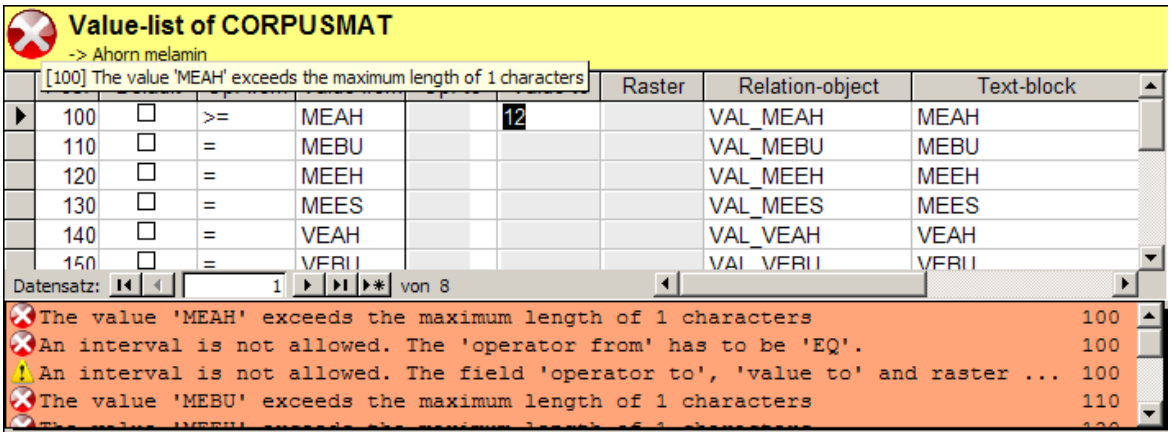
If an older OCD format version is used for export, these dates will not have any effect.

If the dates of validity are not set, the values are always valid.

Specifying a validity period of property values has to be considers if distribution regions are exported with multiple price lists. If the property value affects prices (extra charges) but is only valid in on of the exported price lists, it should be considered to specify the validity period accordingly. Otherwise a user could select the property value, although the currently used price list does not contain the according extra charge price.

Validation

While entering property value the records are validated according to the current property's settings. An additional icon appears, if the property value list contains any errors.



The description of the first error is always shown in the tooltip of this icon. A complete list of all errors can be opened using a double-click on this icon. Double-clicking an entry in this error list, selects the according property value has got this error.

5.3.9 Dialog Property groups

Properties of articles[87] are defined and organized in property classes[101]. Property classes define the groups and order of properties as they are shown in property editors.

Sometimes technical restrictions affect the definition of property classes have to be taken into account. The appearance of the technical view of property classes some times does not match the expectations of good usability of the data at point of sales.

Using optional property groups another more sales oriented view can be defined for articles[91]. The property groups just group and order the properties of property classes differently when shown in property editors.

Fields

Name	This field defines the unique language independent internal name of the property group. This values is only used for internal identification of the group in OCD data model. An optional text block defining a title which appears in property editor, can be defined when the is assigned to an article in Dialog Article master.[91]
Pos.	This field indicates the relative position in the order of the properties of this group.
Class	This field contains the name of property class, where the property is defined. It is possible to use the character * instead of a real name of any property class. This means this record reference a property with the same name from any of an article's property classes..
Property	Contains the name of a property.

Property groups are only exported when OCD 4.3.0 export format[80] or newer is enabled in product line settings.

It is not possible to define new properties in property groups! The properties must always be defined in property classes first. Properties in groups which are not present in any class of an article are just ignored at runtime.

5.3.10 Dialog Relational objects

Relational objects group one or several relations[110]. The type of using the relations is determined here.

Fields

Relation object	This field contains a unique name to identify the relation object.
Pos	The relative position of a relation within the order of the relations of the object is indicated here.
Domain	A relation is assigned to a domain, which is defined in this field.
Type	This field determines the type of the relation.
Relation	It is referred to a defined relation[110] containing the logic.

Domain

Configuration	Relations of this domain are evaluated during generation as well as during each configuration step. They define the configuration options.
Pricing	Relations of this domain are evaluated to calculate the price[125] for the current configuration of the article. The relations of this domain must be of the type Action.
Packaging	Relations of this domain are evaluated to determine the packaging information[126] of the current configuration of an article. The relations of this domain must be of the type Action.
Taxation	Relations of this domain are evaluated while determining the tax. The relations of this domain must be of the type Action. Taxation relations[128] can be used to change the tax category of articles depending on the current article configuration.

Type

Precondition	Preconditions are boolean expressions describing the condition[116] under which a property or a property value can be selected. If this condition is evaluated to false, this property does not exists and won't be displayed.
Select condition	Select conditions are boolean expressions describing the condition[116] under which the user hat to assign a value to an optional property.
Action	Actions allow you to assign values to properties[117]. Actions on properties are only evaluated, if the property is not hidden by a precondition. On property values they are only executed, if the property value has been selected.
Constraint	They serve to check the consistency of an article configuration as well as to assign property values or to restrict property value lists. They are a complex language construct[120]. Constraints can only be assigned to relation objects being used on article level.
Reaction	Reactions are used to assign a value to a property[117]. Reactions which are bound to a property are evaluated right after the user changed the value of this property. If they are bound to an article they will be evaluated only once while article is created. Reactions are always evaluated first before all other relations.
Post-Reaction	Post-Reactions can be bound like Reactions to properties. They can assign a value to a property[117], right after a user has changed the property's value. In opposite to Reactions Post-Reactions are evaluated last after all other relations.


Reactions, which are assigned to articles, are evaluated first. For this reason optional conditions in article bound reactions cannot refer to other properties if their value is assigned by other relations.

5.3.11 Dialog Relations

This dialog manages the relations of the product lines.

Fields

Relation	A unique name of the relation. It is identified by this name.
Relation code	This field contains the code of the relation.

Relations are referenced from relation objects 108 by their names.

Comment lines can be used in relation code. These lines must start with the character * .

The relation code is dependent on the coding language which has been selected for each product line 80.

The features and syntax of each coding language is explained in detail in the OCD specification. The following sections are just a brief overview of the supported language elements according to the OCD 4.0 language definition.

5.3.11.1 Variables & Constants

Following variables and constants can be used in logical expressions.

FALSE	Logical false
$BAN	With the help of this variable the current base article number can be evaluated in expressions.

5.3.11.2 Operators

Relational operators

=	Is equal to (EQ)
<>	Is not equal to (NE)
>=	Is greater than or equal to (GE)
<=	Is less than or equal to (LE)
>	Is greater than (GT)
<	Is less than (LT)

IN	Verifies whether the value of a property is contained within a certain value quantity.
	
	Usage:
	
	`<Property> IN (<Value1>, <Value2>, <Value3>, ...)`
	
	`NOT <Property> IN (<Value1>, <Value2>, <Value3>, ...)`
	
	Since OCD 4.1 open or closed ranges can be used instead of discrete values. Closed ranges are defined as `<LowerLimit>-<UpperLimit>` . Open ranges are defined using the according operator in front of the boundary value using < or <= for downwards open ranges or > or >= for upwards open ranges.
	
	Example:
	
	`WIDTH IN ( 5-10, 20, 30, >50 )`
	
	In case of restrictable values, this condition will only be executed if the property has been restricted to exactly one value.
	
	Constant character string values within the value range can contain the wildcard characters:
	
	* - multiple arbitrary characters
	
	? - exactly one arbitrary character
	
	as long as wildcard comparison is activated in product line management 80. Otherwise these special character are evaluated like normal characters.

The operand types on both sides must be identical.

Logical operators

AND	Logical and
OR	Logical or
NOT	Negating a logical expression. Usage: `NOT <Expression`

SPECIFIED	Determines if the article has a certain property and this property has got any value. If the property is restrictable 102, this condition is fulfilled if the property has been restricted to exactly one value.
	
	Usage:
	
	`SPECIFIED <Property>`
	
	`NOT SPECIFIED <Property>`

First of all, the AND operations are evaluated and after that the OR operations. The order of execution can be controlled with the help of round brackets as well.

Arithmetic operators

+	Addition
-	Subtraction
*****	Multiplication
/	Division

Character string operators

+	Concatenating character strings `$VARCOND = 'UP_PG_' + STRING(PG)`

5.3.11.3 Builtin functions

Arithmetic functions

pow(x, y)	Calculating the power x to y. *Parameters:* x Float y Int *Return value:* Float
sqrt(x)	Calculating the positive square root of x *Parameters*: x Float *Return value:* Float
fabs(x)	Calculates the amount of x *Parameters*: x Float *Return value:* Float
ceil(x)	Rounding off x to the smallest integral value which is greater than x. *Parameters:* x Float *Return value:* Int
floor(x)	Rounding off x to the biggest integral value which is less than x. *Parameters:* x Float *Return value:* Float
sign(x)	Determines the preceding sign of x *Parameters:* x Float *Return value:* Int (-1 or +1)
trunc(x)	Determines the integral part of x *Parameters:* x Float *Return value:* Float

frac(x)	Determines the decimal part of x *Parameters:* x Float *Return value:* Float

Type conversion functions

STRING(x)	This function converts a numeric value x into a character string. *Parameters:* x Int / Float *Return value:* String
FLOAT(x, y)	Converts the value x to a fractional number and returns the result. The parameter x can be an expression of the types integer, fractional number or character string. If the value x cannot be converted, the fallback parameter y will be returned in case y is a constant fractional number. Otherwise a syntax error will occur at runtime.
INT(x, y)	Converts the value x to an integer and returns the result. The parameter x can be an expression of the types integer, fractional number or character string. If the y is a fractional number only the integer part is returned. If the value x cannot be converted, the fallback parameter y will be returned in case y is a constant integer number. Otherwise a syntax error will occur at runtime.

Please note conversions of character strings to numeric values, expects a decimal point as separator of integer and fractional parts. Seperators of groups of thousands are not allowed. The values can have a leading minus.

Character string functions

SUBSTR(x, y, z)	Determines the partial character string with the length z from position y out of the character string x. *Parameters:* x String y Int z Int *Return value:* String The first character's position is 0. Parameter z (Length of the substring) is optional. It can be omitted. In this case the substring from position y to the end of the original character string x is returned.
SIZE(x)	Returns the number of characters of the character string x.
ToUpper(x)	Returns a character string where all characters of string x are changed to upper case.
ToLower(x)	Returns a character string where all characters of string x are changed to lower case.
TRIM(x)	Returns a character string where all white space characters at the beginning and at the end of value x are removed.
LTRIM(x)	Returns a character string where all white space characters at the beginning of value x are removed.
RTRIM(x)	Returns a character string where all white space characters at the end of value x are removed.

User defined messages

USER_MESSAGE(x)	Displays a user defined message. The parameter x defines the name of the text block 95 which is displayed. *Parameters:* x String *Return value:* String The name of the text block has to be given as constant character string enclosed in single quotation marks.

User defined messages are a useful method to inform the users about inconsistencies of the configuration or just to display additional hints. They can be used within actions, reactions or post-reactions 108. However displaying user defined messages interrupts the configuration workflow and may be annoying. It should be considered to use them cautiously.

Special functions

ABORT()	The function ABORT can be used in reactions[108] and post-reactions[108] of properties[102] since OCD 4.1 . This function aborts the execution of relations at runtime under specific conditions. The article's state is automatically resetted to its state before the property change. e.g. `ABORT() IF PROPERTY = 'INVALID'`

A positive user experience should be preferred in data models. Using logic which informs the user about invalid input or which resets the user's selection should be avoided. It should be perfered to show the users only valid values for selection.

5.3.11.4 Conditions

Conditions are defined as logical expressions, which are evaluated with TRUE or FALSE. These are

- Comparisons
- Negations
- Special conditions

Logical operators allow you to create complex logical expressions.

Example: Complex condition

```
WIDTH > 600 AND NOT (DEPTH = 600)
```

If this expression is used as a precondition for a property value, this property value can only be selected if the property `WIDTH` is greater than`600` and if the property `DEPTH` is not set to the value `600`.

The evaluation of an expression will produce errors if it access properties which does not exist (e.g. if it is hidden by a precondition). Using the operator operators[111] `SPECIFIED` this configuration situation can be taken into account.

```
SPECIFIED FRONTTYPE AND FRONTTYPE = 'glass'
```

5.3.11.5 Value assignments

Within actions or relations values can be assigned to a property. The assigment operator is the equal sign = . Multiple assignments within one action are separated by a comma.

Each assignment can be equipped with a condition[116], under which the assignment is carried out.This condition is introduced with the keyword `IF` behind the assignment.

Example: An action to assign a value to a property

```
CABLETRAY = 'X0' IF WIDTH <= 600,
CABLETRAY = 'X5' IF WIDTH >= 1600
```

Such an action sets the value of the property `CABLETRAY` to the value `X0`, if the property `WIDTH` is less than or equal to the value `600` .
If the property `WIDTH` has a value which is greater than or equal to `1600` , the property `CABLETRAY` obtains the value `X5` .

Furthermore the assignment can use calls to value combination tables[118].

Example: Assign a property value in an action

```
TABLE PG_FRONT (COL_FRONTMAT = FRONT, COL_PG = $SELF.PG)
```

This example derives a value with the help of the value combination table[139] `PG_FRONT` for the property `PG` in dependence of the property `FRONT`. The table therefore contains the two columns `COL_FRONTMAT` and `COL_PG`.

Since OCD 4.1 assignments with the same condition can be grouped in actions, reactions and post-reactions[108] using curly brackets. It is not necessary to repeat the condition for each assignment.

Example: Grouped assignments

```
{
    SOCKET = 'S',
    WALLMOUNT = 'Y'
} IF HEIGHT > 1800
```

In this example a value will be assigned to the properties `SOCKET` and `WALLMOUNT` if the property `HEIGHT` is greater than `1800`.

5.3.11.6 Table calls

Within constraints and actions, 108 calls to value combination tables 139 can be used.

The access is effected with the help of the keyword TABLE.

```
TABLE <TableName> (<ParameterList>)
```

The paramater list determines the linkage of property variables with individual columns of the table, separated by a comma.

<Column name> = <Property variable>

Table calls within actions

Value combination tables within actions allow you to assign values for properties. The result of the table access must be unequivocal.

In the parameter list, properties, for which the value is assigned, are marked with the prefix "$SELF." . Within actions of price relations, accessing tables allows you to assign a value for the variable $VARCOND.

Example: Assign a property value in an action

```
TABLE PG_FRONT (COL_FRONTMAT = FRONT, COL_PG = $SELF.PG)
```

This example derives a value with the help of the value combination table 139 PG_FRONT for the property PG in dependence of the property FRONT. The table therefore contains the two columns COL_FRONTMAT and COL_PG.

Example: Pricing relation using a value combination table

```
TABLE UPCHARGES_PG_WIDTH (COL_WIDTH = WIDTH, COL_PG = PG, COL_VARIANT
= $VARCOND)
```

This example activates a price within a price relation. The table 139 UPCHARGES_PG_WIDTH is accessed. It contains three columns. The column COL_VARIANT contains the variant conditions, which are assigned in dependence of the properties WIDTH and PG.

Table calls within constraints

Within constraints 120, tables can be used in the paragraph Restrictions

- to check the consistency of an article configuration
- to restrict the value range of properties

- to assign a value to a property

If a value for a non-restrictable property is assigned, the value combination table must return a unique value, otherwise the constraint is not fulfilled.

Table calls in preconditions

Value combination tables can be accessed from preconditions. The precondition is true if the table returns exactly one valid value combination otherwise the expression is undefined. As long as at least property, which is used to access the table, has not got any value, expression is not evaluated.

Table calls in article code schemes

Since OCD 4.0 value combination tables can be used in user defined article code schemes136. In this case no addtional relation is required. The table call can be used directly within the code scheme (see here136).

5.3.11.7 Constraints

Constraints are complex language constructs. They consist of the following components, which are introduced with the respective keyword followed by a colon. Each part is finished with a period.

Objects:

This part is obligatory. Variables have to be declared for the property classes, by which statements are made in the constraint. This is done by means of the construct.`IS_A` :

```
<Variable> IS_A <Property class>
```

Furthermore, you have the option to declare separate variables for individual properties. This is done after the keyword `WHERE` after the declaration of the property class variables:

```
<Variable> = <Property>
```

Multiple variable declarations for properties are separated by a semicolon.

The section Objects: is optional since OCD 4.0 .

Condition:

This optional paragraph allows you to define a condition in form of a boolean expression. This expression describes the condition, which must be fulfilled first, so that the constraint is evaluated.

Restrictions:

This paragraph contains the property dependencies of the article, which have to be fulfilled, so that the configuration is considered as valid. A comma separates multiple dependencies. Restrictions can be defined in form of.

- Comparing,
- Checking on value ranges, or
- Accesses on value combination tables

The OFML runtime environment ensures that the article cannot be ordered or that a configuration step, which leads to an inconsistent state, cannot be executed.

Inferences:

In this paragraph the properties are named to which values are assigned or whose value range is restricted. Several properties are separated by a comma. The value assignment or restriction is done with the help of the dependencies in the restriction paragraph.

The constraint will not be evaluated if one of the property classes declared in the paragraph Objects has not been assigned to the article.

When creating data, you have to make sure that the initial configuration of an article is valid, otherwise the article will not be generated.

Properties whose value ranges are restricted, have to be marked as restrictable in the property table 102.

Example: Check article consistency using a value comparision

```
Objects:
   x IS_A CUPBOARD.

Condition:
   x.FRONT IN ('WE', 'NB').

Restrictions:
   x.HANDLE = 'KN3'.
```

This simple example checks that the property `HANDLE` of the class `CUPBOARD` is set to the value `KN3` if to the property `FRONT` the value `WE` or `NB` is assigned. If this is not the case, the last configuration step, which would lead to an inconsistent state, will be denied.

Example: Check article consistency using a value combination table

```
Objects:
   x IS_A CUPBOARD.

Restrictions:
   TABLE FRONT_HANDLE ( COL1_MAT = x.FRONT, COL2_HANDLE = x.HANDLE ).
```

This example accesses the value combination table `FRONT_HANDLE`. This table contains the two columns `COL1_MAT` and `COL2_HANDLE`. These columns are connected with the properties `FRONT` and `HANDLE` of the property class `CUPBOARD` and contain the valid combinations of these property values. If the current combination is not listed as valid combination in this table, the last configuration step, which would lead to an inconsistent state, will be denied.

Example: Value assignment

```
Objects:
   x IS_A CUPBOARD.

Restrictions:
   x.FRONT = 'EH' IF x.CORPUS = 'EH',
   x.FRONT = 'BI' IF x.CORPUS = 'BI'.

Inferences:
   x.FRONT.
```

This constraint derives values for the property `FRONT`. It is ensured that `FRONT` obtains the same value as the property `CORPUS`, if `CORPUS` is set to `EH` or `BI`.

Example: Restriction of a discrete property value range

```
Objects:
    x IS_A CUPBOARD_C
    WHERE
        corpus_mat = CORPUS;
        front_mat = FRONT.

Condition:
    corpus_mat = 'NB'.

Restrictions:
    front_mat IN ('NB', 'CM').

Inferences:
    front_mat.
```

This example restricts the value range of the property `FRONT`. The restriction only takes place if the value of the property `CORPUS` is `NB`. This restriction is done with the help of the function `IN` to check a quantity. The property `FRONT` must have been marked as restrictable.

Example: Restriction of a discrete property value range using a value combination table

```
Objects:
   x IS_A CUPBOARD.

Restrictions:
   TABLE MAT_CORPUS_FRONT ( COL1_CORPUS = x.CORPUS, COL2_FRONT =
   x.FRONT ).

Inferences:
   x.FRONT.
```

This example restricts the value range of the property`FRONT` of the class `CUPBOARD` depending on the current value of the property `CORPUS` to valid values. The value combination table `MAT_CORPUS_FRONT` contains the valid combinations of the two properties.

5.3.11.8 Pricing relations

Activation pricing components

Prices with a variant condition are activated with the help of assignments within an action 108 which is used as pricing relation. They are activated by assigning the variant condition of this price to the special variable `$VARCOND` .

Example: A pricing relation

```
$VARCOND = 'PG1_0800' IF PG = 'PG1' AND WIDTH = 800,
$VARCOND = 'PG2_1000' IF PG = 'PG2' AND WIDTH = 1000
```

In this example, the price module is activated with the variant condition `PG1_0800` , if the property `PG` has the value `PG1` and if the property `WIDTH` has the value`800` . If `PG2` is the current value of the property `PG` and if `1000` is the current value of the property `WIDTH`, the price module `PG2_1000` will be activated.

Example: Pricing relation using a value combination table

```
TABLE UPCHARGES_PG_WIDTH (COL_WIDTH = WIDTH, COL_PG = PG, COL_VARIANT
= $VARCOND)
```

This example activates a price within a price relation. The table 139 `UPCHARGES_PG_WIDTH` is accessed. It contains three columns. The column `COL_VARIANT` contains the variant conditions, which are assigned in dependence of the properties `WIDTH` and `PG`.

Within a pricing relation different values can be assigned to the variable `$VARCOND` under different conditions. If multiple assignments are done, because their conditions are valid, each of the according pricing components is activated. A second assignment of a value to this variable does not overwrite the previous value.

The special variable `$VARCOND` is the default variable to activate pricing components. The name of this variable may be optionally configured in product line management 80 since OCD 4.0 .

Price factors

In pricing relations factors can be multiplied to the value of activated pricing components with the help of the special function `$SET_PRICING_FACTOR:`

```
$SET_PRICING_FACTOR(<VariantenKondition>, <Faktor>)
```

Example: Applying a pricing factor

```
$VARCOND = 'UPCHARGE_HANDLE_A',
$SET_PRICING_FACTOR('UPCHARGE_HANDLE_A', 2) IF WIDTH > 600
```

This example demonstrated the activation of a pricing component with variant condition `UPCHARGE_HANDLE_A`. The value of this price is doubled as soon as the value of the property with is greater than `600` .

5.3.11.9 Packaging relations

Activation of packaging data entries

Packaging data entries 93 of articles which have got a variant condition, are activated by packaging relations of type Action 108. These entries are activated by an assignment of the according variant condition to the special builtin variable `$VARCOND`.

Example: Activation of a packaging data entry with a variant condition

```
$VARCOND = 'WGT_HEADREST' IF HEADREST = 'Y'
```

This example will activate a packaging data entry with variant condition `WGT_HEADREST` if the property `HEADREST` of the article is set to the value `Y`.

Within a packaging relation different values can be assigned to the variable `$VARCOND` under different conditions. If multiple assignments are done, because their conditions are valid, each of the according packaging entries is activated. A second assignment of a value to this variable does not overwrite the previous value.

The special variable `$VARCOND` is the default variable to activate pricing components. The name of this variable may be optionally configured in product line management 80 since OCD 4.0 .

Packaging data factors

Additional factors can be multiplied with individual fields of packaging data records. The factors can be set in packaging data relations using the special function `$SET_PCKG_FACTOR` .

```
$SET_PCKG_FACTOR(<VariantenCondition>, <Data element>, <Factor>)
```

The first parameter is a constant string or expression which specifies the variant condition of the packaging data record where the factor will be applied. The second parameter is the name of the data element or field where the factor is applied. The last parameter is the numeric constant or a numeric expression which defines the factor itself.

The second parameter can be one of the following values

NETWEIGHT	The weight of the individual article which is defined in the packaging data record
TARAWEIGHT	The tare weight of the packaging unit
DEPTH	The depth of the packaging unit
HEIGHT	The height of the packaging unit
WIDTH	The width of the packaging unit
VOLUME	The volume of the packaging unit
ITEMSPERUNIT	The count of packaging units
PACKUNITS	The count of articles in each packaging unit

Additional conditions can be defined after this function using the keyword `IF` to limit the conditions when the factor will be applied.

Example: Apply a constant value factor to the article's weight

```
$VARCOND = 'WGT_COVER',
$SET_PCKG_FACTOR  ('WGT_COVER', 'NETWEIGHT', 1.5 )
```

First in this example the relation activates the packaging data record with the variant condition `WGT_COVER` . Then a constant factor 1.5 is applied to article weight which is defined in this record.

Example: Use an expression as factor for the article's weight under specific condition

In following example we assume the article may have a configurable cover (property: `TOPCOVER`). Furthermore the article's width is variable. The weight of the article's optional cover is saved in a separate packaging data record using the variant condition `WGT_COVER` . The weight itself is given per meter in this record. The following packaging relation will activate the weight of the article's cover element.

```
$VARCOND = 'WGT_COVER' IF TOPCOVER = 'Y',
$SET_PCKG_FACTOR ('WGT_COVER', 'NETWEIGHT', WIDTH / 1000 ) IF TOPCOV
```

First the relation will activate the package data record with variant condition `WGT_COVER` if the property `TOPCOVER` is equal to the value `Y`. Then a factor is applied to the article weight of this packaging record using the same condition. The factor itself is given as a numeric expression. It takes the current width from the numeric property `WIDTH` . Assuming this value will be given in millimeter it is converted to the expected unit of meters to be used as factor for the article weight.

5.3.11.10 Tax calculations

Taxation schemes[92] define how taxes are calculated for each article. The taxation schemes for the article in its base configuration. They are independent from the articles configuration (variant).

Using relations of domain "Taxation" and type "Action"[108] it is possible to modify the tax calculation at runtime depending on the current article configuration.

Change the tax category

```
SET_TAX_CATEGORY(<tax type>, <tax category>)
```

This function activates another new tax category for as specific tax type. The parameters `<tax type>` and `<tax category>` are character strings. Valid values are the same as used to define the taxation schemes[146].

Example

```
SET_TAX_CATEGORY(ECO_FR, workplace_metal95) IF TableTop = 'None'
```

A desk has got a property TableTop, which allows a configuration without the plate (just the frame). In this case the amount of materials of the desk changes in a relevant way for the ECO-Tax calculation in France. In this case another tax type is used for the ECO-Tax than the base article usually has git.

OCD 4.3.0 export format[80] or newer must be enabled to use relations for tax calculations in a product line.

5.3.12 Dialog Article encoding

In this dialog the codification schemes for the generation of final article numbers can be maintained.

A codification scheme defines how the final article number will be built from the base article number and the so-called variant code, which specifies the configuration of the article.

Fields

Name	This field contains the symbolic identifier of the scheme. This can be a freely selected name for a user-defined scheme or one of the predefined schemes can be selected.
VarCode sep.	In case of predefined schemes, this field defines a character string, which is integrated into the final article number in order to separate base article number and variant code.
Value sep.	The character string is defined here, which is used for predefined schemes to separate property values from one another.
Display	For predefined schemes it is determined which properties shall be contained in the variant code: • all - All configurable properties are contained • current- Only the currently valid and visible properties are contained
$ <> visible	This field defines a replacement character used for currently invalid or non-visible properties. As many characters will be displayed as defined in the field Total length of the respective property 102.
$ <> selected	This field contains the replacement character for currently not evaluated optional or restrictable properties. As many characters will be displayed as defined in the field Total length of the respective property 102.
Trim	This field defines whether property values are displayed in a shortened form if they consist of less characters than defined in the field Total length of the respective property. Selecting this option removes characters not assigned at the end of the value.
Scheme	The encoding rule of a user-defined scheme is entered here.

To be able to unequivocally reconstruct an article from the final article number, no replacement character, leading to a regular value of optional or restrictable properties, must be used in the fields $ <> visible and $ <> selected.

If the option Trim has been selected, the values of the fields $ <> visible and $ <> selected must be different from Value sep.

If properties, for which the user can freely enter values, in the final article number have to be encoded for an article, the option Trim must not be used.

Using a single space character in fields *$ <> visible* oder *$ <> selected ,* this character has to be enclosed in single quote characters.

Using a white space character as separator or replacement character, this white space character has to be enclosed in single quotation marks.

5.3.12.1 KeyValueList

The KeyValueList maps the variant code as pairs of property and property values in the defined order of the article properties. The mapping is done as follows:

```
<Property class>.<Property>=<Property value>
```

The following fields do not affect this scheme:

- ***Value-Sep.***
- ***Display***
- ***$ <> visible***
- ***$ <> selected***

Properties are always separated with a semicolon.

Non-evaluated properties obtain the internal value VOID. Property values are always displayed in a shortened form.

The variant code is displayed on orders as part of the final article code. In product line management an option can be used to show only the base article code of all articles in a productline. This option can be useful in cases the variant code is just important for the interal processing of the data,

Example: Creation of a variant code using KeyValueList

Applying this scheme to an article with the base article number:

```
C2080TK
```

with the configuration:

Class	*Property*	*Value*
CUPBOARD	**SOCKET**	L7
	INTERIOR	B5
MAT	**CORPUS**	NB
	FRONT	CM

results in the following variant code:

```
CUPBOARD.SOCKET=L7;CUPBOARD.INTERIOR=B5;MAT.CORPUS=NB;MAT.FRONT=
CM
```

If the property INTERIOR was an optional property and had not been evaluated, this would result in the following variant code:

```
CUPBOARD.SOCKET=L7;CUPBOARD.INTERIOR=VOID;MAT.CORPUS=NB;MAT.FRON
T=CM
```

5.3.12.2 ValueList

Only the current property values are mapped taking the parameters set for the scheme into account. The order is defined by the position of the property in the property table 102.

Example: Creation of variant code using ValueList

The following settings were used for the encoding scheme:

VarCode-Sep.	#
Value-Sep.	
Display	all
$ <> visible	-
$ <> selected	X
Shorten	no

Applying this scheme to an article with the base article number:

```
C2080TK
```

with the configuration:

Class	*Property*	*Length*	*Value*
CUPBOARD	**SOCKET**	**4**	L7
	INTERIOR	**4**	B5
MAT	**CORPUS**	**2**	NB
	FRONT	**2**	CM

results in the following final article number:

```
C2080TK#L7  B5  NBCM
```

Assuming that the property `INTERIOR` is an optional property and has not been evaluated, this would result in the following variant code:

```
C2080TK#L7  XXXXNBCM
```

5.3.12.3 EAPos

The final article number is generated according to the following mechanism in the EAPos scheme.

1. The final article number starts with the base article number.

2. The current value of the property is in full length taken over on exactly the position as it is defined in the field EA-Pos 102 in the property table. Depending on the assignment of the field EA-Pos, parts of the base article number can be overwritten.

3. The values of the field EA-Pos have to be selected in a way that property values do not overlap. A property whose value overlaps with the value of a previous property is ignored. This is recorded in the log file during the export.

The EAPos scheme is no native OCD encoding scheme. According to the settings made for this scheme, pCon.creator automatically creates user-defined schemes mapping this in OCD.

Example: Creation of variant code using EAPos

Applying this scheme with the settings

Display	all
$ <> visible	-
$ <> selected	X

to an article with the base article number:

```
C2080TK
```

with the configuration:

Class	*Property*	*EA-Pos*	*Value*
CUPBOARD	**SOCKET**	**6**	L7
	INTERIOR	**10**	B5
MAT	**CORPUS**	**14**	NB
	FRONT	**16**	CM

results in the following final article number:

```
C2080L7  B5  NBCM
```

If the optional property `INTERIOR` has not been evaluated, the final article number is:

```
C2080L7  XXXXNBCM
```

5.3.12.4 User-defined schemes

In the field Scheme you can store an individual coding rule.

The places of the final article number are defined separated with a comma from the left to the right. The processing is done from the left to the right.

Permitted characters

@	The next place of the base article number is printed out at this place and is also read from the left to the right.
<Property class>:<Property>	The value of the specified property is printed out at this place. The property name has always to be indicated fully-qualified via the property class.
Any character	The character is statically printed out at this place. However, the character @ is reserved and must not be used.

Table call	Some digits of the final article code can be defined using a value combination table139. The table call is defined in the following form:
	
	`TABLE <TabeleName>`
	`(<COLUMN_1>=<PROPERTY_1>[,...,`
	`<COLUMN_N>=<PROPERTY_N>)`
	
	The table must contain an additional column `$FAN` which contains the value which is inserted into the final article code. This column is not used in the parameter list of the table call. The return value of the table must be unique. The base article code can be referenced in the parameter list of the table call using the special parameter `$BAN`.

Example: User defined coding scheme

A user-defined scheme is defined as follows:

```
@,@,@,@,@,CUPBOARD:SOCKET,CUPBOARD:INTERIOR,@,@,-
,MAT:CORPUS,MAT:FRONT
```

while the following settings are determined:

Display	all
$ <> visible	-
$ <> selected	X
Trim	no

Applying this scheme to an article with the base article number:

```
C2080TK
```

with the configuration:

Class	***Property***	***Length***	***Value***
CUPBOARD	**SOCKET**	**4**	L7
	INTERIOR	**4**	B5
MAT	**CORPUS**	**2**	NB
	FRONT	**2**	CM

results in the final article number

```
C2080L7  B5  TK-NBCM
```

If the optional property `INTERIOR` has not been evaluated, the final article number is as follows:

```
C2080L7  XXXXTK-NBCM
```

Example: Table call in userdefined code scheme

The following user defined code scheme displays the five first digits of the base article code followed by a value which is dependent on the width and the depth of the article:

```
@,@,@,@,@,TABLE          CODE4711(COL_WIDTH=WIDTH,COL_DEPTH=DEPTH),-
,MAT:CORPUS,MAT:FRONT
```

This value is defined by the value combination table 139 `CODE4711` . The parameter list of this table call connects the column `COL_WIDTH` with the property `WIDTH` and the column `COL_DEPTH` with the property `DEPTH`. The table must contain an additional column `$FAN` . This column contains for each value combination the valid character string which is inserted into the final article code.

Please note: For demonstration purposes this example uses properties which define article dimensions. It should be taken care, that a value combination table must be defined completed. Therefore using value combination tables is only usefull in relation with discrete properties.

5.3.13 Dialog Value Combination Tables

Value combination tables within relational knowledge 118 can be used to check the consistency, to assign values, or restrict the value range of properties. Furthermore value combination tables can be used from user defined article encodings 136 to create final article codes.

This form allows you to create these tables. The values of the individual cells are edited in Excel. An Excel column is created for each column defined. Furthermore, a column with the title # precedes the Excel worksheet, where the logical line numbers are defined.

 Each logical row contains a valid combination of values. If several values of a property can be combined with one or several particular values of another property, they can be combined in a logical row and it is not necessary to permute out each combination. Therefore the row number in Excel has to be respectively entered. Only Excel lines with a valid logical row numbers are taken over.

Fields

Table	This field in the left area of the form contains the name of the table as table name. The name should follow the rules of a valid OFML identifier.
#	This field in the right area of the form defines the relative position of a column.
Column	The individual columns of the value combination table have to be defined in this field in the right area of the form. These are symbolic identifiers.

 The table and column names must only consist of alphanumeric characters and the underscore. The first character must not be a numeric character. Special characters and blank characters are not allowed..

Buttons

	This button allows you to export the current table to Excel in order to edit the values. The exported table is opened immediately.
	This button opens a value combination table already exported to Excel.
	After having edited the value combination table in Excel, this button imports it back to pCon.creator.

The tables exported are stored in the local resource directory of the workspace. It is absolutely necessary to make sure that an edited table is imported back, otherwise the modified values are not taken into account when executing the OFML export.

