EDMtruePLM ™

True Product Lifecycle Management

User Manual
for Version 3.4

Role	Name	Date
Written by	Mona Chaure & Alexey Zorin	16.12.2022
Verified by	Jochen Haenisch & Pål Huse	31.12.2022
Approved by	Kjell Bengtsson	31.12.2022

Document revision history

Rev.	Description	Status	Date	Authors
1.3	Updated	Final	17.01.2022	Jotne
1.4	Updated	Final	18.04.2022	Jotne
1.5	Updated	Final	30.06.2022	Jotne
1.6	Updated	Final	16.12.2022	Jotne

Proprietary notice:

Information in this publication is subject to change without notice. The software described in this publication is furnished under a license agreement or non-disclosure agreement. The software may be used only in accordance with the terms of those agreements. No part of this publication may be distributed to third parties, copied, duplicated, in any form, whole or in part, without the prior written permission of Jotne EPM Technology AS, Grenseveien 107, N-0607 Oslo, Norway.

Express Data Manager™, *EDMserver™*, *EDMapplicationServer™* and *EDMtruePLM ™* are registered trademarks of Jotne EPM Technology in Norway. Please contact helpdesk@jotne.com for details.

Note to the reader of this document

The reader of this document should be accustomed to engineering projects and to the use of web-browsers for engineering applications. A good understanding of the principles of product data management (PDM), product lifecycle management (PLM) and document management is recommended. A working knowledge of CAD, CAE (FEA or CFD), Logistics, and ERP solutions is an advantage, but not a necessity to benefit from using EDMtruePLM with the help of this user manual. This document uses artificial test data that represents a bike product model in most cases with some exceptions.

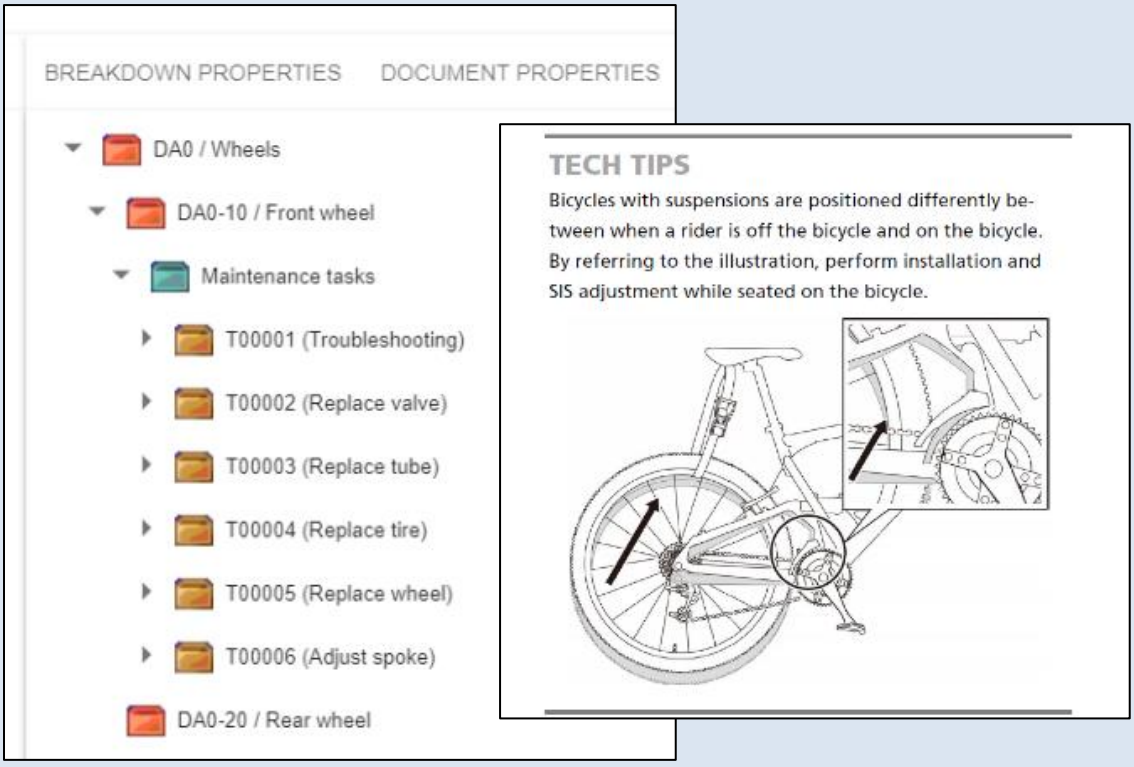


Figure 1. A bike example is used throughout this User Manual

Table of Contents

1 Introduction 12
1.1 Purpose of this document 12
1.2 Scope of EDMtruePLM 12
1.3 Contents of this document 12
2 Login to the EDMtruePLM 13
2.1 Login with 2FA 14
2.2 SSO (Single Sign On) 14
2.2.1 Using your Google Account 15
2.2.2 Using Keycloak 15
3 EDMtruePLM Product Management GUI 17
3.1 Overall Client View 17
3.1.1 Closing tabs 19
3.2 Administration menu 19
3.2.1 Projects 20
3.2.2 Breakdown templates 21
3.2.3 Project templates 21
3.2.4 Administration 21
3.2.4.1 Create project folder 22
3.2.4.2 Create project 22
3.2.4.3 Create project from PDM file 23
3.2.4.4 Create project from text 24
3.2.4.5 Create project from baseline 25
3.2.4.6 Create project from AP242 Domain 26
3.2.4.7 Create project from ASD DEX1 27
3.2.4.8 Change user password 28
3.2.4.9 API Tokens 28
3.2.5 Download file by Hash 28
3.2.6 About (for a project user) 28
3.2.7 Log Out 29
3.3 Breadcrumb menu 29
3.4 Breakdown Structure 30
3.4.1 Notification of changes by other clients 31

3.4.2 Breakdown element functionality 32
3.4.2.1 Breakdown versioning 32
3.4.2.2 Child node functionalities 33
3.4.2.2.1 Rename element 34
3.4.2.2.2 Rename by attribute 34
3.4.2.2.3 Edit element 35
3.4.2.2.4 Delete element 35
3.4.2.2.5 Cut element 36
3.4.2.2.6 Copy element 36
3.4.2.3 Parent node functionalities 36
3.4.2.3.1 Create element 38
3.4.2.3.2 Paste element 40
3.4.2.3.3 Paste element as relating 40
3.4.2.3.4 Paste element as related 41
3.4.2.3.5 Add Organisation 42
3.4.2.3.6 Put Product 43
3.4.2.3.7 Generate document 44
3.4.2.3.8 Add data file 45
3.4.2.3.9 Paste data file 46
3.4.2.3.10 Show element in context 46
3.4.2.3.11 Show Structure Tree 47
3.4.2.3.12 View Change log 48
3.4.2.3.13 Create baseline 48
3.4.2.3.14 Subdomain info 49
3.4.2.3.15 Subdomain management 49
3.4.2.3.16 Import from STEP PDM file 54
3.4.2.3.17 Import from AP242 Domain model 55
3.4.2.3.18 Import from Req-IF file 55
3.4.2.3.19 Export to zipped folder 56
3.4.3 Data file functionalities 56
3.4.3.1 Checkout file 57
3.4.3.2 Undo checkout 57
3.4.3.3 Checkin file 57

3.4.3.4 Set file read only / Clear file read only 58
3.4.3.5 Download 58
3.4.3.6 Open 59
3.4.3.7 VCollab Visualization 59
3.4.3.8 View history 60
3.4.3.9 Delete 60
3.4.3.10 Copy file 60
3.4.3.11 Cut file 61
3.4.3.12 Paste file as affected 61
3.4.3.13 Paste file as affecting 61
3.4.3.14 Sticky notes 61
3.4.3.15 Dependencies 62
3.4.4 Icons used for breakdown elements and data files 64
3.4.5 Navigation 65
3.5 Property window 65
3.5.1 System properties of breakdown elements 67
3.5.2 System properties of data files 68
3.5.3 Product properties 69
3.6 Project menu 71
3.6.1 Blockchain 71
3.6.2 Make comparison 71
3.6.3 Baselines 73
3.6.4 Search 73
3.6.4.1 Breakdown element 74
3.6.4.2 Data files 76
3.6.4.3 Search result tabs 80
3.6.4.4 Sorting & exporting search results 81
3.6.5 Catalogues 81
3.6.5.1 Notification 81
3.6.5.1.1 For a Task or info type 82
3.6.5.1.2 For the change in the Breakdown element properties 84
3.6.5.2 Product 86
3.6.5.3 Organisation 87

3.6.6 Reference data definitions ... 88
3.6.6.1 Opening the RDL page ... 88
3.6.6.2 Reference data GUI ... 88
3.6.6.3 Adding reference data ... 90
3.6.6.4 Deleting reference data ... 90
3.6.6.5 Adding document property ... 91
3.6.6.6 Adding breakdown element type ... 92
3.6.6.7 Adding breakdown element property ... 93
3.6.6.8 Adding derived property ... 96
3.6.6.9 Adding an aggregate structure ... 97
3.6.6.10 Adding a filter key to aggregate data ... 98
3.6.6.11 How to upload existing sensor data in TruePLM ... 99
3.6.6.12 Reference Data Uniqueness ... 100
3.6.6.13 User Defined and System Properties sequencing (Order Elements by) ... 100
3.6.7 Language selector ... 101
3.7 Project root context menus ... 102
3.7.1 Project root menu ... 102
3.7.1.1 Rename element ... 102
3.7.1.2 Rename by attribute ... 103
3.7.1.3 Edit element ... 103
3.7.1.4 Versions ... 103
3.7.1.5 Export to STEP PDM-file ... 103
3.7.1.6 Export structure to text file ... 104
3.7.1.7 Export project to DEX ... 104
3.7.1.8 Export to AP242 domain ... 105
3.7.1.9 Import CSV file ... 105
3.7.1.10 Current project phase ... 107
3.7.2 Project root selected menu ... 107
3.7.2.1 Create element ... 108
3.7.2.2 Paste element ... 108
3.7.2.3 Show element in context ... 108
3.7.2.4 Show structure tree ... 108
3.7.2.5 Send message ... 108

