MagicDraw Training for Developers, Integrators and Testers in EZ1

* Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Version | Comment |
| Miguel Neves | 30/08/2022 | 0.1 | Creation of Draft version which includes only Versioning, Baselining |
| Miguel Neves | 30/09/2022 | 0.2 | Addition of information regarding how to work with MD, like branches; how to find elements; how to check attributes and Change Management |
| Miguel Neves | 10/10/2022 | 0.3 | Added Revision History table and Summary to document |
|  |  |  |  |

Summary

Document containing necessary steps in order for developers, testers and integrators to successfully navigate the tool MagicDraw and operate within their needs for project EZ1.

Training content:

* How to work with projects and branches in MagicDraw
* How to find elements in MagicDraw(Requirements and diagrams)
* How to check requirement attributes
* Versioning
* Baselining
* Change Management
* How to work with projects and branches in MagicDraw
  + MagicDraw uses TWC, which is a cloud based solution to store and manage the projects, branches and elements used in EZ1.

In order to open a project and subsequent branch please do the following steps:

* + - Click Collaborate, and then Open Server Project

Graphical user interface

Description automatically generated with medium confidence

* + - After selecting your project in the below menu, click the 3 dots button in order to open the branch window where one can select which branch to use.

Graphical user interface, application

Description automatically generatedAvailable project list on TWC

Graphical user interface, application

Description automatically generated

Branch selection window

* + - After clicking OK, MagicDraw will open selected project in chosen branch.

- How to find elements in MagicDraw(Requirements and diagrams)

* + If you wish to find a certain element or requirement inside a project, this can be done very quickly with the Quick Find functionality inside MagicDraw.

All elements are located in the tree structure that can be found in the Containment Window, see picture below.

A picture containing graphical user interface

Description automatically generated

Containment window in Body project

WhiBAS requirements are to be found in Transversal project and SWComponent requirements are to be found inside each specific domain project.

So, for checking both sets a change between projects will have to be performed in MD via the Collaborate menu, followed by Open Project Server option, like explained already in this document.

To use Quick Find, first select the package you want to search for the requirement/element and then click Ctrl + Alt + F.

In the following example Headlamp Management was highlighted, but as long as requirement/element is inside the folder selected, Quick Find can search through all folders inside the selected one. In case you have no idea where to start the search, use the top folder for the domain, in this case Body SW Archi, this way you are guaranteed to find what you are looking for.

Graphical user interface, application

Description automatically generated

Quick Find in MagicDraw

* + After the Quick Find textbox appears, one can select the text, requirement name or id. In the following example ExternalHighbeam was chosen to locate a req in that function.

Graphical user interface, application

Description automatically generated

Searching for ExternalHighBeam reqs using Quick Find

* + After selecting the wanted requirement from the possible values displayed from Quick Find list, we are directed to the Containment window on the left where the searched req is highlighted and we can also see where it is located in the tree structure of the project.