After having imported back a value combination table from Excel to the workspace, the Excel workbook can be deleted. It is not needed any more.

Please note to use a decimal point to enter real values in value combination tables for numeric properties 101 with decimal digits.

Example: Value-Combination table with two columns and two logical rows

	A	B	C
1	#	COL_FRONTMAT	COL_PG
2	1	BI	PG1
3	1	CM	
4	2	EH	PG2
5	2	NB	

In this example, a value combination table with the columns `COL_FRONTMAT` and `COL_PG` has been defined. It contains two logical rows, which are interpreted as follows:

The values `BI` or CM in column `COL_FRONTMAT` are valid with the value `PG1` of the column `COL_PG`.

The values `EH` or `NB` in column `COL_FRONTMAT` are valid with the value `PG2` of the column `COL_PG`.

Example: Value-Combination table to be used in article code schemes

	A	B	C	D
1	#	COL_WIDTH	COL_DEPTH	$FAN
2	1	600	600	066
3	2	800	600	086
4	3	800	800	088
5	4	1000	600	106
6	5	1000	800	108
7	6	1200	800	128

This example shows a value combination table which calculates some digits of a final article code, which depend on the width and the depth of an article (see columns `COL_WIDTH` and `COL_DEPTH`). The column `$FAN` contains the value which is inserted into the final article code. The name of this column is fixed and cannot be defined freely. It is important that the table always returns a unique value for a specific value combination. This table has to be used in a user defined code scheme for final article code generation 136.

5.3.14 Dialog Global prices

Global prices are price modules, which are article-independent within a product line. They are taken into account for all the articles when calculating the price.

They are distinguished into the following price levels:

- Upcharges
- Discounts

The price list, in which the global price modules are captured, is selected from the selection list. As default, this control element is set to the active price list of the distribution region.

Fields

Variant-condition	Indication of the variant condition under which the price module is valid. The valid variant conditions are determined at runtime by evaluating price relationships.
Price	This field contains the value of the price module in the defined currency.
Currency	This field determines the currency used for indicating the price. The suggestion value of this field corresponds to the currency of the price list the price module is assigned to. The value "%" is permitted in this field to generate a relative price.
Type	The type 70 of the price module is determined here.
Text block	In this field, a price module can be assigned to a text block 95. The text block is used by the runtime environment for displaying the order list.
Rounding	Using this field a optional rounding rule 144 can be assigned to each pricing component.
Valid from	This field defines the date of the first day from which on the price is valid.
Valid to	This field defines the date of the last day until which the price is still valid.
Rule	Indicating a calculation rule modifies the manner of the price calculation.

Pricing rules

The following calculation rules are allowed for relative prices of the discount level:

1	The price is calculated in relation to the base price
2	The price is calculated in relation to the price accumulated during the price calculation.

No calculation rules are supported for global upcharges. For discounts a rule has to be specified.

Index prices

An OCD price list normally contains real prices which consist of a discrete value and currency. Alternatively so called index prices can be used. Index prices do not have any discrete value and currency. Instead an abstract price index is assigned to them which can be used to look up in an external price table for a real value.

A field besides the price list selector can be used to switch between real and index prices.

The price index must be a positive integer value.

Currently it is not recommended to use index prices in standard data creation and distribution processes using pCon.creator ORG-Module and pCon.update. Index prices and the mentioned external tables are not part of the OCD specification. Furthermore the processes of creation, distribution and processing of index prices are not standardized. Index prices are just a special extension to realize special requirements of the German living furniture market.

5.3.15 Dialgo Global packaging

Article independent packaging information, which is used for all articles in a product line, can be maintained in this dialog.

Global packaging data can be useful, to centrally maintain packaging information 93, which is the same for all articles. e.g. additional weights of accessory parts which is just selected as variant of an article e.g. a chair's head rest.

The fields and their meaning is exactly the same like for article specific packaging information 93. Though it is mandatory to specify a unique variant condition. This is a character string, which is used in packaging relations 126 to activate the packaging data record at runtime.

5.3.16 Dialog Roundings

A special rounding rule, which is defined in this dialog, can be assigned to each article related 87 or global pricing 142 component.

Each rounding rule is identified by its name and can contain multiple rules to round pricing components.

Fields

#	This field is used to define the relative order of rounding rules. It contains a number which specifies the relative position of the record in the order of rules.
Rounding	This field defines the rounding type (see below).
Precision	The float value in this field controls the precision of the rounding operation. The value defines the amount which can devide the price value without any residue. (see below)
Surcharge before	The value of this optional field is added to the price value before rounding.
Surcharge after	The value of this optional field is added to the price value after rounding.
Prices from	This field is used to filter the pricing components to which this rounding rule is applied. The rule is just applied to price values greater equal than the value in this field.
Prices to	This field is used to filter the pricing component to which this rounding rule is applied. The rule is just applied to price values less than the value in this field.

Rounding types

commercial	Values, which end to x.5, are rounded up.
down	
round to even	Values, which end to x.5 are rounded to the next even value.
up	

Examples: Rounding precision settings

A factor sets the rounding precision. The following examples illustrate the function of this factor.

1	Rounding to whole values (to whole euros)
0.1	Rounding to one place after the decimal point
0.01	Rounding to two places after the decimal point (exactly to the cent)
0.5	Rounding to half values (accurate to 50 cents)
0.25	Rounding to fourth values (accurate to 25 Cents)
0.05	Rounding to twentieth values (Rappen-rounding)
10	Rounding to the second place before the decimal point (to ten euros)
100	Rounding to the third place before the decimal point (to hundred euros)

5.3.17 Dialog Taxation schemes

Sales data does not contain specific tax values. So called taxation schemes define the tax types and categories which are used to calculate tax values at runtime. Taxation schemes are optional. This dialog is just used to define taxation schemes. They have to be assigned to articles in article master 92.

Fields

Label	A unqiue internal name which identifies the taxation scheme. Each taxation scheme can define different tax types for different countries.
Country	The country for which the tax type is defined. The value has to be a valid two digit ISO-3166-1 country code.
Region	Optionally this field can define a specific region within a country, where the tax type is applied. This field can be left empty or it has to contain the region specific part of a valid ISO-3166-2 region code.
Tax type	This field defines the tax type (e.g. value-added tax)
Tax category	This field defines a category which is used for product of the specific tax type.
#	Defines a relative order in which the tax types are used if multiple tax types are defined. Finally the OFML application can define another order of evaluation depending on legal restrictions.

The standard VAT rate is used for all articles which don't have got a taxation scheme which defines the VAT taxation category.

Currently the calculation of Value-Added Tax (VAT) cannot be controlled using taxation schemes. OFML runtime applications use always the standard rate for value-added tax calculation. (State of 2013-05-30)

Taxation schemes are used to define tax categories to calculate ECO-Contribution in France. Currently OFML runtime applications only support calculation of the ECO-Contribution as defined by Valdelia for the french office furniture market.

5.3.18 Dialog Classification systems

This dialog is used to defined the classification systems which are used to optionally classify articles 147.

The classification systems are defined for each manufacturer (→ See the context menu of the manufacturer node in pCon.creator Explorer)

Following classification systems are supported:

- ECLASS
- UNSPSC

Licensing notes

Some of the supported classification systems are managed and owned by other organizations. Classifying products may require to obtain a valid license from these organizations. EasternGraphics is not the responsible to grant that you have permission to use these classification systems. Before using a particular classification system it is recommended to check the licensing terms provided by the owning organizations of these classification systems, and verify that your organization is allowed to classify products.

Notes - ECLASS

According to OCD Specification articles can be classified using ECLASS. Using ECLASS it is essential which ECLASS version is used. The <Major>.<Minor> version must be defined if field version. It is possible to use different ECLASS versions if necessary.

Please note: According to the ECLASS definition, different ECLASS versions may have incompatible structures. After updating or changing to a newer versions, it may become necessary to update the classification of some articles, which are assigned to a ClassId which became incompatible.

Notes - UNSPSC

Articles can be classified using the UNSPSC classification system. According to OCD specification defining the particular version of the USNPSC is not supported. The field version must be left empty.

Notes - Export OCD Format version

The classification of products in OCD is supported since pCon.creator 2.19 . It is only supported for OCD product lines which are using the OCD Export format version OCD 4.x or newer.

The classification data is not exported for product lines which are still using older OCD Export format versions e.g. 2.x. Upgrading such product lines to a newer OCD Format Version 80 is necessary to be able to export the classification data.

5.3.19 Dialog Export filters

This entry form allows you to define the filters which are applied to the distribution regions[79] to be exported when exporting the data. They are mainly used for the export of derived distribution regions[65].

Fields

Name	A unique name of the filter is indicated here.
Description	Here you can enter a description of the filter for a more detailed explanation.
Include empty fields	All records which do not have any filter marking or which are marked with * are automatically included in the filtering.
Include-criteria	Patterns of filter markings with included records. The following placeholders can be in these patterns: * - multiple arbitrary characters ? - exactly one arbitrary character
Exclude-criteria	Analogously to the include-criteria, patterns where the records are excluded during the filtering are entered here.

Prior to the data creation, the specific filter markings and their meaning for the records to be filtered have to be defined project-specifically.

A record to be filtered on export will be exported if none of the exclude criteria an at least one include criteria matches its filter tag.

Product lines[80], articles[86], property values[105] and article properties[90] can be filtered for each distribution region[79].

5.3.20 Dialog Status definitions

Status definitions allow you to label records according to individual criteria for further processing. Furthermore, these status definitions can be linked with predefined states in transactions or synchronizations. The results of transactions are thus persistently displayed.

Fields

Label	A unique label for the status definition is entered here.
Color	A color, which is used as background color in the visualization, is selected here. The following values are possible: • 1 – Red • 2 – Yellow • 3 – Green
Transaction	A status definition can be optionally assigned to a specific transaction status 150 in this field.

In data creation dialogs an additional column, containing a records user defined status, can be shown or hidden using the keyboard shortcut F12.

Several function to synchronize or import data make use of status definitions to show the result for each record. Using such functions may remove already applied status definitions to the records in the database.

5.3.20.1 Transaction status

The transaction states are distinguished between the places where they are set.

Source data

Deleted (Source)	The record does not exist in the destination data.
Changed (Source)	The record is different in source and destination data.
Appended (Source)	The record was appended to the destination data.
Identical (Source)	The record is identical in source and destination data.

Destination data

New (*Destination*)	The record is new in the destination data.
Changed (*Destination*)	The record is different in source and destination data.
Appended (*Destination*)	The record was appended to the destination data.
Identical (*Destination*)	The record is identical in source and destination data.

5.3.21 Dialog Export to OFML

The Explorer context menu includes the menu item "Export to OFML". The export can be launched on the following levels of the OCD data format in the workspace:

- Module
- Manufacturer
- Distribution region
- Product line

To export one or several commercial product lines, an export path has to be indicated as target. In the target directory, the data are created according to the storage structure defined in DSR Specification.

The export path of the target data is centrally and user-specifically stored in the workspace. The last export path is automatically set in the OCD module as well as in the ODB and OAS module for the next export performed from the same workspace.

UNC paths are not supported as destination path for the export.

Exporting one product line

When exporting one individual product line, mere partial exports 153 can be optionally performed. In this case, only the selected areas are exported.

Exporting multiple product lines

If the export is launched on the level of the distribution region, manufacturer, or module, you can select the product line to be exported. Initially, the product lines below the level on which the export is performed are already selected.

Buttons

	Using this button settings to automatically run an OFML application after successful export can be configured. (see also)

5.3.21.1 Supplement exports

XCF test catalog

A simple flat XCF catalog is created, which contains all the articles of the product line exported. This catalog is appropriate to test the data in an OFML runtime environment.

Catalogs can be created with the help of the OAS module in pCon.creator.

DSR registration

This option generates from a default template a simple DSR product line registration for each product line exported. This registration allows you to register the data in an OFML runtime environment.

The generated registration file is appropriate to test the catalog. For a release-capable data version, a product line registration file has to be created, which takes all the dependencies of a complete OFML package into account.

5.3.21.2 Partial exports

Selecting individual sections of a product line for export reduces the export time. This allows you to quickly verify individual corrections of a package.

The following sections can be selected:

- Article master
- Article codeschemes
- Classes, properties and values
- Relation-objects and relations
- Value combination tables
- Texts
- Rounding rules

- Article type mapping
- ODB parameter mapping
- Material mapping

 A product line has to be exported completely before using a partial export the first time. To ensure completeness of the data at the end of a data maintenance cycle the product lines should exported completely at the cycle's end.

5.4 OAM user interface

5.4.1 Dialog Article mappings

On this dialog an article is assigned to an OFML planning type representing it at runtime. Furthermore, an ODB object can be assigned to it for the graphical visualization. Additional parameters for the graphical visualization can be entered as well.

If an ODB database is available, the preview image and the description of a selected ODB object are displayed in the detail view.

Fields

Article code	This field shows the article number.
OFML type	The OFML planning type[157] representing the article is determined here. For instance the OFML Type may decide whether the article has got a graphical representation from ODB or is just a sales only article without a graphic from ODB. The look up list usually contains following default values if the types have not been manually defined, yet: • ANY - a sales only article without graphical representation from ODB • ODB - an article with graphical representation from ODB
ODB manufacturer	The manufacturer code of the OFML package of which an ODB object is used for the graphical visualization of the article.
ODB product line	The serial code of the OFML package of which an ODB object is used for the graphical visualization of the article.
ODB name	The name of an ODB object graphically representing the article can be determined here. If an article is mapped to an ODB object, an according OFML type must be selected.
Mapping columns	The assigned mapping columns[156] are displayed with their respective names. The respective value can be flexibly entered.
ODB parameters	Additional ODB parameters to generate the ODB object are entered in this field, according to the syntax of the OAM version used.

On this dialog you cannot create or edit articles. This has to be done in the article master[86].

Sales only articles don't have any graphical representation which is defined in the ODB data. If a sales only article is inserted into a graphical planing, the OFML application may use a default graphic to visualized them. This default graphic may be simple cube or a small product information label showing the catalog image including the article code and text.

The field OFML type is optional. If the OFML type is not selected, the OAM export will map the article automatically as sales only article without graphical representation from ODB. In this case the export uses the special OFML type `::ofml::xoi:xOiDummyArticle` instead the `::ofml::xoi:xOiAnyArticle` as sales only article. On the one side this special export behavior allows using and testing new OCD article in OFML applications without finalizing the article mapping. On the other side quality assurance applications like EGR-TestSuite can log warnings about missing or incomplete article mappings, which should be verified and fixed before publishing the data.

5.4.1.1 Material mappings

The material mappings allows you to determine which properties control which material categories.

Fields

Property	This field allows you to select a property of the article controlling a material.
#	This field determines the relative order of the records used for the material mapping.
Material category	This field contains the material category controlled by the property.
Mapping table	The mapping table 158 containing the respective assignments of a property value to a material is selected here.

Instead of costly maintenance of additional material mapping tables, in ODB graphical data creation it is possible to use functions to evaluate the sales properties directly to determine according materials and colors. But for this approach the data creation project should ensure that the sales property values of colors and materials are chosen in a way that they are unique and are directly applicable as material names.

5.4.2 Dialog Mapping column assignment

In this form, you can product line-centrally define up to nine ODB parameters, whose values can be maintained in separate mapping columns during the article mapping.

Furthermore, you can define a separate ODB parameter to transfer the article number.

Fields

Column name	The column has to be selected in this field. Up to nine columns are available for the selection, plus the column with the article number.
Column description	The name, under which the column is displayed in the article mapping, is entered here.
Variable type	The type of the corresponding ODB parameter has to be selected.
Variable name	The name of the ODB parameter has to be defined here. This is a symbolic identifier. Numerical and alphanumerical characters as well as the underscore are permitted. The first character must not be a number. Furthermore, a variable name must not be identical to a name already assigned to a property.

Variable types

- character
- number
- symbol

5.4.3 Dialog OFML types

The OFML planning types, which represent the articles in the OFML planning system at runtime, are defined here.

Fields

Label	This field determines an identifier, under which the OFML planning type is used within the article mapping 154.
Manufacturer	This field contains the OFML code of the manufacturer of the OFML class
Product line	This field contains the OFML code of the product line containing the OFML class.
Name	The name of the OFML class has to be entered.

The article planning types entered in pCon.creator CLS module can be look up the fields Manufacturer, Product line and Name.

Default Types

Usually it is not necessary to manually enter OFML article planning types on this dialog, except the OFML data development project uses special types. pCon.creator ensures that the default planning types are automatically available to select on article mapping 154 dialog. This includes the following default planning types:

- sales only articles without graphical representation from ODB (→ `::ofml::xoi::xOiAnyArticle` - *ANY*)
- articles having an ODB object for visual representation (→ `::ofml::oi::OiOdbPlElement` - *ODB*)

In old product lines which still use OAM 0.1 articles having an ODB object are mapped using the old type `::ofml::xoi::xOiBTGPlElement3` (supporting the BTGM list with OAM 0.1). The type `xOiBTGPlElement3` should not be used to map articles in up to date projects using OAM 1.0. The type `::ofml::oi::OiOdbPlElement` should be preferred instead.

5.4.4 Dialog Value material mappings

The tables for the material mapping[155] are defined here. These tables contain the assignments between property values and materials.

Fields

Name	This field contains the name of the mapping table.
Value	This field defines a property value.
Material	This field determines the material name, on which the property value is mapped.

The field material can contain a simple material name without a package identifier. In this case the material is loaded from the central material package which is defined in product line management dialog[84]. But the value may also represent a full qualified material name including an information about the package from which the material will be loaded. Usually it is recommended to have one central material package and omit the manufacturer package prefix in material mappings.

Example: Full qualified material name

A full qualified material name has got the following scheme:

```
::<ManufacturerCode>::<PackageCode>::<Name>
```

e.g.

```
::xyx::basics::mebu
```

5.4.5 Dialog Global material mappings

Global material mappings provides article independent control of assignments of properties to material categories. These assignments are valid for all articles within a product line.

Fields

Property	This field allows you to select a property of the article controlling a material.
#	This field determines the relative order of the records used for the material mapping.
Material category	This field contains the material category controlled by the property.
Mapping table	The mapping table 158 containing the respective assignments of a property value to a material is selected here.

Part VI

Catalog data development

EasternGraphics
visualize your business

6 Catalog data development

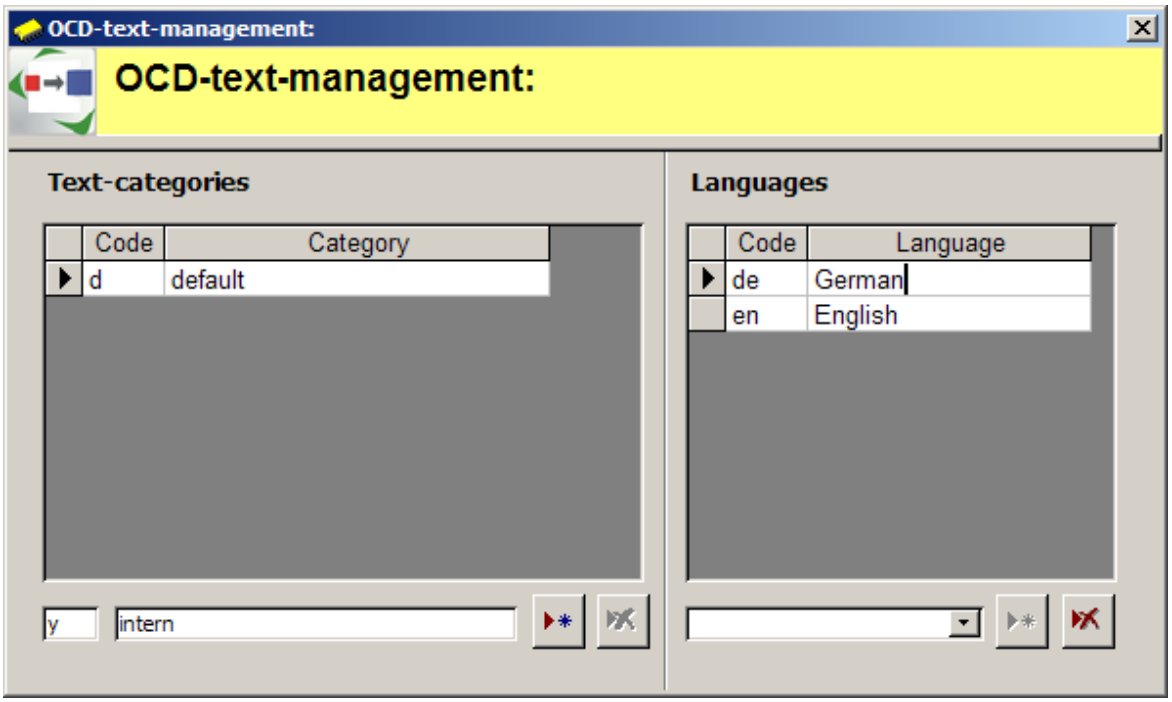
6.1 Overview

The OAS module enables the data format OFML Article Selection (OAS) for catalog data entry in pCon.creator. The catalog data can be exported to XCF (Extensible catalog format).

Several verification routines, export filters and export of derived distribution regions simplify the process of generating consistent catalogs for different markets.

6.2 OAS user interface

6.2.1 Text management



This entry mask manages the text categories 165 and languages 166 in the current workspace.

Several text categories can be created, which can contain several different languages. The categories are entered in the left area, while the languages of a selected category can be edited in the right area of the form.

It is obligatory to create at least one text category with at least one language.

6.2.1.1 Text categories

Text categories are used to distinguish the use of texts in distribution regions[171]. This allows you, for instance, to create texts describing the catalog entries in the specialized trade as well as texts which are intended for an internal use only. When defining derived distribution regions[172], the text categories valid therein can be determined.

Fields

Code	A unique one-digit key to identify the category is entered.
Category	The name of the respective category is entered here.

Buttons

	Adding the new text category.
	Delete the selected text category.

A text category can only be deleted if it does not contain any language.

Text categories, which do not contain languages, are automatically removed when exiting the text management. A note will be displayed.

6.2.1.2 Languages

A text category can contain several different languages. They can be entered by means of the two-digit ISO-639 language abbreviation or selected from the selection list.

Fields

Code	A unique two-digit code according to the standard ISO 639 is indicated to identify the language.
Language	This field shows the name of the language. The language cannot be edited but directly depends on the language code.

Buttons

	Add the new language to the selected text category
	Delete the selected language
	Synchronize the text categories and languages with OCD data in the current workspace.

6.2.2 Export filter

This entry mask allows you to define the filters which are applied to the distribution regions to be exported when exporting the data. They are mainly used for the export of derived distribution regions.

Fields

Name	A unique name of the filter is indicated here.
Description	Here you can enter a description of the filter for a more detailed explanation.
Include empty fields	All records which do not have any filter marking or which are marked with * are automatically included in the filtering.
Include-criteria	Examples of filter markings with included records. The following placeholders can be used: * - multiple arbitrary characters ? - exactly one arbitrary character
Exclude-criteria	Analogously to the incluce-criteria, examples where the records are excluded during the filtering are captured here.

Prior to the data creation, the specific filter markings and their meaning for the records to be filtered have to be defined project-specifically.

A record to be filtered on export will be exported if none of the exclude criteria an at least one include criteria matches its filter tag.

Product lines 173 and entries in catalog structure 182 can be filtered.

Articles already being filtered in OCD won't be filtered separately in OAS. Please use OCD-Articlesynchonization 218 on export to automatically synchronize the exported catalog.

6.2.3 Status definitions

Status definitions allow you to label records according to individual criteria for further processing. Furthermore, these status definitions can be linked up with predefined states in transactions or synchronizations. The results of transactions are thus persistently displayed.

Fields

Label	A unique label for the status definition is created here.
Color	A color, which is used as background color in the visualization, is selected here. The following values are possible: • 1 – Red • 2 – Yellow • 3 – Green
Transaction	A status definition can be optionally assigned to a specific transaction status in this field.

The linking up with transaction states is currently without function and intended for future versions.

6.2.3.1 Transaction status

Die Transaktionszustände werden nach dem Ort unterschieden, wo sie gesetzt werden.

Quell-Daten

Gelöscht (Quelle)	Der Datensatz existiert nicht in den Zieldaten.
Geändert (Quelle)	Der Datensatz unterscheidet sich in den Quell- und Zieldaten.
Angehängt (Quelle)	Der Datensatz wurde an die Zieldaten angehängt.
Identisch (Quelle)	Der Datensatz ist in den Quell- und Zieldaten identisch.

Ziel-Daten

Neu (Ziel)	Der Datensatz ist neu in den Zieldaten.
Geändert (Ziel)	Der Datensatz unterscheidet sich in den Quell- und Zieldaten.
Angehängt (Ziel)	Der Datensatz wurde an die Zieldaten angehängt.
Identisch (Ziel)	Der Datensatz ist in den Quell- und Zieldaten identisch.

----------OLD_TEXT----------

The transaction states are distinguished between the places where they are set.

Source data

Deleted (Source)	The record does not exist in the destination data.
Changed (Source)	The record is different in source and destination data.
Appended (Source)	The record was appended to the destination data.
Identical (Source)	The record is identical in source and destination data.

Destination data

New (*Destination*)	The record is new in the destination data.
Changed (*Destination*)	The record is different in source and destination data.
Appended (*Destination*)	The record was appended to the destination data.
Identical (*Destination*)	The record is identical in source and destination data.

6.2.4 Text profiles

Text profiles allow you to define schemes for catalog texts. The actual texts of the catalog entries of the type Article are made with the help of these templates during the export by replacing the actual values.

Fields

Label	A unique label to identify the text profile.
Template	This field contains a template defining the structure of the catalog text. It can contain the following placeholders which are replaced with the real values during the export.

Placeholder

${ArticleCode}$	This placeholder is replaced with the current article number of the catalog entry.
${VariantKey}$	This placeholder is replaced with the current variant key of the catalog entry.
${CatalogText}$	This placeholder is replaced with the catalog text of the catalog entry in the respective text category and language.

Example

Using the template

```
"[${ArticleCode}$] ${CatalogText}$"
```

when exporting an article with the article number `C1234` and the German catalog text "office swivel chair" results in the following catalog text during the OFML export:

```
"[C1234] Office swivel chair"
```

6.2.5 Distribution region management

Distribution regions are a logical combination of product lines with specific validity. They usually map different markets.

Fields

OFML code	A unique abbreviation to identify and register the distribution region. It can be selected from a selection list of predefined codes for regions.
Label	A name for the distribution region.
Derived from	The master distribution region from which the present is derived 172.
Text category	The text category valid in this distribution region is entered here.
Default language	This value defines the default language of the distribution region.
Product line filter	Defines the filter for product lines to be applied during the export.
Structure filter	Defines the filter for the catalog structure to be applied during the export.

The defined default language is used to preview texts on the further forms within the respective distribution region.

6.2.5.1 Derived distribution regions

Derived distribution regions allow an efficient creation and management of different data versions from a central data state.

Active export filters can be assigned to a distribution region.

The following data can be separately filtered:

- Product lines 173
- Catalog structure 174

A filtering of articles is done in the OCD module. The synchronization 218 during the export automatically keeps the catalog consistent with the available commercial data.

No product lines can be created in derived distribution regions.

It is not possible either to derive a distribution region, which already contains product lines, from another one. If you still desire to do so, you first of all have to delete the productlines.

6.2.6 Product line management

Fields

OFML code	A unique product line abbreviation to identify the product line. The value in this field must correspond to the rules of a valid OFML designator and has to be in small letters.
Label	This field contains a label of the product line.
Version-Major	Shows the major version number of the product line. It starts with 1.
Version-Minor	Shows the minor version number of the product line. It starts with 0.
Version-Build	Shows the build number of the product line. It starts with 0.
Release date	Date of the release of the product line.
Remarks	Remarks about the product line can be stored here.
Text profile	This field allows you to select the Text profile 170 applied to the catalog texts when exporting articles.
Export filter	A character string (filter marking) which is evaluated when applying the product line filter of the distribution region of this product line.
Image path	Instead of using the workspace's local resource path to save external images, an explicit path can be specified in this field, where catalog images are maintained centrally.

Buttons

	The release date is set to the current system date.

The complete version of the product line is made up of the individual fields to capture the version number. It is used to register the product lines in an OFML application system.

The correct data distribution of data updates requires the diligent maintenance and incrementation of the version number prior to an export. This recommendation can be found in the DSR specification.

6.2.7 Catalog structure

The entry mask Catalog structure is divided into two parts. The left area displays the catalog tree with a fixed node of the current manufacturer on the topmost level. The structure of the current catalog is always below this node. This area allows you to build and edit the structure.

Catalog entry types

The different catalog entry types are marked with the following symbols in the tree structure.

	Article 185
	Folder 186
	Information 187
	Metaplanning 189
	Function call 190
	Geometry 191
	Group 212
	Scene 193
	Method call 194
	Metatype 195

Detail view

The detail view in the right area is divided into the following areas:

- **Selection list**
  The selection list contains the currently captured catalog entries and allows different filter operations of this list to efficiently insert several entries into the catalog structure.

- **Catalog entry**
  The tab shows the details of the currently selected entry in the tree structure. The view corresponds to the detail view of the form catalog entries 183. Furthermore, the number of references 181 to this catalog entry is displayed. Filter markings for the structure filter 182 are captured in the lower area of this view,

- **Resources**
  This tab allows you to access thegeneric resource entries 204 of the currently selected catalog entry.

View modes of the tree view

The tree view features the following display modes:

- Display of the article numbers
- Display of the catalog texts

The button allows you to switch between these display modes. 

The catalog texts are displayed in the default language of the distribution region.

6.2.7.1 insert entries

Selection list

The tab Selection list contains all catalog entries[183] already captured. The catalog entries to be inserted can be selected in this list. They will be inserted at the place currently selected in the tree with the help of one of the following steps:

- Using the button
- Key combination SHIFT and CURSOR LEFT

The selection list can be filtered in the following way:

- acc. to a catalog entry type
- Hiding catalog entries already contained in the structure
- Applying a user-defined filter
- Using a quick filter
- Show only new articles
- Show articles of specific product data packages

Using the quick filter restricts the selection list just as it is typed in.

The filter to show only articles, which are new, can be activated from the context menu of the selection list. This filter evaluates the status of the last article synchronization[215].

The filter to show only articles of specific product data packages, can be activated from the context menu of the selection list, too. It is only available if the catalog package contains articles from multiple different product data packages.

Selection dialog

The context menu allows you to insert catalog entries directly in the catalog structure. A selection dialog appears. This dialog looks up already existing catalog entries when defining a catalog entry. If a catalog entry is taken over, a respective reference to a found entry will be inserted. If no entry was found, a new one will automatically be created.

6.2.7.2 modify entries

Modifying the details of a catalog entry

When modifying entries in the tree structure, please note that catalog entries are stored as references in the tree structure. Modifications of the settings within the detailed view thus affect all the references. The reference itself remains unchanged.

Modifying the reference to a catalog entry

The funtion to modify the reference of a catalog entry is in the context menu of the respective node in the tree structure. The F2 key allows you to access this function via the keyboard. A selection dialog will appear to define the catalog entry whose reference is being modified.

If an already existing catalog entry is successfully found, the reference to it will be taken over. Otherwise the newly defined entry will be created and the reference to this one will be taken over.

The partial tree below a modified folder remains unchanged.

6.2.7.3 move entries

Moving with Drag & Drop

A node can be moved in the structure with the left mouse-key held down. If the node is dropped on a folder node or on the manufacturer node, the dragged entry will be inserted as its last child, while it will be inserted as the previous neighbour when dropping it on target nodes of another node type.

You can modify the paste behavior by holding down the following keyboard keys during Drag & Drop:

CTRL	The dragged entry is always pasted as previous neighbour of the target node.
ALT	The dragged entry is always pasted as subsequent neighbour of the target node.

Modifying the paste behavior has no validity for the manufacturer node as the target.

If an entry is dropped in an empty area of the tree structure, the manufacturer node is automatically the target.

If a partial tree is moved to another product line of the same workspace when two forms are open, it will not be moved but copied.

Moving with the help of the keyboard

You can move a selected node within the tree structure with the help of the cursor keys and the CTRL key held down.

CTRL + CURSOR UP	The selected node is moved one position up in its hierarchy level. It becomes at most the first child of its folder.
CTRL + CURSOR DOWN	The selected node is moved one position down within its hierarchy level. It becomes at most the last child of its folder.
CTRL + CURSOR RIGHT	The selected node is indented by one hierarchy level. It becomes the last child of the previous folder.
CTRL + CURSOR LEFT	The selected node is unindented by one hierarchy level. It becomes the next neighbour of the folder in which it was located.

6.2.7.4 copy entries

Partial trees can be copied within a structure. You can also copy a partial tree into another product line of the same workspace when two forms are open.

Copying with Drag & Drop

Catalogs can be copied within the tree structure by means of Drag & Drop. The behavior is analogous to moving an entry with Drag & Drop178. To copy the SHIFT key has to be held down during the Drag & Drop operation.

You can modify the paste behavior by holding down the following keyboard keys during Drag & Drop:

CTRL	The dragged entry is always pasted as previous neighbour of the target node.
ALT	The dragged entry is always pasted as subsequent neighbour of the target node.

Modifying the paste behavior has no validity for the manufacturer node as the target. If an entry is dropped in an empty area of the tree structure, the manufacturer node is automatically the target.

Copying with Copy & Paste

CTRL + C	The selected entry is copied to the clipboard.
F8	see CTRL + C
CTRL + V	An entry previously copied to the clipboard is pasted at the currently selected place in the tree structure.
F9	see CTRL +V

6.2.7.5 delete entries

Deleting an entry from the structure

The DEL key allows you to delete a catalog entry from the structure. You also have access via the context menu of the respective node.

Deleting the subelements of a folder

All subentries of a folder can be deleted with the help of the central function in the context menu of the respective node in the tree structure.

The key combination ALT+DEL executes this operation on a selected node as well.

6.2.7.6 search for entries

Fields

Search	The pattern according to which the search is performed is entered in this field.

Placeholder

The following characters can be used as placeholders when defining the search pattern.

?	This placeholder stands for any character
*****	The asterisk placeholder stands for any number of characters

Buttons

The first entry found is selected in the tree view. If the search result contains several entries, the buttons allow you to navigate between the entries found.

	Go to the next entry found
	Go to the previous entry found

Depending on the currently selected representation mode of the tree view, the search is performed either according to article number or according to catalog text.

6.2.7.7 navigate between references

Catalog entries are inserted into the tree structure as references - either by pasting or by copying them. The catalog entry itself exists only once, even though it appears at different places within the structure. Modifications of the settings of a catalog entry thus affect all the references.

The detailed view shows you the position of the current reference and the total number of references of the catalog entry currently selected in the tree structure.

Buttons

The buttons allow you to switch between the references to the current catalog entry.

	Go to next reference
	Go to previous reference

6.2.7.8 Structure filter

In the catalog structure, records contain markings which are evaluated by the structure filter during the export. These markings are captured in the detailed view of the catalog entry selected.