3.7.2.6 View change log 108
3.7.2.7 Create baseline 109
4 EDMtruePLM Administration GUI 110
4.1 Projects Administration 112
4.1.1 Add user 112
4.1.2 Edit user 113
4.1.3 Remove user 113
4.1.4 Populate new project by import 114
4.1.5 Download STEP pack 115
4.1.6 Upload STEP pack 116
4.2 Breakdown template list 116
4.3 Project template list 117
4.4 User list 118
4.4.1 Create user 119
4.4.2 Deleting user 119
4.4.3 Editing user 119
4.4.4 Changing user password 120
4.4.5 Two Factor Authentication (2FA) 120
4.5 Project list 120
4.5.1 Add project 121
4.5.2 Deactivate project 122
4.5.3 Reactivate project 122
4.5.4 Delete project 123
4.6 About (for an admin) 123
4.7 Log out 123
5 Annex Guide – Text file structure 124
6 Reference Material 127

Figures

Figure 1. A bike example is used throughout this User Manual 2
Figure 2. Registration, login, and language selection 13
Figure 3. Login with 2FA 14
Figure 4. API Tokens 15
Figure 5. SSO Window 15
Figure 6. Keycloak user interface 16
Figure 7. Navigation Page 17
Figure 8. Properties assignments as per STEP standards 18
Figure 9. Closing tabs 19
Figure 10. Administration menu – ordinary user 20
Figure 11. Active Project 20
Figure 12. Breakdown template 21
Figure 13. Project template 21
Figure 14. Administration tab 22
Figure 15. Example of a project folder 22
Figure 16. Create new project with active user login 23
Figure 17. Project from PDM file 24
Figure 18. Project from text file 25
Figure 19. Project from zipped baseline 26
Figure 20. Project from AP242 Domain 27
Figure 21. Project from DEX 27
Figure 22. Change user password 28
Figure 23. Download file by Hash 28
Figure 24. About current user and project details 29
Figure 25. Breadcrumb menu 30
Figure 26. Breakdown Structure 30
Figure 27. Breakdown Element with details 31
Figure 28. Change notification 32
Figure 29. Breakdown structure versions 33
Figure 30. Child node context menu 34
Figure 31. Rename element 34
Figure 32. Rename by attribute 35
Figure 33. Edit element 35
Figure 34. Delete element 35
Figure 35. Two copies in the same parent 36
Figure 36. Parent node context 37
Figure 37. Parent node context menu 37
Figure 38. Create element 38
Figure 39. Requirement element 39
Figure 40. Requirement node menu 39
Figure 41. Requirement types 40
Figure 42. Establishing a backward/related relationship 41
Figure 43. Establishing a direct/relating relationship 42
Figure 44. Add organization to a breakdown element 43

Figure 45. Organisation system property with details 43
Figure 46. Product details 44
Figure 47. Add data file 45
Figure 48. Show element in context 47
Figure 49. Show Structure Tree 48
Figure 50. Breakdown element change log 48
Figure 51. Create baseline 49
Figure 52. Subdomain info 49
Figure 53. Subdomain management 50
Figure 54. Subdomain options 50
Figure 55. Create subdomain 51
Figure 56. Delete subdomain 51
Figure 57. Subdomain roles 52
Figure 58. Subdomain members 53
Figure 59. Add subdomain members 53
Figure 60. Subdomain members details 54
Figure 61. Import from STEP PDM 54
Figure 62. Import from AP242 Domain model 55
Figure 63. Import from Req-IF 55
Figure 64. Data file menu 56
Figure 65. Checkout of a file 57
Figure 66. Checkin of a file 58
Figure 67. Read only file 58
Figure 68. Convert to WCAX file 59
Figure 69. VCollab Web viewer 59
Figure 70. View History 60
Figure 71 Delete Document 60
Figure 72. Sticky Notes 61
Figure 73. Create new note 62
Figure 74. Sticky note form 62
Figure 75. Dependencies 63
Figure 76. Red Flag 63
Figure 77. Removed red flags 64
Figure 78. Dependencies tab menu 64
Figure 79. Property window 66
Figure 80. Project Menu 71
Figure 81. Blockchain 71
Figure 82. Version and baseline comparison tab 72
Figure 83. Version comparison window 72
Figure 84. Baseline tab 73
Figure 85. Search tab 74
Figure 86. Quick Node Search 74
Figure 87. Node search 75
Figure 88. Quick document search 77
Figure 89. Document search and by content 78
Figure 90. Document search result 80

Figure 91. Node search result 80
Figure 92. Sorting & Exporting search result 81
Figure 93. Catalogues 81
Figure 94. Send personal message 82
Figure 95. Message Form Info Type 82
Figure 96. User Notification 83
Figure 97. Message Form Task type 83
Figure 98. Notification 84
Figure 99. Notification Menu 84
Figure 100. Reference Data for Notification 85
Figure 101. Product menu 87
Figure 102- Organisation menu 88
Figure 103. Reference data 89
Figure 104. Add reference data 90
Figure 105. Delete reference data 91
Figure 106. Add document property 92
Figure 107. Add breakdown element subtype 93
Figure 108. Add breakdown element type 93
Figure 109. Add property for specific breakdown element type 93
Figure 110. Breakdown property input form 94
Figure 111. Create new breakdown element property 94
Figure 112. Create derived property 96
Figure 113. Derived property example 97
Figure 114. Aggregate structure 97
Figure 115. Create new aggregate property type 98
Figure 116. Aggregate type 99
Figure 117. Reference Data Uniqueness 100
Figure 118. Drag & Drop property order 101
Figure 119. GUI language selection 101
Figure 120. Initial project root context menu 102
Figure 121. Export to DEX input form 105
Figure 122. Example of an import file in csv-format 106
Figure 123. Bike structure after csv import 107
Figure 124. Project root context menu after selection 108
Figure 125. User Interface 111
Figure 126. Project Administration 112
Figure 127. Add User 113
Figure 128. Edit User 113
Figure 129. Delete User 114
Figure 130. Import data 114
Figure 131. Breakdown Template 116
Figure 132. Create breakdown template type 117
Figure 133. Project Template 117
Figure 134. Create project template type 118
Figure 135. User list 118
Figure 136. Create User 119

Figure 137. Delete user 119
Figure 138. Edit user 120
Figure 139. Change Password 120
Figure 140. Project list 121
Figure 141. Create project 121
Figure 142. Deactivate project 122
Figure 143. Activate project 122
Figure 144. Delete project 123
Figure 145. About (admin) 123

Tables

Table 1. Icons for breakdown elements and data files 64
Table 2. Breakdown element system properties 67
Table 3. Data file system properties 68
Table 4. Product properties 69
Table 5. Document property types 92
Table 6. Breakdown element property types 94
Table 7. Aggregate property types 98
Table 8. List of User Types 110

1 Introduction

1.1 Purpose of this document

The purpose of this document is to describe what you can do with *EDMtruePLM*™ and how to do it.

This user manual covers the needs of the general end-user and of administrators of the application and of projects.

1.2 Scope of EDMtruePLM

The *EDMtruePLM* solution is a product model server for integrating, storing, and accessing data for types of products and for individual products over their lifetime in a standards compliant fashion. The solution scope covers product structure data and product properties, streamed sensor data, and documents / data files and their properties, such as, CAD files, manuals and structured documents. *EDMtruePLM* is intended for product lifecycle management. Data and documents may be collected and categorized from early design to operation and disposal. Product data may be collected into technical data packages, which can be exported and imported in different formats. Technical data package functionality eases communications with customers and within the supply chain. All data except for document files are stored in the format of ISO 10303-239, Product Lifecycle Support (PLCS).

EDMtruePLM has been designed for the concurrent engineering requirements of the space sector. However, care has been taken to enable adaptation to other engineering domains. This has been achieved by so-called 'reference data'. For each project, a project manager may define specific names for properties, roles, lifecycle phases, types of products, breakdown elements and documents etc. Thus, project managers may define properties to attach additional information to products, breakdown elements (also called nodes) and documents.

EDMtruePLM supports integration with IoT frameworks to collect live data from sensor devices.

Note: EDMtruePLM uses a web-client and is accessible through a web-browser; suggested browser is Google Chrome.

1.3 Contents of this document

After the details in chapter 2 of logging into an EDMtruePLM server, Chapter 3 addresses the needs of an ordinary user who wants to manage product and project data. Chapter 4 lists the functions needed to administrate the application. Chapter 5 describes a special import/export file format.

Installation Guide and Release Notes are available in separate documents.

2 Login to the EDMtruePLM

The EDMtruePLM client is started through a web-browser, such as, Google Chrome (recommended) or Firefox.

The user selects an EDMtruePLM server by pasting the server specific URL into the URL-field of the web-browser. The URL-string is defined by the team who installed the server. If you installed the server on the same machine that you are starting the client on, you use the following URL:

http://localhost:8080/EDMtruePLM.

