

Application Notes (2022-02-01)

AN-2019-01: Support of SAP features in the OCD implementation

Contents

1	Introduction	2
2	Standard language elements.....	2
3	Additionally supported language elements	3
3.1	set_default.....	3
3.2	del_default	5
3.3	type_of	5
3.4	Concatenation of strings.....	5
3.5	Alternative identifiers for comparison operators.....	5
3.6	Alternative negation of IN comparisons.....	5
4	Unsupported language elements.....	6
5	Other features.....	6
6	Differing behavior of SAP vs. pCon.....	6
6.1	Divergent data	7
6.2	Backward dependencies.....	7
6.3	Hidden errors in the data	8
	Appendix.....	9
A.1	Example for a hidden data error	9
A.2	Document history.....	10

Legal remarks

© 2022 EasternGraphics GmbH | Albert-Einstein-Straße 1 | 98693 Ilmenau | GERMANY

This work (whether as text, file, book or in other form) is copyright. All rights are reserved by EasternGraphics GmbH. Translation, reproduction or distribution of the whole or parts thereof is permitted only with the prior agreement in writing of EasternGraphics GmbH.

EasternGraphics GmbH accepts no liability for the completeness, freedom from errors, topicality or continuity of this work or for its suitability to the intended purposes of the user. All liability except in the case of malicious intent, gross negligence or harm to life and limb is excluded.

All names or descriptions contained in this work may be the trademarks of the relevant copyright owner and as such legally protected. The fact that such trademarks appear in this work entitles no-one to assume that they are for the free use of all and sundry.

1 Introduction

The OCD specification allows – with restrictions – to use a language for coding of relationship knowledge, which is also used in the variant configuration of the SAP ERP for the coding of relationship knowledge. This supports projects in which the OCD data is generated or exported from the SAP ERP of the manufacturer. However, according to the OCD specification (4.3), by default only those language constructs are supported that are syntactically and semantically identical to the corresponding constructs from the OCD language definitions (section 2). But the providers of OCD implementations are free to support further language constructs. This Application Note describes the SAP language constructs that are supported in the OCD implementation of EasternGraphics beyond the standard (section 3) and the behavior in case of unsupported language elements (section 4). Furthermore, other special features are described (section 5) and possible causes for a different behavior of SAP vs. pCon are examined (section 6).

The statements in this Application Note refer to the most recent versions of the main applications of EasternGraphics at the time the document was created:

- pCon.configurator 5.9
- pCon.planner 8.6u2
- pCon.basket 1.13.6
- EAIWS 4.10 (online Apps)

2 Standard language elements

According to the OCD specification (4.3), the following SAP language elements and constructs are supported by default:

- Relation type: *Precondition Selection condition, Procedure¹, Constraint*
- Condition part: `IF`
- Logical operators: `AND, OR, NOT`
- Built-in conditions: `IN, SPECIFIED`
- Comparison operators: `< or LT, <= or LE, = or EQ, <> or NE, >= or GE, > or GT`
- Arithmetical operators: `+, -, /, *`
- Arithmetical functions: `sqrt(), abs(), sign(), frac(), trunc(), ceil(), floor()`
- Price factors: `SET_PRICING_FACTOR()2`

¹ is supported only to the extent, as defined for OCD relation type *Action*

² slightly different signature

- Evaluation of variant tables: `TABLE()`

Note:

Contrary to the variant configuration in SAP ERP, no evaluation alternatives are used resp. supported in OCD when using a table call to derive values!

- When used in *constraints*, the evaluation in OCD is as if there is an evaluation alternative for every possible combination of the properties specified in *Inferences* that have an assigned value.
- When used in *actions*³, values are derived for those properties from the set of actual parameters that are qualified with `$self`. In contrast to the variant configuration in SAP ERP, this means that key properties *must not* be qualified with `$self`⁴!

3 Additionally supported language elements

Preliminary note:

Before using these language elements in a SAP-based project, it must be made clear to the respective customer (manufacturer) that the OCD data may not be fully functional in an application from another software vendor!

3.1 set_default

Old behavior⁵

Using option `@SetDefaultMode` in the control data table `epdfproductdb` (see Application Note AN-2006-01), two different ways of handling the SAP language construct

```
$set_default($self, <property>, <expression>)
```

can be specified:

`@AssignValue`

The value of the expression is assigned to the property if the property currently has no value. (This is the default behavior if the option is not given.)