Fields

Export filter	This field contains the filter marking of the entry in the catalog structure.

see also

6.2.8 Catalog entries

This form centrally captures all catalog entries contained in the catalog. This includes the entries in the catalog structure as well as the entries which can be accessed in an OFML application system, e.g. by means of the article search only.

Catalog entries are classified according to the types explained below. These types can be selected in the upper area of the form. The left area contains a list in form of a table featuring the entries of a specific type. All necessary data can be captured in columns.

The right area contains the detail area displaying all information of an entry selected in the table. This detail area is divided into two tabs. The page "Catalog entry" displays all fields of the entry grouped into ... for editing purposes. Another tab allows you to capture additional generic resources 204, which are assigned to a catalog entry.

The division of the table view and of the detail area can be customized to your individual requirements by moving the splitter.

Fields for identification

Name	The language-independent key to access the catalog entry.
Variant	Different catalog entry types are allowed to appear multiple times within a product line. They are distinguished with the help of this variant key. e.g. articles

The combination of name and variant is the key of the catalog entry. The keys must be unique in their combination within one product line.

Catalog entry type specific

Each catalog entry type 184 has specific fields which are valid for it.

General fields

Furthermore fields of the following general groups can be entered. They are valid for the different catalog entry types:

- Catalog images 196
- Catalog text 197
- Visibility 203
- Interaction modes 203

Command buttons

	Using this command button the export[198] and import[199] of catalog texts can be started. Furthermore the Synchronization with OCD data[215] can be started when articles are selected.

6.2.8.1 Catalog entry types

Article

Catalog entries of the type article correspond to the commercial articles in the catalog as they can be created in the OCD module of pCon.creator. The name of the catalog entry corresponds to the base article number.

An article can appear multiple times in different variants within a catalog.

Fields

Name	The name of the catalog entry corresponds to its base article number.
Variant	Defines a variant key for this catalog entry. Please note that the combination of name and variant must be unique throughout the entire catalog.
Manufacturer	This field contains the manufacturer code of the OFML package containing the commercial article.
Product line	This field contains the product line abbreviation of the OFML package containing the commercial article.
Parameter	Is an optional field to indicate a product data-dependent variant code of the article during the generation. The content of this field follows the coding scheme of the OCD article. This can be, for instance, the concrete KeyValueList or the final article number in case of user-defined schemes.

Buttons

	This button launches the article synchronization 215 with the OCD data of the current workspace. Furthermore the export 198 and import 199 of catalog texts can be started using this button

The fields Manufacturer and Product line define an external package containing the commercial article. These fields are optional and can remain void, if the article is contained in the same package as the catalog.

Please note that external packages used in the catalog are also taken into account when maintaining the dependencies of the catalog product line within the DSR product line registration.

Folder

Folders are elements to structure the catalog. They are the only catalog entries that can contain subelements.

Empty folders in the catalog structure are not exported during the XCF export.

A folder is considered to be empty if it does not contain any subentries or if all its subentries are of the folder type only.

Information

A catalog entry of the type Information displays further information about the product line or the planning for the user. It is represented by an HTML page, PDF file or an external web address in the internet.

Fields

Type	Defines the type of the information entry. One of the following types can be chosen for each information catalog entry: • HTML page • PDF file • URL After choosing one of these types the resource which represents the information can be defined in one of the following fields.
HTML page	Select a HTML page from the external file resources 209, if the type HTML page was chosen.
PDF file	Select a PDF file from the external file resources 210, if the type PDF file was chosen.
URL	The website address, if the type URL was chosen. Additionally alternative language specific URLs can be specified on Generic Resources 204 tab.

A double-click on the field HTML page or PDF file allows you to switch to the entry mask to capture external file resources of this type.

Normally HTML pages and PDF files should be available in a specific version for each used language. Such language specific versions are assigned directly to the selected catalog resource. It is not necessary to define it explicitly for the catalog entry here.

Typical

A typical represents a group of pre-configured OFML articles.

Typicals can be used to provide users predefined solutions of complex configuration and planning situations in OFML catalogs. A user can add this solution into the current planning as a starting point for reconfiguration and planning with installed OFML data. The typical itself is saved in the catalog as external resource file in pCon Exchange Container Format 211 .

Fields

Resource	Referenced the resource file in pCon Exchange Container format 211.
Sync	After starting the command "Synchronize typicals" this field shows the current state of each catalog entry. It shows states if the resource file is missing or the synchronization with resource files has added new catalog entries for new resource files.

Command - Synchronize typicals

The command button  opens a context menu from which the "Synchronize typicals" command can be started.

The synchronization process automatically synchronizes the external PEC resource files from resource folder of one catalog package. For each new resource file a new catalog entry is added automatically. These new entries are tagged with the Sync state "New" afterwards. In case the referenced PEC resource file of a catalog entry is missing this entry is automatically tagged with the sync state "Deleted".

Market specific Typicals

Catalogs containing typicals can be developed using one master distribution region from which derived distribution regions 172 are exported for different markets. Since typicals contain market specific information about articles, the typicals must be used which are up-to-date to the OFML data set for the specific market.

The resource folder 211 of the workspace can contain copies of the typical resource files which have been updated for each derived distribution region. The catalog export will use automatically the PEC resource files of the specific derived distribution region. For each derived distribution region a specific resource folder 211 is used. The name of the resource file for one typical must be the same in each distribution region.

If a separate resource folder for typicals of the derived distribution region is present, the typical resource files are exported from this folder. All typical resource files which are used in the derived distribution region must be available there. If the workspace does not contain an according resource folder for typicals in the derived distribution region, the catalog export will use the resource folder of the master distribution region to export the PEC resource files.

PEC resource files can be updated easily for different markets in batch mode using the pCon.planner ArticleUpdate Plugin. For further questions about the pCon.planner ArticleUpdate Plugin please ask your contact person at EasternGraphics GmbH or send your request to our Support.

Metaplanning

A catalog entry of the type Metaplanning activates a special metaplanning workflow.

Fields

Name	The key in the catalog for this catalog entry.
Parameter	The parameters to call the metaplanning workflow are stored in this field.

Structure of the parameters

The value of the field Parameter must correspond to the following structure.

```
<::package::class::@MPWorkflowID>,<::package::class::MPClassName>[,<Argument>]
```

The indication of the symbol for the MP workflow ID and the MP class must absolutely be fully-qualified.

The indication of an argument is optional.

Function call

Catalog entries of this type call global OFML functions for the currently selected objects of the scene.

Fields

Parameter	This field defines the call of the OFML function.

Parameter syntax

The value of the field Parameter has the following structure:

```
<PackageName>; <Function>(<Parameters>)
```

The package name has to be indicated fully-qualified (`::<Manufacturer>::<Program>`)

An optional placeholder `%SELECTION%` is replaced by the runtime system with a list of the currently selected elements prior to the call (e.g. `[t.e1, t.e2]`). If no element is selected, an empty list is transferred.

Geometry

These catalog entries represent geometries which are saved in external geometry files of specific geometry formats. These geometry files are inserted into a planning instead of a real OFML article. This approach can be used to simply provide non-configurable typicals or preferred preconfigured combinations of products in catalogs. Geometry files can be provided online on pCon.catalog. This catalog entry type provides the possibility to distribute the same symbols to use them offline in an OFML catalog.

A geometry file has to be assigned to this catalog entries e.g. in DWG format.

Fields

Geometry file	Selecting a geometry file from the external file resources 211.

The catalog entry type "geometry" is not supported by all OFML applications. It is dependent from the specific application which geometry formats it supports. Applications which don't support geometry catalog entries, just ignore them and don't present them in the catalog view.

DWG files (Typicals) can be created with pCon.planner from OFML product data, these files contain the information to link them back to OFML product data. This means user who insert these geometries into a planning in pCon.planner, can update the products which are used in these files and reconfigure the products if the according product data is installed. If OFML product data is updated it should be considered in data creation projects, that geometry files in catalogs which depend on the data should be updated, too.

DWG based geometries should not be used as typicals in OFML catalogs since introducing new catalog entry type Typical. pCon Exchange Container (PEC) based typicals provide the benefit to be supported in more OFML applications than DWGs this includes both - offline and online OFML applications.

Group

When inserting a catalog entry of this type, an OFML group is loaded into the scene of the OFML application system. An OFML group is a predefined grouping of objects of one scene, which is persistently saved as a file with the extension .ogrp.

Fields

OFML group	Selecting an OFML group from the external file resources 212.

A group can also be a graphical representation of an article so that using the fields as for the type Article is valid for this type, too.

Important note: The concept of OFML groups is obsolete. OFML groups are only supported for backward compatibility reasons in pCon.creator OAS Module. This feature is subject to be removed in future versions of pCon.creator. In new OFML catalogs this old resource type should not be used anymore. It should be preferred to use catalog entries of type Typical 188 instead. Catalogs which are still using OFML groups should be migrated to use Typicals 188 in PEC file format.

When using groups in catalogs, you have to make sure that the articles grouped therein are compatible with the current commercial and geometric data states.

A double-click on the field OFML group allows you to switch to the entry mask to capture external file resources of this type.

Scene

A catalog entry of the type Scene allows you to replace the current scene in an OFML application by a predefined scene. A persistently saved scene file with the file extension .fml will be loaded.

Fields

OFML scene	Selecting an OFML scene file from the external file resources 212.

Important note: The concept of OFML scenes is obsolete. OFML scenes are only supported for backward compatibility reasons in pCon.creator OAS Module. This feature is subject to be removed in future versions of pCon.creator. In new OFML catalogs this old resource type should not be used anymore. It should be preferred to use catalog entries of type Typical 188 instead. Catalogs which are still using OFML scenes should be migrated to use Typicals 188 in PEC file format.

A double-click on the field OFML scene allows you to switch the entry mask to capture external file resources of this type.

Method call

Catalog entries of this type call specific OFML methods on the selected object in the scene of the OFML application.

Fields

Parameter	This field describes the calling of the OFML method on the object selected in the scene.

Parameter syntax

The value of the field Parameter has the following structure:

```
<::package::class | @Interface>; <methodename>(<Parameters>)
```

The fully-qualified type or the interface is defined on which the OFML method defined after the semicolon is called. Furthermore the parameters to be transferred at the call are to be indicated.

The method is only called with its parameters if the selected object is of the type indicated (::package::class) or if it supports the interface indicated.

Metatype

Catalog entries of the type Metatype are direct references to a metatype in the catalog. They can be stored in different variants.

Fields

Name	The key in the catalog for the metatype
Variant	Defines a variant key for this catalog entry. Please note that the combination of name and variant must be unique throughout the entire catalog.
Manufacturer	This field contains the manufacturer code of the OFML package containing the OFML metatype.
Product line	This field contains the product line abbreviation of the OFML package containing the metatype.
Parameter	Is an optional field to indicate specific parameters to generate the metatype.

The fields Manufacturer and Product line define an external package containing the metatype. These fields are optional and can remain void, if the metatype is contained in the same package as the catalog.

Please note that external packages used in the catalog are also taken into account when maintaining the dependencies of the catalog product line within the DSR product line registration.

6.2.8.2 Catalog images

Images for visualization at runtime can be assigned to catalog entries.

Fields

Catalog image	Selecting an image used to visualize the catalog entry within the OFML application.
Print image	A separate image with a higher resolution can be selected for the catalog entries of the type Article, Metatype, and Group, which is used to print a form.

The selectable images must have been captured as an external file resource 207.

A double-click on these fields allows you to switch to the entry mask to capture the external file resources.

Automatic Image Assignment

It is possible to assign images 207 from the resources automatically to catalog entries. This function can be found in context menu of the grid cells in column Catalog Image and Print Image. The images must be already imported to the catalog package.

Following rules to look up for an image file name can be used:

Article Code	`<ArticleCode>.<FileExtension>`
Article Code and Variant	`<ArticleCode>___<Variant>.<FileExtension>`

These rules are equal to the file names created by EGR-GeometryCreator .

It is important to take care that the image file names are unique. And an image file is not used with the same name but different file name extension. Otherwise unique assignments may not be possible.

Preview of the images

The detail area displays the selected images in a preview. This is done on a scale of 1:1. If the selected image is larger than the space available in this form, the image is cut at the rims and displayed with a red dashed frame.

If the external file resource cannot be loaded, no preview is displayed.

If the external file resource is available but cannot be displayed, the following symbol is shown:

6.2.8.3 Catalog texts

A text is assigned to each catalog entry. This text can be captured in different categories or languages. A field at the end of the table view is available for each category-language combination.

The detailed view allows you to display a text in two languages at the same time for editing purposes. The area Catalog texts therefore provides respectively one selection field to select the language and one text field to edit the text in the defined language.

Localized catalog texts can be exported to Excel for external translation. They can also be reimported back. Using the command button  the export 198 and import 199 of catalog texts can be started.

The default settings in the detailed view are always the text category and language selected for the distribution region. The selected language is also preserved upon switching the catalog entry as well as the catalog entry type until the form is closed.

Export catalog texts

Translatable texts can be exported to translate them externally. Currently just a specific excel data format 200 is supported. The catalog text of each catalog entry type is exported. It is not required to export e.g. catalogtexts of articles and folders separately.

Settings

Text catagory	In this field the text category has to be selected of which the translatable texts are exported.
Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
Source language	This field configures the source language. The texts of this language are export as reference for the translators.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
all packages	Normally the text blocks of the current product line are exported. With this option the text blocks of all product lines in the current distribution region are exported in one step.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Export	Start the export after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

Import catalog texts

After external translation of exported texts [197] this updated texts can be reimported to pCon.creator. The used data format is the same like the exported excel data format [200].

Settings

Text catagory	In this field the text category has to be selected of which the translatable texts are exported.
Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
all packages	Normally the text blocks of the current product line are exported. With this option the text blocks of all product lines in the current distribution region are exported in one step.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Import	Start the import after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

The excel file format contains additional information to identify the text blocks correctly. For this reason only excel files which have been exported [197] can be reimported. For this reason it can be possible that excel files cannot be reimported if e.g. these files have been created newly while the external translation process. Please contact support in such cases.

The excel files have to be closed. Otherwise they cannot be imported in any case.

The import automatically tags the modified text records with a status definition [168] "Updated". Maybe the field with the user defined status has to be shown explicitly.

Excel text data format

The following section describes the supported Excel data format for export and import of translatable texts.

Filenames

A specific excel workbook is created for each language. The file name of these workbooks identifies the text category and language. It is build in the following scheme:

```
pcr_text_sel_oas_<category>_<language>.xls
```

where <category> matches the single-digit code of the text category and <language> matches the two-digit target language code (ISO-639).

Workbook

A workbook contains all translatable texts of multiple product lines and catalog entry types. Translatable texts of different catalog entry types are saved in different worksheets.

Worksheet fields

A	***Product line***	The OFML code of the product line.
B	***Name***	The name of the catalog entry.
C	***Variant***	The variant key of the catalog entry
D	***Source***	The translatable text in the source language.
E	***Target***	The translated or updated text in target language.

Using the Excel workbook to transfer translations please note:

- Translations have to be entered in column E. Only values in this column should be changed. Values in other columns must not be changed.
- Additional Values must not be added in additional columns.
- It is not recommended to add or delete records. Data developers must add translatable records in workspaces first. The purpose of the Excel workbook is just to transfer translations.
- A simple worksheet protection is enabled to ensure these recommendations.
- If users do not comply with these recommendations, problems and errors may occur re-importing the translations to pCon.creator.
- The content of Excel workbooks with translations should not be copied into new empty workbooks. These new Excel workbooks can't be re-imported to pCon.creator directly. Since pCon.creator expects additional essential information in the workbooks to be re-imported (see below for further details).

Custom document properties

The excel workbook contains additional custom document properties to identify the data. Following properties are used:

pcr_appid	An identificator of the application system. The value "pcr" is fixed for this property.
pcr_dbtype	The data format type. The value "text" is fixed.
pcr_FormatDomain	The code of the pCon.creator data format domain. The value "SEL" is fixed.
pcr_FormatName	The name of the pCon.creator data format. The value "OAS" is fixed.
pcr_FormatVersion	The version of the excel data format. The value is "1.0.0"
pcr_TextCategoryCode	The single-digit code of the text category where the text blocks belong to.
pcr_LanguageCode	The two-digit target language code (ISO-639)
pcr_LocItemTable_<texttype>	For each text type these property define the name of the excel worksheet where the according text blocks are saved. See below for supported <texttype> values.

Supported text types

article	Catalog texts of articles
folder	Catalog texts of folders
info	Catalog texts of information catalog entries
metaplanning	Catalog texts of metaplanning catalog entries
function	Catalog texts of function calls
metatype	Catalog texts of metatype catalog entries
group	Catalog texts of OFML groups
scene	Catalog texts of FML scenes
method	Catalog texts of method calls to OFML classes
geometry	Catalog texts of geometry catalog entries

6.2.8.4 Visibility

A catalog entry can be visible or hidden in different situations when visualizing the catalog in an OFML application system.

Fields

3D - Planning	Visibility of the entry in 3D representation of the planning mode of the OFML application
2D - Planning	Visibility of the entry in 2D representation of the planning mode of the OFML application
Planning general	Visibility of the entry in the planning mode of the OFML application
configuration	Visibility of the entry in the configuration mode of the OFML application
not in basket	Hiding the entry in the shopping basket mode of the OFML application

In the default settings, the catalog entry is visible in the planning in general, both in the 3D and 2D mode, in the configuration mode as well as in the shopping basket.

In table view a context menu of the visibility fields provides additional functions to switch the according visibility for all records.

6.2.8.5 Interaction modes

Fields

free 2D - positioning	The object hangs on the crosshair cursor to determine the position in the 2D planning.
free 3D - positioning	The 3D feedback mode is used to determine the position of the object in the 3D planning.
free 3D - positioning (programmed mode)	An especially programmed mode to determine the position of the object is used in the 3D planning.

In the default settings, the catalog entry is visible in the planning in general, both in the 3D and in the 2D mode, in the configuration mode as well as in the shopping basket.

The use of the 3D interaction modes excludes each other.

6.2.8.6 Generic resources

On the tab "Resources", so-called generic resources can be additionally entered for each catalog entry.

Fields

Type	This field specifies one of the below described types of the resource record.
Language	Some of the resource types can be specified language-dependently. This field contains the language code for which this record is valid.
Resource	This field contains the value of the generic resource. The value itself depends on the resource type.

Resource types

Addon key	An optional addon key for addon articles	***AD***
Appending parameter for the article creation in the 3D	Additional attach point parameters for interactive 3D article insertion. (3D feedback mode)	***AP***
Description	An additional language-dependent description of the catalog entry. This text is displayed as hint in OFML runtime applications.	***DS***
Geometry name	A fully-qualified or local geometry name can be specified. This geometry represents the catalog entry during free 3D positioning.	***ZN***
Graphical variant	This resource contains additional ODB parameters which specify the graphical variant of the article. `<Key1>=<Value1>, <Key2>=<Value2>` e.g.: `ALIGN=@L`	***OP***
MIME type	Using this resource additional objects / files can be integrated into the catalog. The value of this resource will be specified as: `<MIME-Typ>;<FileName>`	***MT***
Partial planning	Article number and variant key of partial planning type.	***PD***
Parameter for the article creation in 2D	This resource can contain the following values • `W,REF` The article will be placed on top of the underlying article. • `WKL,x` The article will be rotated by x degrees. • `U,x` The article will be moved in X direction by x m • `V,x` The article will be moved in Y direction by x m • `W,x` The article will be moved in Z direction by x m • `W,PUN` Use 3D placing algorithm; automatic placing at height 0.0 will be suppressed. • `POS,MANU` Force manual placing of ADDON articles. • `WKL,MANU` Force manual angle selection, even on attach points.	***UV***
Parameter for the transformation into a special article	This resource can contain the following values • `COM` Sales properties can be modified • `MAT` Materials can be modified • `GEO` Components can be moved, scaled, or deleted Multiple values have to be separated by a comma.	***SA***
Tags	This resource can be used freely to categorize catalog entries. It is not processed directly. But Online-Configurator will pass this information to processing clients. Currently B.S.O. is going to specify a new catalog data format which takes categorization and keywords for catalog entries into account. This resource key does not provide such a general method in advance and it is not guaranteed that this resource will be migrated in the future to such standardized categorization methods.	***TAG***
URL	Can be used to add additional language specific URLs to catalog entries of type Information 187 (URL). The language must be specified.	***URL***

6.2.9 External files

A catalog can reference different external resources. These resources are managed in this form. Each record corresponds to an external file, which is stored in the resource path of the workspace.

Fields

Sync	This field shows the state of the individual records after the last synchronization.
File name	The file name of the resource is captured in this field.

Buttons

	Importing files of the file system as external resources in this product line.
	Synchronization of the records with the external files in the resource directory of the workspace.

6.2.9.1 File types

Images

Images are referenced by catalog entries. They can serve the OFML runtime environment as catalog image in the article selection or be used to print the order list in form of an image with a higher resolution.

Preview of the images

The detail area displays a selected image in a preview. This is done in the ratio of dimensions 1:1. If the selected image is larger than the space available in the form, the image is cut at the rims and displayed with a red dashed frame.

If the external file resource cannot be loaded, no preview is displayed.

If the external file resource is available but cannot be displayed, the following symbol is shown:

HTML pages

HTML pages are referenced by catalog entries of the type Information 187.

Multiple languages

The OAS module manages a language-independent HTML page. Furthermore HTML pages can be stored for several languages.

The HTML pages have to be stored in the subdirectory HTML of the resource directory of the product line in the workspace. The language-independent file is on the topmost level. Translated copies of the HTML page can be stored in further subdirectories. The names of the subdirectories must correspond to the respective two-digit ISO-639 code of the language.

Files referenced on the HTML page have also be stored in the resource path of the HTML page. Thus, the content of this folder is copied 1:1 to the catalog during the export.

When creating the HTML pages, relative path names have to be used.

Each language is highlighted depending on the status of the external resource file. It will be highlighted in green, if the language specific file exists in resource folder. If the file does not exist, it is highlighed in red.

Buttons

	Imports a languages specific version of selected HTML file. The new file is copied to the workspace's resource folder and automatically renamed.
	Opens a preview of the file in standard web browser. This button does not have any function if no web browser is installed.
	Shows the file in windows explorer

PDF files

PDF files are referenced by catalog entries of the type Information 187.

Mehrsprachigkeit

The OAS module manages a language-independent PDF file (main file). Additionally for each PDF file language specific versions can be added for each language, which is used in the workspace. These additional versions are maintained in detail view of the dialog for each record. In OAS catalog the language specific PDF files always have the same file name like the main PDF file, but they are saved in language specific sub folders of the resource directory.

Each language is highlighted depending on the status of the external resource file. It will be highlighted in green, if the language specific file exists in resource folder. If the file does not exist, it is highlighed in red.

Buttons

	Imports a languages specific version of selected HTML file. The new file is copied to the workspace's resource folder and automatically renamed.
	Opens a preview of the file in standard web browser. This button does not have any function if no web browser is installed.
	Shows the file in windows explorer

pCon Exchange Container

Files in pCon Exchange Container Format are used to exchange graphical and sales planning data between different OFML applications.

Such files can be included in OFML catalogs as resource files to represent Typicals 188.

Typicals can be created in pCon.planner and exported as PEC file. These files can be imported 212 into a catalog package in OAS module.

The PEC resource files are saved in a pCon.creator workspace in sub folder:

```
\resource\SEL_OAS\$ManufacturerCode\$DistributionRegionCode\$ProgramCode\pec\
```

The variables `$ManufacturerCode`, `$DistributionRegionCode` and `$ProgramCode` must match the current catalog package's OFML codes on the level manufacturer, distribution region 171 and product line 173.

PEC files which have been copied manually or saved with pCon.planner into this subfolder directly, can be synchronized 213 into the pCon.creator workspace database.

The import 212 and synchronization 213 commands operate on the resource folder for the current distribution region. If catalogs are developed in a master distribution region and exported to market specific derived distribution regions 172, additional PEC resource file sub folders can be added for each derived distribution region to save the PEC files which were updated for each specific market. The sub folder structure described above must be used for each derived distribution region. The name of the resource file for one typical must be the same in each distribution region. Additional information to work with market specific typicals can be found in topic Catalog entry Typical 188.

Geometry files

External geometry files in formats DWG, DXF and 3DS can be used as catalog resources.

Users of the catalog can insert these geometries in a planning using catalog entries of type "geometry 191".

OFML groups

OFML groups are files with the extension OGRP. They represent one or several grouped articles as a persistently stored element.

Important note: The concept of OFML groups is obsolete. OFML groups are only supported for backward compatibility reasons in pCon.creator OAS Module. This feature is subject to be removed in future versions of pCon.creator. In new OFML catalogs this old resource type should not be used anymore. It should be preferred to use catalog entries of type Typical 188 instead. Catalogs which are still using OFML groups should be migrated to use Typicals 188 in PEC file format.

OFML groups are, for instance, created with the function "Save element" of pCon.planner.

The articles in an OFML group always correspond to the data state that was active at the time when they were saved. I.e. if the concerned data were updated, those OFML groups in the catalog containing articles from these updated data, might need to be recreated.

OFML scenes

OFML scenes are files with the file extension FML. They represent a complete planned scene.

Important note: The concept of OFML scenes is obsolete. OFML scenes are only supported for backward compatibility reasons in pCon.creator OAS Module. This feature is subject to be removed in future versions of pCon.creator. In new OFML catalogs this old resource type should not be used anymore. It should be preferred to use catalog entries of type Typical 188 instead. Catalogs which are still using OFML scenes should be migrated to use Typicals 188 in PEC file format.

6.2.9.2 import files

A dialog to select files in the file system is displayed. The selected files will be copied to the resource directory of the current resource type and a record referencing to it will be created.

The records created during this function contain the synchronization status:

If a file with the name of the file to be imported already exists, you will be asked whether you want to overwrite this file.

6.2.9.3 sychronize files

The records of external file resources will be synchronized with the external resource directory.

Every record is labeled with a specific synchronization status after this operation.

Synchronization status

OK	The file exists in external resource path.
Deleted	The file does not exist in external resource path.
New	The record has been created by last operation. The file does exist in external resource path.

6.3 OCD article synchronization

The synchronization of the catalog entries of the type Article with the OCD data availabe in the workspace is a powerful feature of the OAS module. It allows you to efficiently verify and ensure the consistency of the catalogs created.

This function can be used at two points of time during the catalog creation.

- Capturing the catalog entries 215
- During the export 218

6.3.1 Synchronize catalog entries

All catalog entries are checked for whether there is a corresponding article stored in the OCD. The export filters stored in the current distribution region of the OCD format on product line and article level are taken into account.

This function is startet using the button above the article list  on catalog entries dialog185.

Synchronization status

During the synchronization, each record in the form catalog entries is marked with a status.

OK	The article is available in the OCD data.
Deleted	The article does not exist in the OCD data.
New	The article was reimported.

Article import

This area allows you to select the OCD product lines, from which the articles are taken over as catalog entries. It is automatically checked whether the articles are already available as catalog entries. Articles are not taken over multiple times. Catalog entries with different variants are automatically created for articles from different series but with the same article number.

Settings

Import short texts of new articles	When importing new articles as catalog entries, their short texts are also taken over from the OCD data as catalog text. Texts of already existing articles remain unchanged.
Synchronize catalog texts	This option takes over the short text of the OCD article as catalog text for all catalog entries. The catalog text of a catalog entry remains unchanged, if no OCD article is available for it.
Remove deleted articles	By default, articles are only marked as deleted, if they are not available in the OCD data. This option automatically removes the articles marked as deleted from the list of the catalog entries and the catalog structure. Attention: This operation is not reversible. If this option is activated, an additional warning is displayed.

A correct assignment of the text categories165 and languages166 requires them to be identical both in the OCD and in the OAS. The synchronization can be done in the text management164.

6.4 Export to XCF

The Explorer context menu features the menu item "Export to XCF". The export can be launched on the following levels of the OAS data format in the workspace:

- Module
- Manufacturer
- Distribution region
- Product line

To export one or several catalog product lines, an export path has to be indicated as target. In the target directory, the data are created according to the storage structure defined in the DSR.

UNC paths are not supported as destination path for the export.

Exporting an individual product line

When exporting an individual product line, mere partial exports 219 can be optionally performed. In this case, only the selected areas are exported.

Exporting multiple product lines

If the export is launched on the level of the distribution region, manufacturer, or module, you can select the product lines to be exported. Initially, the product lines below the level on which the export is performed are already selected.

Buttons

	Using this button settings to automatically run an OFML application after successful export can be configured. (see also)

6.4.1 Supplemental exports

DSR registration

This option generates from a default template a simple DSR product line registration for each product line exported. This registration allows you to register the data in an OFML runtime environment.

The generated registration file registers the exported product line as mere catalog product line.

OCD article synchronization

This option activates the synchronization of the articles with the OCD data during the export. The export filters of the OCD data are evaluated. Only the catalog entries of the type article are exported, which are also available in the OCD data after evaluation of the filters.

This option must be activated if catalogs from derived distribution regions are exported and if the OCD data are also defined in a derived distribution region, which export filters are applied to.

Update release date

This option automatically sets the release date of the exported product line to the current system date. This is also taken into account during the supplement export DSR registration.

No transfer of external resources

With this function no external resource files are copied to the export directory during the export. This can be useful when exporting extensive catalogs for test purposes. This function does not affect the export of the XCF table resource.csv.

6.4.2 Partial exports

Selecting individual parts to be exported of a catalog product line reduces the export time. This allows you to quickly verify individual corrections of a catalog product line without having to wait for the complete export to be finished.

The following partial exports can be selected.

XCF partial export

- Catalog entries
- Catalog structure
- Resources
- Texts

Partial exports can only be selected, if the export is launched on product line level.

Part VII

Graphic data development

EasternGraphics
visualize your business

7 Graphic data development

7.1 Overview

The ODB module provides functions for graphical data creation in pCon.creator. Direct and interactive access to AutoCAD makes the creation of parameterized graphical and functional objects for sales article visualization very efficient.

7.1.1 System requirements

The general system requirements of pCon.creator have to be fulfilled.

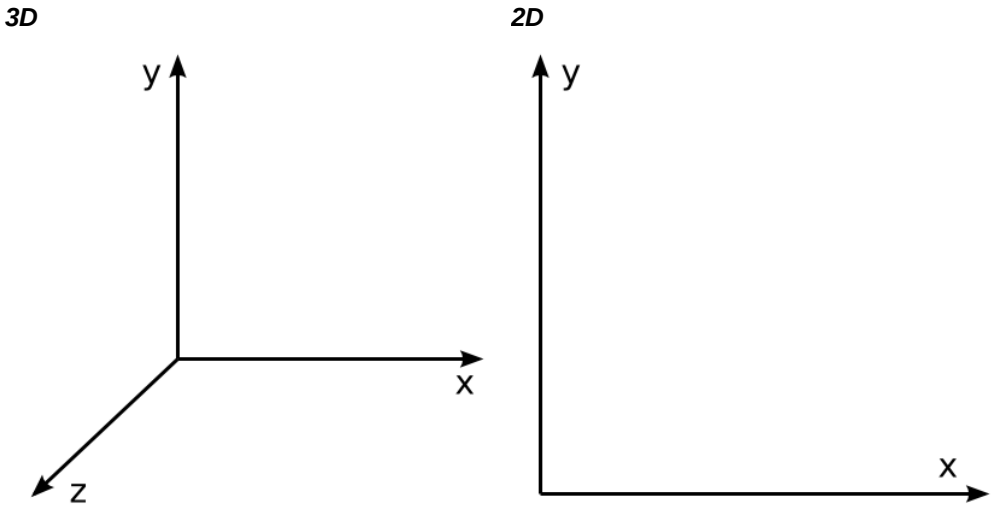
CAD-System

Efficient ODB data development is made possible by importing geometries and selection graphical components from an external CAD application. In settings dialog 273 pCon.planner Pro can be selected as CAD system to access external geometries from the component library DWG 276.

7.1.2 Units & coordinate systems

In ODB data development process the following coordinate systems and units are used.

OFML coordinate system



Please note: in 3D different coordinate systems[276] are used in OFML and DWG. Each geometric parameter is used in pCon.creator according to the OFML coordinate system.

Units

Lengths	The base unit in OFML is meter. Each value of position, length or scaling parameters have to be entered in this unit.
Angles	Angles are used in degree. Each rotation value has to be entered in this unit.

7.2 Main functions

7.2.1 Management of graphical data

The ODB module provides functions and dialogs to create and maintain parametrized graphics in data format OFML database ODB. These graphical objects can be connected with sales articles in OCD module's OAM user interface.

A specific parametrized graphic for sales article visualization is called an ODB object. This object can also be seen as a hierarchical parametrized bill of graphical materials. The components or graphical parts of an ODB object can be imported interactively from a CAD system 226.

Before creating the first ODB object 231 in a new workspace a new product line has to be entered. For this purpose following steps have to be done:

- Add a new manufacturer 43
- Create the product line 227

Afterwards the data entry dialogs for this product line can be accessed from its context menu in pCon.creator - Explorer.

Using the data in an OFML runtime application (e.g. pCon.configurator/planner) it has to be exported 274 first.

7.2.2 CAD Interface

The ODB supports direct and interactive access to geometries, which were drawn in AutoCAD.

Each product line contains has got a component library 276, which contains the geometric parts as blocks for 2D and 3D view. In product line management 227 the layer structure 277 which is used in this component library has to be configured,

The 2D part 272 and 3D part 270 management dialog contains a function to import the available components from AutoCAD. This process automatically imports the 2D and 3D layers, which can be mapped to OLAYERS and material categories in the 2D layer management 268 or 3D layer management dialog 266.

The object structure 231 can be build with interactive access to the modelspace of the component library 276. In each node of the structure a function can be started to select the component references and import them. The geometrical parameters like relative position, orientation and scaling are automatically imported from the selected block references. This function translates these parameters to the OFML coordinate system 276, too.

Furthermore it is possible to measure specific graphical parameters of the ODB object structure interactively in modelspace of the component library.

The component library is accessed using a CAD runtime application.pCon.planner PRO is supported as CAD runtime application.

On each workstation the used CAD System must be specified in Settings dialog 273.

The additional pCon.planner Plugin X3G-CadSystemAdapter is used to access the component library in pCon.planner PRO. This plugin is automatically installed with pCon.creator setup.

Formerly Autodesk AutoCAD was supported as CAD System until pCon.creator 2.20. AutoCAD is no longer supported in future pCon.creator version since the components to use this interface are no longer maintained to support future AutoCAD versions. Please note: Using pCon.creator with any 64bit version of Microsoft Access to import 2d or 3d geometries only pCon.planner can be used.

7.3 User interface

7.3.1 Dialog Product lines

In this dialog the ODB product lines of a manufacturer are maintained.