With a valid URL you will see a login pop-up similar to the one on the left side of Figure 2

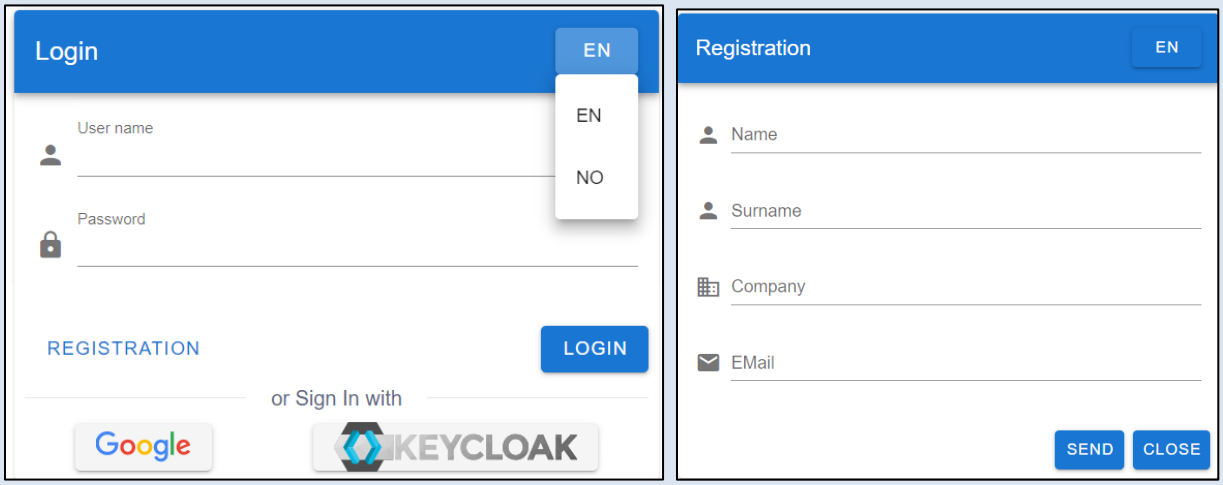


Figure 2. Registration, login, and language selection

The login page allows you to login to EDMtruePLM using given credentials. If you do not have credentials, use the “Registration” link to request access. Fill in the registration form as shown above with required details and push “Send”. The form is then forwarded to the EDMtruePLM admin team who will share the credentials to the email of the requesting user.

For login you need a username and a password. Such user details are provided to you by the EDMtruePLM system administrator; see 4.1.

You can change the language of the client by the Language selector on the upper right of the left-hand screenshot in Figure 2. You may select between English and Norwegian.

The following options to login to EDMtruePLM are described in the following sub-sections:

1. Two-factor authentication (2FA)
2. Single sign on (SSO) using Google or Keycloak platforms

2.1 Login with 2FA

Two-factor authentication (2FA) requires you to login via two devices, which gives additional security. 2FA is activated by the system administrator per user. You then need to use both your ordinary user credentials (username and password) plus a one-time password (OTP) generated by the authenticator system like Microsoft Authenticator or Google Authenticator.

Once 2FA authentication is enabled, a QR code is generated for use with the Authenticator application. You need to install such an Authenticator tool and to create a new account by means of the QR code. Whenever you provide username and password for login, the authenticator tool will generate a six-digit one-time password, which needs to be entered into the web-browser window to get access to the requested EDMtruePLM server. See Figure 3, below, for screenshots.

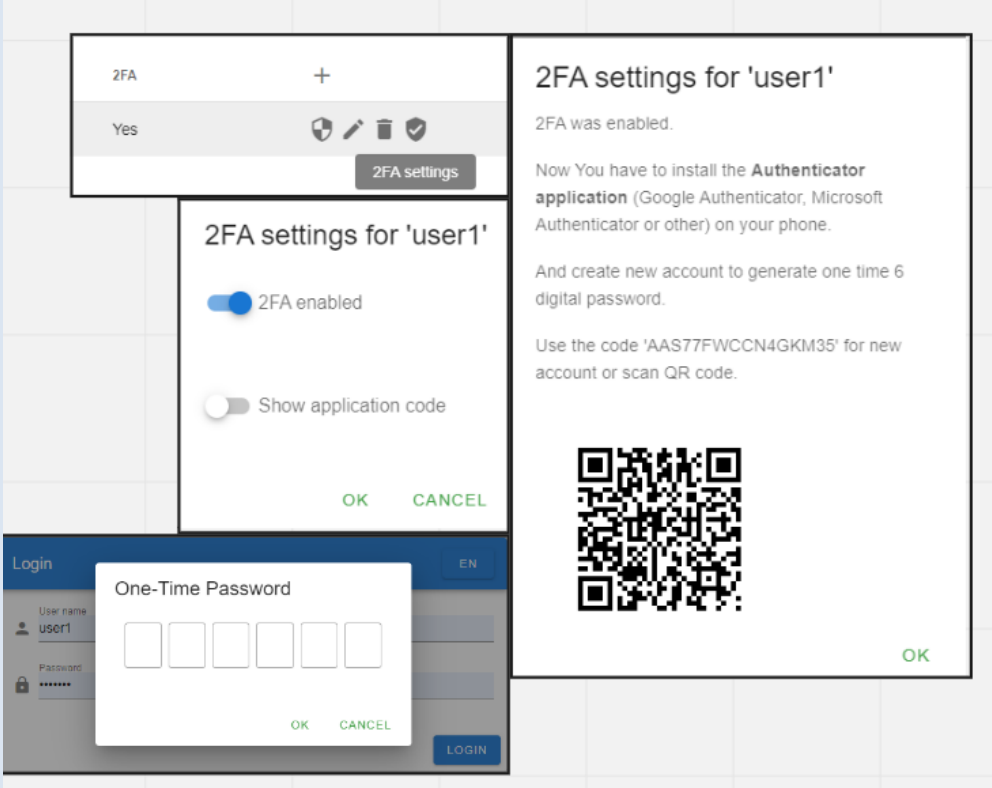


Figure 3. Login with 2FA

2.2 SSO (Single Sign On)

With SSO, each account login can be linked to either a Google user or a KeyCloak user. Both will create an API Tokens log under the administration tab as shown in Figure 4.
To disconnect an existing EDMtruePLM SSO user account, delete the API token and logout from EDMtruePLM. To connect again, repeat the registration process that is described below.

Figure 4. API Tokens

2.2.1 Using your Google Account

Access via a Google account is enabled by the system administrator; see the EDMtruePLM Installation guide for details. Once this is enabled, you will see a Google button on the Login page, as shown in Figure 2. Click on the Google button and login with your personal Google credentials. The SSO login page will appear, as shown below. Enter your EDMtruePLM credentials that were shared with you by your system administrator.

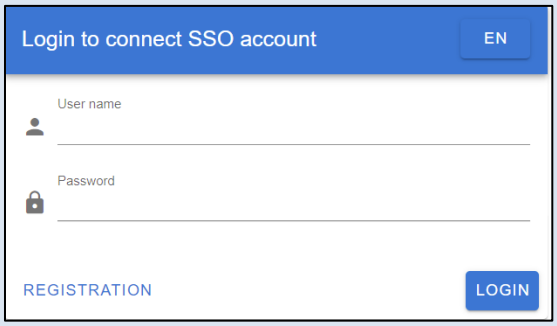


Figure 5. SSO Window

2.2.2 Using Keycloak

Also access via a Keycloak account is enabled by the system administrator; see the EDMtruePLM Installation guide for details. Like the Google button, a Keycloak button will appear on the user login page. Click on the Keycloak button and login with your personal Keycloak credentials; see Figure 6, below. Then, enter your EDMtruePLM credentials that were shared with you by your system administrator.

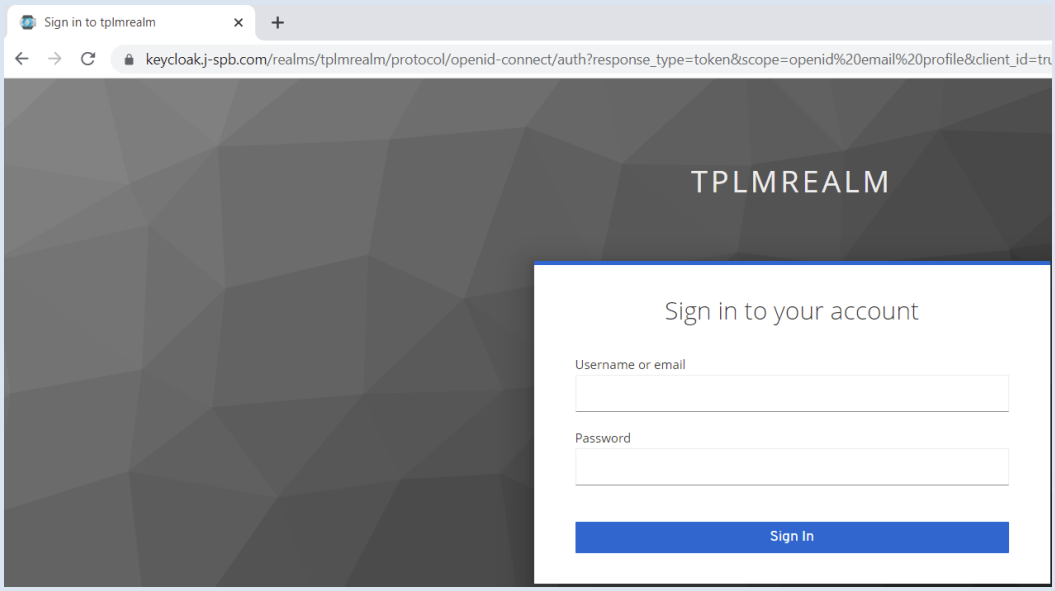


Figure 6. Keycloak user interface

3 EDMtruePLM Product Management GUI

The Product Management GUI is the client view for the ordinary user, that is, for users without system administrator access rights.