`@SetDefault`

The value of the expression is marked as the default value for the property.

So there is no direct value assignment to the property, however, for restrictable properties the default value can be effective in the context of a subsequent automatic value selection (see option `@PreselectRestrictable`).

For optional properties this mode has no meaning. The behavior for these properties is always as in mode `@AssignValue`.

³ procedures in SAP

⁴ In SAP ERP, the object variable can be omitted for key properties (or object variable `$root` is used).

⁵ Up to the following versions: pCon.planner/configurator 5.8.10, pCon.planner 8.1u2, pCon.basket 1.12.1, pCon.xcad 2.25.1, EAIWS 4.6.3

New behavior

Due to unsatisfactory results of the previous handling in many situations, the treatment of the language element was revised. With this, option `@SetDefaultMode` mentioned above is no longer needed (ignored).

1.

With *configurable restrictable* properties, the new behavior generally corresponds to the old handling in mode `@SetDefault`.

2.

With all other properties, `set_default` statements conceptually are treated as reactions in OCD. Specifically, this means:

- a) During *article initialization*, `set_default` statements in actions bound to the article or to properties lead to an appropriate value assignment, but only in the first cycle of the evaluation of the relationship knowledge.
- b) After a *property change*, only `set_default` statements in actions bound to the changed property lead to a value assignment, again only in the first evaluation cycle⁶.
However, an assignment to the changed property itself is prevented.

This basic approach is supplemented by the following rules:

- An assignment to *non-configurable* properties only takes place if the affected property is not (yet) valuated at that time.
- `set_default` statements in actions bound to properties also are evaluated in later evaluation cycles if properties have become valid/visible (again) in the previous cycle due to preconditions. However, a (possible) assignment takes place only for these properties.
- If the evaluation of the relationship knowledge takes place during a re-configuration (with a given variant code), `set_default` assigns only if the target property is not valuated at this time.
- When evaluating relations in the context of an automatic value selection⁷ of a restrictable property *after* a property change, `set_default` in actions bound to properties is ignored.

Note on data creation:

In the original SAP data, the `set_default` statements are often found (collected) only in procedures bound to the article. However, according to regulation b) these do not become effective after a change of relevant master properties! Then, the corresponding code snippets would have to be created also as an action (or procedure in SAP) bound to the relevant leading properties⁸.

This approach (necessary for OCD) is a little unpleasant if a target property is dependent on more than one leading property. Then, the code must be included in an action bound to all respective properties.

⁶ This regulation ensures that a default value “overwritten” by the user can be withdrawn only after a change of the master property.

⁷ see option `@PreselectRestrictable` in control data table `epdfproductdb` (Application Note AN-2006-01)

⁸ Where appropriate, these code snippets then can/should be removed from the action/procedure bound to the article.

3.2 del_default

The SAP language construct

```
$del_default($self,<property>,<expression>)
```

in the old applications in mode *@SetDefault* (see section 3.1 above) as well as in the current applications with *configurable restrictable* properties removes the default-value indicator for the value of the expression.

In all other cases `del_default` statements have no effect.

3.3 type_of

The type to be queried in the SAP function

```
type_of (<object>,<type>)
```

can be specified in two different ways:

1. (<class type>)<class name>
2. (<object type>)< class type> (<object key>)

The first form is supported with restrictions:

- a) The object specification is ignored, i.e., the query is always related to the currently configured article.
- b) The class type is ignored or assumed to be (300), i.e., it is expected that a variant class is queried.

The 2nd form is *not* supported and leads to a syntax error (see section 4)!

3.4 Concatenation of strings

The operator `||` for the concatenation of strings is supported. In the OCD standard (language definition `OCD_2`), this corresponds to the operator `+`.

3.5 Alternative identifiers for comparison operators

In addition to the standard identifiers for comparison operators listed in section 2, the following identifiers also are supported: `><` (unequal), `=<` (less than or equal to), `=>` (greater than or equal to)

3.6 Alternative negation of IN comparisons

By default, a negation of an `IN` comparison, as with all logical expressions, is done by preceding it with the `NOT` operator. Alternatively, the form

```
<property name> NOT IN (<value set>)
```

also is supported.

4 Unsupported language elements

The following language elements do not cause a syntax error, but have no effect:

- `pfunction`

All other language elements not covered in section 3 lead to a syntax error!