General fields

OFML-Code	A unique product line abbreviation to identify the product line. The value in this field must correspond to the rules of a valid OFML identifier and has to be in small letters. Furthermore, the abbreviation must not contain an underscore.
Label	This field contains a label of the product line.
Version-Major	Shows the major version number of the product line. It starts with 1. Incrementing the major number means significant modifications such as adding or deleting complete product lines.
Version-Minor	Shows the minor version number of the product line. It starts with 0. Incrementing the minor number is used e.g. in case of removing or adding articles, properties, property values etc. A regular price list update requires a minor increment as well.
Version-Build	Shows the build number of the product line. It starts with 0. Incrementing the build number means a minor modification in terms of error eliminations (graphic, price etc.).
Release date	Date of the release of the product line of this product line version.
Remarks	Remarks about the product line can be stored here.

Fields - ODB-Export

Format-Version	Defines the ODB format version which is used to export the ODB data. Default value is 1.0.
OFML Geometry format	Defines which file format of 3d OFML geometries is used in the package and generated importing the 3d components 270 from component library. OBJ is defined as default. For reasons of backward compatibility the old geometry format GEO is still available. It is recommended to use pCon.planner as CAD System to import geometries if OBJ is used.
Proxy2 DWG	This option activates the export of Proxy2 DWGs.
Egms	For each product line this option can be used to deactivate the generation and export of geometries in Egms format for 2d parts. Egms export is enabled as default for all new product lines. If the Egms export is disabled Proxy2 DWG files must be used.
Vertex normals	This option allows the generation and export of additional vertex normals. This setting is only relevant and available using the old 3d OFML geometry format GEO. Vertex normals are exported into additional VNM files. Using OBJ geometry file format the vertex normals are always saved inside the OBJ geometry files. Vertex normal are used to improve the visualization in applications which uses OFML geometries.

Please note, using ODB export format 1.1 to support new layer based features like acoustic information in OFML data requires up to date runtime application. Such data cannot be used by dealers which don't have updated their OFML runtime applications (like pCon.planner) after April 2013.

With Proxy-DWGs the articles are visualized using the natively in AutoCAD drawn geometries, if the articles are created in CAD based OFML applications like pCon.xcad or pCon.planner 6.x . Proxy-DWGs allow a higher presentation quality of the articles especially if further the articles would be modified in these applications.

Fields - Layer structure

CAD library	In this field the layer structure of the CAD component library is configured.
OFML-Export	This field specifies the layer structure which is used in the exported OFML data. The layer structures `OLAYERS 1.1` or FOS (in old product lines) can be selected.
2D block prefix	This field contains a prefix of block names in component library, which identifies them as 2D parts. This field is just used if the neutral structure is active.
3D block prefix	This field contains a prefix of block names in component library, which identifies them as 3D parts. This field is just used if the neutral structure is active.
2D layer prefix	This field contains a prefix of layer names in component library, which identifies them as 2D layers. This field is just used if the neutral structure is active.
3D layer prefix	This field contains a prefix of layer names in component library, which identifies them as 3D layers. This field is just used if the neutral structure is active.

These fields are just used to create the connection to the drawings in CAD library 276. Modifications of these values are not applied automatically to the block or layer names in the component library. After modification of any of these fields the parts have to be imported newly.

Layer naming convention FOS is outdated and is not supported anymore since pCon.creator 2.15. This obsolete layer naming convention may still be active for reasons of backwards compatibility. After deactivating the FOS layer structure it cannot be reset in pCon.creator user interface.

Fields - CAD Geometry

Precision	Defines the decimal digits used to round transformation parameter values (position, rotation, scaling) when blocks are selected in cad system to be added into an ODB Objects 231 2d or 3d structure. This option is limited to a maximum number of 8 decimal digits.
Quality of tessellation	This option defines the level of quality which is applied to tessellate solids when OFML Geometries are imported (GEO, OBJ).
Preview image	Importing parts 270 from component library 276 preview images are generated to show them in pCon.creator UI. This option can be used to disable the generation and display of these preview images in several product lines.

The tessellation quality setting should be a good compromise on visualization quality and data amount. Higher levels usually produce more geometric data. Unnecessary large geometric files, may also have negative impact on the runtime performance of the OFML data.

After modification of the geometric export settings the parts have to be imported newly from the component library.

The formerly so called "Compatibility mode" for generation of OFML geometries is not supported since pCon.creator 2.15. The "Compatibility mode" option may only be visible at old product lines for compatibility reasons. After deactivation the compatibility mode option cannot be reactivated in pCon.creator user interface.

7.3.2 Dialog Objects

ODB objects are the parametrized graphics, which are used to visualize sales articles.

Each ODB object is distinguished into a separate structure for a 2D and 3D view. A 2D structure has to be created for each root object.

The structures can be seen as parametrized hierarchical bills of geometric materials. The structures are build of nodes. Elements define the several graphical components the object consists of. Each node and element can be parametrized in relation to the article's configuration.

Fields

Root	This field either defines the object is a root object, which is exported to OFML, or just a macro definition.
Name	A unique identifier of the Object.
Description	In detail view of the current ODB object a description can be entered into this field to describe the purpose of this object and its parameters.

Commands

The context menu of each node or element in the object's structure contains the following functions. Please note according to the node's type the available functions may be restricted.

Select elements in CAD	Block references from the CAD component library 276 can be imported interactively as sub elements. (Key `CTRL+INSERT`)
New element	A new element is created. (Key: `INSERT`)
New node	A new node is created. (Key `SHIFT+INSERT`)
Delete	The selected node and its sub elements is deleted. (Key `DELETE`)

The interactive selection of block references of the modelspace in the component library is performed with the following steps, which can be seen in the AutoCAD command line, too:

- Select the base point (or insertion point)
- Select the components
- Confirm the selection

pCon.creator creates a temporary OFML user coordinate system. After confirmation the selection of components this user coordinate system can remain optionally active. The default behavior is to reactivate the previously used user coordinate system. Hence this prompt can be skipped with key `ENTER`.

Buttons

	Using this button the view of the object's 3D structure is activated. The 3D structure is already visible as default, if the dialog is opened.
	Using this button the objects structure can be switched to the 2D structure view.
	This button switches the detail view on the right side of the selected node. As default the node's graphical parameters are visible. Alternatively the node's sub nodes and elements can be shown in a tabular view using this button.
	With this button the current view to the modelspace of the component library 276 is saved as preview image of the selected object. Please note: The preview of the current selected ODB Object can be removed, using the context menu of the preview image.
	With these buttons several geometric parameters like position, scaling or rotation can be measured in the modelspace of component library 276 to import these values.

7.3.2.1 3D-Nodes

Node entries are the structural elements of an ODB object. They can contain sub nodes and sub elements.

Fields

Name	A name of the node. The value has to be unique on each level.
Visible	This field can contain an expression 256 which defines the condition under which the node is visible. This can be an ODB expression itself or user defined ODB function 255. Using user defined ODB functions should be preferred. The node is always visible if this field is empty.
Position	The relative position of the node in relation to its parent's node local coordinate system. The coordinates on x-, y- and z- axis of the OFML 3D coordinate system have to be entered.
Rotation	The orientation of the node in relation to its parent node. The elementary rotations around the x- , y- and z-axis of the local coordinate system of the node can be entered, whereas the rotations are applied in this order. It is recommended to use a separate node for each elementary rotation to keep control of the order and prevent accidental side effects.
Node-type	Each node may be of a special type for which additional parameters can be defined.

The position and rotation values can be constant values, expressions or a call to a user defined function.

The position and rotation values have to be entered in OFML units 224.

Node types

Nodes are distinguished into the following types:

- Node
- Class 235
- Reference 237
- Macro 236
- CSG - Constructive solid geometry 238

- CSG Union 240
- CSG Difference 240
- CSG Intersection 240

Class

A node can be represented by an OFML class at runtime. e.g. this can be functional classes (GO-Types).

Fields

Manufacturer	The OFML code of the manufacturer of the package, where the class is defined. As default the manufacturer *ofml (EasternGraphics GmbH)* is available. It contains GO- and OFML types and OFML extensions. Additional classes can be maintained in CLS module.
Product line	The OFML code of the package where the class is defined. As default the manufacture *ofml* is available who contains the packages: go: GO-Types oi: OFML-Types xoi: OFML-Extensions
Name	The name of the class.
Parameter	The parameters of the class. The parameters which have to be provided are dependent on the used class. The documentation which parameters have to be supplied explained in the class's description.
Material parameter	Depending on the used class additional material parameters have to be supplied, which have to be entered in this field. The required parameters are explained in the class's description. In general material parameters are just required for classes which create a programmed geometry. Functional classes like GO-Types normally don't have material parameters.

Buttons

Picture	This button shows the preview image of the used class.
Description	This button shows the class's description about its purpose and parameters.

Macro

Macros are references to other ODB objects at design time. While export the 3D structure of the referenced ODB object is copied to the macro's position. Macros cannot contain sub elements.

Fields

Manufacturer	The OFML code of the manufacturer of the package, where the referenced ODB object is defined.
Product line	The OFML code of the package, where the referenced ODB object is defined.
Name	The name of the reference ODB object.

If the referenced ODB object is defined in another product line, the used ODB functions 255 have to be defined in the current product line, too.

Buttons

Picture	This button shows the preview image of the referenced ODB object.
Description	This button shows the description of the referenced ODB object.

Reference

ODB references refer to other ODB objects. Contradicting to macros ODB references are evaluated at runtime.

Fields

Manufacturer	The OFML code of the manufacturer of the package, where the referenced ODB object is defined.
Product line	The OFML code of the package, where the referenced ODB object is defined.
Name	The name of the referenced ODB object.
Parameter	In this field the arguments P_0, P_1 to P_{n-1} of the referenced ODB object can be entered. Multiple parameters are separated by a comma. Each argument can be a constant value or an ODB Expression 256 using ODB Functions 255. If arguments and how many arguments have to be specified depends on the referenced ODB Object.

Buttons

Picture	This button shows the preview image of the referenced ODB object.
Description	This button shows the description of the referenced ODB object.

Usage of Parameters

An ODB object which is used as ODB Reference cannot access any sales property or OAM parameter directly using Variable References 259. The referenced ODB Object can only access its own parameters P_0, P_1 bis P_{n-1}.

These arguments can be used in expressions like regular Variable References 259. e.g. `$P1` provides the second argument; `$P0` the first. and `$P2` the third parameter.

It is not recommended to use ODB-References.

ODB references are a special feature of the ODB specification. Special considerations and pitfalls have to be taken in to account modeling the ODB data using them. In pCon.creator based ODB data development projects it has become practical to prefer Macros 236.

ODB References can only be used in 3d.

An ODB Object used as ODB Reference can only access its own parameters. Properties of the current sales article configuration cannot be accessed directly in this object. They must always passed to the object by the P_0, P_1 bis P_{n-1} parameters. The referenced objects must be modeled using this extra style to be used in ODB references.

Constructive solid geometry (CSG)

Using the following operations of so called constructive solid geometry (CSG) it is possible to build complex geometries from boolean combinations different individual partial geometries:

- Union 240 of multiple different partial geometries
- Difference 240, subtracting parts from a geometry
- Intersection 240 of shapes of multiple geometries
- Stretch 240 at a cutting planing
- Macro 240

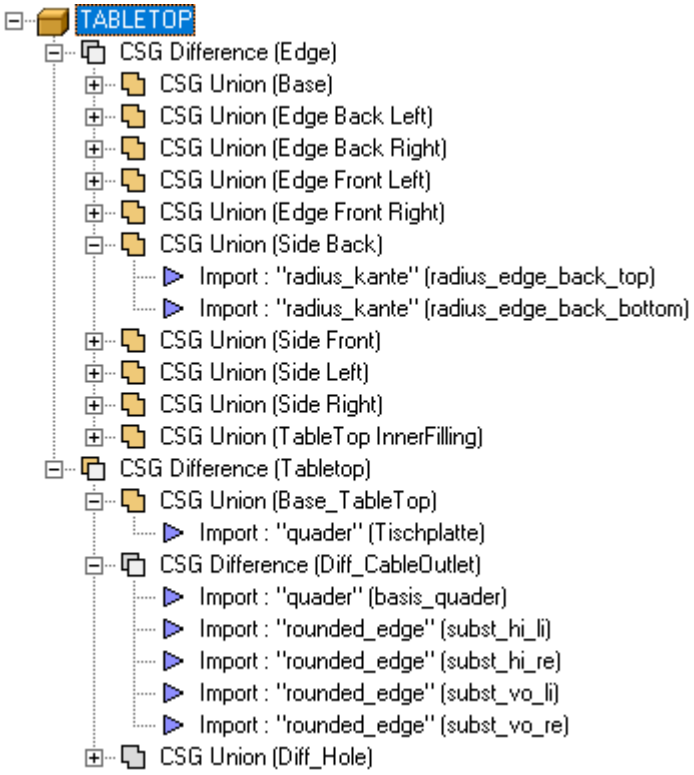
These operations are defined as nodes in 3d object structure. Sub nodes of these CSG nodes are operands / partial geometries, which are processed by these operations. These partial geometries can be defined as simple partial 3d Element 241 types e.g. components selected in CAD component library 276.

Furthermore CSG operations can be nested. For instance a node of type CSG Difference can have operands which are Unions of geometries. Using this approach it is possible to build complex parametrized scalable components within the 3d structure of an ODB object, which may contain for instance:

- non-scalable parts like rounded edges
- parametrized cut outs or holes
- ...

Each partial geometry can be parametrized according to the features the individually used Element Type 241 supports. For instance its visibility, relative position, rotation or scaling factors may depend on the current article's configuration.

 Example



Using CSG operations following issues have to considered:

- CSG nodes can be used as sub nodes of other regular nodes. The top most CSG node of a CSG sub structure represents a single 3d component have a one and only layer and material assignment.
- This means the top most CSG node of a CSG sub structure contains the material and layer assignment. Individual layer and material assignments within the sub structure below this top most CSG node don't have any effect and are ignored.
- The result of some CSG operations depend on the exact sequence of its operands (sub nodes and sub elements). The name of sub elements or nodes defines the ascending alphanumeric sorted sequence of the nodes on each sub level.
- On each sub level within a CSG sub structure either CSG sub nodes or sub elements have to be used. It may be necessary to insert an additional CSG Union sub node containing only one geometry element.
- Within a CSG sub structure it is only possible to use CSG sub nodes or elements. Other regular node types like macros 236, class 235 or odb references 237 are not allowed and cannot be used.

CSG Union

Using the node type CSG Union multiple individual geometries (sub elements) are merged into one component.

CSG Difference

Using the node type CSG Difference it is possible to cut out or subtract parts from geometries.

The sequence of the operands (sub nodes and elements) is essential.

The first operand represents the geometry from which other shapes are subtracted. This means all other operands 2..n represent shapes which are cut out from the first operand.

CSG Intersection

The intersection produces a shape which is the common portion of the overlapping geometries of all operands (sub elements) of this operation.

CSG Stretch

The CSG operation "Stretch" defines a cutting plane by its normal vector and distance from the origin of the local coordinate system (offset). All geometries below this node are cut at this plane and stretched along the normal vector of the cutting plane. The stretched distance can be defined as scale factor using a constant factor or ODB expression / function 255.

CSG Macro

The CSG Macro references another ODB object (Template). The referenced object is exported as copy at the location of the macro node.

The CSG Macro operates like regular macros 236 below other CSG nodes, but with the limitations of nested CSG operations 238. Since below a CSG node only other CSG nodes and geometries are allowed. The referenced ODB object can only consist of CSG nodes and geometries, too.

7.3.2.2 3D-Elements

3D elements are the several geometric parts of which the ODB object consists in 3D view.

Field

Name	A name of the element. The value has to be unique on each level.
Visible	This field can contain an expression 256 which defines the condition under which the node is visible. This can be an ODB expression itself or user defined ODB function 255. Using user defined ODB functions should be preferred. The node is always visible if this field is empty.
Layer	The layer 266 which is assigned to this element. Layers are used material categorization and assignment.
Mat.-Param.	Additional parameters to the material e.g. special mapping function can be assigned in this field. The value may be a constant character string according to OMATS or a reference to an ODB function 255.
Position	The relative position of the node in relation to its parent's node local coordinate system. The coordinates on x-, y- and z- axis of the OFML 3D coordinate system have to be entered.
Rotation	The orientation of the node in relation to its parent node. The elementary rotations around the x- , y- and z-axis of the local coordinate system of the node can be entered, whereas the rotations are applied in this order. It is recommended to use a separate node for each elementary rotation to keep control of the order and prevent accidental side effects.
Element-Type	Each element may be of a special type for which additional parameters can be defined.

The material parameters, position and rotation values can be constant values, expressions or a call to a user defined function.

The position and rotation values have to be entered in OFML units 224.

Element types

Following special element types can be used:

- Import 242
- Cube 242
- Frame 243

- Cylinder 243
- Sphere 243

Import

This element references an external geometry (Block in component library 276).

Fields

Geometry	The name of the external block/geometry 270.
Scaling X, Y, Z	An element can be scaled along the x-, y- and z-axis. The value can be a constant value. Assigning an ODB functions 255 is possible, too.

The name of the geometry is entered as a ODB expression. Normally this is a constant character string enclosed in double quotation marks. The available 3D parts 270 can be selected directly.

Scaling with factor 0 is not allowed. Errors will occur if a scaling factor is 0.

Cube

This element creates a rectangular cube which is extended along the positive axis of the local coordinate system.

Fields

Width, Height, Depth	Constant values, expressions or references to user defined ODB functions 255 can be entered.

Sphere

This element creates a sphere around the origin of the local coordinate system.

Fields

Radius	A constant value, expression or a reference to a user defined ODB function 255 can be entered.

It is recommend to avoid using this element type in new project. This element type was defined and implemented in the early days of OFML. The rendering quality of this element is not as good as it is usually expected in all OFML runtime applications today. A better high quality can be achieved drawing this shape in a block in the CAD component library 276 and use this external geometry with an import 242 element.

Cylinder

This element creates a cylinder which is symmetric rotated around the y-axis. Its base point is the origin of the local coordinate system and it is extended along the y-axis.

Fields

Height, Radius	Constant values, expressions or references to user defined ODB functions 255 can be entered.

It is recommend to avoid using this element type in new project. This element type was defined and implemented in the early days of OFML. The rendering quality of this element is not as good as it is usually expected in all OFML runtime applications today. A better high quality can be achieved drawing this shape in a block in the CAD component library 276 and use this external geometry with an import 242 element.

Frame

This element creates a frame which is extended along the positive axes of the local coordinate system.

Felder

Width, Height, Depth, Thickness	Constant values, expressions or references to user defined ODB functions 255 can be entered.

7.3.2.3 2D-Nodes

Node entries are the structural elements of an ODB object. They can contain sub nodes and sub elements.

Fields

Name	A name of the node. The value has to be unique on each level.
Visible	This field can contain an expression 256 which defines the condition under which the node is visible. This can be an ODB expression itself or user defined ODB function 255. Using user defined ODB functions should be preferred. The node is always visible if this field is empty.
Position	The relative position of the node in relation to its parent's node local coordinate system. The coordinates on x-, y-axis of the OFML 2D coordinate system have to be entered.
Rotation	The orientation of the node in relation to its parent node.
Scaling	A 2D node including its sub elements can be scaled along the x- and y-axis.
Node type	Each node may be of a special type for which additional parameters can be defined.

The position, rotation and scaling values can be constant values, expressions or a call to a user defined function.

The position, rotation and scaling values have to be entered in OFML units 224.

Scaling with factor 0 is not allowed. Errors will occur if a scaling factor is 0.

Node types

- Node
- Macro 245
- Stretch 245

Macro

Macros are references to other ODB objects at design time. While export the 2D structure of the referenced ODB object is copied to the macro's position. Macros cannot contain sub elements.

Fields

Manufacturer	The OFML code of the manufacturer of the package, where the referenced ODB object is defined.
Product line	The OFML code of the package, where the referenced ODB object is defined.
Name	The name of the reference ODB object.

If the referenced ODB object is defined in another product line, the used ODB functions 255 have to be defined in the current product line, too.

Buttons

Picture	This button shows the preview image of the referenced ODB object.
Description	This button shows the description of the referenced ODB object.

Stretch

The sub geometries of this node are stretched.

The parameters normal vector and offset (distance from the origin of the local coordinate system) define an intersecting line at which the sub geometries are stretched. The normal vector also shows the direction of the stretch operation. The scale factor is the stretched distance. It can be a constant factor or ODB expression / function 255.

7.3.2.4 2D-Elements

2D elements are the several geometric parts of which the ODB object consists in 2D view.

Fields

Name	A name of the node. The value has to be unique on each level. A name of the node. The value has to be unique on each level.
Visible	This field can contain an expression 256 which defines the condition under which the node is visible. This can be an ODB expression itself or user defined ODB function 255. Using user defined ODB functions should be preferred. The node is always visible if this field is empty.
Position	The relative position of the node in relation to its parent's node local coordinate system. The coordinates on x-, y-axis of the OFML 2D coordinate system have to be entered.
Rotation	The orientation of the element in relation to its parent node.
Scaling	The scaling of the element along x- and y-axis.
Element type	Each element may be of a special type for which additional parameters can be defined.

The position, rotation and scaling values can be constant values, expressions or a call to a user defined function.

The position, rotation and scaling values have to be entered in OFML units 224.

Scaling with factor 0 is not allowed. Errors will occur if a scaling factor is 0.

Element types

- Import 247
- Point 247
- Text 248
- Square 250
- Lines 251
- Circle 251
- Ellipse 252
- Arc 252

Import

This element references an external geometry (Block in component library 276).

Fields

Geometry	The name of the external block/geometry 272.

The name of the geometry is entered as a ODB expression. Normally this is a constant character string enclosed in double quotation marks. The available 2D parts 272 can be selected directly.

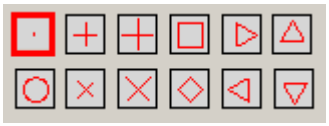
Point

This element inserts a simple point.

Fields

Color	This element can have a specific color 254. The color components red, green and blue have to be entered.
Layer	The layer 268 which is assigned to this element.
Point size	In combination with the following point styles the point can be scaled using the value in this field.

Point styles



With these buttons the point style can be defined. Multiple styles can be combined. The red frame highlights the selected styles.

Please node that only one global point style is supported DWG based OFML planning applications. The style of an individual OFML point won't have any effect in these systems.

Text

Two different element types to create a text in 2D view can be used.

- Text
- Formatted text

Using a formatted text element, additional values e.g. return values from functions can be displayed in a formatted style according to their data type.

Fields

Alignment	This field defines the horizontal alignment of the text at the insertion point: -1 - left aligned 0 - centered 1 - right aligned The base line of the text intersects the insertion point. This field is mandatory. It must not be left empty.
Content	An expression, which defines the displayed text. This can be a constant character string enclosed in double quotation marks or a reference to an ODB function which returns the character string. Please note the special content parts of formatted text elements below.
Color	This element can have a specific color 254. The color components red, green and blue have to be entered.
Layer	The layer 268 which is assigned to this element.
Font height	The height of the text. The value has to be entered in OFML units 224. A value of 1.0 creates a text with a height of 1 meter.
Font scale	This value scales the text horizontally. Values below 1.0 compress the text, while values above 1.0 stretch it.

The height of the text should be set using the font height parameter. Scaling a text using the general scaling parameters is not recommended. The default height is 0.1 .

Formatted Text

The content of a formatted text element consists of multiple parts which are separated with a comma.

```
<Fromatstring>, <Parameter 1> , .... <Parameter n>, <Parametercount>
```

The format string is a constant character string which will be displayed. This format string may contain placeholder which are replaced by the parameters 1 to n from left to right. The last part of the content defines the count of parameters.

Instead of specifying all parts in the content field an ODB function with multiple return values 262 can be used.

Following placeholders can be used in the format string:

`%d` integer

`%f` real number

`%s` character string


Real numbers (%f) are displayed in the shortest possible style. This means an application may display the value in scientific notation using mantissa and exponent in case this style is shorter than printing all total and decimal digits.


Example- Formatted text

An ODB function which specifies the content of formatted texts has to return the following values:

- the format character string
- the values which will be formatted
- the number of values which will be formatted

A function `sDimensions` creates a dimension character string e.g. 1600x800x720. This function is defined as follows:

```
"%dx%dx%d", fWidth * 1000, fDepth * 1000, fHeight * 1000, 3
```

The first return value is the format character string. The formatting placeholders %d define that the values are formatted as integer values. The value are read from left to right. The values for width, depth and height are returned by separate functions. The last value defines that 3 parameters are formatted.

Square

This elements creates squares and rectangles which are extended along the axes of the local coordinate system.

Fields

Color	This element can have a specific color254. The color components red, green and blue have to be entered.
Layer	The layer268 which is assigned to this element.

The size of the rectangle is defined by the scaling parameters246.

This element can have a specific linestyle253.

Lines

Horizontal line

Horizontal lines are extended along the x-axis. The length is specified by the scaling factor along the x-axis 246.

Vertical line

Vertical lines are extended along the y-axis. The length is specified by the scaling factor along the y-axis 246.

Diagonal line

Diagonal lines are extended from the lower left to the upper right corner of a 1 to 1 meter square, which is located on the axes of the local coordinate system. The scaling parameters 246 specify the length and gradient.

Fields

Color	This element can have a specific color 254. The color components red, green and blue have to be entered.
Layer	The layer 268 which is assigned to this element.

This element can have a specific linestyle 253.

Circle

This element creates a circle with radius 1m around the origin of the local coordinate system. The size can be scaled 246.

Fields

Color	This element can have a specific color 254. The color components red, green and blue have to be entered.
Layer	The layer 268 which is assigned to this element.

This element can have a specific linestyle 253.

Ellipse

This element creates a ellipse around the origin of the local coordinate system.

Fields

Color	This element can have a specific color[254]. The color components red, green and blue have to be entered.
Layer	The layer[268] which is assigned to this element.
x-Radius	The radius of the ellipse along the x-axis.
y-Radius	The radius of the ellipse along the y-axis.

 This element can have a specific linestyle[253].

Arc

This element creates an arc of a circle with radius 1m around the origin of the local coordinate system. The size can be refined using the scaling parameters[246].

Fields

Color	This element can have a specific color[254]. The color components red, green and blue have to be entered.
Layer	The layer[268] which is assigned to this element.
Start angle	The angle from which the arc is drawn.
End angle	The angle until which the arc is drawn.

 Start and End angle are entered in relation to the x-axis counter clock wise.

 This element can have a specific linestyle[253].

Linestyles

A linestyle can be applied to the following 2D elements:

- Square 250
- Lines 251
- Circle 251
- Ellipse 252
- Arc 252

Parameters

A linestyle is defined by the following parameters.

Line width	This value defines the strength of the line in pixel. The default value is 1px .
Line type	In this field a specific line type can be selected (see below).
L-type scale	The value is used to scale the line type. It is given in pixel, but its definite meaning depends on the line type.

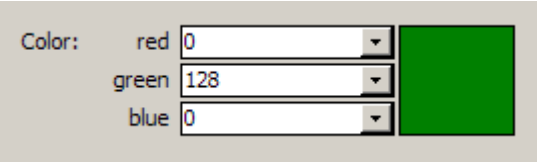
Line types

_____	***continuous***	
_ _ _ _ _ _	***dashed***	The scaling factor defines the length of the visible and hidden segments.
. . . . .	***dotted***	The scaling factor defines the distance between the middle epicenters of two neighbor dots.
_.-_.-_.-_.-_.	***dash-1dot***	The scaling factor defines the distance of the visible segment and the half length of the invisible segments' distance.
_..-_..-_..-_..	***dash-2dots***	The scaling factor defines the distance of the visible segment and one third of the invisible segments' distance.
_...-_...-_...	***dash-3dots***	The scaling factor defines the distance of the visible segment and one fourth of the invisible segments' distance.

Furthermore the predefined default line type can be selected. Normally this is a continuous line.

Color assignments

Several 2D elements can have a specific color.



Constant values

The color components red, green and blue have to be set. Each component is specified with a value in the range from 0 (none) until 255 (max.).

A single click on the color preview control opens a dialog which can be used to define the color exactly. After confirmation the color components are set automatically.

Using functions

Alternatively user defined ODB function can be used to set the color dynamically. This functions have to return a real value between 0.0 (none) and 1.0 (max.) .

It is possible to use one function with multiple return values which specify the color components in the exact order red, green, blue. If such a function is used, it has to be entered for the field of the red color component. In this case any other field has to be left empty.

Explicit color values should be used carefully. It is preferred to use layers. Users can modify a layers color, but they cannot change constant color values.

7.3.3 Dialog Functions

User defined ODB functions, which can be used for dynamic parameterization of the ODB object's structure, are managed in this dialog. These user defined functions can be used like built-in functions and constants in any expression.

Fields

Name	The unique name of the function is entered in this field. The name must correspond to the rules of a valid OFML identifier.
ODB expression	The body which of the function is entered in this field. The expression is entered in infix notation.
Comment	A comment which describes the purpose of the function can be entered in this field.

 ODB expressions and functions are case-sensitive.

7.3.3.1 ODB expressions

ODB functions always have a return value. The types of expressions are distinguished by their return types.

Simple expressions can be just a constant value, but using

- Variables 259
- Operators 261
- Built-in constants 263
- Built-in functions 264
- User defined ODB functions 255

more complex expressions can be defined. Using round brackets the evaluation order of terms in complex arithmetic or boolean expressions can be modified.

Numeric expressions

The return value of numeric expressions is an integer or a real number normally in meter. (f_)

Alpha numeric expressions

The return value of alpha numeric expressions is a character string. Constant character strings within expressions have to be enclosed in double quotation marks. (s_)

Boolean expressions

Boolean expressions may be comparisons and logical expressions. The return value is the logical state true or false. A boolean expression is represented numerically with 1 (true) and 0 (false). For this reason it is possible to use boolean expressions as numeric in complex expressions. (b_)

Variables 259

Return a value from an external source e.g. commercial property. The return value is not specified by the prefix of the function, because of this they should only be used inside other functions to make changes without bigger impact possible. (p_)

Example: Comparison with a numeric constant

The following expression compares that with is lower equal 2 meters. It uses the ODB function `f_width` which returns the current article's width in meter. `f_width` access a commerical property using a variable reference 259.

```
f_width <= 2.0
```

Example: Comparison with a constant character string

If a function returns a character string a comparison with this value could look like:

```
p_socket == "FT"
```

This example uses the ODB function `p_socket`, which accesses a commercial property. The expression is true if the current vaule `FT` is selected.

Example: Complex boolean expressions using round brackets

In following expression two terms are combined with logical AND. This means it is true, if the numeric return value of ODB function `f_width` is greater equal `1.2` AND the character string return value of function `f_socket` is not `FR`.

```
( f_width >= 1.2 ) && ( f_socket <> "FR" )
```

Using round brackets the evaluation of terms in an expression can be modified. The following expression

```
f_heightleft == 1.0 || f_heightright == 1.0 && f_depth == 0.8
```

is true in case functions `f_heightright` is `1.0` and `f_depth` is `0.8` or `fHeightLeft` is `1.0`. But it is also true if `f_heightright` is not equal `1.0` as long as `f_heightleft` is equal `1.0`.

Logical AND is always evaluated before logical OR. The following expression emphasizes this behavior:

```
f_heightleft == 1.0 || ( f_heightright == 1.0 && f_depth ==
0.8 )
```

The evaluation order can be modified using the round brackets. The following expression ensures that the return value will be true, if f_depth is equal `0.8` and at least one of the functions `f_heightleft` OR `f_heightright` is equal `1.0`.

```
( f_heightleft == 1.0 || f_heightright == 1.0 ) && f_depth ==
0.8
```

It is allowed to use round brackets around terms even if they are not required to modify the evaluation order. This may be useful to improve readability of complex expressions. White space characters between operators, operands and round brackets help improving the readability:

```
( ( f_heightleft == 1.0 ) || ( f_heightright == 1.0 ) ) &&
( f_depth == 0.8 )
```

7.3.3.2 Variables

The configuration of sales articles parameterizes ODB objects. Using variable references in ODB expressions the actual values of the sales properties and in OAM defined ODB parameters can be determined.

Variable references start with the character $ followed by the name of the according OCD property or ODB parameter in OAM.

```
$<PropertyName>
```

An error occurs if the property or parameter does not exist.

This case can be considered using a default value when accessing the property. In the following way:

```
${<PropertyName>:-<DefaultValue>}
```

The default value can also be an expression. With this approach alternative properties can be accessed as fallback, if the first does not exist.

The type of the variable reference is dependent on the property definition in OCD.

Example - Accessing a sales property (WIDTH)

The following expression may be used in an user defined ODB function `fWidth` to retrieve the value of the sales property `WIDTH`.

```
$WIDTH
```

an error occurs if this property does not exist. For this reason the following expression always returns the default value `1000` in cases the property `WIDTH` does not exist:

```
${WIDTH:-1000}
```

Assuming the property `WIDTH` is the width of the article in millimeter, the following expression transforms this return value in the OFML unit meter:

```
${WIDTH:-1000} / 1000
```

Example - Accessing a temporary missing property

Assuming the visibility of a property `EDGE`, which defines the color of a table top's edge is, controlled by an OCD precondition. In cases the property is hidden the edge has got the same color like the table top, which is configured with the property `PLATE`.

The following expression of an user defined ODB function `mat_EDGE` first determines the value of the property `EDGE`. If this fails because the property is hidden, the value of the property `PLATE` is determined instead:

```
${EDGE:-${PLATE:-"ERRORMATERIAL"}}
```

Furthermore this example uses the final constant default value `"ERRORMATERIAL"` in cases the property `PLATE` cannot be determined, too.

7.3.3.3 Operators

Following operators can be used in expressions

Relational operators

==	Is equal to
!=	Is not equal to
>=	Is greater than or equal to
<=	Is less than or equal to
>	Is greater than
<	Is less than

Logical operators

&&	Logical and
||	Logical or
!	negation of the following expression

Arithmetische Operatoren

+	Addition
-	Subtraction
*****	Multiplication
/	Division
%	Modulus operator. This operator determines the integer modulus of a division of its operands.

Character string operators

+	Concatenation of two character strings  `"::xyx::basics:xy": + ${FRONTMAT:-"ERRORMATERIAL"}`

7.3.3.4 Parameters

A user defined function can optionally expect parameters, which have to be supplied when referencing them.

If a function expects some parameters this the number of parameters have to be defined in the first term of the function definition:

```
argc(<ParameterCount>)
```

The specific value of each parameter can be retrieved within the function definition by the numbered variable references `$0`, `$1`, ... where the variable `$0` contains the value of the first parameter.

Example: Verification of a value in a specific range

A user defined function `bWidthIn` can have following definition:

```
argc(2), ($0 <= $WIDTH) && ($WIDTH <= $1)
```

This ODB function expects two parameters. It determines if the value of the variable 259 `$WIDTH` is in the cloesd range which is defined by the two parameters of the function. The function can be used e.g in the following form:

```
bWidthIn(1, 1.2)
```

7.3.3.5 Return values

User defined functions always have at least one return value.

A function can also return multiple values. These return values are separated by commas in the function's definition.

Functions with multiple return values are useful in cases they are used to retrieve multiple parameters for elements.

e.g. the ODB element "formatted text 248" expects multiple parameters which define its content. These parameters be retrieved with a function, which returns multiple values.