3.1 Overall Client View

Below is the overall view of the web client with its different sections after successful login and after the user has opened a project; see 3.2 for opening existing projects and for creating local projects.

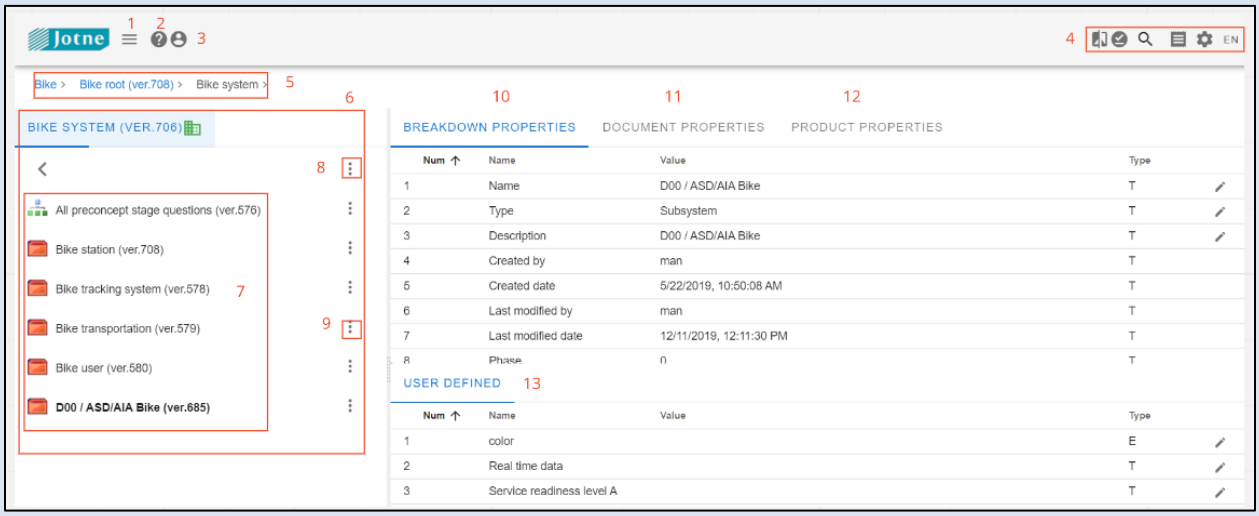


Figure 7. Navigation Page

The above figure shows the EDMtruePLM project page. Below is a short description of the numbered items of Figure 7:

1. System menu button: Includes project selection, project administration, system info etc.
2. Offline help document: a PDF version of the online document for the given version
3. User info: It shows user details such as user, role, name, email and organization
4. Project menu: Functionalities that will be available to the current active project
5. Breadcrumb address bar: The path to the selected element within the breakdown structure
6. Breakdown parent node name: The name of the parent element and some other information about the element
7. Breakdown structure: Representation of the product elements and their data files, shows the child element
8. Context menu of current node: Includes all functionality for the current node
9. Context menu of child node: Includes functions for child nodes, such as copy, move etc.
10. Breakdown properties: System defined meta data of the selected breakdown element along with assignments definitions as per STEP standard.

The properties are divided into two sections: fixed and flexible.

- fixed (counted) so named "system" node attributes collected from a predefined set of the breakdown element assignments
- flexible node attributes consist of 5 subsections see below Figure 8:
    - "Identifiers" - DEXlib template assigning_identification (assigning_identification.id_class_name = "Role", assigning_identification.id = "ID", assigning_identification.org_id = "Context")
    - "Classified by" - DEXlib template assigning_reference_data (superclass = "Role", assigning_reference_data.class_name = "Class")
    - "Described as" - DEXlib template assigning_descriptor (assigning_descriptor.class_name = "Role", assigning_descriptor.descr = "Text")
    - "Timestamps" - DEXlib template assigning_time (assigning_time.date_class_name = "Role", assigning_time.year, assigning_time.month, assigning_time.day = "Date")
    - "Persons" - DEXlib template assigning_person_in_organization (assigning_person_in_organization.person_role_class_name = "Role", assigning_person_in_organization.login = "Name", assigning_person_in_organization.org_id = "Organization")

Note: The attributes represent all assignments applied to the breakdown_element, breakdown_element_version or breakdown_element_definition representing the node. Every subsection is collapsible (expanded in the first appearance). The data of the assignment is used to populate attributes in fixed division, so some duplication exist.

— Identifiers	Role	ID	Context
	UID	1gvEFMUWeHxO00051MkOdM	EDMtruePLM
	Version	1	EDMtruePLM
	Name	D00 / ASD/AIA Bike	EDMtruePLM
— Classified by	Role	Class	
	Type	Subsystem	
	Phase	0	
— Described as	Role	Text	
	Description	D00 / ASD/AIA Bike	
— Timestamps	Role	Date	
	Modification time	12/12/2022, 11:47:48 AM	
— Persons	Role	Name	Organisation
	Version creator	jotne_mc	EDMtruePLM

Figure 8. Properties assignments as per STEP standards

11. Document properties: System defined meta data of the selected document
12. Product properties: System defined meta data of the selected product

13. User defined properties: Properties of the current node that were defined by the user (see “Reference data”).

3.1.1 Closing tabs

Additional tabs that appear, for example, after a search, are closed by a right-click on the tab and selecting the "Close tab" option.

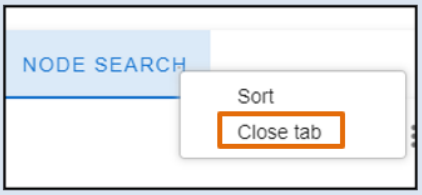


Figure 9. Closing tabs

3.2 Administration menu

Ordinary users without specific system administration rights have access to a limited set of project management functions; see Figure 10 for an overview of those. General project management is done by the system administrator; see chapter 4.

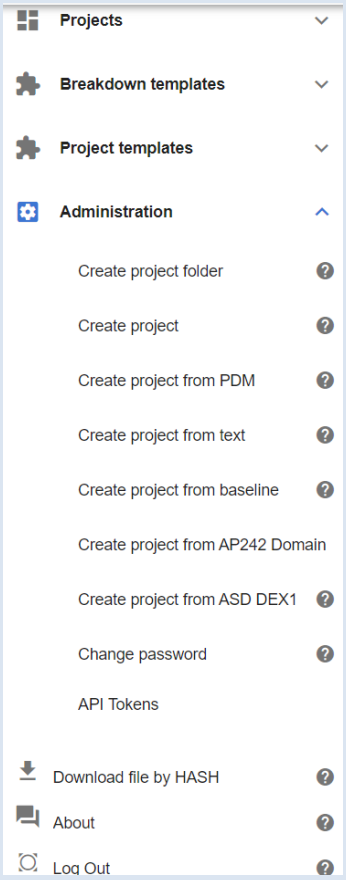


Figure 10. Administration menu – ordinary user

3.2.1 Projects

Shows all active projects that the user has access to. By clicking on a project, the project will be opened (see Figure 11).

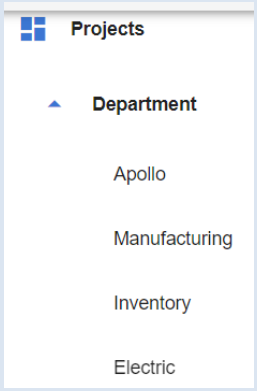


Figure 11. Active Project

3.2.2 Breakdown templates

Shows all the breakdown templates that the user is assigned to. By clicking on the template, the template will open.

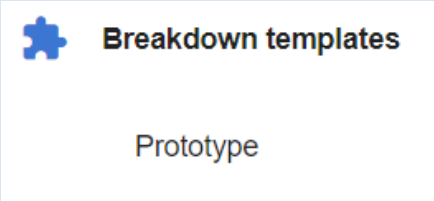


Figure 12. Breakdown template

3.2.3 Project templates

Shows all the project templates that the user is responsible for. By clicking on the template, the template will open.

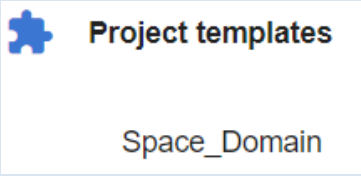


Figure 13. Project template

3.2.4 Administration

Allows the user to create a project folder (see 3.2.4.1) or to change login password (see 3.2.4.8). A project that is created here instead of in the system administration area, cannot be assigned to other users.

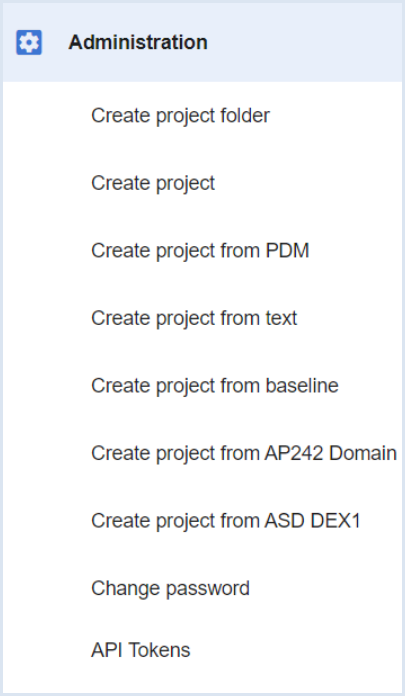


Figure 14. Administration tab

3.2.4.1 Create project folder

One can create a project folder to store related projects under a single branch. This allows grouping of projects based on user preferences.