In the case of a syntax error, the behavior during the evaluation of relationships depends on the type of the affected relation⁹:

- A syntax error in a *constraint* aborts the evaluation of the complete relationship knowledge.
- A syntax error in *price relationships* (actions¹⁰) leads to the abort of the price determination, i.e., a zero price will be delivered¹¹.
- A syntax error in all other relationship causes the relationship to be ignored, but the other relationships continue to be evaluated, i.e., the behavior is as if the relationship would not exist.

5 Other features

- Missing points at the end of a *constraint* section (*Objects, Condition, Restrictions, Inferences*) are tolerated.

6 Differing behavior of SAP vs. pCon

Occasionally there are support cases where different behavior is reported when evaluating the relationship knowledge (supposedly) based on the same data. This section describes and explains some possible causes. In principle, an absolutely identical behavior is rather to be regarded as the ideal case, since the applications SAP ERP and pCon differ in some aspects even if the standard language elements mentioned in section 2 are used exclusively, e.g.:

- Differing objectives in user guidance due to different user groups:
SAP focuses more on experienced office staff, while pCon focuses on dealers, field staff and end customers.
- The partly declarative relationship knowledge (e.g. constraints) requires a cyclic evaluation of the different types of relationships. Even if the (same) semantics of these relationship types result in similarities in the processing sequence, differences in the details of the implementation are practically unavoidable.
- The different application architectures are accompanied by, among other things, a different handling of persistence (storing and restoring) of configurations.

⁹ this applies to any syntax error, i.e., this is not SAP-specific

¹⁰ procedures in SAP

¹¹ which facilitates the detection of errors related to the "sensitive" price topic

6.1 Divergent data

In very few cases the data (properties, relationships, etc.) is actually the same. When converting the OCD data from the SAP ERP, usually modifications take place that affect the processing of the data. In addition, special handlings can be enabled by options in the control data table `epdfproductdb`.

Frequent modifications in the OCD data compared to the original SAP data are:

- **Scope *RG* instead of *R***

In SAP, there is no equivalent to scope `RG` in OCD. If the scope `RG` (instead of the actual equivalent `R`) is assigned to a non-configurable property when exporting from SAP, the way in which this property is initialized and persisted changes (see also the OCD specification).

- **Filtering out non-sales-relevant properties**

When exporting from SAP, the properties that are not relevant to sales¹² are often filtered out. This is the recommended procedure to reduce the amount of data and to improve the performance when processing the data in pCon applications. However, for smooth processing of the data in pCon, relationships that refer to the filtered out properties must be taken into account. In contrast to SAP, in pCon the use of an unknown property in a relationship (fortunately) does not lead to a syntax error, but (only) to the fact that the (partial) expression in which the property occurs is evaluated as undefined. Among other things, this can lead to an assignment not taking place or a precondition being regarded as "not violated".

Therefore, if properties are filtered, the relationships in the data model in SAP should generally be defined in such a way that the sales-relevant properties exported to OCD are not dependent on non-sales-relevant properties (which are filtered). If, however, there is a dependency on a property that is not relevant to sales, expressions that could be undefined due to the filtering of properties, if necessary, must be safeguarded using the `SPECIFIED` condition.

- **Options in der control data table `epdfproductdb`¹³**

Options in this table can be used to enable special handlings for which there is no equivalent in SAP and which, therefore, can lead to a different behavior.

Of particular importance is the option `@PreselectRestrictable` for the automatic value selection of restrictable properties. Without an automatic value selection, as in SAP, the (initial) status (value) of a restrictable properties is "not specified" ("???").

6.2 Backward dependencies

A recurring problem are so-called backward dependencies. These are characterized by the fact that the value set of a property is influenced by the value of another property, which is further down in the property list. In the configuration dialog of SAP, such dependencies are supported with the help of a "trick" (see below). With the standard processing in pCon, however, in extreme cases it can happen that the dependent property (up in the list) has only one valid value after the property further down in the list has been changed and, therefore, can no longer be changed.

¹² e.g. properties that are only required for the BOM resolution when processing a sales order

¹³ See Application Note AN-2006-01

A typical example are two properties, "X Finish Group" and the property "X Finish" below in the list, which are related via a constraint (and an associated variant table) in such a way that a mutual restriction is possible.