7.3.3.6 Builtin constants

The following constants are built in and can be used in expressions.

M_PI	
M_PI_2	
M_PI_4	
M_2PI	
M_1_PI	
M_2_PI	
M_2_SQRTPI	
M_E	e
M_LN10	ln 10 = $\log_e 10$
M_LN2	ln 2 = $\log_e 2$
M_LOG10E	lg e = $\log_{10} e$
M_LOG2E	1/ln 2 = $\log_2 e$
M_SQRT1_2	
M_SQRT2	

7.3.3.7 Builtin functions

The following predefined functions are supported by OFML runtime applications an can be used in ODB expressions. A description of the supported functions can be found in the ODB specification.

Mathematical functions

acos(x)	calculates the arc cosine of x
asin(x)	calculates the arc sine of x
atan(x)	calculates the arc tangent of x
atan2(x, y)	calculates the arc tangent of x/y
ceil(x)	rounds the numeric value x up
cos(x)	calculates the cosine of x
cosh(x)	calculates the hyperbolic cosine of x
exp(x)	calculates euler's number e to the power of x
fabs(x)	calculates the absolute amount of the numeric value x
floor(x)	round the numeric value x down
fmod(x, y)	calculates the floating point modulus of the division x / y
log(x)	calculates the natural logarithm
modf(x)	This function divides the value x into an integer part y and the fractional part z, whereas each value has got the same sign. The return values are y and z
neg(x)	negates x
pow(x, y)	calculates x to the power of y
sin(x)	calculates the sine of x
sqrt(x)	calculates the square root of x
tan(x)	calculates the tangent of x
tanh(x)	calculates then hyperbolic tangent of x

Character string functions

substr(x, y, z)	Retrieves a partial character string *Parameter:* x String The source character string y Int The start position of the partial character string z Int The length of the partial character string *Return value:* String
size(x)	Retrieves the length of a character string *Parameter:* x String The source character string *Return value:* Int
string(x)	Converts a numeric value into a character string representation *Parameter:* x Int / Float *Return value:* String

7.3.4 Dialog 3D-Layer

In this dialog the 3d layers which are used in a product line are maintained. Material assignments on material category level can be managed here.

 In pCon.creator the names of layers and blocks are used neutrally according to any layer structure. Therefore they do not contain any special prefix in the database. Specific layer structures are just used at the interfaces from CAD component library[276] to pCon.creator and the export to OFML.

Fields

Sync	This field displays the layer's status after the last synchronization.
Layer	This field contains the unique name of the 3d layer.
Mode	This field configures the mode of the layer according to OLAYERS 1.1
Tag	This field configures a special tag of the layer according to OLAYERS 1.1.
Material category	In this field a OFML material category is assigned to the layer. Material categories are used for central material mapping.
Default Material	In this field a default material can be defined for the material category. The value can be a constant character string or a reference to a user defined function[255].
Parameter	Additional material parameters can be entered in this field. This can be a constant character string according to OMATS or a reference to user defined function[255]. This field can be used to add special mapping parameters to control texture orientations.
Comment	A description of the purpose of the layer can be entered in this field..
OFML Export	This field shows the name of the layer after export according to the configured layer structure[227].

 Layers are automatically imported when geometries are selected from CAD component library[231]. Normally the material category name is equal to the layer name.

 As long as OLAYERS 1.1[280] or OLAYERS 1.1 (without Prefix)[278] is used in CAD component library[227], the Mode and Tag are automatically filled depending on the entered layer name.

Buttons

	Synchronize the layers. This function verifies the usage of the layer in the database or the drawings. New layers in the drawings are imported, but this function does not create new layers in the CAD component library 276. The status of the layers are set by this function.

Layer status

A layer can have one of the following status after the synchronization.

New	The layer has been imported newly from component library.
Unused	The layer does not exist in the component library and the database does not contain any ODB object which used this layer.

If the layer exists in the component library and there is at least one ODB object saved in the database which uses this layer, the field ***Sync*** will be empty. In this case no special status is displayed.

7.3.5 Dialog 2D-Layer

In this dialog the 2d layers which are used in a product line are maintained. This dialog is used to assign specific modes to a layer. These modes are used according to OLAYERS 1.1. They are set automatically if one of the OLAYERS base layer structures is used in component library 227.

 In pCon.creator the names of layers and blocks are used neutrally according to any layer structure. Therefore they do not contain any special prefix in the database. Specific layer structures are just used at the interfaces from CAD component library to pCon.creator and the export to OFML.

Fields

Sync	This field shows the status of the layer after the last synchronization.
Layer	The name of the layer is entered in this field.
Mode	This field configures the mode of the layer according to OLAYERS 1.1
Tag	This field configures a special tag of the layer according to OLAYERS 1.1.
Comment	A description of the purpose of the layer can be entered in this field.
OFML export	This field shows the layer name how it is build by export and used at runtime. It is build according to the selected layer structure on export setting 227.

 Layers are automatically imported when geometries are selected from CAD component library 231.

 As long as OLAYERS 1.1 280 or OLAYERS 1.1 (without Prefix) 278 is used in CAD component library 227, the Mode and Tag are automatically filled depending on the entered layer name.

Buttons

	Synchronize the layers. This function verifies the usage of the layer in the database or the drawings. New layers in the drawings are imported, but this function does not create new layers in the CAD component library 276. The status of the layers are set by this function.

Layer status

A layer can have one of the following status after the synchronization.

New	The layer has been imported newly from component library.
Unused	The layer does not exist in the component library and the database does not contain any ODB object which used this layer.

If the layer exists in the component library and there is at least one ODB object saved in the database which uses this layer, the field ***Sync*** will be empty***.*** In this case no special status is displayed.

7.3.6 Dialog 3D-Parts

In this dialog the available 3d parts are maintained.

The geometries are imported from the component library 276. For this reason they cannot be entered manually in this dialog. While import the Proxy-DWG and OFML Geometry files. According to the product line settings 227 the OBJ format or old OFF format (*.geo) files are created for each component.

In pCon.creator the names of layers and blocks are used neutrally according to any layer structure. Therefore they do not contain any special prefix in the database. Specific layer structures are just used at the interfaces from component library to pCon.creator and the export to OFML.

Fields

Name	The name of the 3d geometry. This field is read only. The geometries are just synchronized from external resources using the geometry import.

The geometry name should only consist of characters, numbers and underscores. Other special characters or white space is not allowed.

Buttons

	The import functions are accessed from this button.
	With this function the currently selected geometry, can be inserted as block reference into the modelspace of the component library 276.

Import geometries

While import of the geometries the OFML geometry files and PROXY-DWG files are created from the component library. These files are saved in the workspace's resource path. Following import functions can be used:

Import all parts	Any 3d geometry is imported newly.
Import new parts only	Only new 3d blocks are imported. If an existing block just has been modified in the library, currently all blocks have to be imported newly.
Select view point	This function opens a dialog to select a viewpoint which is used to create preview images of the blocks while import.
Import geometries	Imports 3DS geometry files

The selected viewpoint is saved user dependently in the workspace configuration.

ODB objects still may use geometries, which blocks have been deleted in the component library. According warnings are logged.

Import of 3DS geometry files

Additional geometries in 3DS format can be imported to the workspace. These files are just imported as external resource files. They are not added to the component library 276.

Only characters, numbers and the underscore are allowed in file names. Special characters and the underscore are not allowed. If the 3ds file names contain an OLAYERs manufacture - product line - prefix, this prefix will be automatically removed. OFML export automatically takes care of adding the correct prefix depending on the current product line.

If an image file is saved besides the selected 3ds files, they are automatically imported as preview images. The file name of these image files must be equal to the according 3ds files. Supported image formats are WMF, JPG or BMP.

Usually ODB data creation can be done most efficiently using the component library. This approach also has the benefit that solid models in DWG files can be resolved at runtime with a user specific resolution setting. Normally using DWG files in data (see Proxy-2 DWGs 227) a high great visualization quality can be achieved. Normally just in special cases it might by useful to use 3DS geometries instead.

Using additional 3ds geometry files is not supported any more since supporting OBJ file format in pCon.creator 2.16. This features is only still available in conjunction with using GEO file format for backwards compatibility reasons. In the past additional 3ds files could be used to circumvent the disadvantages of GEO and native DWG format in special visualization scenarios. Now using pCon.planner it is possible to join geometries containing vertex normal and texture coordinates in the component library pcr_geolib.dwg. After importing these geometries from component library in pCon.creator this information is made available to OFML runtime applications in a simplified and unique way just using OBJ and PROXY2-DWG format. An additional devious route using additional 3ds file format is not necessary anymore using OBJ.

7.3.7 Dialog 2D-Parts

In this dialog the available 2d parts are maintained.

The geometries are imported from the component library276. For this reason they cannot be entered manually in this dialog. While import the Proxy-DWG and OFML geometry files in EGMS format are created as defined in the product line settings227.

In pCon.creator the names of layers and blocks are used neutrally according to any layer structure. Therefore they do not contain any special prefix in the database. Specific layer structures are just used at the interfaces from component library to pCon.creator and the export to OFML.

Fields

Name	The name of the 2d geometry. This field is read only. The geometries are just synchronized from external resources using the geometry import.

The geometry name should only consist of characters, numbers and underscores. Other special characters or white space is not allowed.

Buttons

	The import functions are accessed from this button.
	With this function the currently selected geometry, can be inserted as block reference into the modelspace of the component library276.

Import geometries

While import of the geometries the OFML geometry files and PROXY-DWG files are created from the component library. These files are saved in the workspace's resource path. Following import functions can be used:

Import all parts	Any 2d geometry is imported newly.
Import new parts only	Only new 2d blocks are imported. If an existing block just has been modified in the library, currently all blocks have to be imported newly.

ODB objects still may use geometries, which blocks have been deleted in the component library. According warnings are logged.

7.3.8 Dialog Settings

In this dialog specific settings can be configured. These settings are distinguished into options related to the ODB module in general and options which are just used in the current workspace.

ODB module settings

The following settings are used by pCon.creator in all workspaces:

CAD System	The CAD system which is used to select geometries from component library 226 or import components from this library can be selected in this field. Following CAD applications are supported: • pCon.planner PRO • ~~Autodesk AutoCAD~~
Executable file	In this field the absolute path of the executable file of the CAD application must be defined: • using pCon.planner PRO the path to planner[64]_pro.exe • ~~using Autodesk AutoCAD the path to acad.exe~~
Parameter	In this field additional command line arguments can be defined to start the CAD application.

The additional pCon.planner Plugin X3G-CadSystemAdapter is used to access the component library in pCon.planner PRO. This plugin is automatically installed with pCon.creator setup.

Formerly Autodesk AutoCAD was supported as CAD System until pCon.creator 2.20. AutoCAD is no longer supported in future pCon.creator version since the components to use this interface are no longer maintained to support future AutoCAD versions. Please note: Using pCon.creator with any 64bit version of Microsoft Access to import 2d or 3d geometries only pCon.planner can be used.

7.3.9 Dialog Export to OFML

The Explorer context menu includes the menu item "Export to OFML". The export can be launched on the following levels of the OCD data format in the workspace:

- Module
- Manufacturer
- Product line

To export one or several commercial product lines, an export path has to be indicated as target. In the target directory, the data are created according to the storage structure defined in DSR Specification.

The export path of the target data is centrally and user-specifically stored in the workspace. The last export path is automatically set in the OCD module as well as in the ODB and OAS module for the next export performed from the same workspace.

UNC paths are not supported as destination path for the export.

Exporting one product line

When exporting one individual product line, mere partial exports 275 can be optionally performed. In this case, only the selected areas are exported.

Exporting multiple product lines

If the export is launched on the level of the distribution region, manufacturer, or module, you can select the product line to be exported. Initially, the product lines below the level on which the export is performed are already selected.

Buttons

	Using this button settings to automatically run an OFML application after successful export can be configured. (see also)

7.3.9.1 Partial exports

Selecting individual sections of a product line for export reduces the export time. This allows you to quickly verify individual corrections of a package.

The following sections can be selected:

- 2d structure
- 3d structure
- Functions
- 2d parts
- 3d parts

If the export is started from the menu bar of the objects dialog 231, it is possible to select only the current object for export.

A product line has to be exported completely before using a partial export the first time. To ensure completeness of the data at the end of a data maintenance cycle the product lines should exported completely at the cycle's end.

7.4 Component library

The component library `pcr_geolib.dwg` is the base for creation parametrized graphical objects. It contains the drawings of the several geometric parts of the products. This CAD library is saved in resource path the workspace.

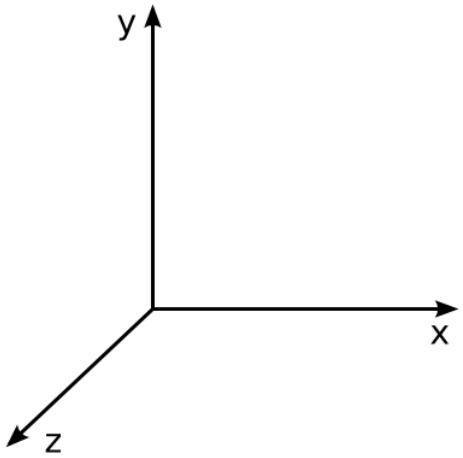
This file is saved in the workspace in following path:

`resource\geo_odb\<ManufacturerCode>\<ProductLineCode`227`>\`

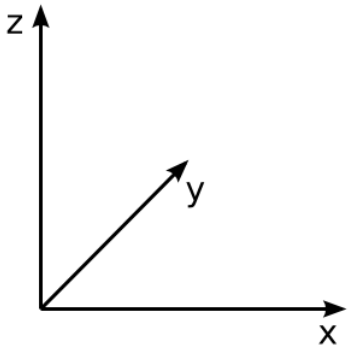
Coordinate systems

The coordinate system in DWG is rotated about 90° around the x-axis in relation to the OFML coordinate system (see figure below). The drawing in modelspace have to be created in the OFML unit meter224.

OFML coordinate system



DWG coordinate system



Although component library contains the blocks in the DWG coordinate system, the geometrical parameters are entered in relation to the OFML coordinate system in pCon.creator. These parameters are transformed automatically when selecting the geometries or parameters interactively from the modelspace of the component library in CAD-System226.

7.4.1 Layer naming convention

A naming convention of 2d and 3d layers has to be followed. The used layer naming convention has to be configured in product line management dialog 227.

Following conventions are supported by pCon.creator in the component library:

- OLAYERS 1.1 (without Prefix) 278
- OLAYERS 1.1 280
- Neutral 281

It is recommend to use OLAYERS 1.1 (without prefix) in new data creation projects.

The outdated layer naming convention FOS is not supported anymore since pCon.creator 2.15.

7.4.1.1 OLAYERS 1.1 (without Prefix)

The parts are drawn according to the OLAYERS 1.1 specification, but fully qualifying prefix is not used. The block and layer names contain only the significant mode and tag, which are described in the OLAYERS specification.

The import into pCon.creator cuts the non significant prefix. This means the name of 2d layers appears in pCon.creator 1:1 like in the drawings. But the modus is cut off 3d layers and 2d and 3d block names.

Example - 2d layers

	Name in CAD drawing	*Name in pCon.creator*	*Mode*	*TAG*
Snap layer	D2SNAP	D2SNAP	D2SNAP	
Graphik detail layer	D2DETAIL_ANY	D2DETAIL	D2DETAIL	ANY
Graphik layer	D2_ANY	D2_ANY	D2	ANY

Example - 2d blocks

Name in CAD drawing	*Name in pCon.creator*
D2_TABLETOP	TABLETOP
D2_CABLETRAY	CABLETRAY

Example - 3d layer

Name in CAD drawing	*Name in pCon.creator*
D3_TB_TABLE_TOP	TB_TABLE_TOP

Example - 3d blocks

Name in CAD drawing	*Name in pCon.creator*
D3_TABLETOP	TABLETOP
D3_TABLETOPEDGE	TABLETOPEDGE
D3_TRAVERSE	TRAVERSE

7.4.1.2 OLAYERS 1.1

The parts are drawn according to the OLAYERS 1.1 specification including the fully qualifying prefix. This prefix is cut of the block and layer names by import into pCon.creator.

Example - 2d layers

	Name in CAD drawing	*Name in pCon.creator*	*Mode*	*TAG*
Snap layer	72_XYX_DEMO1_D2SNAP	D2SNAP	D2SNAP	
Graphik detail layer	72_XYX_DEMO1_D2DETAIL_ANY	D2DETAIL	D2DETAIL	ANY
Graphik layer	72_XYX_DEMO1_D2_ANY	D2_ANY	D2	ANY

Example - 2d blocks

Name in CAD drawing	*Name in pCon.creator*
72_XYX_DEMO1_D2_TABLETOP	TABLETOP
72_XYX_DEMO1_D2_CABLE TRAY	CABLE TRAY

Example - 3d layer

Name in CAD drawing	*Name in pCon.creator*
72_XYX_DEMO1_D3_TB_TABLE_TOP	TB_TABLE_TOP

Example - 3d blocks

Name in CAD drawing	*Name in pCon.creator*
72_XYX_DEMO1_D3_TABLETOP	TABLETOP
72_XYX_DEMO1_D3_TABLETOPEDGE	TABLETOPEDGE
72_XYX_DEMO1_D3_TRAVERSE	TRAVERSE

7.4.1.3 Neutral

Neutral prefixes to distinguish 2d and 3d block and layer names are used in the drawings. The used prefixes are configured in the product line management dialog 227. They are cut off the names by the import into pCon.creator.

Example - 2d layers

	Name in CAD drawing	*Name in pCon.creator*	*Mode*	*TAG*
Snap layer	2D_SNAP	SNAP	D2SNAP	
Graphik detail layer	2D_DETAIL_ANY	DETAIL	D2DETAIL	ANY
Graphik layer	2D_ANY	ANY	D2	ANY

Example - 2d blocks

Name in CAD drawing	*Name in pCon.creator*
2D_TABLETOP	TABLETOP
2D_CABLETRAY	CABLETRAY

Example - 3d layer

Name in CAD drawing	*Name in pCon.creator*
3D_TB_TABLE_TOP	TB_TABLE_TOP

Example - 3d blocks

Name in CAD drawing	*Name in pCon.creator*
3D_TABLETOP	TABLETOP
3D_TABLETOPEDGE	TABLETOPEDGE
3D_TRAVERSE	TRAVERSE

7.4.2 Construction guidelines

All geometric parts of products and detail drawings are created within blocks in the CAD component library (pcr_geolib.dwg) . Block references of this parts are inserted into the modelspace in an assembly oriented matrix. Project specific a definite layer and block naming convention 277 has to be regarded.

The block and layer names should be entered in upper case. Just characters from A-Z, 0-9 and the underscore are allowed. Other special characters or white space are not allowed in block names and layer names.

The drawing units in modelspace is meter according to the OFML units. The block references in modelspace can be scaled accordingly.

Each block should contain logical insert points at the object. Normally this is the left, bottom point at the back of the part. But it can also be the center of rotation of a swiveling element. Blocks must not contain any scaled block reference.

Dynamic blocks, containing authoring parameters (additional user defined parameters), are not allowed.

A label or description should be provided beneath the block references in modelspace. Optionally this could be also a bill of material number of the manufacturer number.

The assembly approach should be followed to reduce the overall data volume and to be able to create parametrized objects. It is recommended to have a list of the available parts.

The component library `pcr_geolib.dwg` should be saved in DWG-Format 2004 .

7.4.2.1 Guideline for 2D geometry construction

Special characters or white space are not allowed in block and layer names.

The maximum length of a block name without prefix (mode) is limited up to 64 characters.

2d layers are assigned to the drawing entities within 2d blocks.

The maximum length of 2d layer names without prefix and without mode is restricted to 32 characters. (max. length of the layer's tag / see also OLAYERS 1.1)

7.4.2.2 Guideline for 3D geometry construction

Each component of a product which has a individual color or material hast to be drawn as a separate block. If textured materials are used like veneers or fabrics it may be recommended to create separate blocks for covers, edges and side walls of an assembly. Doing so more realistic material representations are possible.

Special characters or white space are not allowed in block and layer names.

The maximum length of block and layer names without prefix (mode) is limited up to 64 characters.

3d layers are just assigned to the block references in modelspace to the 3d blocks.

The layer "0" must be assigned to drawing entities within 3d blocks (color index = 7) . Further more these drawing entities should not contain any material assignment. The color setting of these entities is "from layer".

The 3d geometries must be drawn closed faces (solids or surfaces). Lines, splines or circles cannot be visualized as 3d geometries in OFML.

Part VII

Management of registration data

EasternGraphics
visualize your business

8 Management of registration data

8.1 Overview

The ORG Module enables the data format OFML-Registration (ORG) in pCon.creator to create and maintain registration data of OFML packages and export them to DSR. Furthermore it provides functions to release source cleared OFML datasets which can be used for final data testing processes. Additionally this module provides functions to build installation packages out of this OFML release dataset to be distributed to end users.

8.1.1 System requirements

The general system requirements of pCon.creator are valid.

Furthermore the following additional requrierments have to be complied:

Microsoft .NET Framework

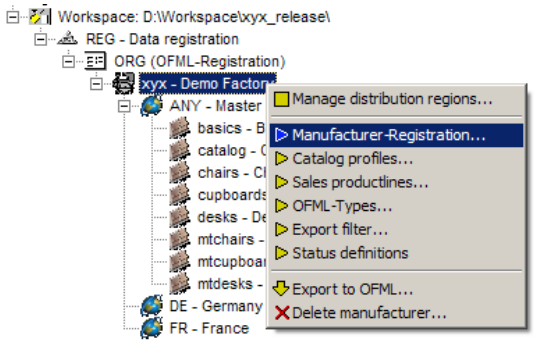
- Version 4.5

8.2 Main functions

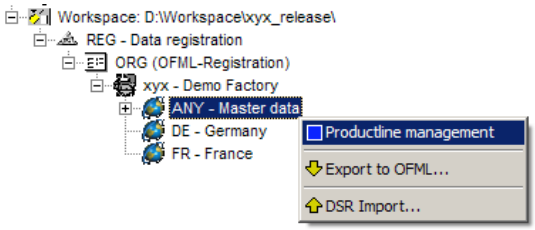
8.2.1 Management of registration data

Using the ORG module registration data of OFML packages can be maintained and exported to DSR. pCon.creator Explorer provides context menues on the different access levels to access the functions on the different access levels to maintain:

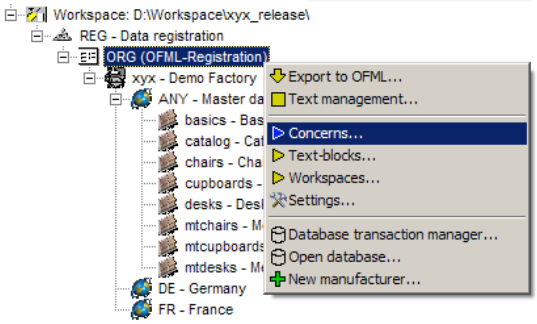
- Manufacturer registration data 316



- Package registration data 329



- optional Concern registration data 305

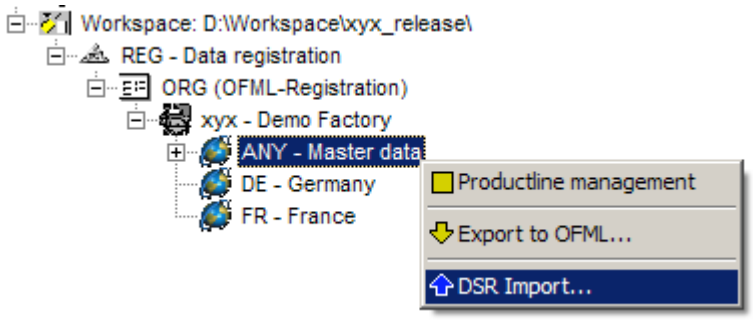


The packages' registration data is maintained within a distribution region 314. Derived distribution regions can be used to easily create registration data for product lines in different markets.

Furthermore in so called catalog profiles 318 the distribution of packages to the different markets can be configured.

8.2.2 Import of registration data

The ORG module provides a function to import DSR package registration files. This function can be started from the context menu of a distribution region in pCon.creator Explorer.



A dialog is opened to select the DSR package registration files for import.

Packages which already exist in pCon.creator will not be overwritten. In this case this package is skipped and an error is written to an import log file.

After import it might be necessary to revise the imported data manually to fit into the further data creation process with pCon.creator. For instance dependencies 333 are imported 1:1 as they are given in the DSR files complete with full version and distribution region. But using derived distribution region for registration data export it is required to remove the concrete versions and distribution regions from dependencies to sales data packages or catalogs, because they are just fixed while export automatically.

Information which could not be imported from a DSR file are documented with a warning to the import log file.

8.2.3 Release OFML data

OFML data is created within or exported to a so called source dataset[294]. Different users with different tools can be involved in complex distributed OFML data creation processes to save partial OFML data in this dataset.

This source dataset contains data which are required for functional tests while data creation and archiving purposes. But before releasing this data to end users, this dataset should to be cleared up.

The export[341] of registration data has got an option to perform this task. This data release function transfers the data from the source dataset into the so called release dataset[299], which just contains the required information for further data distribution. While this clearing process the databases (ebase), album and different ZIP container files are refreshed.

Based on this release dataset further final data tests can be performed and installation packages[292] can be finally build.

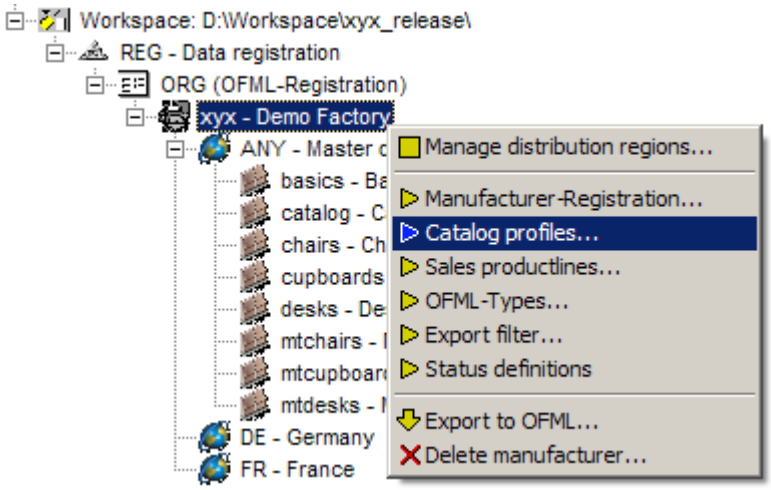
 The ORG module is capable to clear up and release OFML packages which contain data which is not maintained with pCon.creator (e.g. metatypes, materials and textures etc.) To embed this data in the OFML source dataset according instructions about should be project organization[294] noticed

8.2.4 Management of catalog profiles

To distribute OFML packages they are organized in catalog profiles. Normally for each market a separate catalog profile is maintained.

The catalog profile management 318 provides additional functions to verify and update the packages according to the current release data set. Therefore it is possible to build the according data installation packages 292 .

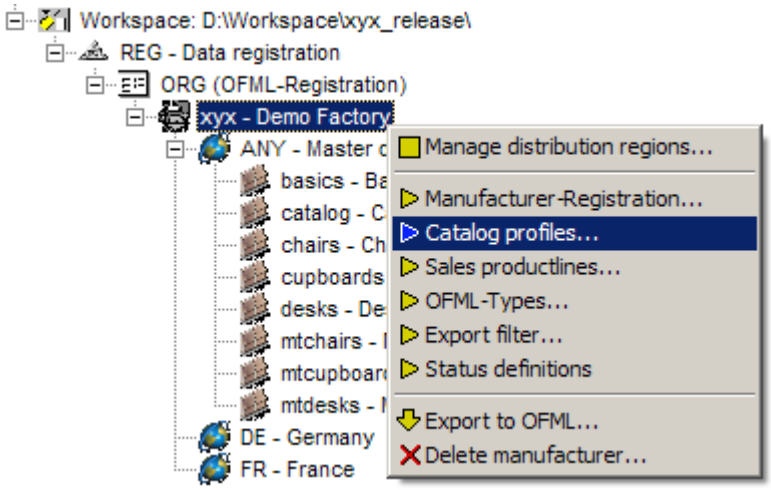
The dialog of catalog profile management can be opened from the context menu of the manufacturer node on pCon.creator Explorer.



Catalog profiles fullfill functions like data profiles and manufacturer profiles as they are mentioned in DSR specification.

8.2.5 Creation of installation packages

The dialog of catalog profile management [318] contains a function to build installation packages of the OFML data to be distributed to end users (e.g. using pCon.update)



This function creates installation packages from the release dataset [299] for each OFML data package including the manufacturer installation package. These installation packages are saved in separated paths for each catalog profile.

8.3 Project organization

pCon.creator ORG module supports different scenarios to be used in OFML data creation projects. The required settings for each scenario have to be configured in settings dialog 313.

Local management of registration data

To each workspace, in which sales, graphical or catalog data is maintained, the data format ORG can be added for maintenance of registration data for the according OFML packages. All functions of the pCon.creator for maintenance of data, creation of source cleared OFML release data and building installation package can be used in this scenario.

 Generally local maintenance of registration data is just useful in small data creation projects where OFML packages are manly contained within one workspace. In larger distributed projects registration data maintenance and release of OFML data has to be better seen as an centrally coordinated task.

Central management of registration data

Centralized management of registration data and release of source cleared OFML data as got the following advantages:

- unique maintenance of central registration data for all OFML packages
- ability for final filtering of internal test data in source dataset while data release (e.g. pure test catalogs within the OFML source dataset)
- decoupling of the data creation and data release process
- central coordination of final data testing and distribution of installation packages

For central registration data management two basic approaches can be used. First a so called repository for workspaces can be used. This is an ordinary central directory in file system where the last final state of pCon.creator workspaces are saved after a data creation cycle. These workspaces are normally saved by the persons responsible for data creation or they are send to a central role (e.g. the project manager).

In a new separate workspace the registration data is maintained for all workspaces within this repository. Each product line 335 can be assigned to one of the external workspaces in the repository. The workspaces in the repository contain besides the pCon.creator database the finally exported OFML source dataset of the productlines in a separate sub folder (e.g. _ofml). Then the data release 290 function automatically collects the OFML source data from each of these workspaces to save the source cleared OFML data in release dataset.

Alternatively the OFML source datasets of all product lines can be saved within a separate central directory together.

8.3.1 OFML-Source dataset

The OFML source data set contains all OFML data parts created with the different tools. It has to be distinguished from the release dataset in the point that it still contains all data in plaintext, while in release dataset the data is summarized in different data containers.

Although the OFML source dataset can be used to perform functional tests while data creation phase, final data tests and the creation of installation packages for distribution to end users should be done using the source data cleared release dataset290 set.

It is essential for correct data release, that the OFML source dataset contains additional meta informationen300 to describe its content. Depending on this information the ORG module can determine which data has to be build to which data containers of the release dataset.

This meta information consists of so called partial content descriptor files. These files are created automatically by the data creation tools while export. These files are normally saved within the corresponding data path and they are not allowed to be deleted manually. Furthermore if data is added manually to the OFML source dataset according content descriptor files have to be supplied, too.

Each pCon.creator module creates the required content descriptor files while export automatically. Currently the data release function of the ORG module supports the following data formats automatically in source dataset without that the user has to supply the according content descriptor files:

- materials and textures
- OFML class and string resource files
- Property value preview images
- Article specific views ASV

Furthermore the data creation tools are using existing content descriptor files for clear up while further exports.

8.3.1.1 Metatypes

The metatype CSV data tables have to be saved in path meta in distribution region of a product line's OFML source dataset. Additionally the partial descriptor files `mt.<tablename>.inp_descr` for EBASE database 300 creation have to be saved, too.

For compatibility reasons meta type data in a packages graphics path are supported alternatively.

Depending on this information the ORG module rebuilds the EBASE database while data release.

If partial descriptor files for each metatype data table are not supplied, but only one complete descriptor file `mt.inp_descr` for all tables this existing file is used to rebuild the metatype database.

8.3.1.2 Metaplanning

The metaplanning CSV data tables have to be saved in graphic path of a product line's OFML source dataset. Additionally the partial descriptor files `mt.<tablename>.inp_descr` for EBASE database 300 creation have to be saved, too.

Depending on this information the ORG module rebuilds the EBASE database while data release.

If partial descriptor files for each metaplanning data table are not supplied, but only one complete descriptor file `mp.inp_descr` for all tables, the data release process 290 uses this existing file to rebuild the metaplanning database.

8.3.1.3 Metadialogs

The metadialog CSV data tables have to be saved in graphic path of a product line's OFML source dataset. Additionally the partial descriptor files `md.<tablename>.inp_descr` for EBASE database creation have to be saved, too.

Depending on this information the ORG module rebuilds the EBASE database while data release.

If partial descriptor files for each metaplanning data table are not supplied, but only one complete descriptor file `md.inp_descr` for all tables this existing file is used to rebuild the metaplanning database.

Furthermore the metadialog implementations and resources have to be saved in graphic path in the product line's OFML source dataset. It is the most common practice to save them in a separate sub folder `md` . Because these files have to be copied to release dataset while data release, an additional partial content descriptor file `alb.md.ext_list` has to be saved too directly in graphic path. This file contains the relative file names to be copied in each line.

Example: Relative paths in partial content descriptor file alb.md.ext_list

```
md\md_demo.dll
md\md_demo.dll.manifest
md\gsr\md_demo_de.gsr
md\gsr\md_demo_en.gsr
md\gsr\md_demo_fr.gsr
md\gsr\md_demo.es.gsr
md\gsr\md_demo.it.gsr
```

8.3.1.4 Metaplacement

The metaplacement CSV data tables have to be saved in path meta in distribution region of a product line's OFML source dataset. Additionally the partial descriptor files `mpl.<tablename>.inp_descr` for EBASE database creation have to be saved, too.

Depending on this information the ORG module rebuilds the EBASE database while data release.

If partial descriptor files for each metaplacement data table are not supplied, but only one complete descriptor file `mpl.inp_descr` for all tables this existing file is used to rebuild the metaplanning database.

8.3.1.5 OFML Aided Planning (OAP)

OAP - OFML Aided Planing is a new concept to develop inter product rules and intuitive interaction concepts. This new type of logics was developed with a focus to be used easily in online and offline OFML applications.

OAP data tables have to be saved in the sub folder oap of a product line's distribution region folder in OFML source data set. Additionally the partial descriptor files `oap.<tablename>.inp_descr` for EBASE database creation have to be saved, too.

Depending on this information the ORG module rebuilds the EBASE database oap.ebase while data release.

If partial descriptor files for each OAP data table are not supplied, but only one complete descriptor file `oap.inp_descr` for all tables this existing file is used to rebuild the OAP database.

The data release operation automatically supports OAP if the oap folder exists in source data set.