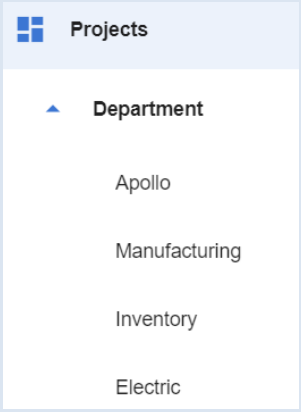


Figure 15. Example of a project folder

3.2.4.2 Create project

Click on the *Create project* item (see Figure 16).

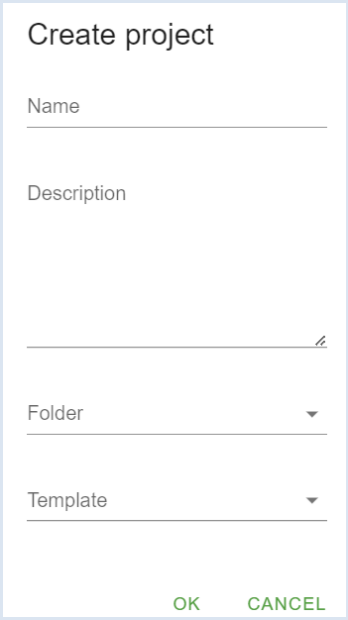


Figure 16. Create new project with active user login

The create project form takes the following information (see 4.5 for project name restrictions):

1. The name of the project (Mandatory)
2. The description of the project (Mandatory)
3. Folder assignment to sort a project (OPTIONAL)
4. A project template that will be loaded as initial population of the project (OPTIONAL).

After project creation the application shows the project root of the breakdown and automatically selects it. The context menu shown in Figure 124 is then available with the functions described in chapter 3.7 .

3.2.4.3 Create project from PDM file

Provide the name, description and the PDM STEP file (ISO 10303-21 format and with PDM data according to ISO 10303-242 or its predecessors).

Note: Only folders that have children are reimported.

The details of the input file are described in 3.7.1.5 .

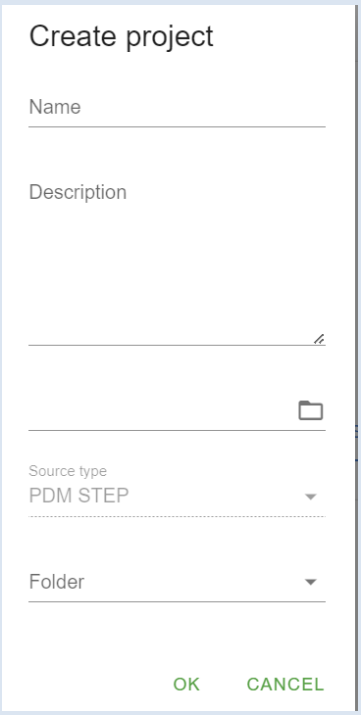


Figure 17. Project from PDM file

3.2.4.4 Create project from text

Provide the name, description, and the zipped file (see Figure 18). The text file syntax is described in 5 “Annex Guide – Text file structure”.

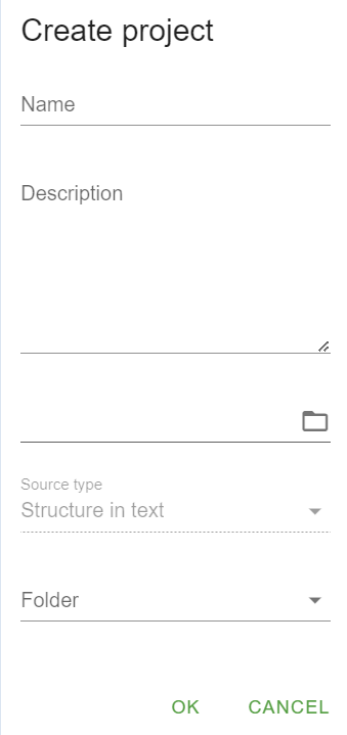


Figure 18. Project from text file

3.2.4.5 Create project from baseline

An exported baseline package may be imported to become a new project of its own. A baseline package includes a single version of the breakdown structure with all associated data and documents.

Whereas all data of the exported baseline will become available in the new project, there may be changes to user access.

The import function described here is available to all users. Here, the current user will be assigned the role of "Project Manager". The user who was Project Manager before the export will be assigned the role of "Project Admin". Thus, also the earlier "Project Manager" will have extensive access rights to the new project. If this is not desirable, the "System Administrator" (superuser) can remove this user from the new project or change his/her role.

Baselines may also be imported by the "System Administrator" (superuser); this is described in chapter 4.1.4 .

In both types of imports, all historical assignments of users are maintained. For example, the usernames in system properties like "Created by" and "Modified by" are the same as before the baseline export. This is independent of whether these users have access to the project or not.

Note: A username in the exported baseline may match a username in the target database of the import even though this is not the same person; the two users may, for example, have different e-mail addresses. For EDMtruePLM these two users are the same person. The username alone identifies a person and gives access to the system; only the "System Administrator" may change that principal access, and "Project Manager" and "Subdomain Leader" may change detailed access.

To execute this import, provide a project name, a description, and the zipped baseline file. The baseline file must have been exported from EDMtruePLM.

Figure 19. Project from zipped baseline

3.2.4.6 Create project from AP242 Domain

Provide the name, description and the AP242 file (eg. stpx or stp or xml format) to create project.

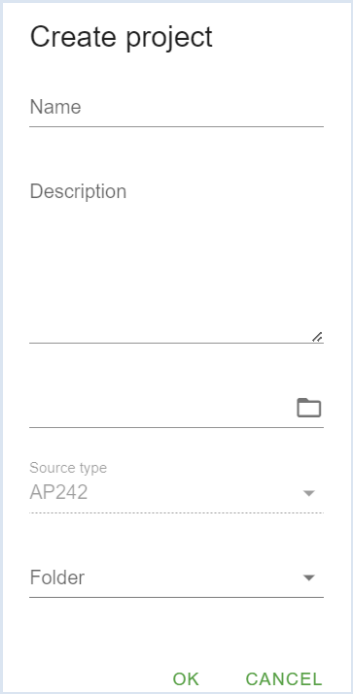


Figure 20. Project from AP242 Domain

3.2.4.7 Create project from ASD DEX1

Provide the name, description and the DEX1 file (see Figure 21). The ASD DEX1 file must be compliant to the DEX1 “Product breakdown for support” subset of the ISO 10303-239 data model.

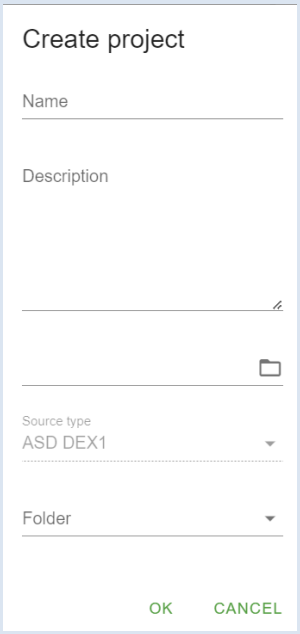


Figure 21. Project from DEX

3.2.4.8 Change user password

This form allows the logged in user to change password. With this all the associated API tokens will be deleted.

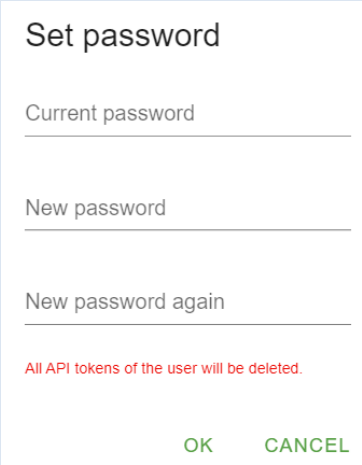


Figure 22. Change user password

3.2.4.9 API Tokens

In this, login log generated via Single sign on. A user can view the log and remove API Tokens if required. The details are described above in section 2.2.

3.2.5 Download file by Hash

This feature is used to identify and download documents with the help of blockchain values. Select the block from the blockchain window and paste the value in the input form; see Figure 23, below. This will download the related version of the document into the browser.

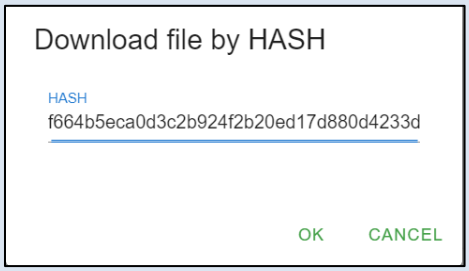


Figure 23. Download file by Hash

3.2.6 About (for a project user)

Shows the details of the current project and user profile, that is:

- project name, phase and its description;
- username and role;
- support email address and
- version of the current EDMtruePLM server; click on the three dots to see the list of all server constituents.