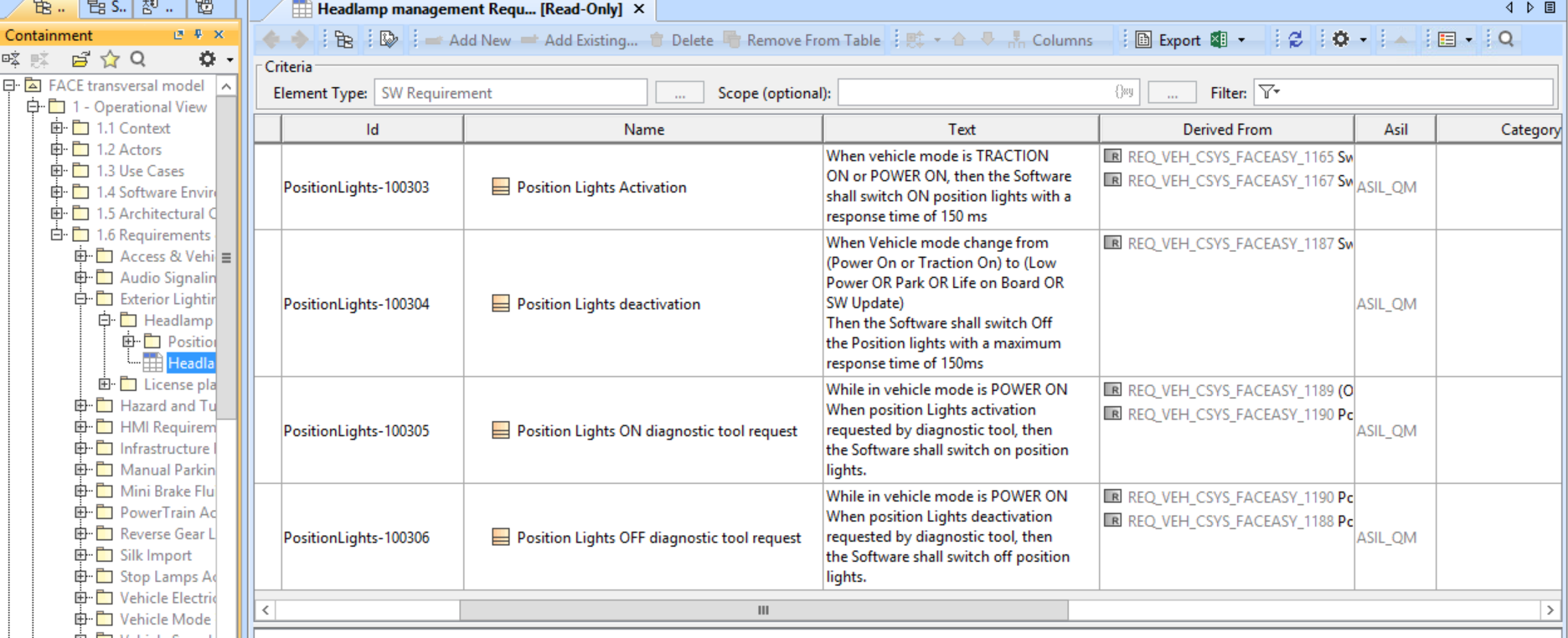
Graphical user interface, application

Description automatically generated

Location of searched req in Containment window

* How to check requirement attributes
  + The best way to check for the attributes value inside requirements is to have a look at the generic table created for that feature’s WhiBAS or SWComponent reqs.

Please find an example below:



Generic Table for Headlamp Management WhiBAS Reqs

* + The most relevant attributes to check are Id(should be unique throughout the database), Text(the req itself that must be implemented), ASIL(in case of Safety req), Lifecycle Status(should be Submitted for reqs under review), Regulation(related to possible legal standard that could be applicable), Version(has the req version, which should be increased with every change).

Several other attributes may be used but these are not mandatory, so their use could be at author’s discretion, like Requirement Type, Category, Validation Method or Testing Coverage.

This can be checked against the AEK document pdf provided by Renault during feature reviews.

* Versioning
  + Input to Version attribute seems to be solely manual until further information surfaces. After checking previous versions of different reqs, version attribute didn’t change even when req did change, see example below.

Only mention to possible Versioning strategy in EZ1 was done in Confluence and mentions that after a requirement change, version attribute should be incremented by 1.

Ex: REQ\_SRS\_ELECHEATWINDSCREEN-01, REQ\_SRS\_FRONTFAN-08.