OAP Resource files

OAP may link images (see oap_image.csv) or preview graphics (see oap_generalinfo.csv). These resource files must be saved in parallel to the oap data tables or in relative sub folders of the oap data path exactly as defined in the data tables. The data release function of ORG module transfers these resource files to Release data set. The partial content description files oap.image.ext_list and oap.generalinfo.ext_list are automatically updated on each data release. ORG module only considers files which are linked in oap_image.csv or oap_general.info csv. This also means images which are not mentioned in oap_image.csv are not transferred automatically.

8.3.1.6 Article Specific Rendering Setup (ARS)

Article Specific Rendering Setups (ARS) allow to distribute additional optional settings in OFML data to render articles in online applications like pCon.configurator Online.

Source data of this optional partial data format is saved in the distribution region's sub folder "ars" . It consists of data tables and texture files which are saved separately in addition sub folders "db" and "image".

If a data package's distribution region folder contains the "ars" base folder, the release function of pCon.creator ORG Modul includes this additional partial data format automatically. Its data files are copied to the release data set accordingly to the ARS specification. For this purpose ORG-Modul automatically updates the following partial content description files in "ars" base folder:

ars.datatable.ext_list

considers the known data table files located in sub folder "db":

- article.csv
- variant.csv
- variant_dag.csv
- setup.csv
- view.csv
- camera.csv
- lighting.csv
- light.csv
- origin.csv
- options.csv

ars.image.ext_list

considers all image files with file extension png located in sub folder "image".

8.3.2 OFML-Release dataset

The source data cleared OFML dataset just contains OFML data packed into their specific data containers 300. Apart from less exceptions it does not contain any OFML data in plaintext form.

Depending on this release dataset final reliable data tests can be performed. Furthermore it will be the basis for building installation packages 292 for distribution to end users.

The main advantages of distributing just the release dataset are:

- less data volume
- no temporary garbage files
- higher runtime performance of the data
- higher data security

8.3.2.1 OFML data containers

Following types of data containers in the release dataset can be distinguished:

Album

The album contains graphic files, materials and textures as well as OFML class files and string resources. The partial content descriptor files contain on each line the relative filename of a file which has to be packaged in the album. They are separated by different scopes of their creation and have got the filename: `alb.<scope>.inp_list`

The following scopes are currently reserved by pCon.creator:

- geometry_2d
- geometry_3d
- ofmltype
- material
- stringres
- addfiles

 The file `alb.inp_list` is refreshed with each data release and an already existing one may be overwritten.

EBASE-Datenbank

The CSV data tables of the following data formats are converted to EBASE data bases and distributed with the release dataset.

Daten format	Database	Content descriptor files
OCD	pdata.ebase	pdata.<tablename>.inp_descr
ODB	odb.ebase	odb.<tablename>.inp_descr
OAM	oam.ebase	oam.<tablename>.inp_descr
Metatypen	mt.ebase (<program>.ebase>)	mt.<tablename>.inp_descr
Metaplanning	<program>_mp.ebase	mp.<tablename>.inp_descr
Metaplacement	metaplacement.ebase	mpl.<tablename>.inp_descr
Metadialog database	md.ebase	md.<tablename>.inp_descr
Data control tables	ofml.ebase	ofml.<tablename>.inp_descr
OAP	oap.ebase	oap.<tablename>.inp_descr

Zip-Container

Following data is saved in ZIP containers within the release dataset. The partial descriptor files contain on each line a relative filename to be included in this container.

Data	Data container file	Content descriptor files
XCF data	xcf.zip	xcf.<scope>.inp_list The scope `catalog` is reserved by pCon.creator.
Catalog images	image.zip	image.<scope>.inp_list The scope `catalog` is reserved by pCon.creator.
Property value preview images	mat.zip	mat.<scope>.inp_list The scope `propinfo` is reserved by pCon.creator.

The files xcf.inp_list, image.inp_list und mat.inp_list refreshed with each data release and already existing ones may be overwritten.

Copied files

Exceptionally several files are just copied to a product lines graphics path in release dataset. e.g. Proxy-1 library DWG files or metadialog resources. The according partial content descriptor files contain on each line the the relative filename of a file which has to be copied.

The name of these content descriptor files in a product lines graphical path are: `alb.<scope>.ext_list`

Following scopes are reserved by pCon.creator

- geometry
- material
- addfiles

Furthermore in following sub folders of the distribution region folder resource files are copied to release data set depending on the specific content descriptor files:

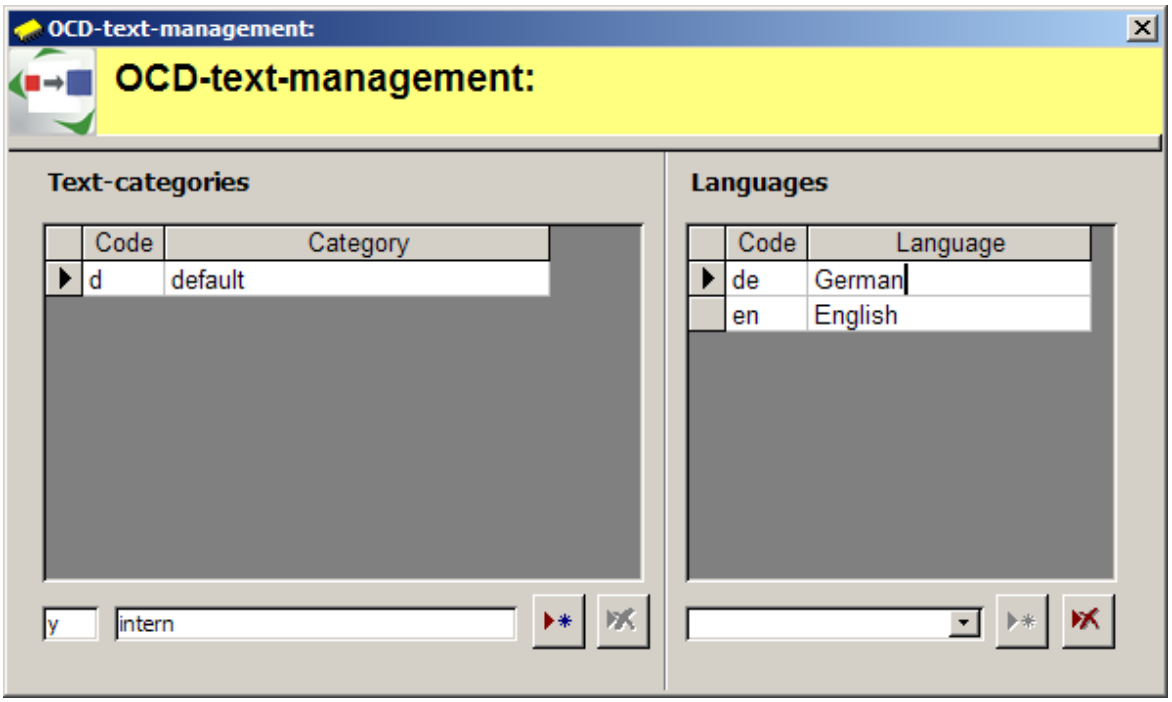
Data	Sub folder in distribution region	Content descriptor files
HTML information pages in XCF catalogs	html	html.<scope>.ext_list The scope `catalog` is reserved by pCon.creator.
Additional XCF catalog resources	etc	etc.<scope>.ext_list The scopes `catalog`, `geometry`, `group`, `scene`, `pdf`, `pec` and `asv` are reserved by pCon.creator.
OAP resource files	oap	oap.<scope>.ext_list The scopes `image` and `generalinfo` are reserved by pCon.creator.
ARS data files	ars	ars.<scope>.ext_list The scopes `datatable` and `image` are reserved by pCon.creator.

The files alb.ext_list, html.ext_list and etc.inp_list are refreshed with each data release and already existing ones may be overwritten.

Please note: Date release transfers only known files to release data set. Project specific additional files are not automatically transfered as long as they are not mentioned in one of the known partial content descriptor files e.g. alb.<anythingelse>.ext_list. Especially additional sub folders in distribution region are not considered to be released.

8.4 User interface

8.4.1 Dialog Text management



This dialog is used to maintain the languages for which localized texts are entered in the current workspace.

It is required to enter at least one language.

Fields

Code	The unique two digit identier according to the international stanard ISO-639-1 is displayed.
Language	This field shows the name of the language. The language cannot be edited but directly depends on the language code.

Buttons

	Add a new language
	Delete the selected language
	Synchronize the languages

 Currently synchronization of languages is still not supported.

8.4.2 Dialog Concern registration

Multiple manufacturers 316 may belong to one Concern or affiliated group of companies. This dialog is used to maintain registration data for this Concern.

Fields - Concern registration

OFML-Code	This field contains the registered unique code. The registered codes of concerns, for which at least one manufacturer is activated with a license, can be selected.
Label	A language independent title of the concern. If this field is left empty the registered concern title will be added automatically.
Text	In this optional field a text block 306 can be assigned, which contains a localized title of the concern..

While adding a manufacturer to the ORG data format in a workspace its optionally registered concern is added automatically.

Manufacturers and concerns are registered in a central database. Please contact our support if your concern cannot be selected, yet.

Fields - Logo

large	This field shows a concern logo. The picture's height has to be 40 pixel and its width is not allowed to exceed 200 pixel. The logo picture can be imported from file system using a double click or the context menu.
small	This field show the smaller version of the concern logo. This picture's height has to be 20 pixel and its width is not allowed to exceed 100 pixel. The logo picture can be imported from file system using a double click or the context menu.

Image files in formats JPG or PNG can be used as logos. The logo files are saved as part of the workspace resources.

Field DLM

The listbox DLM contains the required licensing files to activate the concerns data in OFML runtime applications. Using the context menu DLM files can be added or removed. Furthermore the keyboard shortcuts `INSERT` or `DELETE` can be used. The DLM files themselves are saved in the local resource path of the workspace and are transfered automatically to the OFML release dataset while data release 290.

8.4.3 Dialog Text blocks

Using this dialog localized texts which may used throughout the registration data can be maintained. These text blocks are separated by their usage as:

- Product line name 329
- Manufacturer name 316
- Catalog title 318
- Concern title 305
- Description (catalog 318 / product line 329)
- Copyright 329
- Sales product lines 324

Fields

Name	This field contains a unique language independent identifier of the localized text block.
Text fields	For each language which has been added in Text-Management 303 a separate field is available to maintain the localized description.

 The left area of the dialog lists the texts of the individual languages in form of a table. The right area allows you to oppose the respective texts in two languages and to edit them. The respective languages have therefore to be set in the selection fields. The detailed view shows the maximum length one text line is allowed to have.

The translatable texts can be exported to Excel and of course reimported from Excel. This function can be used to for external translation processes. Using the command button  the Export 307 and Import 308 of texts can be started.

8.4.3.1 Export text blocks

Translatable texts can be exported to translate them externally. Currently just a specific excel data format 308 is supported. Any text block type is exported. It is not required to export e.g. product line names or descriptions.

Settings

Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
Source language	This field configures the source language. The texts of this language are export as reference for the translators.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Export	Start the export after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

8.4.3.2 Import text blocks

After translation of exported texts[306] the updated texts can be reimported to pCon.creator. The used data format is the same like the exported excel data format[308].

Settings

Language	In this field a specific language can be selected of which the translatable texts are exported. The specific value * activates all languages for export.
only empty texts	Activating this field only the text blocks with missing translations are exported. Otherwise any text block including current translations are exported.
Path	This field defines the destination directory where the texts are exported.

Buttons

	Open a dialog to select the destination directory in file system.
Import	Start the import after all settings have been configured. This button is deactivated as long as any setting is missing.
Close	Close the dialog.

The file names of the exported data is fixed by the data format.

The excel file format contains additional information to identify the text blocks correctly. For this reason only excel files which have been exported[306] can be reimported. For this reason it can be possible that excel files cannot be reimported if e.g. these files have been created newly while the external translation process. Please contact support in such cases.

The excel files have to be closed. Otherwise they cannot be imported in any case.

8.4.3.3 Excel text data format

The following section describes the supported Excel data format for export and import of translatable texts.

Filenames

A specific excel workbook is created for each language. The file name of these workbooks identifies the text category and language. It is build in the following scheme:

```
pcr_text_reg_org_<language>.xls
```

where <language> matches the two-digit target language code (ISO-639).

Workbook

A workbook contains all translatable texts of multiple product lines and text types. Different text types are saved in different worksheets.

Worksheet fields

B	***Name***	The name of the text block.
D	***Source***	The translatable text in the source language.
E	***Target***	The translated or updated text in target language.

Custom document properties

The excel workbook contains additional custom document properties to identify the data. Following properties are used:

pcr_appid	An identificator of the application system. The value "pcr" is fixed for this property.
pcr_dbtype	The data format type. The value "text" is fixed.
pcr_FormatDomain	The code of the pCon.creator data format domain. The value "REG" is fixed.
pcr_FormatName	The name of the pCon.creator data format. The value "ORG" is fixed.
pcr_FormatVersion	The version of the excel data format. The value is "1.0.0"
pcr_LanguageCode	The two-digit target language code (ISO-639)
pcr_LocItemTable_<texttype>	For each text type these property define the name of the excel worksheet where the according text blocks are saved. See below for supported <texttype> values.

Supported text types

catalog	Localized catalog name
comgroup	Language specific sales product line / sales group names.
concern	Language specific concern names
copyright	Language specific copyright texts
country	Language specified country names
description	Language specific descriptions
manufacturer	Language specific manufacturer names
program	Language specific product line names

8.4.4 Dialog Workspaces

It is possible to maintain registration data of product lines within the workspaces where sales data and graphics and so on of this product lines are maintained, too. However in large distributed data creation processes where the data creation is split into several workspaces. It is useful to maintain the registration data in a separate central workspace.

Using this dialog the workspace so called repository 293 can be managed. These external workspaces contain the product line's OFML data for which the registration data is maintained in the current central workspace. In package registration data management dialog 329 each product line can be assigned to one of these external workspaces.

Fields

Sync	This field indicates the status of workspaces after reading them from repository. This field may contain the following values: • New The workspace has been just added • Deleted The workspace is missing in repository
Label	A unique identifier of this workspace.
Folder	This field contains the relative path of the workspace in repository. This field may be left empty, if the path name is equal to the workspace's label.
OFML data path	Using this optional field a specific relative path within the workspace can be defined, where the OFML source data is saved. This field may be left empty if a central source data path is used or the local source data path matches the name "_ofml"
Type	This field configures the type of the workspace. Valid values are • Standard a default pCon.creator workspace • OFML a special workspace which just contains OFML source data, but no pCon.creator database

Workspace of type OFML may be used as organizational units to integrate OFML packages into the data release process, which just contain data which is not maintained with pCon.creator. (e.g. base libraries for materials or pure metatype packages)

Commands

Using the context menu following functions can be accessed:

New workspace ...	Adds a new workspace from the repository. The dialog is opened to selected the central workspace database pcr_workspace.mdb. The selected workspace won't be added if it is not saved in the repository or the record already exists.
Read repository ...	This function verifies that all workspaces are still saved in repository. If new workspaces are found in the repository they will be added automatically. The field ***Sync*** displays the status of the workspaces after this operation.

Please note: workspaces of type OFML are not completely supported by the functions described above. These workspace have to be added manually.

8.4.5 Dialog Settings

Repository

If registration data for product lines in external workspace[311] with local OFML source datasets is maintained in a central role. The absolute path to the repository where these workspaces are saved has to be entered here. This setting may be left empty if registration data is not maintained for external workspaces or just a central OFML source dataset is used.

Central OFML source data path

If registration data for external workspaces is maintained, but all OFML source data has been exported to a central directory, the absolute path of this OFML source dataset has to be entered here.

Central OFML release data path

This option contains the absolute path where the source data cleared OFML release dataset will be saved. Alternatively this option can be configured while export[341], too.

OFML application

At this point an OFML runtime application can be selected, which can be started from catalog profile management[318] with a specific catalog profile in OFML release dataset. Additional arguments to run the application with can be set, too. Using the option "Register catalog profile automatically" the selected catalog profile will be temporarily registered for startup in the selected application. Please note: this option just works with pCon.configurator, pCon.planner and pCon.basket. Using other applications the catalog profile has to be registered manually.

Buttons

	Select the according path or file
	Clear the according setting. The file or path itself is not deleted in file system.

8.4.6 Dialog Distribution regions

Distribution regions are a logical compilation of catalogs with specific validity. They usually map different markets.

Fields

OFML-Code	A unique code to identify the distribution region. It can be selected from a selection list of predefined codes for regions. Furthermore, the OFML code is used for registering the packages in the OFML application system.
Label	A name for the distribution region, which is used in pCon.creator for purposes of presentation, e.g. in the Explorer window.
Brand	This optional field can be used to assign a text-block 306 to define a specific manufacturer name as brand for all packages in this distribution region. In general the manufacturer name as it is configured in manufacturer registration management 316 is used for all packages automatically, so this field may be left empty.
Derived from	The distribution region from which the present one is derived 315.
Default-language	This value defines the default language for the distribution region.
Product line-filter	Defines the filter 326 for product lines 333. The filter is evaluated during the export.
Dependency-filter	Defined the filter 326 for dependencies 333. This filter is evaluated during the export.

The selected default language is used to preview texts in the further forms within the respective distribution region.

Using derived distribution regions product line filters should be used, to exclude basis librarries which just contain distribution region independent data such as graphics and materials. The should not be exported for each distribution region but may used together in market specific catalog profiles 318.

Using the field Brand product lines of a distribution region can be informally displayed under a specific manufacturer 316. But please note technically only the OFML code and Sales ID of a article's manufacturer are valid for processing the articles.

If a specific brand is used, the manufacturer name with the distribution region's default language 314 or an english translation will be exported as language independent manufacturer name to DSR package registration file

8.4.6.1 Derived distribution regions

Derived distribution regions allow an efficient creation and management of different data versions from a central dataset.

A special active price list as well as active filters can be assigned to a distribution region. These are used for data export.

The following data can be separately filtered:

- Product lines 329

No product line can be stored in derived distribution regions.

It is not possible either to derive a distribution region which already contains product lines from another distribution region. If you still desire to do so, you first have to delete the product lines.

Base libraries, which just contain distribution region independent data such as graphic or materials should be excluded from derived distribution region using an appropriate product line filter.

8.4.7 Dialog Manufacturer registration

Using this dialog registration data of a manufacturer can be maintained. The manufacturer itself can be added to the ORG data format using the pCon.creator - Explorer.

Fields - Manufacturer registration

OFML-Code	This field contains the registered OFML code of the manufacturer. This is a read only field.
Sales code	This field contains the sales manufacturer id. It is a read only field.
Label	The language independent title of the manufacturer. This is a read only field,
Concern	Using this field the manufacturer can be optionally assinged to Concern or affiliated group of manufacturers 305
Text	In this field a text-block can be optionally assigned, which contains a localized manufacturer title.

The manufacturer code , id an and the language independent title are registered in the central manufacturer database and cannot be changed in pCon.creator. Especially changed manufacturer names have to be registered again. In this case please contact your person in charge or our support.

Fields - Logo

large	This field shows a manufacturer logo. The picture's height has to be 40 pixel and its width is not allowed to exceed 200 pixel. The logo picture can be imported from file system using a double click or the context menu.
small	This field show the smaller version of the manufacturer logo. This picture's height has to be 20 pixel and its width is not allowed to exceed 100 pixel. The logo picture can be imported from file system using a double click or the context menu.

Image files in formats JPG or PNG can be used as loges. The logo files are saved as part of the workspace resources.

Fields - Address

Name	
Street	
Postbox	
ZIP	
City	
Country	
State	
Phone	
Fax	
E-Mail	This field may contain multiple e-mail addresses for contact. Each address has to be entered in a separate line.
Homepage	This field may contain multiple web addresses for contact. Each address ha to be entered in a separate line.

Field DLM

The listbox DLM contains the required licensing files to activate the manufacturers data in OFML runtime applications. Using the context menu DLM files can be added or removed. Furthermore the keyboard shortcuts `INSERT` or `DELETE` can be used. The DLM files themselves are saved in the local resource path of the workspace and are transfered automatically to the OFML release dataset while data release 290.

Fields - Miscellaneous

GLN Global Location Number	In this field the optional Global Location Number (GLN) of the manufacturer can be entered.

While adding a new manufacturer to ORG dataformat in pCon.creator - Explorer its registered Concern is added automatically, too.

Manufacturers and concerns are registered in a central database. Please contact our support if your manufacturer or concern cannot be selected, yet.

8.4.8 Dialog Catalog profiles

On this dialog the different catalog profiles of a manufacturer are maintained. In each catalog profile the data packages are listed, which represent the manufactures complete catalog. In general each catalog profile represents one base distribution region. Several OFML packages may be used as base libraries in multiple catalog profiles.

The management of the packages is independent from the management of the package's registration data. The following operations work directly on the current data in OFML release dataset.

Buttons

	Verify and update the catalog profiles 321
	Save catalog profiles and start a OFML application 322
	Build installation packages 323

Fields - Profile

Catalog ID	A unique identifier to identify the catalog profile. e.g. this code may match the OFML code of the base distribution region to indicate its distribution to a specific market.
Revision	The revision specifies the version of the catalog. The revision must be updated to a higher number if the catalog was already released and must be updated.
Base manufacturer	In this field the base manufacturer can be entered, which is used to update the data packages.
Base distribution region	In this field the base distribution region can be entered, which is used to update the data packages.
Label	An optional text-block, which represents the localized title of the catalog profile.
Description	An optional text-block, which represents the localized description of the catalog profile.
Valid from	This field can be used to enter a date from which the catalog profile can be used.
Valid to	This field can be used to enter a date until which the catalog profile can be used.
Release-date	An additional information when the catalog was released.
Brand	Using this field the catalog can be assigned to another manufacturer which is used as a brand for the catalog's products.

The fields "valid from", "valid to" or the release date do not restrict the usage of the catalog. They are just a general information.

The catalog id cannot contain white space or other special characters, which are invalid in files names. E.g. such invalid characters are: / \ : ? * < > " |

If catalog was already published and an update has to be released, this catalog has to be redistributed using a new (higher) revision.

A duplicate of a catalog profile can be created easily using the clone function (F8 / F9).

Fields - Price profiles

In a catalog an optional price profile can be used. The following fields have to be specified if a price profile is used. Furthermore an additional price profile data package has to be distributed as part of this catalog.

Region	The price profile region which is used in the catalog.
Version	The version of the price profile data package.

Fields - Data Packages

On tabsheet "Data packages" the OFML packages which are used in the catalog have to be listed.

Sync	This field displays the status of a package after applying one of the above described functions.
#	This field defines the relative position a package in the order of the profile's packages.
Manufacturer	The OFML code of the manufacturer of a package
Productline	The OFML code of the package
Distribution region	The OFML code of the distribution region of the package
Major	The major version number of the package
Minor	The minor version number of the package
Build	The build version number of the package
Published	Represent the manually edited published status of the package e.g. pCon.update. A marked package is excluded from the preselection on the dialog to build installation packages. The status will be automatically reset if the version number is raised by the "update" function. The published state will not be automatically set when the installation package is exported. It is necessary to set it manually after these packages where made available on pCon.update. The context menu of this field provides a function to set it for all packages in the current catalog profile.

One package can be added only once within a catalog profile. Multiple distribution regions or versions of a package are not allowed within a catalog profile

To build delete instead installation packages the version number of the package as to be incremented accordingly.

Fields - Information

On tabsheet "Information" URLs to websites or a PDF file in internet can be specified, which provide additional release information about the catalog. For instance the release information can consist of general descriptions or a change history. Individual URLs for different languages can be entered.

Language	The language of the web content. This field can be left empty for language independent URLs. The language can be selected from a list which is defined in Text management dialog 303.
Info url	The url to a website or PDF file in internet e.g. "http://www.EasternGraphics.com"
Format	In this field the content format has to be specified. It may be HTML or PDF.

Users of pCon.update will see this information in pCon.update DataClient

8.4.8.1 Verify and update

 Using this button the packages of a catalog profile can be verified if they are still available in release dataset.

If the current version of a package is not available anymore it will be indicated as deleted or outdated.

Furthermore this button provides a function to update a catalog profile for new versions of packages in release dataset. In this case new packages of the base distribution region in release dataset are appended automatically. Furthermore outdated packages are updated to the highest available version. After this function has been applied successfully, proposals 321 to update the revision of the catalog profile can be confirmed or rejected.

Each package of the updated or verified catalog profile has got an according synchronisation status 318.

Update revision

 The revision of an already published catalog profile has to be incremented if an update e.g. a bugfix of this catalog has to be published. The new revision of a catalog replaces older revisions of the catalog profile with the same catalog id.

While update of catalog profiles 318 the revision increment is automatically proposed according to the modifications of its packages. Data developers can confirmed or rejected this proposal..

Furthermore with manual changes in the packages list the catalog profile's revision number should be manually adapted.

Synchronization status

The field Sync displays the status of packages in catalog profiles after verification and updating. This field may contain the following values:

OK	This package is available with the used version in release dataset.
New	This package has been just added newly.
Deprecated	The current version of the package is deprecated. The release dataset contains a package with a newer version.
Updated	The version of this package has been updated to the highest version available in release dataset.
Deleted	This version of the package is not available anymore in release dataset, but it is possible that an older version still exists there.

Instead of installation packages so called delete packages are build for data packages which are indicated as deleted.

8.4.8.2 Save and execute

 Using this button on catalog profile management dialog 318 the catalog profiles can be saved to release dataset. The saved catalog profiles are verified 321 automatically while this operation.

The catalog profiles are saved in sub folder `profiles` of the release dataset. Their file names are:

```
<ManufacturerCode>_<CatalogID>.cfg
```

Furthermore using this function aftter saving the profile an OFML runtime application can be started automatically with a selected catalog profile.

The application to be started is configured on settings dialog 313. If one of the applications pCon.configurator, pCon.planner or pCon.basket have been selected, the profile may be optionally registered automatically to be loaded with this application. Using other applications manual data registration may be required.

8.4.8.3 Build installation packages

 Using this button the installation packages for further data distribution can be build for the catalog profiles. While this operation the catalog profiles are verified 321 automatically.

A dialog is opened to configure export options and selected the packages to be build. In left area the packages can be activated or deactivated in a treeview. If a delete package will be build for a data package, which is missing in release dataset, the according node is highlighted grey.

Settings

Release folder	This field configures the path of the release dataset from which the installation packages are build.
Export folder	This field defines the destination path where the installation packages are saved. The export routine saves the packages for each catalog profile in a separate sub folder: `<ManufacturerCode>\<CatalogID>`
Clear path	Using this options any data is removed from a catalog profile's export folder before the export itself starts.
No manufacturer package	This option prevents to build manufacturer installation packages.

If a catalog profile's node is activated or deactivated in the selection tree view, this status is applied to all its data packages automatically. To build only manufacturer installation packages these nodes have to be activated while any of its data packages can be deactivated using the option to "Deselect packages" from its context menu.

8.4.9 Dialog Sales product lines

This dialog is used to maintain the sales product lines or also called sales groups of articles.

They are used to group articles logically and may be used for discounting in quotation and accounting systems.

The registration data of an OFML package 329 has to mention the sales groups of the articles it contains.

Fields

Code	This field contains the unique identifier of the sales group.
Label	This field can contain a label to describe the sales group.
Text	In this field a text block 306, which contains a language dependent label of the sales product line, can be assigned.

Sales product lines are also called *Sales groups* in other systems.

8.4.10 Dialog OFML types

This dialog is used to define the OFML Types which are used for different packages' registration data.

Sorts of OFML types

Produkt database	A OFML class, which represents the used product database of the sales data. The parameter has to be left empty.
Metatype wrapper	A OFML class of a metatype into which articles of a product line are wrapped. The parameter of this type defines the static factory functions to create a metatype instance. Furthermore additional static function for initialization may be entered as parameter. These functions have to be entered by a semi-colon. The static functions may be entered without full qualifying package name. In this case the OFML classes manufacturer and packages are automatically added to these function calls while export.
Proginfo object	A OFML class, which represents the so called program information object of a product line providing general settings. The parameter has to be left empty.

Fields

Label	This field contains a unique identifier which is used to assing it to roduct lines 329.
Manufacturer	This field contains the OFML code of the manufacturer of the OFML package where the OFML type is saved.
Productline	This field contains the OFML code of the package where the OFML type is saved.
Name	The name of the OFML class.
Parameter	This field contains parameters to be used with the OFML type. The concrete values may vary depending of the OFML type's sort.

In fields Manufacturer, Productline and Name the OFML types, which are maintained in CLS module for the according OFML type's sort.

To simplify data maintenance the package registration management 329 allows to select default types, which don't have to be entered explicitly in this dialog.

8.4.11 Dialog Export filter

This entry form allows you to define the filters which are applied to the distribution regions[314] to be exported when exporting the data. They are mainly used for the export of derived distribution regions[315].

Fields

Name	A unique name of the filter is indicated here.
Description	Here you can enter a description of the filter for a more detailed explanation.
Include empty fields	All records which do not have any filter marking or which are marked with * are automatically included in the filtering.
Include-criteria	Patterns of filter markings with included records. The following placeholders can be in these patterns: ? - any character * - any number of characters
Exclude-criteria	Analogously to the include-criteria, patterns where the records are excluded during the filtering are entered here.

Prior to the data creation, the specific filter markings and their meaning for the records to be filtered have to be defined project-specifically.

8.4.12 Dialog Status definitions

Status definitions allow you to label records according to individual criteria for further processing. Furthermore, these status definitions can be linked with predefined states in transactions or synchronizations. The results of transactions are thus persistently displayed.

Fields

Label	A unique label for the status definition is entered here.
Color	A color, which is used as background color in the visualization, is selected here. The following values are possible: • 1 – Red • 2 – Yellow • 3 – Green
Transaction	A status definition can be optionally assigned to a specific transaction status 328 in this field.

Currently the transaction manager is not enabled in ORG module.

8.4.12.1 Transaction codes

The transaction states are distinguished between the places where they are set.

Source data

Deleted (Source)	The record does not exist in the destination data.
Changed (Source)	The record is different in source and destination data.
Appended (Source)	The record was appended to the destination data.
Identical (Source)	The record is identical in source and destination data.

Destination data

New (*Destination*)	The record is new in the destination data.
Changed (*Destination*)	The record is different in source and destination data.
Appended (*Destination*)	The record was appended to the destination data.
Identical (*Destination*)	The record is identical in source and destination data.

8.4.13 Dialog Package registration

The registration data for OFML libraries or product lines is maintained on this dialog.

The left area of this dialog provides the fields of the tabs Settings 330 and Miscellaneous 335 in tabular view to support power user data maintenance.

Furthermore for packages with sales data the sales product lines 332 or sales groups, they are containing, have to be listed explicitly. Additionally the dependencies of each package have to be maintained on a separate tab 333.

In the tabular view the field Features 335 is a so called bit vector. The value is the sum of the individual flags. The separate options can be maintained on the tab Miscellaneous 335. At this point the specific value of each flag is documented, too.

Packages should not be deleted before delete packages have been distributed. After these delete package have been build using the according function on dialog catalog profile management 318 they can be removed here.

After confirming to delete a product line its optional external resources are deleted automatically from workspace. Please note its data exported to release dataset is not deleted with this operation.

The field Sync displays the status of each package after the last data synchronization. Currently this operation is still not supported.

8.4.13.1 Settings

General

Label	This mandatory field contains the language independent title of this package.
OFML-Code	This field contains the unique identified of the package.
Category	This option defines the target OFML planning hierarchy, where articles of a product line are inserted. In general the following category has to be selected: • `Product data`
Name	This field can contain an optional text-block 306, which contains the localized title of the package.
Description	This field can contain an optional text-block 306, which contains a localized description of the package.
Copyright	This field can contain an optional text-block 306, which contains a localized copyright information.
Manufacturer (Brand)	This optional field can be used to assign a text-block 306 to define a specific manufacturer name as brand for a package. In general the manufacturer name as it is configured in manufacturer registration management 316 is used for all packages automatically, so this field may be left empty.

Using the field Brand product lines of a distribution region can be informally displayed under a specific manufacturer. But please note technically only the OFML code and Sales ID of a article's manufacturer are valid for processing the articles.

Release

Version-Major	The major version number of a package
Version-Minor	The minor version number of a package
Version-Build	The build version number of a package
Release date	The release date of a package
Release status	Using this field a specific release status can be set for a package, if it is finalized (FINAL) or still in development. This field can be left empty what applies to the finalized status.

The diligent incrementation of the fields of the version and the release date of the product line are absolutely necessary prior to each publication after having updated the data. Please note further guidelines for version increments.

The internal directory structure of an OFML data package is dependent on its major version. If the major version is changed ORG module will release 290 the OFML data to a new sub folder according to the new major version number. Furthermore the source data already must be saved in the source data set according to the new major version, since the release function 290 will search for it. Please refer to DSR specification for further information about the OFML data directory structure.

OFML runtime applications will display a warning message while loading a package, which has not a finalized release status.

Data formats

Product data	This field configures the product data format of the sales data in a package. If a product data format has been selected a according OFML type 325 which represents the used product database implementation has to be used, too. The standard product database class can be selected. (see below)
Mappings	This field configures the used mapping data format in this package.
Catalog	If the package contains catalog data, its used catalog data format has to be selected here.
Metatypes	If the package contains metatype data, it has to be configured in this field.
Metatype class	If the articles of this package are wrapped within a metatype, this wrapping OFML type has to be selected here. The standard Metatype class can be used here. (see below)

Guidelines for version increments Following guidelines to increment the version of an OFML package should be taken into account:

Build	Slight changes have been made to fix errors. (e.g. fixed bugs in prices or graphics)s
Minor	Further modification have been applied. (e.g. adding or removing articles or properties) Regular annual price updates should be indicated with an increment of the package's minor version, too.
Major	An increment of the major version indicates fundamental and incompatible changes of the data.

Standard product databases

If the package contains sales product data, an according OFML type[325] has to be selected, which represents the used product database implementation. The value `Default` can be selected to use default implementation. In this case depending on the used product data format version one of the following OFML types are used for registration data export:

OCD 4.3	`::ofml::xoi::xOiNativeOCDProductDB43`
OCD 4.2	`::ofml::xoi::xOiNativeOCDProductDB42`
OCD 4.1	`::ofml::xoi::xOiNativeOCDProductDB41`
OCD 4.0	`::ofml::xoi::xOiNativeOCDProductDB40`
OCD 2.1	`::ofml::xoi::xOiNativeOCDProductDB21`
OCD 2.0	`::ofml::xoi::xOiOCDProductDB20`
OCD 1.0	`::ofml::xoi::xOiOCDProductDB10`

Standard Metatype class

If articles have to be wrapped into a metatype at runtime, the used OFML-Type[325] which wraps the articles have to be entered here. The value `Default` can be used to indicate that the following default OFML type and factory function is used.

OFML-Type	`::ofml::go::GoMetaType`
Parameter	`goGetMetaType()`

8.4.13.2 Sales product lines

For each package which contains sales product data (OCD) the therein used sales product lines[324] or sales groups have to be entered here.

Fields

Sync	The status after the last data synchronization.
Code	Die ID der kfm. Serie[324]

The field Sync is currently not used.