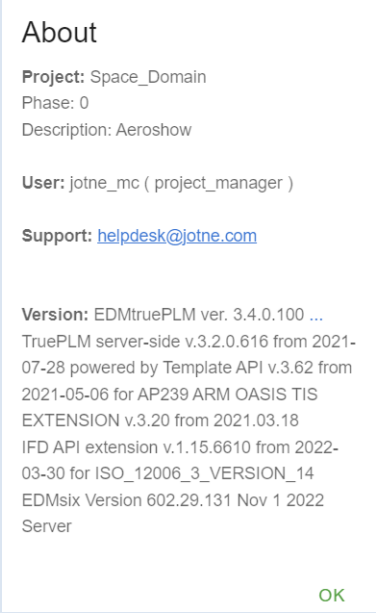


Figure 24. About current user and project details

3.2.7 Log Out

Logs out the current user.

3.3 Breadcrumb menu

The breadcrumb menu shows the path to the current breakdown element within the breakdown structure (see Figure 25). It is, thus, a navigation help. The individual constituents of the breadcrumb are hyperlinked. By clicking on a breadcrumb link the selected breakdown element is made current and is displayed in the structure pane on the left side. The left-most constituent of the breadcrumb path represents the breakdown root node.

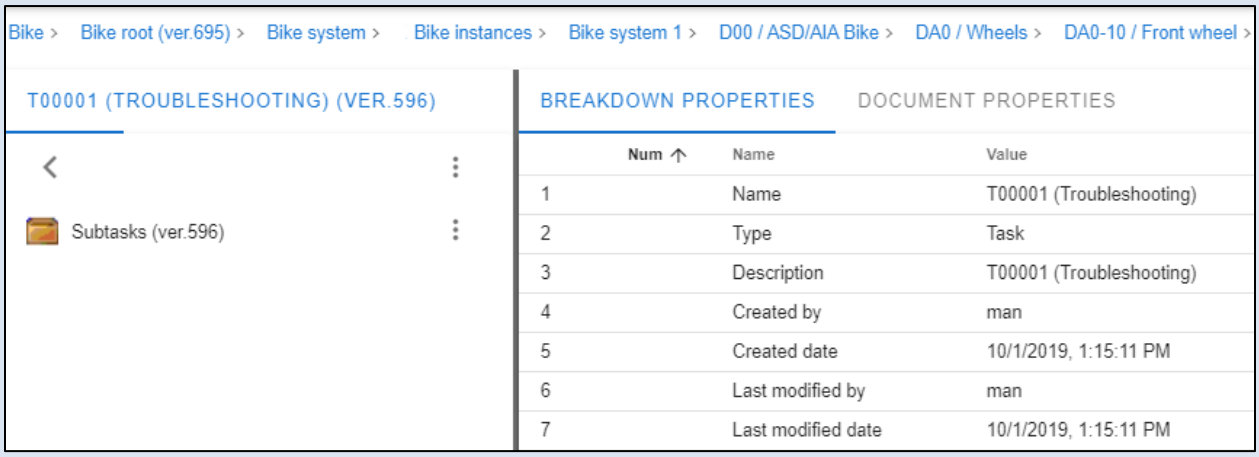


Figure 25. Breadcrumb menu

3.4 Breakdown Structure

The client shows one breakdown element, also called node, at a time in the breakdown structure view of the left pane; the name of the current breakdown element is the blue text on the top of the pane (see Figure 26). The view below that blue name displays the set of children breakdown elements and the documents that belong to the current breakdown element, as described below.

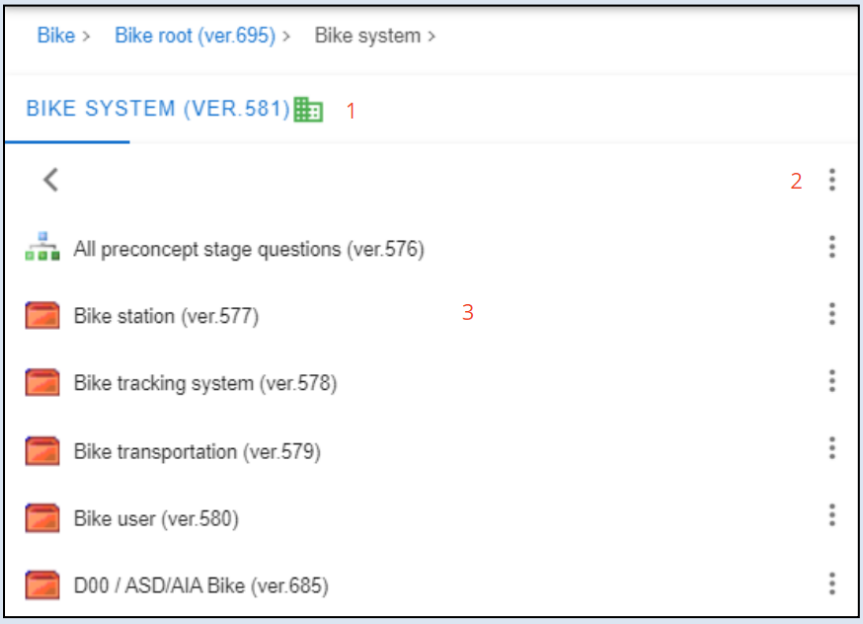


Figure 26. Breakdown Structure

The above picture shows a breakdown structure for a bike. The breakdown structure consists of three main sections (see red numbers 1, 2 and 3 on Figure 26):

1. Parent node name and information: Contains the name of the parent node, the version of the parent node and additional icons such as the subdomain icon.
2. Parent node context menu: Through this menu the user can access different breakdown element and document functionality.
3. Children nodes: All children nodes, that is, breakdown elements and data files, are located in this section. Depending on the type of node the user will have access to breakdown element or data file functionality through the child node context menu to the right of the child name or data file name.

The breakdown element shows the details of the child elements, documents, and subdomain when hover the mouse on it as shown in the below figure.

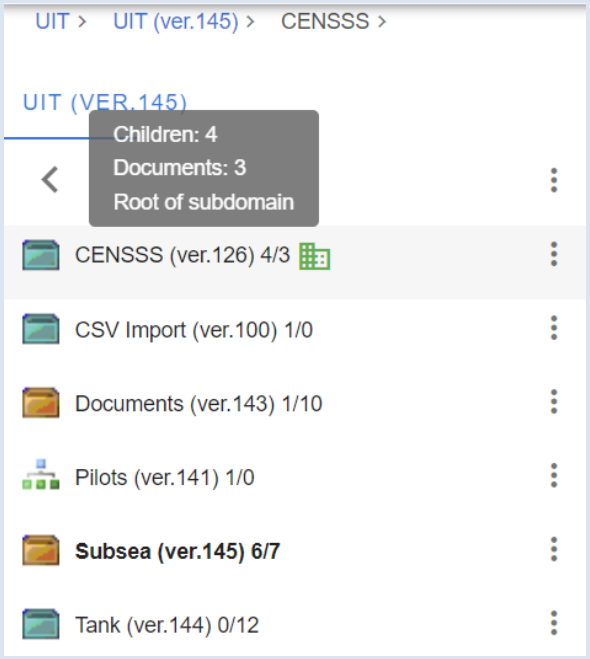


Figure 27. Breakdown Element with details

3.4.1 Notification of changes by other clients

In case other users than the current one has changed the breakdown structure, breakdown elements or documents, the system will provide the notification shown in Figure 28 in the client of the current user. You may select to receive the updates done by others to what you currently see on your screen (“Update”), or you may want to continue with your view undisturbed (“Ignore”). In case of the latter, be aware that your view does not any more reflect the status in the database.

Figure 28. Change notification

3.4.2 Breakdown element functionality

The functionality for managing breakdown elements is distributed over two context menus depending on the role that a breakdown element is playing. For a parent breakdown element, that is, for the root of a breakdown branch, a wider range of functions is available than when a breakdown element plays the role of a child node.

3.4.2.1 Breakdown versioning

Each change within the breakdown structure results in a new version of this structure, except for updates to documents. For each new version the breakdown version number is incremented by one and is assigned to the root of the breakdown structure; see Figure 29, below. With Manual versioning set up, user can increment the version manually.