Graphical user interface, text, application, email

Description automatically generated REQ\_SRS\_ELECHEATWINDSCREEN-01

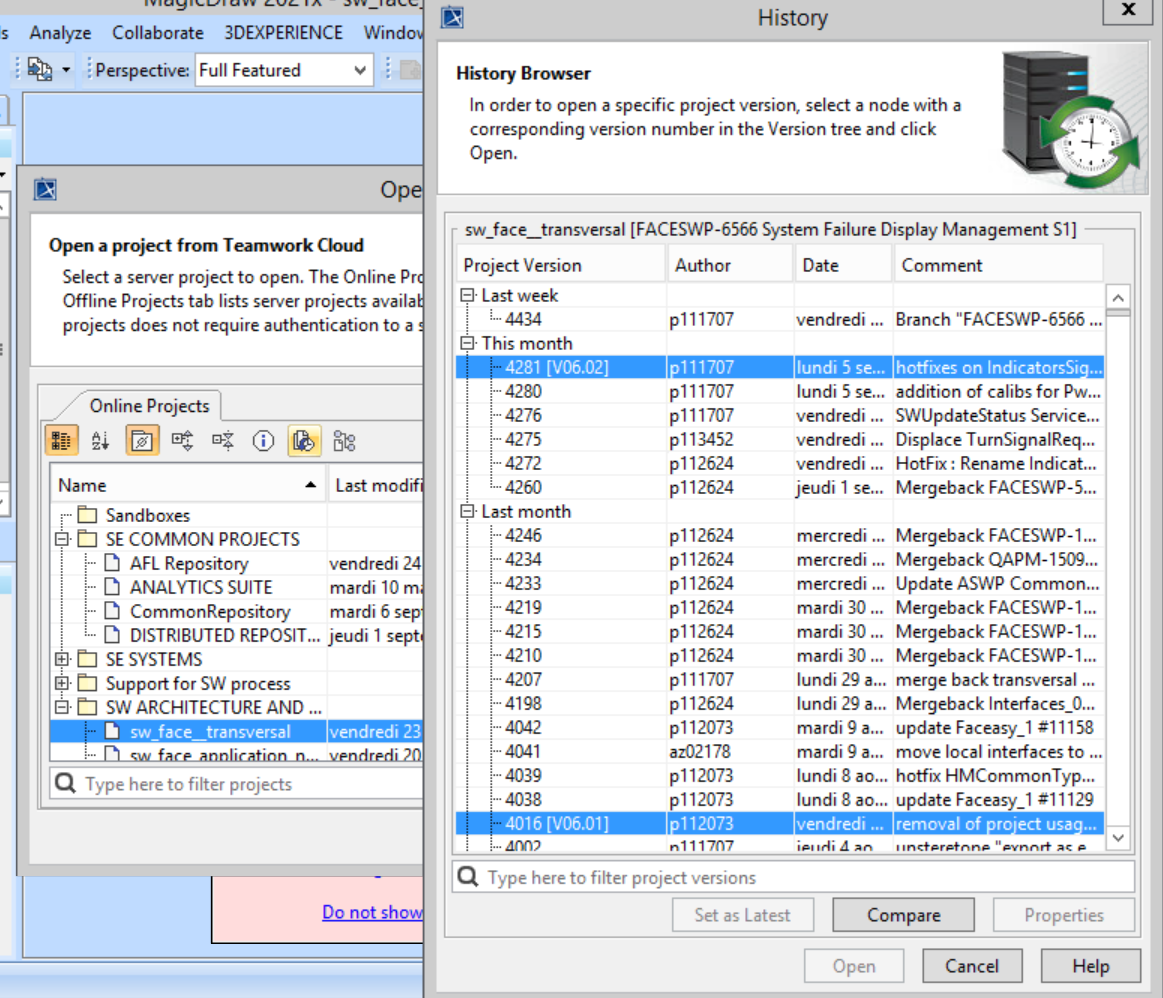
* Baselining
  + Baselines per usual RE definition don't exist in MagicDraw but commits of released project versions do exist and can be accessed via Project History button in Open Server Project after selecting the branch one wants.
    - After selecting your project, click Project History button in Open Server Project window, see picture Available project list on TWC.
    - Navigate History Browser for the baseline you want. The commits that refer to a new baseline will be tagged as such like the V06.01 version/baseline you find in the picture below.

Table

Description automatically generated

List of commits of project sw\_face\_transversal, ez1 branch

* + In case you wish to compare 2 different commits, they must be from the same project, select in History Browser the 2 commits and click the Compare button. This will initiate the comparison tool which will analyse the 2 commits provided for the differences between them. So far this hasn’t worked well and during the analysis there are always some diagrams that are either not compatible or some imports to MD installation that are missing.



Comparison between 2 commits of project sw\_face\_transversal

* Change Management on Requirements
  + To see the changes occurred inside a package of elements, one can right-click and select Content History and wait for the list of changes to load. Beware, as it takes quite a while and only 1 change is loaded.

Graphical user interface, text, application

Description automatically generated

Content History

* + For more, you need to click Show more results in the bottom left corner.

Graphical user interface, application

Description automatically generated

How to display multiple results for Content History

* + After the results appear on the list on the left, select the entry you want to check with double click and the list of changes that occurred in the package selected will appear in a tree format, blue for changed, green for new and red for deleted elements on the right side.

Graphical user interface, text, application, email

Description automatically generated

* + To open a quick diff to see the changes in attributes you can just double-click said element or right-click and select Quick Diff. This will open the Quick Diff window where the changes to the element will be visible and can be compared between current commit and previous commit.

Graphical user interface, text, application

Description automatically generated

Quick Diff window for requirement REQ\_SRS\_ELECHEATWINDSCREEN-01

* + To create a Full Diff, with more information regarding the changes, just select the Full Diff option when right-clicking on the req. This will take a couple of minutes and several popups might surface depending on installed plugins or validation rules applied. The result is the following screen for the same req as above.

Graphical user interface, text, application, email

Description automatically generated

Full Diff window for requirement REQ\_SRS\_ELECHEATWINDSCREEN-01