8.4.13.3 Dependencies

Every package can contain references to other OFML packages. In this case it is dependent from these other packages. For correct data evaluation at runtime it is essential that these dependencies are maintained carefully on the separate tab for each package.

Several types of dependencies are distinguished. Please see below.

Fields

Sync	The status of the dependency after the last data synchronization.
Type	The type of the dependency.
Manufacturer	The OFML code of the manufacturer of the package.
Product line	The OFML code of the package.
Distribution region	The OFML code of the distribution region of the package.
Version-Major	The major version of the package the current is dependent on.
Version-Minor	The minor version of the package the current is dependent on.
Version-Build	The build version of the package the current is dependent on.
Export filter	A character string (filter marking) which is evaluated when applying the dependency filter 326 of the distribution region 314 to this product line.

The distribution region and version fields may be left empty. In this case the current value is determined automatically while export depending on the available packages in package management. Using derived distribution regions these fields are normally to be left empty for dependencies to other sales data or catalog packages. (please see below)

Dependency filters may be required to filter the dependencies of centralized catalog packages to product line packages which are filtered by product line filters for different distribution regions.

Currently the field Sync is not used.

Types of dependencies

OFML-Package	The dependency refers to another OFML package. Using derived distribution regions the guidelines for dependency derivation have to be considered.
OAP-Package	If using the product data of a product line also involves using OFML Aided Planning Data (OAP) from another data package. A dependency of type OAP-Package to the other central package containing the OAP data must be defined. This dependency is exported to the DSR key oap_program. Only one OAP-Package dependency can be defined. Only the manufacturer and product line code must be specified. The distribution region code and version is not required and will be ignored.
Catalog	If package with sales product data does not contain a catalog and the articles are referred just by external catalogs. These external catalog packages have to be entered as catalog dependency. The distribution regions and version fields can be left empty for this dependency type. The catalog dependency helps runtime environments to show the correct package of the articles while searching for them.
Runtime module	OFML packages could be dependent from specific modules of OFML runtime applications. In this case the module id as a combination of manufacturer, package and according version have to be entered here. Normally this type of dependency is not use in general OFML data packages.
SketchUp symbol library	Another OFML package which contains the graphical symbols in SketchUp® format SKP. Using this type of dependency is optional but its only allowed to use one external OFML-package with SketchUp® symbols for each OFML package. The application note *AN-2014-02 How to support SketchUp® with pCon* describes the requirements to use OFML data in SketchUp®. The *pCon.catalog for SketchUp® Starter Package* contains descriptions and tutorials for data creation and distribution. Please ask your contact person at EasternGraphics to receive further information or send an e-mail to info@EasternGraphics.com

Derivation of dependencies

Package can be used in derived distribution regions 315. While export of these derived packages it may be required that dependencies to other packages are derived to this distribution, too.

The distribution region and version fields are optional. If they are left empty, while export the current valid values according to the exported distribution region are determined automatically.

 Example: Dependency a base library

In general pure graphical packages or base libraries, which just contain materials and textures are used distribution region independently. They are normally maintained in the master distribution region ANY . They won't be derived into other market specific distribution regions. Sales product data packages which are dependent from these base libraries and be derived to different distribution regions have to define the fixed distribution region ANY in the dependency to the base library package.

Example: Dependency of a catalog to other sales product data packages

Catalog can refer to articles in other OFML packages. In this case the catalog package is dependent from the OFML package containing the sales product data of these referred articles. Catalogs and product data has to be saved always within the same distribution region. If derived distribution regions are used the dependencies of the catalog package to the product data packages have to be entered with an empty distribution region field. When exporting the current distribution region is determined for these dependencies.

8.4.13.4 Miscellaneous

Features

Transformation into special articles	The articles of this package can be transformed by end users into so called special articles for further modification. A specific article code scheme can be configured for such articles too. (see below)	1
Program information object has static properties	Each package can have a so called program information object to provide general settings. If a special implementation is used which contains own properties which can be configured in Product line management of the OFML runtime application, this option has to be activated. The Default program information object does not have any property.	2
Deny free positioning	Activating this feature the positions of new articles of this package in the scene are just determined by the planning system. The user cannot place them freely.	4

Transformation into special articles	The articles of this package can be transformed by end users into so called special articles for further modification. A specific article code scheme can be configured for such articles too. (see below)	1
Deny free moving in 3D	Users cannot move articles from this package freely which are already placed in a 3D scene.	8
Metadialogs	This package uses metadialogs.	16
May set final article code	This option activates the DSR Feature maySetFinalArticleSpec. Using this option OFML runtime applications are allowed to initialize articles from this package with a partial final article code which the end user can enter. This feature is deactivated by default because runtime errors can occur if it is not supported in sales data. Please refer to the sales data specification regarding additional information about code schemes for final article code generation and processing. Especially user defined code scheme are much often incomplete to initialize a specific article configuration. This feature does not have any effect or limitation on the regular function to create articles or search for articles and article variants in catalogs. It only supports an additional and just optional method to search for articles which are not explicitly present in catalogs after the end user has entered the according final article code.	32

Program information object

Each package can have a so called program information object, which provides central settings to OFML runtime environments. e.g. assignments of properties and property values to specific material preview images, which are shown in property editors of the OFML runtime environment.

The program information object is represented by a specific OFML type 325.

The `Default` OFML Type can be selected, which does not have to be maintained in OFML type management. This Default type is given by the OFML class `::ofml::xoi::xOiProgInfo` .

Furthermore optionally another package can be defined, which contains the program information object's database. This external package has to be defined as a dependency 333 too. Normally this field is left empty which means the database is saved within the current package.

Geometry export parameter

use generated mesh geometries	This option defines that the generated mesh geometries are used instead of Proxy geometries in Teigha™ based OFML applications. Normally this option should be deactivated because using the predefined meshes can result in larger plannings and reduced visualization quality and performance.

Miscellaneous

Persistency	Defines the form which is used to save articles of the product line in a planning internally. The "Standard" value is used for each new product line. This means the articles is saved by state codes. The value "Local scene" can be necessary in some individual data creation projects which make use of some special complex inter product rules. The decision which value has to be chosen throughout the data creation project has to be made in reconciliation between the project manager and the developers of inter product rules. In general different articles which are save in different persistence forms cannot interact with each other. An OFML application may chose another persistency form than configured here.
Workspace	This field can be used to assign this package to an external workspace 311 in repository, which contains the OFML data, of which the registration data is maintained in the central workspace.
Global Trade Item Number	The Global Trade Item Number (GTIN) which is assigned to this package.
Codescheme for special articles	Defines how article codes are formed for special articles (see below).
Export-Filter	A character string (filter marking) which is evaluated when applying the product line filter 326 of the distribution region 314 to this product line.
copy on release	With activation of this option the graphical data and the distribution region of this OFML package are just copied on data release. In this case neither the data is cleared up nor the data containers are refreshed. This option can be used to consider specific packages without access to the source data, because they were maintained with specific service contracts (e.g. metaplacement).

Codescheme for special articles

End users can transform standard articles to special articles in OFML application at runtime. In this process the optional "Codescheme for special articles" formats the resulting article code of the special article.

This scheme can consist of any alpha numeric character. Furthermore the following wildcard characters can be used:

- ? - This wildcard is replaced by the next digit of the original article's base article number
- * - This wildcard is replaced by the complete base article number of the original article

If left empty, the default value of this codescheme is `SPECIAL *` while `SPECIAL` is used language specifically.

Metadialog database

Normally if the option "Metadialogs" is activated. The metadialog database and implementations are saved in the current package. Optionally another package can be configured, which contains this metadialog database and implementations.

8.4.13.5 Visibility

The settings on the tabshet "visiblity" can be optionally used to restrict the visibility of the product line in different OFML applications. Normally each OFML product line is visibility in all OFML, where the data is installed.

Options

Show in all applications	This option is preselected for each new product line. This shows the product line in all OFML applications.
Hide in following applications...	Choosing this optiona, a list of OFML applications can be defined, which describes the applications where the product line is hidden. Each individual application where the product line should be hidden has to be defined using the following fields.

Fields

Module code	The unique code which identifies the application. The module codes of standard OFML applications can be chosen from the selection list.
Major version	The major version of the application. This field is optional.
Minor version	The minor version of the application. This field is optional.
Build version	The revision of the application. This field is optional.

8.4.13.6 Languages

This tabsheet can be optionally used, to define the languages which are used for localized texts in this product line. As default all languages which are defined in the workspace are used for each product line.

 Individual product lines, which are distributed only to some specific markets, may contain texts which are not translated to all languages. Using this function ensures that the OFML applications chooses a fallback default language for theses product lines in case the OFML application uses a language which is not contained in the this product line but in others of the same manufacturer.

Options

all languages	All languages which are defined in the workspace are used by this product line. This option is active as default.
only following languages:	Choosing this option a list the list of used languages can be restricted for this product line. Languages which are not used can be removed and new languages can be added. Please note in any case the usable languages have to be defined in the workspace (see here 303).

Felder

Code	The two digit ISO-639-1 code of the language.
Language	This field shows only the language name. It cannot be edited. It is filled automatically after choosing the language code.

 If the language list contains all languages which are defined in the workspace, the option "all languages" will be activated automatically.

8.4.14 Dialog OFML export

The export of registration data to OFML can be started from each data access level's context menu in pCon.creator - Explorer:

- Module
- Manufacturer
- Distribution region
- Product line

A dialog is opened to configure the options for export and selected the individual packages to be exported.

The packages to be exported can be activated or deactivated of the tree view on left side. Using the context menu of a package's node it can be activated or deactivated for all distribution regions.

Optionen

Following export options can be configured:

- DSR export 342
- Data release 343

8.4.14.1 Options DSR export

Registration data is always exported according to DSR specification. The library, manufacturer and optional concern registration files are exported.

To define the export path of registration data following scenarios have to be considered:

Local management of registration data

The destination directory for export of product lines for which the data is maintained in the current workspace. The value of this field equals the export path as it is used while exports of the different partial OFML data formats.

UNC paths are not supported as destination path for the export.

Central management of registration data

The registration data of each package[329] which source data is saved within an external workspace[311] is exported to the external workspaces OFML source dataset path. A separate export directory cannot be selected for these packages.

Central OFML source dataset

If a central OFML source dataset[313] is used, the registration for all packages is saved in this path.

Options

Codepage	In this field the codepage can be configured to save the DSR files.
Update release-date	This option sets the release data of all exported product lines to the current day.

Currently OFML runtime environments just support ANSI Codepage 1250 (ISO-Latin-1) . Nevertheless it is technically ensured that language specific special characters will be preserved while export of localized texts.

8.4.14.2 Options Data release

Optionally while Export to OFML source cleared OFML data can be exported to release dataset.

This function transfers the source cleared data to release dataset depending on the current information in the OFML source datasets and registration data of the activated packages.

Data containers such as EBASE databases, album and so on are refreshed while this operation.

Beside selecting the target directory of the release dataset following options can be used:

zipped catalogs	Using this option XCF catalogs, catalog images and property value preview images are released within ZIP container files. To improve performance of data runtime and data distribution it should be preferred to use this option, but it should be considered that eventually some older OFML runtime applications cannot process such catalogs. (e.g. pCon.basekt < 1.3)
external textures	Textures and materials are normally distributed within the album, but some older CAD based OFML runtime applications require that the textures are saved beside the album.
clear release data path	Using this option any data is removed from release data path before export.
manage catalog profiles / build installation packages	After successful export the catalog profile management 318 is opened automatically to maintain catalog profiles and build installation packages. This option is deactivated, if the export dialog has been opened from module level. In this case the catalog profile management 318 has to be opened manually from pCon.creator - Explorer

Part IX

Management of classes and logics

EasternGraphics
visualize your business

9 Management of classes and logics

9.1 Overview

The CLS module is a central management tool of OFML types. This data format already contains predefined types, which may be used for OAM article mapping or ODB data entry. These OFML types are available in any OFML runtime application of EasternGraphics GmbH.

OAM and ODB data can be create more efficently after adding the CLS data format to a workspace.

9.2 User interface

9.2.1 Dialog Product lines

Packages which contain programmed OFML classes are managed on this dialog.

Fields

OFML code	A unique product line abbreviation to identify the product line. The value in this field must correspond to the rules of a valid OFML identifier and has to be in small letters. Furthermore, the abbreviation must not contain an underscore.
Label	This field contains a label of the product line.
Version-Major	Shows the major version number of the product line. It starts with 1. Incrementing the major number means significant modifications such as adding or deleting complete product lines.
Version-Minor	Shows the minor version number of the product line. It starts with 0. Incrementing the minor number is used e.g. in case of removing or adding articles, properties, property values etc. A regular price list update requires a minor increment as well.
Version-Build	Shows the build number of the product line. It starts with 0. Incrementing the build number means a minor modification in terms of error eliminations (graphic, price etc.).
Release-Date	Date of the release of the product line of this product line version.
Remarks	Remarks about the product line can be stored here.
Base library	This field defines that this product line is a base library.

Bases libraries are product lines, which do not contain any commercial product data or catalogs. Other product line packages use them as central packages, containing the OFML classes or materials.

Buttons

	The predefined OFML types are updated in a workspace. This button is just available in product line management of the manufacturer OFML:

9.2.2 Dialog OFML types

This dialog lists the OFML types which are contained in a package. These types are provided for look up in OAM article mapping or ODB data development.

Fields

Class	The name of the OFML type.
Description	This field is used to describe the purpose of the class. The parameters which are used by this class should be described here, too.

Please note, OFML is case sensitive. The class name has to be provided with correct upper and lower case letters.

Kinds of OFML types

ODB type	The class provides geometries or geometric behaviors like interactions. It is used in the structure of an ODB objects.
Article planning type	The class represents an article in OFML. It can be used in OAM article mapping.
Product database	The class is a specialized product database.
Program information element	The class is a specialized ProgInfo object.

Buttons

Select picture	Using this button, a picture can be added to the record. A picture may be helpful to illustrate the type's behavior.

A meaningful description and precise explanation of the used parameters simplifies the further usage of the type and avoids mistakes while data development.

9.3 Predefined OFML types

The data format CLS already contains predefined OFML types of the following packages, if it is added to a workspace:

- OFML types 350
- OFML extensions 350
- GO types 351

These packages contain a selection of OFML classes, which can be used for data createion with pCon.creator. These libraries are available in each OFML runtime application of EasternGraphics GmbH. (e.g. pCon.configurator, pCon.planner, pCon.basket and pCon.xcad)

9.3.1 Package: OFML types

Manufacturer code

ofml

Package code

oi

Description

The package OI contains the classes of the OFML specification.

9.3.2 Package: OFML extensions

Manufacturer code

ofml

Package code

xoi

Beschreibung

The package XOI contains classes, which provide addition extension to the package OI 350.

9.3.3 Package: GO types

Manufacturer code

ofml

Package code

go

Description

The package GO contains ODB classes, which provide several functions for geometric components. Using these classes complex behavior of geometric components of ODB objects can be realized. (e.g. height adjustments or moveable fronts)

In ODB-Module these types can be used in nodes of the 3D structure as type class reference.

The GO package contains simple types and complex types. The complex types expect a certain hierarchy of class reference nodes. This expected hierarchy is documented in the description of each type in the type list.

Parameters and units

The parameters and their value range are documented in the description of each type in the type list. As far as not explicitly specified length parameters are expected in OFML unit meter. Angular parameters are expected in degree.

Part X

Import of commercial data

EasternGraphics
visualize your business

10 Import of commercial data

10.1 Overview

The OCD Import module enables functions to Import OCD and XOCD data from CSV tables into pCon.creator workspaces. Herewith it provides an interface to transfer sales data from ERP-System to pCon.creator.

10.1.1 System requirements

The general system requirements of pCon.creator have to be fulfilled.

Furthermore the OCD import module can only be used if the OCD module is installed.

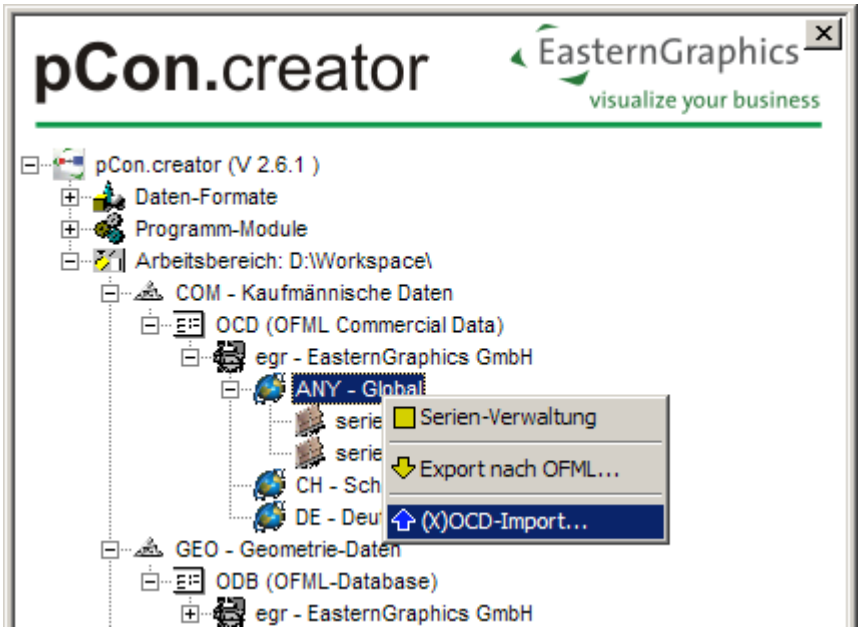
10.2 Main functions

The OCD-Import modules enables the following function in pCon.creator Explorer:

- Start (X)OCD-Import 356

10.2.1 Start (X)OCD-Import

The (X)OCD import can be started from the pCon.creator Explorer. This function can be accessed using the context menu of a distribution region in OCD data of the workspace. This distribution region will be the target for the data being imported.



A dialog 357 will be opened to configure the import settings.

10.3 User interface

10.3.1 Import settings

To import data the following settings have to be set:

- Define the datasource 359
- Select productlines for import 361
- Assign the price lists 362
- Assign the text categories 362

Options

Rename existing productlines	This option renames existing productlines in the workspace before overwriting it with new imported data. The OFML-Code of this backup gets the prefix $_ . It remains saved for further operations (e.g. to save OAM mapping data).
Cancel on errors in source data	The source data is imported in two steps. First the CSV tables are read and validated. After this step the data is transfered into the workspace. If a record is detected as invalid in first step and cannot be imported, the complete import will be cancelled. The data in workspace remains unchanged. This option can be deactivated. In this case, the data is imported into workspace, even if a record in source data was detected as invalid. But in this case pCon.creator will ignore the invalid records. This option is active as default.
Add new languages	The source data may contain localized texts in new additional languages, which are not created in the OCD-database of the workspace, yet. Using this option these new languages are created automatically while import. This option is deactivated as default. Please note: Providing data with new languages has to be well organized in data creation projects. It has to be kept in mind, that texts in other OFML parts may have to be localized, too (e.g. catalog texts or property descriptions of metatypes e.t.c).

Buttons

	This button will start the import.

The start button remains deactivated until all require settings have been made.

Old backup product lines will be deleted when a new backup is created using the option "Rename existing product lines".

10.3.1.1 Datasource

The following fields specify the data source from which the product lines 361 will be imported.

Fields

Source folder	The directory where the source data is saved is specified in this field.
Codepage	This field has to be set to specify the codepage 413 in which the source data is provided. The default value ANSI Latin-1 can be set by double click in this field.
Import type	The type of the data which will be imported. Following values can be selected: • OCD • XOCD
Import-manufacturer-code	Using the import type OCD, the OFML code of the manufacturer of the product lines in the source data has to be set.
Import-distribution-region	Using the import type OCD, the distribution region, from which data is imported from the source, has to be set.
Import-major-version	Using the import type OCD, the major version of the product lines in source data has to be set.

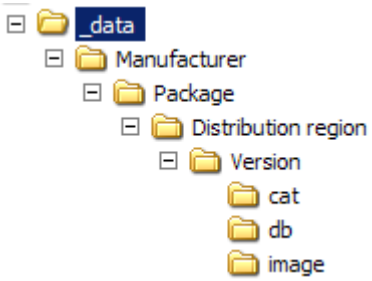
The import determines the specific format version automatically depending on the version information table 395 in the source data.

Source directory for XOCD import

XOCD data tables are saved directly in the source folder

Source directory for OCD import

Using the import type OCD, the OCD data tables have to be saved in the folder "db" of the DSR directory structure within the **Source folder** (e.g: "_data")



This directory structure will be filtered for product lines regarding the specified values in fields **Import-manufacturer-code**, **Import-distribution-region** and **Import-major-version.**

10.3.1.2 Product line selection

On the right side of the dialog the product lines found in Datasource[359] will be listed. For each product line additional information of product line management are displayed and can be maintained.

Fields

Imp	Select the product line for import
Folder	The folder where the source data of the product line is saved. Normally this field is equal to the product lines OFML-code.
!	Indicator if this product line already exists in target distribution region which would be overwritten.
OFML-code	The OFML code of the product line. This value identifies the product line. It must match the rules of a valid OFML identifier in lower case.
Sales-code	This field contains the two-digit abbreviation of the commercial product line.
Label	The label of the product lines
Version-major	The major version number
Version-minor	The minor version number
Version-build	The build version number
Release-date	The date of the release of the product line.
Remarks	Additional comments to the product line.
OCD-Exp.-Ver	The OCD format version which is used to export the data.
Rel.-coding	The field determines the relation coding language, in which the code of the relation knowledge is stored.
Valid from	This field defines the date of the first day from which on the product line is valid.
Valid to	This field defines the date of the last day until which the product line is still valid.
OAM-Exp.-Ver	The version of the OAM format, into which the article mapping data are exported.

The fields **!** and **Folder** are not editable. The other fields can be edited and specify the appearance of the product line in the target distribution region after import.

Values which cannot be read from the source data is saved from the old product line in target distribution region (e.g. OAM-Exp.-Version)

Buttons

	Select all product lines for import
	Deselect all product lines
	Invert the current selection.

10.3.1.3 Price lists

The source data contains prices of different price lists. These price lists in source data have to be assigned to the according price lists of the OCD database. At least one price list has to be assigned to a price list in the data base.

Fields

OCD-database	The price list in the database in workspace
Import-data	The price list in the source dataset.

OCD data does only contain one price list per distribution region. This price list is called default.

10.3.1.4 Text categories

The source data can contain descriptions for different text categories. These text categories in source data have to be assigned to the according text categories used in the pCon.creator workspace.

Furthermore it is possible to assign one text category from source data to multiple categories in database.

Fields

OCD-database	The text category in the pCon.creator
Import-data	The text category in source data

OCD data does only contain texts for one text category. This text category called default.

10.4 Data formats

The pCon.creator OCD-Import module enables functions to import data of the following formats:

- OCD 2.1
- OCD 4,0
- OCD 4.1
- OCD 4.2
- OCD 4.3
- XOCD 2.1
- XOCD 4.0
- XOCD 4.1
- XOCD 4.2
- XOCD 4.3

The source data is subject to some stricter restrictions than the OCD and XOCD specifications provide. The following sections describe these restrictions which meat some technical and practical aspects. They are just refinements for the data processes and don't violate the formats' specifications themselves.

The following sections explain just these refinements e.g. maximum possible field lengths and so on. A detailed description of the fields and their usage can be taken from the format specifications. Furthermore these help describes the supported tables. Some of the optional tables not listed here won't be imported.

The source data has to be provided as CSV tables.

 For data transfer from ERP systems. The XOCD data format should be preferred. XOCD tables are just an extension of the OCD tables. XOCD can contain multiple product lines, price lists and text categories within one product line. Furthermore several tables allow the product line independent usage of specific records (e.g. property classes). XOCD is just an transfer data format and may not be part of runtime OFML data.

10.4.1 OCD 2.1 - Data tables

10.4.1.1 The article table

Table name

Article

Filename

ocd_article.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***ArticleID***	x	Char	80	x	Base article number
2.	***ArticleType***		Char	1	x	Article type: C Configurable article P Plain article (not configurable)
3.	***ManufacturerID***		Char	16	x	Sales Manufacturer ID
4.	***SeriesID***		Char	16	x	Sales Series ID
5.	***ShortTextID***		Char	80	x	Short text number
6.	***LongTextID***		Char	80		Long text number
7.	***RelObjID***		Num			Relational object number
8.	***FastSupply***		Num			Fast supply counter
9.	***OrderUnit***		Char	3		Order Unit
10.	***SchemeID***		Char	30		Identifier of the code scheme for final article number generation

10.4.1.2 The article base table

Table name

ArtBase

Filename

ocd_artbase.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***ArticleID***	x	Char	80	x	Article number
2.	***PropertyClass***		Char	50	x	Identifier of the property class
3.	***PropertyName***		Char	50	x	Identifier of the property
4.	***PropertyValue***		Char	30	x	The property value

10.4.1.3 The property class table

Table name

PropertyClass

Filename

ocd_propertyclass.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***ArticleID***	x	Char	80	x	Article number
2.	***Position***		Num		x	Position of the class
3.	***Name***	x	Char	50	x	Identifier of the class
4.	***TextID***		Char	80		Text number
5.	***RelObjID***		Num			Relational object number

10.4.1.4 The property table

Table name

Property

Filename

ocd_property.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	PropertyClass	x	Char	50	x	Identifier of the property class
2.	PropertyName	x	Char	50	x	Identifier of the property
3.	Position		Num		x	Position of the property
4.	TextID		Char	80		Text number
5.	RelObjID		Num			Relational object number
6.	Type		Char	1	x	Data type
7.	Digits		Num		x	Count of digits (total)
8.	DecDigits		Num			(thereof) count of the decimal digits
9.	Obligatory		Bool		x	Obligatory property
10.	AddValues		Bool		x	allow additional values entered by user
11.	Restrictable		Bool		x	values range restrictable by constraints
12.	Scope		Char	2	x	Scope
13.	TxtControl		Num		x	print control

10.4.1.5 The property value table

Table name

PropertyValue

Filename

ocd_propertyvalue.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***PropertyClass***	x	Char	50	x	Identifier of the property class
2.	***PropertyName***	x	Char	50	x	Identifier of the property
3.	***Position***		Num		x	Position of the property value
4.	***TextID***		Char	80		Text number
5.	***RelObjID***		Num			Relational object number
6.	***IsDefault***		Bool	1	x	Default value
7.	***SuppressTxt***		Bool	1		Suppress printing
8.	***OpFrom***	x	Char	2		Operator from
9.	***ValueFrom***	x	Char	30		Property value from
10.	***OpTo***	x	Char	2		Operator to
11.	***ValueTo***	x	Char	30		Property value to

10.4.1.6 The price table

Table name

Price

Filename

ocd_price.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***ArticleID***	x	Char	80	x	(Base) article number
2.	***Variantcondition***	x	Char	50		Variant condition
3.	***Type***	x	Char	2	x	Type of price: GS Gross sales price NS Net sales price P purchase prices
4.	***Level***		Char	1	x	Price level: B Base price X eXtra charge price D Discount
5.	***Rule***		Char	10		Calculation rule
6.	***TextID***		Char	80		Text number
7.	***PriceValue***		Num		x	Price value / Amount
8.	***FixValue***		Bool	1	x	Fixed amount (or percentage)
9.	***Currency***		Char	3	(x)	Currency of the fixed amount
10.	***DateFrom***		Date	8	x	Valid from
11.	***DateTo***		Date	8	x	Valid to

The import does not distinguish between sales price types *GS* and *NS*. They are just imported as sales price types.

10.4.1.7 The relational object table

Table name

RelationObj

Filename

ocd_relationobj.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	RelObjID	x	Num		x	Relational object number (> 0)
2.	RelName		Char	80	x	Relation name
3.	Type		Char	1	x	Type of the relation: 1 Pre-Condition 2 Selection condition 3 Action 4 Constraint
4.	Domain		Char	4	x	Scope: C Configuration P Pricing relation PCKG Packaging relation

The sequence of records determines the evaluation sequence of the relations for each relation object.

10.4.1.8 The relational knowledge table

Table name

Relation

Filename

ocd_relation.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**RelationName**	x	Char	80	x	Relation name
2.	**BlockNr**	x	Num		x	Code block number
3.	**CodeBlock**		Char		x	Code block

The records have to sorted for the fields ***RelationName*** and ***BlockNr*** .

10.4.1.9 Value combination tables

Filenames

```
<Tablename>_tbl.csv
```

The name of a value combination table is limited to a maximum length of 64 characters. It must be a valid OFML identifier. Special characters or white space characters are not allowed in the name of a value combination table.

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**LineNr**	x	Num		x	Number of the table row
2.	**PropertyName**	x	Char	50	x	Name the table column
3.	**Value**	x	Char	30	x	Value in the table cell

10.4.1.10 The description tables

Table names	Filenames
ArtShortText	ocd_artshorttext.csv
ArtLongText	ocd_artlongtext.csv
PropClassText	ocd_propclasstext.csv
PropertyText	ocd_propertytext.csv
PropValueText	ocd_propvaluetext.csv
PriceText	ocd_pricetext.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***TextID***	x	Char	80	x	Text number
2.	***Language***	x	Char	2	x	Language
3.	***LineNr***	x	Num		x	Line number
4.	***Textline***		Char	80	x	Text line

The records have to sorted for the fields ***TextID***, ***Language*** and ***LineNr***.

10.4.1.11 The code scheme table

Table name

CodeScheme

Filename

ocd_codescheme.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	SchemeID	x	Char	30	x	Unique identifier of the scheme
2.	Scheme		Char			user defined description of the scheme
3.	VarCodeSep		Char	12		Character string to separate base article number and variant code in predefined schemes
4.	ValueSep		Char	12		Character string to separate property value in predefined schemes
5.	Visibility		Char	1		Visibility mode, indicating which properties are included in the variant code in predefined schemes: `0` only the current valid visible properties `1` all configurable properties
6.	InVisibleChar		Char	1		Replacement character indicating invalid or invisible properties. If this field is empty, `'-'` will be used.
7.	UnselectChar		Char	1		Replacement character indicating not selected / evaluated optional properties. If this field is empty, `'X'` will be used.
8.	Trim		Bool	1	x	Trim the property values or print values exactly as defined by the properties digits count field.

10.4.1.12 The packaging table

Table name

Packaging

Filename

ocd_packaging.csv

Fields

Nr.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Erklärung*
1.	***ArticleID***	x	Char	80	x	Base article number
2.	***Variantcondition***	x	Char	90		Variant condition
3.	***Width***		Num			Width of the packaging unit
4.	***Height***		Num			Height of the packaging unit
5.	***Depth***		Num			Depth of the packaging unit
6.	***MeasureUnit***		Char	3	(x)	Unit of measurement of the packaging unit dimensions. The unit of measurement has to be specified if a dimensions is defined. `CMT` Centimeter `FOT` Foot `INH` Inch `MMT` Millimeter `MTR` Meter
7.	***Volume***		Num			Volume of the packaging unit

Nr.	Name	Key	Type	Length	Obligation	Erklärung
8.	VolumeUnit		Char	3	(x)	Unit of measurement of the volume, The unit of measurement has to be specified if the volume is defined. INH Cubic inch LTR Liter MTR Cubic meter
9.	TaraWeight		Num			Weight of the packaging unit
10.	NetWeight		Num			Weight of of the individual article
11.	WeightUnit		Char	3	(x)	Unit of measurement of the weights. The unit of measurement has to be specified if a weight is defined. KGM Kilogramm LBR Pound MGM Milligramm
12.	ItemsPerUnit		Num			Number of articles per packaging unit
13.	PackUnits		Num			Number of packaging units

10.4.1.13 The version information table

Table name

Version

Filename

ocd_version.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***FormatVersion***		Char	14	x	Number of the used OCD format version
2.	***RelCoding***		Char	16	x	used language for relational knowledge
3.	***DataVersion***		Char		x	Database version
4.	***DateFrom***		Date	8	x	Usable from
5.	***DateTo***		Date	8	x	Usable to
6.	***Region***		Char	32	x	Distribution region
7.	***Tables***		Char		x	included tables
8.	***Comment***		Char			free comments

10.4.2 OCD 4.0 - Data tables

10.4.2.1 The article table

Table name

Article

Filename

ocd_article.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***ArticleID***	x	Char	80	x	Base article number
2.	***ArticleType***		Char	1	x	Article type: C Configurable article P Plain article (not configurable)
3.	***ManufacturerID***		Char	16	x	Sales Manufacturer ID
4.	***SeriesID***		Char	16	x	Sales Series ID
5.	***ShortTextID***		Char	80	x	Short text number
6.	***LongTextID***		Char	80		Long text number
7.	***RelObjID***		Num			Relational object number
8.	***FastSupply***		Num			Fast supply counter
9.	***Discountable***		Bool	1		Can discounts be granted?
10.	***OrderUnit***		Char	3		Order Unit
11.	***SchemeID***		Char	30		Identifier of the code scheme for final article number generation

10.4.2.2 The article base table

Table name

ArtBase

Filename

ocd_artbase.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***ArticleID***	x	Char	80	x	Article number
2.	***PropertyClass***		Char	50	x	Identifier of the property class
3.	***PropertyName***		Char	50	x	Identifier of the property
4.	***PropertyValue***		Char	30	x	The property value

10.4.2.3 The property class table

Table name

PropertyClass

Filename

ocd_propertyclass.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***ArticleID***	x	Char	80	x	Article number
2.	***Position***		Num		x	Position of the class
3.	***Name***	x	Char	50	x	Identifier of the class
4.	***TextID***		Char	80		Text number
5.	***RelObjID***		Num			Relational object number

10.4.2.4 The property table

Tabellenname

Property

Dateiname

ocd_property.csv

Felder

Table name

Property

Filename

ocd_property.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	PropertyClass	x	Char	50	x	Identifier of the property class
2.	PropertyName	x	Char	50	x	Identifier of the property
3.	Position		Num		x	Position of the property
4.	TextID		Char	80		Text number of the property text
5.	RelObjID		Num			Relational object number
6.	Type		Char	1	x	Data type
7.	Digits		Num		x	Count of digits (total)
8.	DecDigits		Num			(thereof) count of the decimal digits
9.	Obligatory		Bool		x	Obligatory property
10.	AddValues		Bool		x	Are additional values allowed to be entered by user?
11.	Restrictable		Bool		x	Is the value range restrictable by constraints?
12.	Multioption		Bool		x	Can multiple values be selected by the user?
13.	Scope		Char	2	x	Scope
14.	TxtControl		Num		x	Print control code
15.	HintTextID		Char	80		Text number of a property hint text

The fields ***Multioption*** and ***HintTextID*** are imported, but this information is currently not evaluated by OFML runtime applications in OCD 4.0 implementation stage 1.