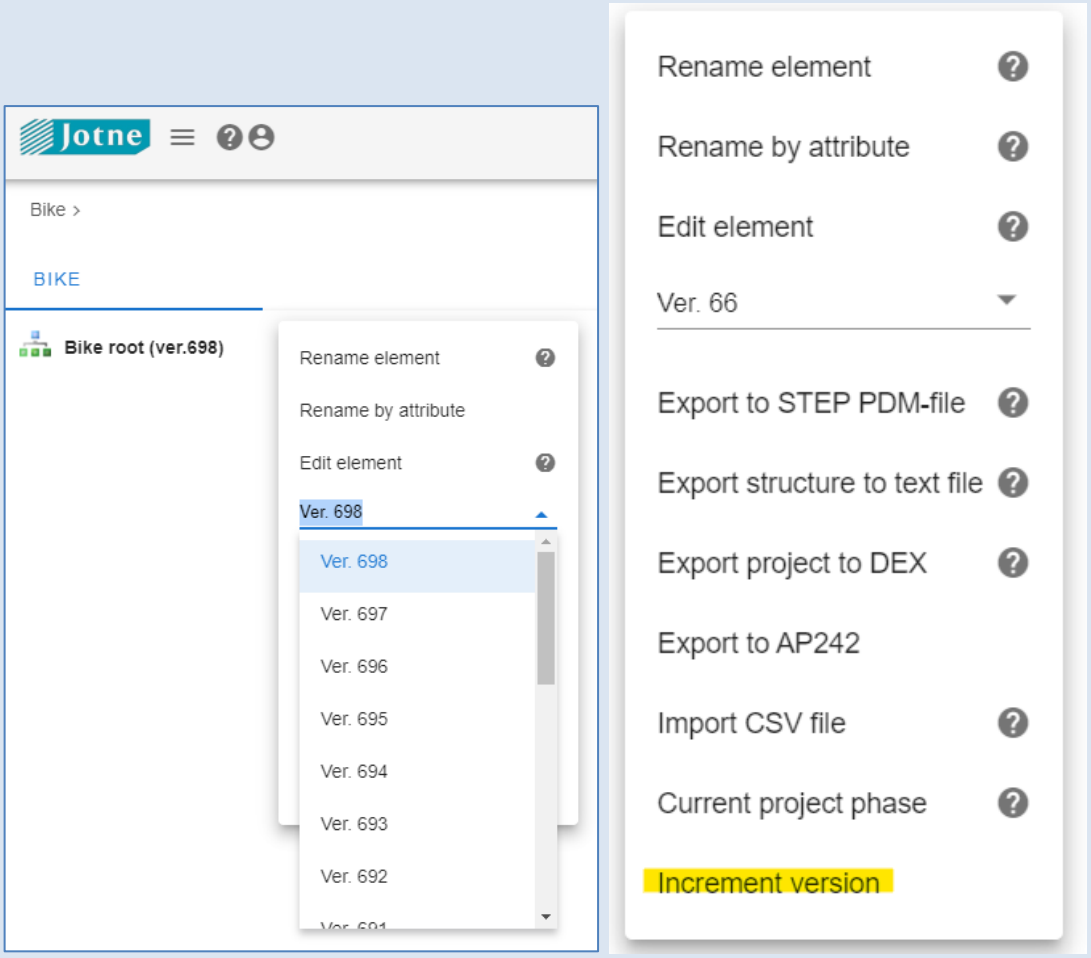


Figure 29. Breakdown structure versions

Any version may be set current by selecting it from the drop-down list. This version of the past may then be browsed with all its breakdown elements, documents, and property values. This may be repeated for any version from version 1 to the latest version. However, only the latest version may be changed; all other versions have read-only access. The read-only access is marked by the yellow lock sign to the right of all parent breakdown names that is also used for baselines, as described in Table 1.

When breakdown elements are created, the version number of the breakdown that their creation resulted in is added in parentheses to the end of the breakdown element name; see, for example, in Figure 35.

3.4.2.2 Child node functionalities

The child node context menu can be accessed by clicking on the icon to the right of the child nodes; see Figure 30. The individual functions are explained below.

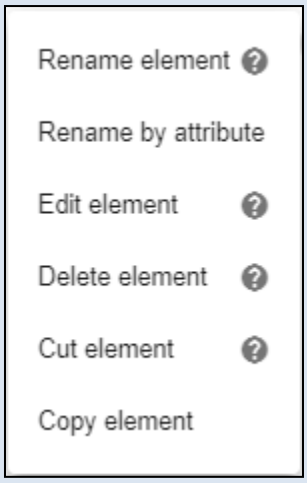


Figure 30. Child node context menu

3.4.2.2.1 Rename element

Allows the user to rename the child node (see Figure 31).

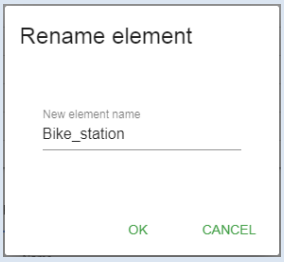


Figure 31. Rename element

3.4.2.2.2 Rename by attribute

This function replaces – temporarily – the name of the breakdown element where this function was invoked and its children by the value of a specified property. If the current breakdown element or some of its children do not have that property, that is, if a user-defined property is specified and this does not exist in all the involved breakdown element types, their names do not change.

The name of the attribute / property needs to be spelled exactly as listed in the breakdown properties panes; it is case sensitive. You may copy it from there and paste it into the input field; see Figure 32.

This change occurs only in the current client; it will be undone by an update of the client with server data, for example, by reloading the client webpage.

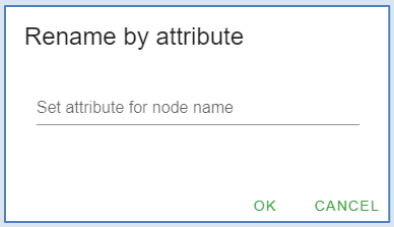


Figure 32. Rename by attribute

3.4.2.2.3 Edit element

Allows the user to change the description or the element type of the node (see Figure 33).

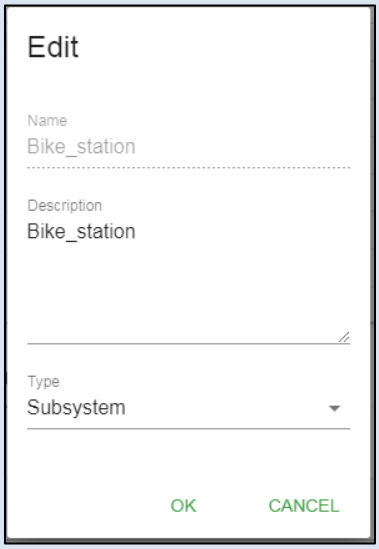


Figure 33. Edit element

3.4.2.2.4 Delete element

Allows the user to delete a breakdown element (see Figure 34).

Note: Remember that this function creates a new version of the breakdown structure in which the selected element has been removed. The removed element is still available, but only for reading, in the previous version of the breakdown.

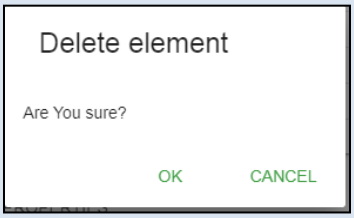


Figure 34. Delete element

3.4.2.2.5 Cut element

This function together with “Paste” can be used to move a breakdown element to a different location.

3.4.2.2.6 Copy element

This function works together with different Paste-functions within the parent node menu; see chapter 3.4.2.3 . Use this command to copy and paste the selected element with all its child elements and documents, that is, a complete branch, to another location.

Note: A branch can be pasted into the same parent breakdown element. In this case you will get two breakdown elements with the same names, but with a sequence number to distinguish them in parentheses (see Figure 35).

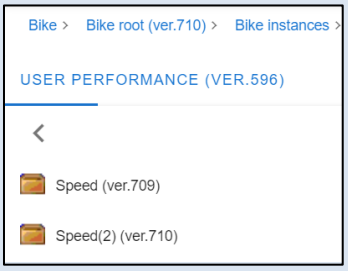


Figure 35. Two copies in the same parent

3.4.2.3 Parent node functionalities

The name of the current breakdown element, also called “Parent node” is listed below the breadcrumb. In the example in Figure 36 it is “BIKE SYSTEM”. When a breakdown element is initially selected, its properties are displayed in the pane named “BREAKDOWN PROPERTIES”. When one of the child elements of the current breakdown element are selected by single click, the “BREAKDOWN PROPERTIES” pane displays the properties of this child breakdown element. A second click on the same child name will make this child the current breakdown element.

Functions for the current breakdown element are available from the context menu right underneath its name (see Figure 36).

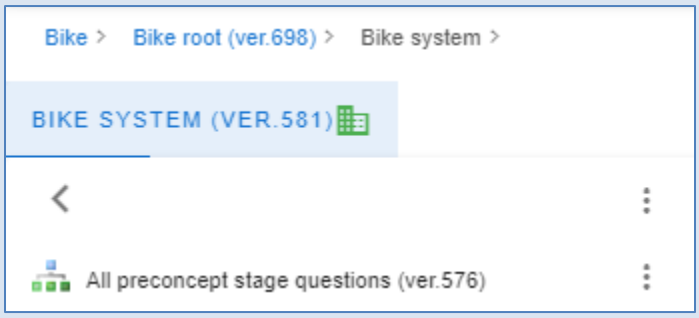


Figure 36. Parent node context

When the three-dot icon has been selected, the following menu in Figure 37 appears.

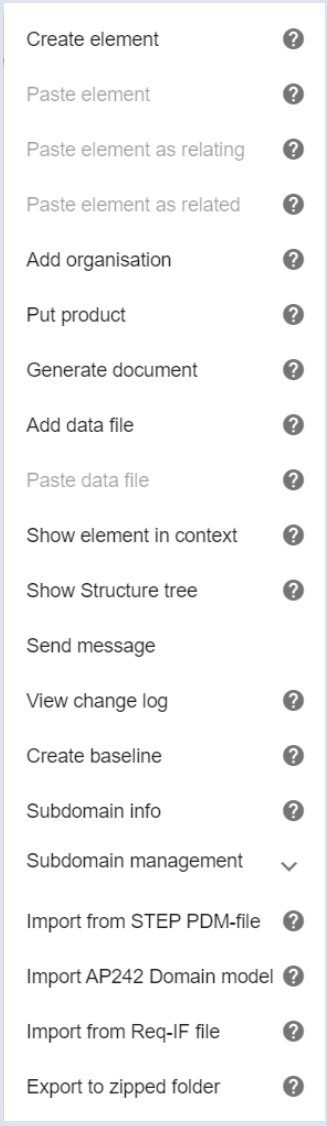


Figure 37. Parent node context menu

3.4.2.3.1 Create element

To add a breakdown element to the breakdown structure, left-click on the main breakdown node and choose "Create element" from the context menu. The input form in Figure 38 appears.

