

Version 1.2

Date: 2023-09-20

Author: Audrius Pundzevicius

ProjectWise attribute helpers

ProjectWise administrators have been using stored procedures and SQL statements in environment attribute configuration. This often resulted in a variety of similar solutions that achieve the same result.

ProjectWise attribute helpers were introduced with ProjectWise Design Integration 23 to provide frequently used tools for attribute configuration in a repeatable manner. They will save time when building new attribute configurations. This version of attribute helpers is focused on replacing all custom stored procedures with standard ones.

Attribute helpers, including stored procedures, are maintained by ProjectWise DI server development team. They are performant and safe to use. They will be supported in future releases of ProjectWise. The source code of attribute helpers will be available for everyone to see.

The list of attribute helpers will grow over time to support new requirements.

ISO19650 specific procedures

Procedures in this chapter are:

- Based on standard ISO19650 procedures proposed by user success team
- Suitable for in-place replacement, i.e. parameters and returned results are the same as for old procs. Only names will have to be changed.
 - Should be suitable for migrating older BS1192 stuff to, but may require more changes
- Extended to support known variations by adding extra default parameters.

Document attribute lookup lists

Performs lookup in lkp_pw_attributes table. The environment has to be provided by administrator. It is not created by PWDI automatically.

```
PROCEDURE dms_ISO19650_attr_lkp
```

```
    [@typecode=] 'typecode'  
    , [@filter=] 'filter'
```

```
PROCEDURE dms_ISO19650_attr_lkp_state
```

```
    [@typecode=] 'typecode'  
    [@cde_state] 'state name'  
    , [@filter=] 'filter'  
    , [@state=] 'state'
```

Parameters:

- `typecode` – document, model, or drawing that will be translated in the lookup table....
- `filter` – value type filter
- `cde_state` – CDE state or PW document state name (depends on picklist value configuration).

List of code and description, sorted by sort order and then by code (if sort order is not provided it will be sorted by code at least).

Lookup environment

Table name: `lkp_pw_attributes`

Columns:

Column name	Type	Notes
<code>pw_code</code>	Varchar	Returned
<code>pw_description</code>	Varchar	Returned
<code>pw_model</code>	Int	Type filter
<code>pw_drawing</code>	Int	Type filter
<code>pw_document</code>	Int	Type filter
<code>pw_filter</code>	Varchar	Text filter
<code>cde_state</code>	Varchar	Only required if function that takes state as filter is used
<code>pw_sortorder</code>	int	Sorting number

User attribute calculation

This stored procedure is used to generate and populate user pick list environment table. The SP takes User Name and Surname as a single parameter then the separator character and then the Initials or Title block switch. Returns a string of user initials or TB name. Input “John Smith” output initials “JS”, Output TB name “J.Smith”.

This function only performs calculations. It does not read any tables, so there are no database objects that have to exist for it to execute.

```
FUNCTION dms_ISO19650_user_name_calc
    (@fullName=] 'name surname'
    , [@separator=] \.]
    , [@returnType=] \T'
```

Parameters:

- `fullName` – User full name, ‘First Middle Surname’

- separator – separator character for name parts
- returnType – 'I' for initials; 'T' for Title Block Name

User actions

Very specific function to make WRE and ISO work....

Functions returns user information stored in lkp_user_attributes table. Returns either Initials or Title block name (parameter value T) if the parameter value is 1. If the parameter is not 1 – returns “---”.

Generally, it is used in the user creator/checker/approver user information field or in Revision history.

```
PROCEDURE dms_ISO19650_action_names
    [@userName=] 'user name'
    , [@trigger=] 1 | 0
    , [@returnType=] 'I' for initials; 'T' for Title Block Name
```

Parameters:

- userName – User name to return matched data from the lkp_user_attributes table.
- trigger – return data trigger value
 - 1 – returns user name or initials.
 - 0 – returns ‘---’
- returnType – Return type switch
 - T – returns user short name like J.Smith
 - I – returns user initials, like JS

See procedure source for details.