Then, the SAP configuration dialog allows for the following configuration scenario:

1. Initially all values are valid for "X Finish Group".
2. If value "Finish Group A" is selected, in the choice list for "X Finish" all matching values are offered under *Valid*, all others under *Invalid*¹⁴.
3. If the user now selects an *invalid* (!) value for "X Finish", the value of "X Finish Group" is changed according to the constraint (overwriting the value set in step 2).
4. If the choice list for "X Finish Group" now is opened again, in contrast to step 1, now only a single value (the result of the restriction in step 3) is shown as valid, all others as invalid.

If now an invalid value is selected, the scenario continues with step 2.

Such a configuration scenario may be reasonable for a SAP-trained back office employee, but not for the target user group of pCon applications. Apart from the fact that the display of currently invalid values in a choice list in pCon (currently) is not possible technically resp. not supported, the scenario described above is not acceptable from the user perspective of a pCon application for the following reasons:

- The user should see only the valid values.
- The display of all values (valid and invalid) often, especially when it comes to finish groups, leads to very long lists (like 1000 values!) through which the user would have to "dig".
- The logical configuration process is from top to bottom. If the change of a property causes the change of a property further up in the list, which has already been selected by the user (as in step 3), this is at least confusing, but in the worse case it can be overlooked by the user (which is fatal especially for price-relevant group properties).

Thus, for a smooth processing of the data in pCon, such backward dependencies should be avoided in the SAP data model. In the example above, in ideal circumstances (only) the property "X Finish" should be dependent on the property "X Finish Group".

If this is not possible and if the properties that imply a backward dependency are restrictable properties, option `@UnlockBackwardRestriction` in control data table `epdfproductdb` can be used¹⁵ to enable an exception handling in the OCD implementation: A temporary reset of the current value of the changed property and an additional evaluation cycle of the relationship knowledge unlocks properties higher up in the list¹⁶.

6.3 Hidden errors in the data

Occasionally it happens that errors in the OCD data are not detected because they remain without consequences in the current OCD implementation of pCon due to special circumstances, i.e. they do not lead to an undesired behavior. Due to a bug fix in the OCD implementation these errors then possibly can become noticeable through a changed and undesired behavior. Although this is not limited to data that is exported from SAP, it can also lead to the impression that pCon (now) behaves incorrectly compared to SAP.

Appendix A.1 describes an example for illustration purposes.

¹⁴ In the pCon applications, here "only" the valid values appear.

¹⁵ starting with the pCon releases from November 2021

¹⁶ For more details see Application Note AN-2006-01.

Appendix

A.1 Example for a hidden data error

As mentioned in section 6.3, a bug fix in the OCD implementation can reveal previously undiscovered data errors by a changed, undesired behavior. The example described below from the pCon releases of fall 2021 is intended to illustrate that many facts in the data and in the OCD implementation must coincide for such a scenario to occur.

Properties that have the scope `RG` and, therefore, are persistent must retain their value when the user changes another property. In earlier pCon versions, however, the value was (incorrectly) overwritten by the default value.

A support case revealed that the bug fix for this error can lead to an undesired behavior under the following circumstances:

- a) there is a price-relevant¹⁷ property AB with scope `RG`
- b) there is a leading configurable property AA, that has an action that makes an assignment to AB depending on its own value

Assuming

- c) that the default value of AB does not lead to setting of the variant condition

the problem can occur from the fall releases 2021

- d) if the action (b) assigns a value to AB that causes the variant condition to be set¹⁸.

It is/becomes a problem if setting the variant condition is not desired, i.e. leads to an invalid surcharge. Conversely, this means that the action assigns an incorrect value to AB.

In older pCon applications, the problem is not noticeable if, as in the concrete case, the variant condition depends on another, configurable property C: When the user assigns a value to property C, the value of AB set in the action (b) by mistake¹⁹ is replaced by its default value. Then assumption (c) takes effect.

In the specific case, the assignment of an incorrect value to AB in the end was the result of another rare error: Several values were marked as default values for property AA (where there can/may only be one). The OCD implementation of pCon then uses the last value from the list. However, this value did not match the default value of AB and then led to the assignment of another value to AB by action (b).

¹⁷ i.e., variant conditions can be set depending on the value of the characteristic

¹⁸ if the other conditions for setting this variant condition are also met

¹⁹ see the bug mentioned above

A.2 Document history

2022-02-01:

- More precise description of the usage of function `TABLE ()` (section 2).
- New section 6 (including an example in appendix 1) discussing possible causes for a differing behavior SAP vs. pCon.
- Start of the document history in this appendix.