

Manual to use the Polarion TestDocument Workitems

Basic Information

For every object in a HVDC plant a specific testing document number is declared. It consists of the project id, kind of document, aspect key, objectflag, dccflag, language and the revision counter. These numbers used to be saved in an excel-sheet called MTP (Master Test Plan). This overview is now transferred to Polarion.

A dcc class describes a type of document.

A otc type describes a type of object.

If there is any lack of clarity check the info base for rds-ts:

<https://wse07.siemens.com/content/P0007504/rdsts/SitePages/Home.aspx>

If any questions or concerns contact: julian.leupold@siemens.com

Polarion

For any kind of “how to’s” about Polarion look up the wiki:

<https://wiki.siemens.com/pages/viewpage.action?spaceKey=en&title=Siemens+Polarion+Service+Portal>

First import of data

To the date this manual is written the first import will be executed by the person, who sets up the new project. First of all, it is mandatory to set the “project id” default value on the current project id. Navigate to the administration tab and choose “Work Items”. Then navigate to “Custom Fields” and click edit at the “all types” workitem type. There has to be a field with the id “projectid”. Set the default value in the second to last column. Now you can start importing data as microsoft excel workbook. After setting the mapping click on preview before starting the import. You will get a preview, after the scripts for this workitem ran. They will be described later in this document. If the data you try to import is not enough or not correct, the scripts will tell you by a pop-up why. Simply close it, change the data and try again. After importing your data should be visible and some fields are filled with some extra information. After the import of the data the end user can work with the data, if he has the required user rights.

Plugin

Up to date there is a currently not usable plugin which will multiply the workitems of aspect-key type product and location based on the information in the rds-ts info base. The plugin checks if the workitem is a c&p object and will create multiple new workitems based on the current one, which just differs in the dcc class value. Additionally, it sets the enums of optionality and the enum on which site (on/off-site) the protocol is needed. This plugin is fully functional, but just if it is started by the IDE it is written in (in this case: eclipse). So up to date its not possible to execute the plugin on a

polarion server. For the plugin to run there must be a workitem of custom type "properties". There must not be more than one!

The fields of this workitem have to be filled with data from the rds-ts base. It is shown in the next picture.

workitem-21 - properties DONT CHANGE!; there has to be only one of this kind

Created: 2020-06-08 14:24, Updated: 2020-06-18 13:39

Type: **Properties** ITS IMPORTANT TO CHECK THE CASE SENSITIVITY; OTCS HAVE TO BE UPPER CASE, THE REST LOWER

Severity: **Critical**

Author: **julianeupold**

Project: **test**

Categories: **all otc codes which are describing a c&p object**

Description

Custom Fields

Project IDs: **test**

OTC Codes for C&P objects: **AF300;BA300;BC300;BG300;BK300;BK310;BM300;BR300;BT300;BT311;BU300;BU310;BU311;BU312;BU313;BU314;BU315;BU316;BU317;BU321;BU331;BU332;BU341;BU351**

DCC list for non C&P Objects: **dc080;dc081;dc082;dc083;dc085;qc040;qc041;qc042;qc043;qc045;dc030;dc031;dc032;dc036;dc038;dc180;qc050;qc051;qc052;qc055;qc056;qc057;qc058;qc150**

DCC list for C&P Objects: **dc080;dc081;dc082;dc083;dc085;qc010;qc020;qc040;qc041;qc042;qc043;qc045;dc030;dc031;dc032;dc038;qc050;qc051;qc052;qc056;qc057;qc058**

dcc count for offsite cp: **12**

dcc count for offsite noncp: **10**

optional items list c&p: **dc081;qc041;dc085;qc045;dc030;qc050;qc056**

optional items list non c&p: **dc081;dc082;dc083;dc084;dc085;qc041;qc042;qc043;qc044;qc045;dc030;qc050;qc056;dc035;qc055**

amount of offsite protocols in the dcc list for cp/noncp; the first, in this case, 12 dcc codes are for offsite protocols

these dccc are optional in case of cp/noncp

As it is currently not working, it is not needed to create this type of workitem.

Workitemtype “testdocument”

19.6.2020

localhost:81/polarion/print/test/workitem-405?filter=false

workitem-405 - Single Line Diagram, variant for FAT (SubSIM),

Type: Test Document	Assignee(s):	VSC: xD
Severity: Normal	Resolution:	LCC: xD
Project: test	Custom ID:	Object Type: Non Physical Object
Project ID: P-012345	Internal remark:	DMS Identifier:
		TestingSite:

Description

Single Line Diagram, variant for FAT (SubSIM)

Documenttitle: Single Line Diagram, variant for FAT (SubSIM)

DCC description: Overview Diagram

Serialnumber:

Document Type: E	Station number:	Level E:
Aspect Key: C	Level B:	Level F:
OTC.Subtype: 30011.A002	Level C:	Level G:
	Level D:	Level H:

Code Letter: &E	DCC Class: FA010
OTC Code: 30011	
Subtype: A002	
OTC Code explicit for Location Aspect:	Level D, L:
Station number, L:	Level E, L:
Level B, L:	Level F, L:
Level C, L:	Level G, L:
	Level H, L:

Language: EN
Revision: 00
Page Counter:

Document Number: P-012345_EC_30011.A002&EFA010_EN_00

Document File Name: P-012345_EC_30011.A002_EFA010_EN_00

Issued for information:
Issued for approval:
Issued for design:
Issued for manufacture:
Issued for construction:
as built:

Contact Person at Siemens:	Responsible Department:
Manufacturer:	SubPM:

Contact Person at Manufacturer:

Author: JulianLeupold	Reviewer:	Approver:
Work Progress: 60%	Review Status:	Approval Status:
Due Date:	Due Date:	Due Date:

Created: 2020-08-18 13:42 , Updated: 2020-08-17 13:54

This is the layout for the custom workitemtype “testdocument”.

Basically, all of the data will be created by importing base data from the sources given by the creator of the project. Currently all objects with aspect key type “C” will be imported from the MTP-template and all of type “P” and “L” from the PAM-database.

Scripts

There are running scripts, written for the FMC-WorkItemSave Extension:

<https://extensions.polarion.com/extensions/134-fmc-work-item-save>

This type of workitem uses 8 scripts, which are running one after another to ensure a correct workflow. They are running in the order they are displayed here. The scripts are executed every time you press the save button in polarion.

`not_save_wo_requiredfield.js`

the script checks if all the required fields are filled with data; if not it will return the missing fields; it will not even let the other scripts run.

`docfile_num_builder.js`

the script builds the final document file- and document number-name from the given data.

`otc_subtype_builder.js`

the script builds one string out of two; the otc code and the optional otc-subtype are imported separated; for further use they have to be together.

`doc_title_from_otc.js`

the script fills the field “document title” based on the enum description of the field “OTC.Subtype”. for every otc.subtype element a description is set on the value in the corresponding otc list.

`title_set_name.js`

the script sets the title of the workitem to the otc description plus the whole documentnumber.

`dcc_description_setter.js`

the script fills the field “dcc description” based on the enum description of the field “DCC Cass”. for every dcc class element the description is set on the value in the corresponding dcc list.

`non_physical.js`

the script sets the enum which describes the physicality of an object, based on the otc code.

`issued.js`

The script set the values for the issued for section enums, based on logic gained from Paul-Heinz Esters. These need a dcc class value to work, so the fields will be still empty if no dcc classes are set.