Drawn By	<input type="text" value="---"/>
Checked By	<input type="text" value="---"/>
Approved By	<input type="text" value="---"/>
Authorised By	<input type="text" value="---"/>
Rej/Rev By	<input type="text" value="P.Admin"/>

Archive Revision Notes

		^ v				
		^ v				
		^ v				
		^ v				
C01	xcvxc	^ v	PWA 2021-07-29	PWA 2021-07-29	PWA 2021-07-29	PWA 2021-07-29
P02.03	asdas	^ v	---	---	---	---
REV	REVISION NOTE		DRN	CHK	APP	AUT

Get Folder attribute

The stored procedure returns folder name or folder description of a “parent” folder in relation to the document in question. The stored procedure usually is used to determine discipline folder or similar.

```

PROCEDURE dms_ISO19650_root_folder_attribute
    [@folderNo=] 5000
    , [@level=] 2
    , [@property=] 'N' | 'D'
    
```

Parameters:

- folderNo - PW variable \$Vault\$ to determine current folder.
- level - 0 for Root level, 1 for first level, etc.
- property - argument to determine to return folder name or description
 - 'N' - Folder name
 - 'D' - Folder description

Get formatted date if user name, or '---'

Used to set current date or remove the date (change it to “---”) when a related attribute is updated. If the related attribute is “---” then returned value is “---”. If the value is other value, then the date is returned in a selected format. Selected format is passed as a parameter.

```
PROCEDURE dms_ISO19650_date
```

```
    [@flag=] '---'
```

```
    , [@format=] 23
```

Parameters:

- `Flag` – Related/triggering attribute
- `Format` – format of the date

Available formats (see T-SQL CONVERT):

FORMAT #	FORMAT	SAMPLE
1	mm/dd/yy	12/30/22
2	yy.mm.dd	22.12.30
3	dd/mm/yy	30/12/22
4	dd.mm.yy	30.12.22
5	dd-mm-yy	30-12-22
6	dd-Mon-yy	30 Dec 22
7	Mon dd, yy	Dec 30, 22
10	mm-dd-yy	12-30-22
11	yy/mm/dd	22/12/30
23	yyyy-mm-dd	2022-12-30
31	yyyy-dd-mm	2022-30-12
32	mm-dd-yyyy	12-30-2022
33	mm-yyyy-dd	12-2022-30
34	dd-mm-yyyy	30-12-2022
101	mm/dd/yyyy	12/30/2022
102	yyyy.mm.dd	2022.12.30
103	dd/mm/yyyy	30/12/2022
104	dd.mm.yyyy	30.12.2022

105	dd-mm-yyyy	30-12-2022
106	dd Mon yyyy	30 Dec 2022
107	Mon dd, yyyy	Dec 30, 2022
110	mm-dd-yyyy	12-30-2022
111	yyyy/mm/dd	2022/12/30

WRE

Currently `wreIncrementVersionNumber` will not be replaced with new version during database upgrade if an existing variant was found.

This is not really an attribute helper, but a WRE procedure for version numbering. This procedure will be initially delivered in its initial state. If extensions will be required, they can be made by external parametrization.

```
PROCEDURE wreIncrementVersionNumber
```

ProjectWise core product feature Workflow Rules Engine uses SQL Stored procedure to calculate next revision/version number. The stored procedure takes current revision/version and requested update as parameters and returns next revision/version.

The Stored Procedure is called to generate the next Version/Revision number. It is called by WRE Action "CHANGE_REV_NO" and parameter passed to identify how the revision would be increased.

This Stored Procedure is not called by ProjectWise environment attribute.

Supported revision formats:

- P01.01 - P – revision prefix letter (It can be any letter), 01 – numerical major revision, 01 numerical minor revision.
- A.01 – Alphanumerical major revision A, 01 numerical minor revision.
- AA.01 - Alphanumerical major revision AA, 01 numerical minor revision.
- 1.01 - 1 – numerical major revision, 01 numerical minor revision.
- P01 - P – revision prefix letter (It can be any letter), 01 – numerical major revision.

Supported parameters to pass in WRE sheet:

- INC_MAJ – Increases major revision by 1 and adds first minor version.
 - P01.01 -> P02.01
 - A.01 -> B.01
 - AA.01 -> AB.01
 - 1.01 -> 2.01
- INC_MIN- Increases minor version by 1.
 - P01.01 -> P01.02
 - A.01 -> A.02
 - AA.01 -> AA.02
 - 1.01 -> 1.02

- PUB – Increases major revision, does not return minor version
 - C01 -> C02
 - P01 -> P02