10.4.2.5 The property value table

Table name

PropertyValue

Filename

ocd_propertyvalue.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	PropertyClass	x	Char	50	x	Identifier of the property class
2.	PropertyName	x	Char	50	x	Identifier of the property
3.	Position		Num		x	Position of the property value
4.	TextID		Char	80		Text number of the property value text
5.	RelObjID		Num			Relational object number
6.	IsDefault		Bool	1	x	Default value?
7.	SuppressTxt		Bool	1		Suppress printing
8.	OpFrom	x	Char	2	x	Operator from
9.	ValueFrom	x	Char	30	x	Property value from
10.	OpTo	x	Char	2		Operator to
11.	ValueTo	x	Char	30		Property value to
12.	Raster	x	Char	30		Raster in interval

10.4.2.6 The price table

Table name

Price

Filename

ocd_price.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***ArticleID***	x	Char	80	x	(Base) article number
2.	***Variantcondition***	x	Char	80		Variant condition
3.	***Type***	x	Char	1	x	Type of price: S sales price P purchase prices
4.	***Level***		Char	1	x	Price level: B Base price X eXtra charge price D Discount
5.	***Rule***		Char	10		Calculation rule
6.	***TextID***		Char	80		Text number of a price text
7.	***PriceValue***		Num		x	Price value / Amount
8.	***FixValue***		Bool	1	x	Fixed amount (or percentage)
9.	***Currency***		Char	3	(x)	Currency of the fixed amount
10.	***DateFrom***		Date	8	x	Valid from
11.	***DateTo***		Date	8	x	Valid to
12.	***ScaleQuantity***		Num			Scale price quantity unit
13.	***RoundingID***		Char			ID of the pricing component's rounding rule

The field ***ScaleQuantity*** is not evaluated by OFML runtime applications using OCD 4.0 implementation stage 1. For this reason values in this field won't be imported.

10.4.2.7 The rounding rule table

Table name

Rounding

Filename

ocd_rounding.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	ID	x	Char	50	x	ID of the rounding rule
2.	Number	x	Num		x	The relative position in evaluation sequence
3.	Minimum	x	Char			the smallest amount, which is still rounded
4.	Maximum		Char			The largest amount, which is not rounded anymore
5.	Type		Char	4	x	Rounding type: `DOWN` round down `UP` round up `COM` commercial rounding `ECOM` round to even / banker's roundings
6.	Precision		Num		x	Precision of the rounding
7.	AddBefore		Num		x	A fix amount which is added before rounding.
8.	AddAfter		Num		x	A fix amount which is added after rounding.

10.4.2.8 The taxation scheme tables

Table names	Filenames
TaxScheme	ocd_taxscheme.csv
ArticleTaxes	ocd_articletaxes.csv

Fields of the TaxScheme table

No.	Name	Key	Type	Length	Obligation	Description
1.	***TaxID***	x	Char		x	Identifier of the taxation scheme
2.	***Country***	x	Char	2	x	Country code (ISO-3166-1)
3.	***Region***	x	Char	3		Region code (ISO-3166-2)
4.	***Number***	x	Num		x	Relative position in evaluation sequence if the scheme contains multiple taxation types.
5.	***TaxType***		Char	8	x	Taxation type: VAT Value added tax

6.	***TaxCategory***		Char	24	x	Taxation category: for type VAT
						`standard_rate`
						`reduced_rate`
						`super_reduced_rate`
						`parking_rate`
						`services`
						`zero_rate`
						`exemption`

Fields of the ArticleTaxes table

No.	Name	Key	Type	Length	Obligation	Description
1.	***ArticleID***	x	Char		x	(Base) article number
2.	***TaxID***	x	Char		x	Identifier of the taxation scheme
3.	***DateFrom***	x	Date	8		Valid from
4.	***DateTo***	x	Date	8		Valid to

10.4.2.9 The relational object table

Table name

RelationObj

Filename

ocd_relationobj.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***RelObjID***	x	Num		x	Relational object number (> 0)
2.	***Position***		Num		x	The relative position of the relation in the relations objects evaluation sequence.
3.	***RelName***		Char	80	x	Relation name
4.	***Type***		Char	1	x	Type of the relation: 1 Pre-Condition 2 Selection condition 3 Action 4 Constraint 5 Reaction
5.	***Domain***		Char	4	x	Scope: C Configuration P Pricing relation PCKG Packaging relation

10.4.2.10 The relational knowledge table

Table name

Relation

Filename

ocd_relation.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***RelationName***	x	Char	80	x	Relation name
2.	***BlockNr***	x	Num		x	Code block number
3.	***CodeBlock***		Char		x	Code block


The records have to sorted for the fields ***RelationName*** and ***BlockNr*** .

10.4.2.11 Value combination tables

Filenames

```
<Tablename>_tbl.csv
```

The name of a value combination table is limited to a maximum length of 64 characters. It must be a valid OFML identifier. Special characters or white space characters are not allowed in the name of a value combination table.

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***LineNr***	x	Num		x	Number of the table row
2.	***PropertyName***	x	Char	50	x	Name the table column
3.	***Value***	x	Char	30	x	Value in the table cell

10.4.2.12 The description tables

Table names	Filenames
ArtShortText	ocd_artshorttext.csv
ArtLongText	ocd_artlongtext.csv
PropClassText	ocd_propclasstext.csv
PropertyText	ocd_propertytext.csv
PropHintText	ocd_prophinttext.csv
PropValueText	ocd_propvaluetext.csv
PriceText	ocd_pricetext.csv
UserMessage	ocd_usermessage.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**TextID**	x	Char	80	x	Text number
2.	**Language**	x	Char	2	x	Language
3.	**LineNr**	x	Num		x	Line number
4.	**LineFormat**		Char	1		Line formatting code
5.	**Textline**		Char	80	x	Text line

The records have to sorted for the fields ***TextID***, ***Language*** and ***LineNr***.

The field ***LineFormat*** is not evaluation by OFML runtime applications using OCD 4.0 implementation stage 1. For this reason values in this field are not imported.

10.4.2.13 The code scheme table

Table name

CodeScheme

Filename

ocd_codescheme.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	SchemeID	x	Char	30	x	Unique identifier of the scheme
2.	Scheme		Char			User defined description of the scheme
3.	VarCodeSep		Char	12		Character string to separate base article number and variant code in predefined schemes
4.	ValueSep		Char	12		Character string to separate property value in predefined schemes
5.	Visibility		Char	1		Visibility mode, indicating which properties are included in the variant code in predefined schemes: `0` only the current valid visible properties `1` all configurable properties
6.	InVisibleChar		Char	1		Replacement character indicating invalid or invisible properties. If this field is empty, `'-'` will be used.
7.	UnselectChar		Char	1		Replacement character indicating not selected / evaluated optional properties. If this field is empty, `'X'` will be used.
8.	Trim		Bool	1	x	Trim the property values or print values exactly as defined by the properties digits count field.
9.	MO_Sep		Char			Character string to separate the selected values of a multi option property.
10.	MO_Bracket		Char	2*12		Characters strings, which represent brackets around the selected values of a multivalue property. The first half characters represent the left bracket, while the last half represent the right bracket.

10.4.2.14 The packaging table

Table name

Packaging

Filename

ocd_packaging.csv

Fields

Nr.	Name	Key	Type	Length	Obligation	Erklärung
1.	***ArticleID***	x	Char	80	x	Base article number
2.	***Variantcondition***	x	Char	90		Variant condition
3.	***Width***		Num			Width of the packaging unit
4.	***Height***		Num			Height of the packaging unit
5.	***Depth***		Num			Depth of the packaging unit
6.	***MeasureUnit***		Char	3	(x)	Unit of measurement of the packaging unit dimensions. The unit of measurement has to be specified if a dimensions is defined. `CMT` Centimeter `FOT` Foot `INH` Inch `MMT` Millimeter `MTR` Meter
7.	***Volume***		Num			Volume of the packaging unit
8.	***VolumeUnit***		Char	3	(x)	Unit of measurement of the volume, The unit of measurement has to be specified if the volume is defined. `INH` Cubic inch `LTR` Liter `MTR` Cubic meter
9.	***TaraWeight***		Num			Weight of the packaging unit
10.	***NetWeight***		Num			Weight of of the individual article
11.	***WeightUnit***		Char	3	(x)	Unit of measurement of the weights. The unit of measurement has to be specified if a weight is defined. `KGM` Kilogramm `LBR` Pound `MGM` Milligramm
12.	***ItemsPerUnit***		Num			Number of articles per packaging unit
13.	***PackUnits***		Num			Number of packaging units

10.4.2.15 The version information table

Table name

Version

Filename

ocd_version.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***FormatVersion***		Char	14	x	Number of the used OCD format version
2.	***RelCoding***		Char	16	x	used language for relational knowledge
3.	***DataVersion***		Char		x	Database version
4.	***DateFrom***		Date	8	x	Usable from
5.	***DateTo***		Date	8	x	Usable to
6.	***Region***		Char	32	x	Distribution region
7.	***VarCondVar***		Char	16		User defined pricing variable.
8.	***PlaceHolderOn***		Bool		x	Activation of wildcards in comparisons with the IN operator.
7.	***Tables***		Char		x	Included tables separated by a comma
8.	***Comment***		Char			free comment

10.4.3 OCD 4.1 - Data tables

The structure of the importable OCD 4.1 data tables is described in the OCD 4.1 specification. Additional information about max length of fields of type char can be found in this help file in section about the OCD 4.0 data tables.

The OCD 4.1 specification contains tables which are not imported. Only tables which are listed in section OCD 4.0 data tables in this help are supported to be imported.

10.4.4 OCD 4.2 - Data tables

The structure of the importable OCD 4.2 data tables is described in the OCD 4.2 specification. Additional information about max length of fields of type char can be found in this help file in section about the OCD 4.0 data tables.

The OCD 4.2 specification contains tables which are not imported. Only tables which are listed in section OCD 4.0 data tables in this help are supported to be imported.

10.4.5 OCD 4.3 - Data tables

The structure of the importable OCD 4.3 data tables is described in the OCD 4.3 specification. Additional information about max length of fields of type char can be found in this help file in section about the OCD 4.0 data tables.

The OCD 4.3 specification contains tables which are not imported. Only tables which are listed in section OCD 4.0 data tables in this help are supported to be imported.

Additionally the new tables defining Property Groups are supported from OCD 4.3 too:

- PropertyGroup (ocd_propertygroup.csv)
- Article2PropGroup (ocd_article2propgroup.csv)

Note - Classification table

Since pCon.creator 2.19 the classification table ocd_classification.csv is supported. The classifications of articles according to the classification systems ECLASS and UNSPSC are imported.

This table is also implicitly imported using older OCD 4.x versions.

pCon.creator supports only classification of articles according to the classification systems like mentioned above. In OCD classification systems themselves are not defined. Accordingly the additional tables ocd_classificationdata.csv and ocd_classificationtext.csv are not supported.

10.4.6 XOCD 2.1 - Data tables

10.4.6.1 The series table

Table name

Programs

Filename

xocd_programs.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key
2.	**Program_ID**	x	Char	16	x	Sales code
3.	**Label**	x	Char	64		Productline name

10.4.6.2 The text category table

Table name

TextCategories

Filename

xocd_textcategories.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**TextCategory**	x	Char	1	x	Text category key
2.	**Label**	x	Char	64		The name of the text category

10.4.6.3 The price list table

Table name

PriceLists

Filename

xocd_pricelists.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***PriceList***	x	Char	2	x	Price list key
2.	***Label***	x	Char	32		Label of the price list

10.4.6.4 The article table

Table name

Article

Filename

xocd_article.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***Program***	x	Char	32	x	Product line key 397
2.						
11.						

OCD article table 364

10.4.6.5 The article base table

Table name

ArtBase

Filename

xocd_artbase.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***Program***	x	Char	32	x	Product line key 397
2.						
5.						

OCD article base table 365

10.4.6.6 The property class table

Table name

PropertyClass

Filename

xocd_propertyclass.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key 397
2.						
6.						

 **OCD property class table** 366

10.4.6.7 The property table

Table name

Property

Filename

xocd_property.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key 397
2.						
14.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

OCD property table 367

10.4.6.8 The property value table

Table name

PropertyValue

Filename

xocd_propertyvalue.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key 397
2.						
12.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

OCD property value table 368

10.4.6.9 The price table

Table name

Price

Filename

ocd_price.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key 397
2.	**Pricelist**	x	Char	2	x	Price list key 398
3.	**ArticleID**	x	Char	80	x	Base article number
4.						
13.						

The wildcard character * instead of a specific product line can be saved in field ***Program*** . In this case the price is valid for all product lines. This wild card character can only be used for article independent prices. Article independent prices are upcharges or discounts which are indicated with the wildcard character * in field ***ArticleID***.

OCD price table 369

10.4.6.10 The relational object table

Table name

RelationObj

Filename

xocd_relationobj.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Program***	x	Char	32	x	Product line key 397
2.						
5.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

OCD relational object table 370

10.4.6.11 The relational knowledge table

Table name

Relation

Filename

xocd_relation.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	***Program***	x	Char	32	x	Product line key 397
2.						
4.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

The records have to sorted for the fields ***Program, RelationName*** and ***BlockNr*** .

OCD relational knowledge table 371

10.4.6.12 Value combination tables

Filename

The filename of a value combination table has got the product line key as prefix, which specifies to which product line it belongs to.

```
<Program>_<TableName>_tbl.csv
```

Value combination tables can be defined product line independent. In this case they belong to all product lines in the data set. In this case the prefix is $

```
$_<TableName>_tbl.csv
```

The name of a value combination table is limited to a maximum length of 64 characters. It must be a valid OFML identifier. Special characters or white space characters are not allowed in the name of a value combination table.

Fields

The format of value combination tables is equal to the format of value combination tables in OCD.

 **OCD value combination tables** 371

10.4.6.13 The description tables

Table names	Filenames
ArtShortText	xocd_artshorttext.csv
ArtLongText	xocd_artlongtext.csv
PropClassText	xocd_propclasstext.csv
PropertyText	xocd_propertytext.csv
PropValueText	xocd_propvaluetext.csv
PriceText	xocd_pricetext.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	**Program**	x	Char	32	x	Product line key 397
2.	**TextCat**	x	Char	1	x	Text category key 397
3.						
6.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

The records have to sorted for the fields ***Program***, ***TextCat***, ***TextID***, ***Language***, and ***LineNr*** .

OCD description tables 372

10.4.6.14 The code scheme table

Table name

CodeScheme

Filename

xocd_codescheme.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Program***	x	Char	32	x	Product line key 397
2.						
9.						

The wildcard character * instead of a specific productline can be saved in field ***Program*** . In this case the record is valid for all product lines.

OCD code scheme table 373

10.4.6.15 The packaging table

Table name

Packaging

Filename

xocd_packaging.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Program***	x	Char	32	x	Product line key 397
2.						
14.						

OCD packaging table 374

10.4.6.16 The version information table

Table name

Version

Filename

xocd_version.csv

Fields

The format of this table equals its format in OCD.

OCD version information table 376

10.4.7 XOCD 4.0 - Data tables

The structure of the importable XOCD 4,0 data tables is described in the XOCD 4.0 specification. The import interface respects the field lengths, which are specified in this document.

Nevertheless some additional restrictions have to considered according to the current implementation stage of OCD 4.0 in OFML runtime environments.

The property table

The fields ***Multioption*** and ***HintTextID*** are imported, but the information of these fields is not evaluated by OFML runtime applications using OCD 4.0 Implementation stage 1.

The relational knowledge table

The records have to sorted for the fields [***Program,] RelationName*** and ***BlockNr*** .

Description tables

The records have to sorted for the fields [***Program***,] ***TextCat***, ***TextID***, ***Language***, and ***LineNr*** .

10.4.8 XOCD 4.1 - Data tables

The structure of the importable XOCD 4.1 data tables is described in the XOCD 4.1 specification. The import interface respects the field lengths, which are specified in this document.

Nevertheless some additional restrictions have to considered according to the current implementation stage of OCD 4.0 in OFML runtime environments.

The property table

The fields ***Multioption*** and ***HintTextID*** are imported, but the information of these fields is not evaluated by OFML runtime applications using OCD 4.0 Implementation stage 1.

The relational knowledge table

The records have to sorted for the fields [***Program,] RelationName*** and ***BlockNr*** .

Description tables

The records have to sorted for the fields [***Program***,] ***TextCat***, ***TextID***, ***Language***, and ***LineNr*** .

10.4.9 XOCD 4.2 - Data tables

The structure of the importable XOCD 4.2 data tables is described in the XOCD 4.2 specification. The import interface respects the field lengths, which are specified in this document.

Nevertheless some additional restrictions have to considered according to the current implementation stage of OCD 4.0 in OFML runtime environments.

The property table

The fields ***Multioption*** and ***HintTextID*** are imported, but the information of these fields is not evaluated by OFML runtime applications using OCD 4.0 Implementation stage 1.

The relational knowledge table

The records have to sorted for the fields [***Program,] RelationName*** and ***BlockNr*** .

Description tables

The records have to sorted for the fields [***Program***,] ***TextCat***, ***TextID***, ***Language***, and ***LineNr*** .

10.4.10 XOCD 4.3 - Data tables

The structure of the importable XOCD 4.3 data tables is described in the XOCD 4.3 specification. The import interface respects the field lengths, which are specified in this document.

Nevertheless some additional restrictions have to considered according to the current implementation stage of OCD 4.3 in OFML runtime environments.

The property table

The field ***MultiOption*** is imported, but only internal multivalued properties (Scope R) are supported in OFML runtime applications. Configurable multivalued properties are not supported.

The field ***HintTextID*** *is* imported, but it is not evaluated by OFML runtime applications.

The relational knowledge table

The records have to sorted for the fields [***Program,] RelationName*** and ***BlockNr*** .

Description tables

The records have to sorted for the fields [***Program***,] ***TextCat***, ***TextID***, ***Language***, and ***LineNr*** .

Notes - XOCD 4.3.1 - Classification table

Since pCon.creator 2.19 the 1. revision XOCD 4.2.1 is supported - including the classification table xocd_classification.csv .

This table is also implicitly imported using older XOCD 4.x versions.

10.4.11 Codepage

The source data tables are saved with a specific codepage. It is required to specify the used codepage in import settings 359 for correct import.

Currently OFML data supports only data in ANSI-Codepage ISO-8859-1 (Latin 1 / Windows Codepage 1252) This codepage is set as default value. Language dependent texts in the description tables in OCD or XOCD are saved in the language typical ANSI codepage. This language typical codepage will be automatically considered on import to preserve special characters.

Unicode support

The import function supports OCD or XOCD source data in Unicode. The source data can be provides in UTF-8 or in UTF-16 Little Endian.

If any of the supported Unicode codepages is recognized, the data table will be imported automatically according to this Unicode codepage instead using the one defined in import settings 359.

Data tables in UTF-16 will be automatically detected by the Byte-Order-Mark.

Data tables in UTF-8 will only be detected automatically if the optional Byte-Order-Mark is provided.

The codepages used in source data have to be installed in the operation system on which the import is executed.

Part XI

Import of catalog data

EasternGraphics
visualize your business

11 Import of catalog data

11.1 Overview

Using the OAS import module XCF catalogs can be imported into a workspace. Starting with data of existing XCF catalogs simplifies the beginning of new catalog data creation projects.

11.1.1 System requirements

The general system requirements of pCon.creators have to be fulfilled.

Furthermore the OAS import module can only be used if the OAS module is installed.

11.2 Main functions

Das OAS-Import Modul erweitert den pCon.creator Explorer um die Funktion:

- XCF-Import starten 419

Es wird nicht empfohlen den XCF Import zum Austausch von Katalogdaten zwischen verschiedenen Arbeitsbereichen zu verwenden. Im OAS Modul besteht die Möglichkeit, Katalogstrukturen zwischen verschiedenen Arbeitsbereichen zu kopieren. Diese Funktion ist besser für diesen Einsatzzweck geeignet.

----------OLD_TEXT----------

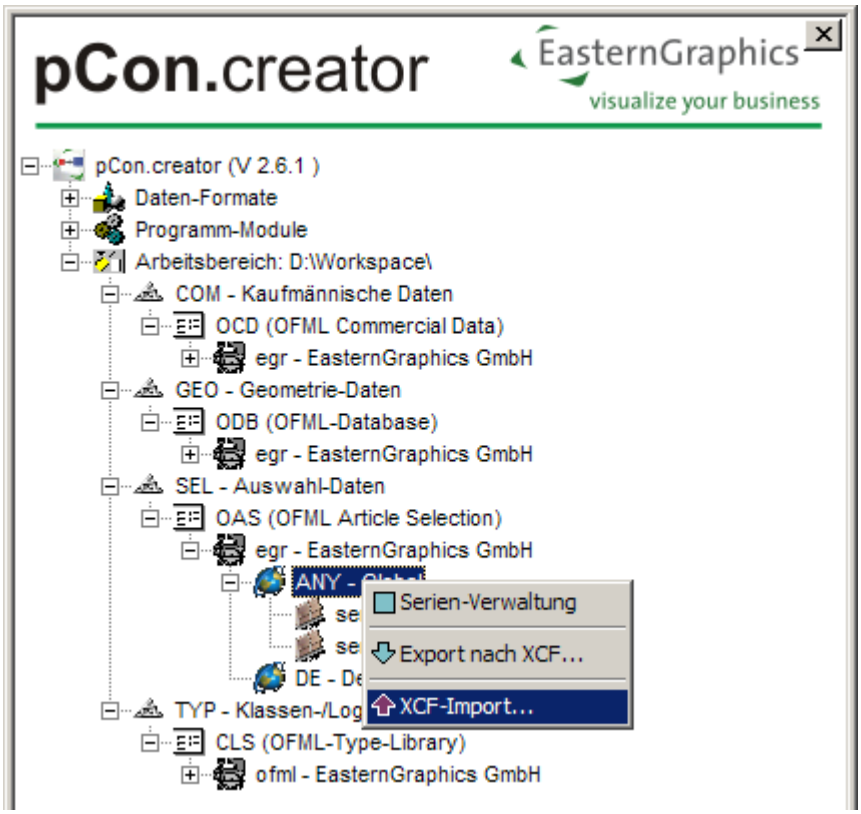
The OAS import module extends the pCon.creator Explorer with the function:

- Start XCF import 419

It is not recommended to use the XCF import for catalog data exchange between different pCon.creator workspaces. In OAS module it is possible to copy catalog structures from one workspace to another. This function suits better for this purpose.

11.2.1 Start XCF import

XCF import function can be accessed from the context menu of a distribution region node in OAS data format of the workspace. The XCF catalogs will be imported into this distribution region.



A dialog 420 appears to configure the settings for import.

11.3 User interface

11.3.1 Dialog Import settings

To import data the following settings have to be set:

- Define the datasource 421
- Select the catalogs for import 423
- Assign the text categories 424

Options

Rename existing productlines	This option renames existing productlines in the workspace before overwriting it with new imported data. The OFML-Code of this backup gets the prefix $_ .
Check classification	The catalog entries in XCF catalogs are classified automatically to match the catalog entry types used in OAS module. This option shows a dialog after this operation to verify and adjust this automatic classification of catalog entries.

Buttons

	This button starts the import.

The command button Start is disabled until all required settings have been made.

Old backup product lines will be deleted when a new backup is created using the option "Rename existing product lines".

11.3.1.1 Datasource

Die folgenden Felder legen die zu importierenden Quelldaten fest.

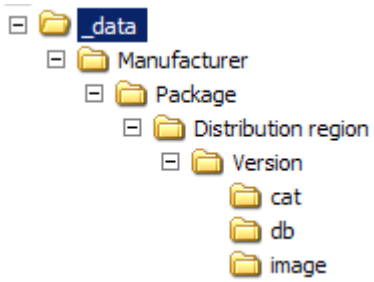
Fields

Source folder	The directory where the source data is saved is specified in this field. .
Codepage	This field has to be set to specify the codepage 432 in which the source data is provided. The default value ANSI Latin-1 can be set by double click in this field.
Import type	The type of the data which will be imported. Currently this value is fixed: • XCF
Import manufacturer code	The OFML code of the manufacturer of the catalogs in the source data has to be set.
Import distribution region	The distribution region, from which data is imported from the source, has to be set.
Import major version	The major version of the catalogs in source data has to be set

The source data tables has to be saved as CSV tables in a DSR directory structure.

Structure of the source directory

The source data can be saved an a free selectable **Source directory** (e.g. *_data*). This source directory has to contain a sub directory structure as it is specified in DSR - Data structures and regisration. The XCF data tables are saved in the folder *cat*. Catalog images are saved in the sub folder *image*.



The catalogs in this source directory are listed according to the fields *Import manufacturer code*, *Import distribution region* and *Import major version*.

11.3.1.2 Catalog selection

On the right side of the dialog the product lines found in Datasource 421 will be listed. For each product line additional information of product line management are displayed and can be maintained.

Fields

Imp	Select the product line for import
Folder	The folder where the source data of the product line is saved. Normally this field is equal to the product lines OFML-code.
!	Indicator if this product line already exists in target distribution region which would be overwritten.
OFML-code	The OFML code of the product line. This value identifies the product line. It must match the rules of a valid OFML identifier in lower case.
Label	The label of the product lines
Metatype filter	This field can be used to define a pattern of articles codes which are classified as metatype partial planings instead of sales articles. Following wild card characters can be used: ? - exactly one arbitrary character * - multiple arbitrary characters
Version-Major	The major version number
Version-Minor	The minor version number
Version-Build	The build version number
Release-Date	The date of the release of the product line.
Remarks	Additional comments to the product line.

The fields **!** and **Folder** are not editable. The other fields can be edited and specify the appearance of the product line in the target distribution region after import.

Buttons

	Select all catalogs for import
	Deselect all catalogs
	Invert the current selection.

Metatype filters

Articles in an catalog may be metatype partial plannings instead of real OCD sales articles. To classify these special entries correctly a pattern which matches the article codes for these entries can be configured.

11.3.1.3 Text categories

The source data can contain descriptions for different text categories. These text categories in source data have to be assigned to the according text categories used in the pCon.creator workspace.

Furthermore it is possible to assign one text category from source data to multiple categories in database.

Fields

OAS-database	The text category in the pCon.creator
Import-data	The text category in source data

XCF data does only contain texts for one text category. This text category called default.

11.3.2 Dialog Classification

If the import option check classification 420 is activated, this dialog appears while import, after the XCF tables have been read and the catalog entries have been automatically assigned to the OAS catalog entry types. In any case this dialog appears, if after automatic classification at least one entry remained unclassified.

In case of data errors in XCF data tables correct automatic classification cannot be ensured.

Usually manual classification is just necessary if the catalog contains metatype partial planning entries, which cannot be matched using a metatype filter 423.

Filter

The catalog entry list of a selected package can be restricted to show only entries of specific types. Additionally following special filters are selectable:

*****	All entries are shown without filtering. This filter is active as default.
unclassified	Only entries are shown, which are not classified, yet.

11.3.2.1 Package

On the left side of the classification dialog the imported product lines are listed.

Fields

Name	The label of the product lines. The value in this field cannot be modified.

11.3.2.2 Catalog entry

On the right side of the classification dialog the catalog entries of each product line are listed.

Fields

Name	This field shows the name or the article code of the catalog entry.
Variant	This field shows the variant key of the catalog entry.
Type	This field shows the automatically assigned catalog entry type. This field may be modified.

The fields *Name* and *Variant* cannot be modified.

11.4 Data formats

pCon.creator OAS import module provides functions to import catalog data from the data fromats:

- XCF 2.5 - 2.10

The source data is subject to some stricter restrictions than the CSV file specifications provide. The following sections describe these restrictions which meat some technical and practical aspects. They are just refinements for the data processes and don't violate the formats' specifications themselves.

The following sections explain just these refinements e.g. maximum possible field lengths and so on. A detailed description of the fields and their usage can be taken from the format specifications. Furthermore these help describes the supported tables. Some of the optional tables not listed here won't be imported.

The source data has to be provided as CSV tables.

Only elements which are supported in pCon.creator OAS Module are imported. The import ignores obsolete insertion types or resource keys which are not supported in pCon.creator OAS Module.

11.4.1 XCF-Tabellen

11.4.1.1 The article table

Table name

Article

Filename

article.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Article number***	x	Char	50	x	The base article code or name of the catalog entry
2.	***Variant key***	x	Char	50		
3.	***Article Category***		Num			This field is not processed.
4.	***Position number***		Char	50		This field is not processed.
5.	***Insertion type***		Char	2	x	
6.	***Visibility***		Num		x	
7.	***Package***		Char	50		The name of a foreign package, if the article is defined there instead of the current one.

11.4.1.2 The variant table

Table name

Variant

Filename

variant.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Article number***	x	Char	50	x	The base article code or name of the catalog entry
2.	***Variant key***	x	Char	50		
3.	***Variant codes***		Char			The content of this field is dependent on the sales product data. It may contain the variant code or the final article code.

11.4.1.3 The structure table

Table name

Structure

Filename

structure.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Article number***	x	Char	50	x	The base article code or name of the catalog entry
2.	***Variant key***	x	Char	50		
3.	***Level***		Num		x	
4.	***Type***		Char	1	x	The type of the structure entry.
5.	***reserved***		Char	50		

11.4.1.4 The text table

Table name

Text

Filename

text.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Article number***	x	Char	50	x	The base article code or name of the catalog entry
2.	***Variant key***	x	Char	50		
3.	***Language***	x	Char	2	x	The two digit ISO language code (ISO-639).
4.	***Article text***		Char			

11.4.1.5 The resource table

Table name

Resource

Filename

resource.csv

Fields

No.	*Name*	*Key*	*Type*	*Length*	*Obligation*	*Description*
1.	***Article number***	x	Char	50	x	The base article code or name of the catalog entry
2.	***Variant key***	x	Char	50		
3.	***Language***	x	Char	2		The two digit ISO language code (ISO-639)
4.	***Type***		Char	2	x	The resource type
5.	***Resource***		Char			

11.4.1.6 The co-resource tables

Filenames

resource_AP.csv

resource_FM.csv

resource_ZN.csv

Fields

No.	Name	Key	Type	Length	Obligation	Description
1.	Article number	x	Char	50	x	The base article code or name of the catalog entry
2.	Variant key	x	Char	50		
3.	Language	x	Char	2		The two digit ISO language code (ISO-639)
4.	Type		Char	2	x	The resource type
5.	Resource		Char			

11.4.2 Codepage

The source data tables are saved with a specific codepage. It is required to specify the used codepage in import settings 421 for correct import.

Currently OFML data supports only data in ANSI-Codepage ISO-8859-1 (Latin 1 / Windows Codepage 1252) This codepage is set as default value. Language dependent texts in the XCF text table is saved in the language typical ANSI codepage. This language typical codepage will be automatically considered on import to preserve special characters.

Unicode support

The import function supports XCF source data in Unicode. The source data can be provides in UTF-8 or in UTF-16 Little Endian.

If any of the supported Unicode codepages is recognized, the data table will be imported automatically according to this Unicode codepage instead using the one defined in import settings 421.

Data tables in UTF-16 will be automatically detected by the Byte-Order-Mark.

Data tables in UTF-8 will only be detected automatically if the optional Byte-Order-Mark is provided.

The codepages used in source data have to be installed in the operation system on which the import is executed.

Part XII

Keyboard shortcuts

EasternGraphics
visualize your business

12 Keyboard shortcuts

General

F1	Open help system
SHIFT+ENTER	Save the current edited record in a datasheet.
SHIFT+F2	Opens a dialog to edit the value of the field which has the focus. This function is useful to edit a long value, which cannot be displayed completely, because of the controls limited space.
CTRL+F6	Cycles between the open dialogs of a pCon.creator module. This keyboard might be usefull when using Access 2007 or higher. These Access versions prevents showing the dialogs in the windows taskbar. Until Access 2003 the dialogs are shown in the taskbar. The system keyboard shortcut ALT+TAB can be used to cylce between the dialogs.

Clone records

F8	The selected records are copied from a datasheet to the clipboard for cloning.
F9	Inserts records which have been copied to clipboard for cloning (F8) into a datasheet. This function creates deep clones of these records including the referenced records.
CTRL+SHIFT+C	see F8
CTRL+SHIFT+V	see F9

ODB Objects

SHIFT	Modifies the Drag&Drop operation in object structure. The dragged item is copied instead of being moved.
DEL	Delete the selected element from the object structure.
INS	Insert a new node in the object structure.
CTRL+EINFG	Select elements from AutoCAD to insert them in the object structure.
SHIFT+EINFG	Inert a new element in the object structure.
CTRL+SHIFT+NUM PLUS	Open the next two sub levels of the current node.
CTRL+SHIFT+NUM MINUS	Close all sub nodes of the current node.
CTRL+NUM MINUS	Close all nodes.
F5	Refresh the object structure.

OAS Catalog structure

SHIFT	Modifies the Drag&Drop operation in the catalog structure. The dragged item is copied instead of being moved.
CTRL	Modifies the Drag&Drop operation in the catalog structure. The dragged item is inserted as previous neighbor when dropped. The insertion mode modification can be used in combination with SHIFT.
ALT	Modifies the Drag&Drop operation in the catalog structure. The dragged item is inserted as next neighbor when dropped. The insertion mode modification can be used in combination with SHIFT.
DEL	Delete the selected item from the catalog structure.
ALT+DEL	Delete all sub items of the selected folder.
CTRL+C	Copy the selected entry to the clipboard.
CTRL+V	Paste an item from the clipboard including is sub items.
CURSOR UP	Select the previous item.
CURSOR DOWN	Select the next item.
CURSOR LEFT	Select the parent folder and collapse it.
CURSOR RIGHT	Expand the current folder and select its first child item.
CTRL+CURSOR UP	Move the selected item up. This operation can only be used within one folder level.
CTRL+CURSOR DOWN	Move the selected item down. This operation can only be used within one folder level.
CTRL+CURSOR LEFT	Unindent the selected item. It becomes the next neighbor of its current parent folder.
CTRL+CURSOR RIGHT	Indent the selected item. It becomes the last child of its previous neighbor folder.
SHIFT+CURSOR LEFT	Copy the selected items from the selection list to the catalog structure. If the currently selected item in the structure is a folder they will be inserted as its last children otherwise the will be inserted as the previous neighbor.
CTRL+SHIFT+NUM PLUS	Open the next two sub levels of the current folder.
CTRL+SHIFT+NUM MINUS	Close all sub nodes of the current folder.
CTRL+NUM MINUS	Close all folders.
F2	Change the current item in the catalog structure. Just the referenced to a catalog entry is changed. The catalog entry itself remains unmodified.
F5	Refresh the catalog structure.
F8	see CTRL + C

F9	see CTRL + V

OAS Catalog entries

CTRL+CURSOR UP	Select the previous picture in the catalog picture selection controls.
CTRL+CURSOR DOWN	Select the next picture in the catalog picture selection controls.

Additional information

F10	Toggle the visibility of the database record id (#ID) in all datasheets.
SHIFT+F10	Toggle the visibility of the source or temporary record id (#SrcID in all datasheets after a transaction.
F11	Toggle the visibility of the internal transaction status (#TAG) in all datasheets.
F12	Toggle the visibility of the user defined status in all datasheets.

Individual datasheet layouts

F7	Save the current layout of a datasheet.
SHIFT+F7	Delete a saved datasheet layout.
F6	Restore a previously saved datasheet layout.
CTRL+S	see F7
CTRL+SHIFT+S	see SHIFT+F7
CTRL+R	see F6