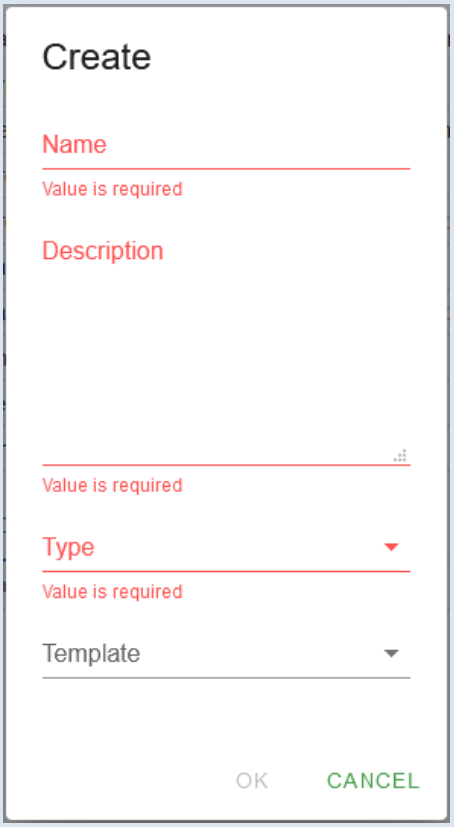


Figure 38. Create element

- Name: The name of the node.
- Description: Description of the node.
- Type: The type of the node. The values in the dropdown list are defined as breakdown element types in the reference data section.
- Template: A breakdown template that may be selected from a list of available templates to copy it to become the new breakdown element; children elements and documents of the template will also be copied.

Note: Breakdown elements have a system property called “Phase”. The value of Phase is set by the project root menu item “Current project phase” (see chapter 3.7.1.10).

To create a requirement, use the Create element function with “Requirement” as type. The description of the element is used as requirement text (Figure 39). Any resources that are associated to this requirement can later be added to the element.

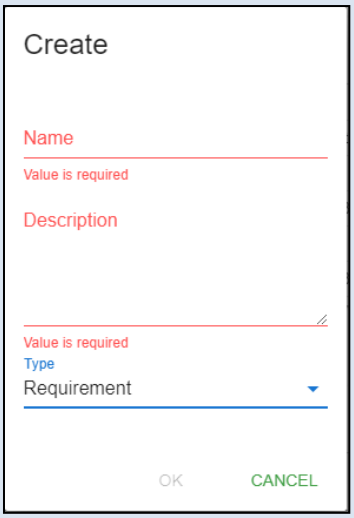


Figure 39. Requirement element

The parent node functionality of a requirement node is different than other breakdown elements (see Figure 40).

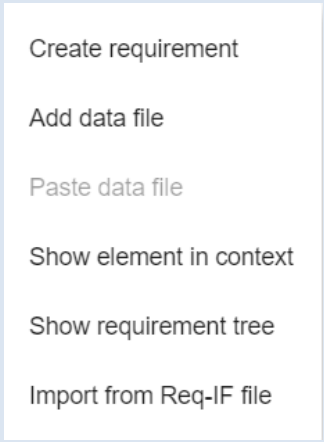


Figure 40. Requirement node menu

When using "Create requirement" only breakdown elements of type requirement or its subtypes will be available (refer Figure 41).

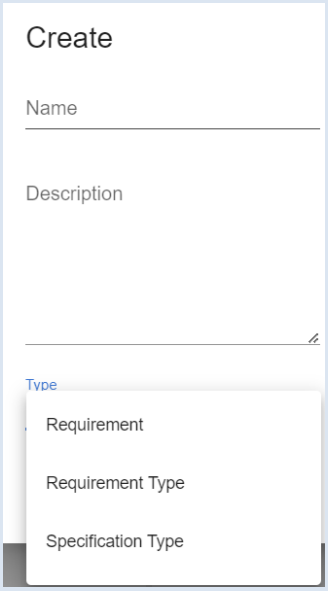


Figure 41. Requirement types

3.4.2.3.2 Paste element

This function is used after a Copy element or Cut element function; see "Child node functionalities" in chapter 3.4.2.1 . It copies or moves the element as child into the current breakdown element.

3.4.2.3.3 Paste element as relating

This function relates the pasted element as "relating" to the current breakdown element. The current breakdown element is then considered to be "related" to the pasted element. The difference between "related" and "relating" is that if one element of the relationship is dependent upon the other, it is the "related" element; see definition in ISO 10303-41. The matter of dependency is given by the semantics of the "Role" attribute.

Select "Copy element" from a child node context menu. Go to another element and click on "Paste as a relating" and fill the form. The selection of "Role" values is defined in the reference data section. The attribute "Remark" may be left empty.

The new relationship is viewed by clicking on the info button of the "Links" breakdown element property, as shown below in Figure 42. "Relating" relationships are always marked as "direct" links and "related" relationships are marked as "backward" links.

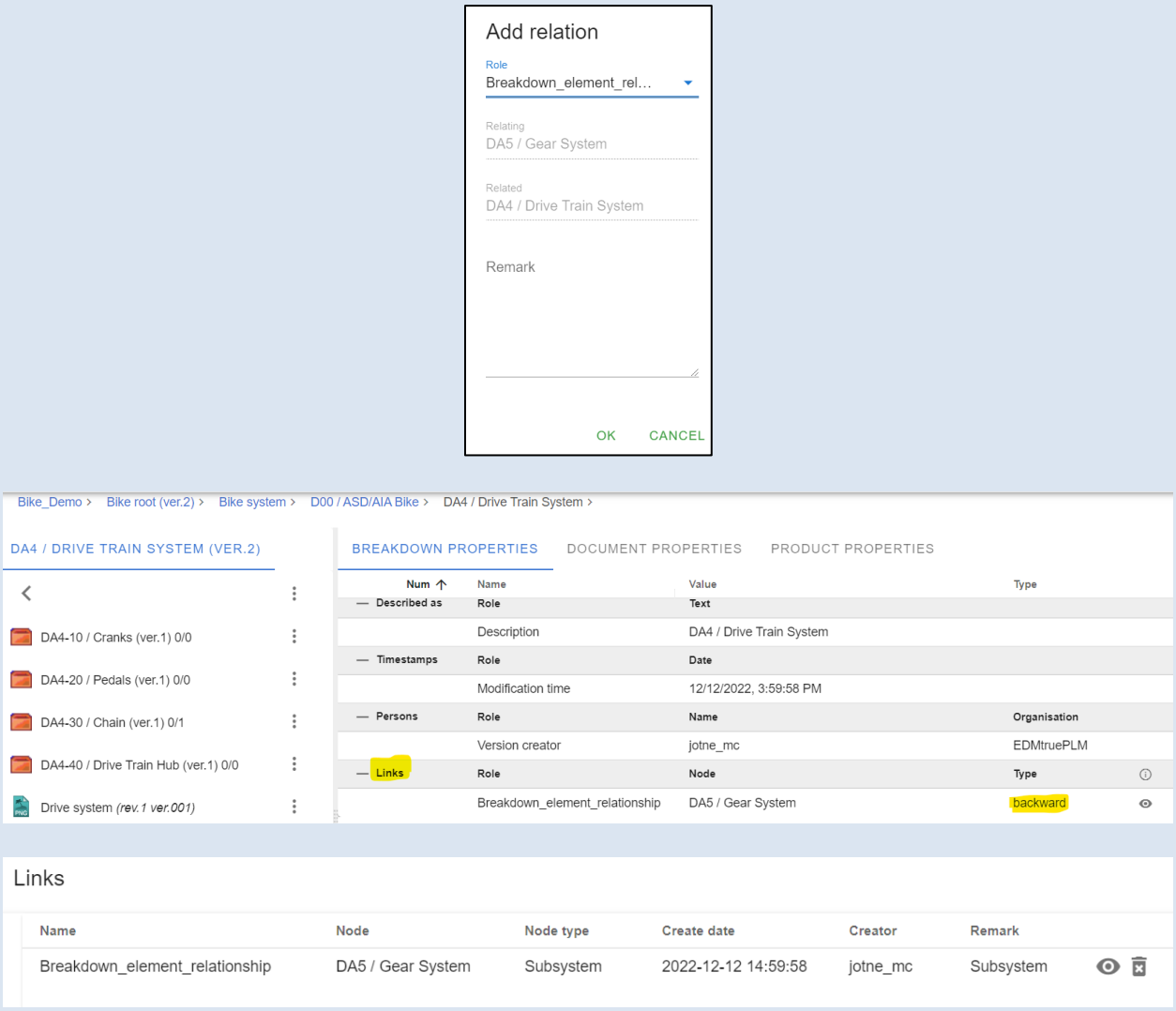


Figure 42. Establishing a backward/related relationship

3.4.2.3.4 Paste element as related

This function relates the pasted element as “related” to the current breakdown element. The current breakdown element is then considered to be “relating” to the pasted element. The difference between “related” and “relating” is that if one element of the relationship is dependent upon the other, it is the “related” element; see definition in ISO 10303-41. The matter of dependency is given by the semantics of the “Role” attribute.

Select “Copy element” from a child node context menu. Go to another element and click on “Paste as a related” and fill in the form. The selection of “Role” values is defined in the reference data section. The attribute “Remark” may be left empty.

The new relationship is viewed by clicking on the info button of the “Links” breakdown element property, as shown below in Figure 43. This “related” relationship is marked as “backward” link

in the copied and pasted element; the corresponding “relating” relationship is marked as “direct” in the target element.

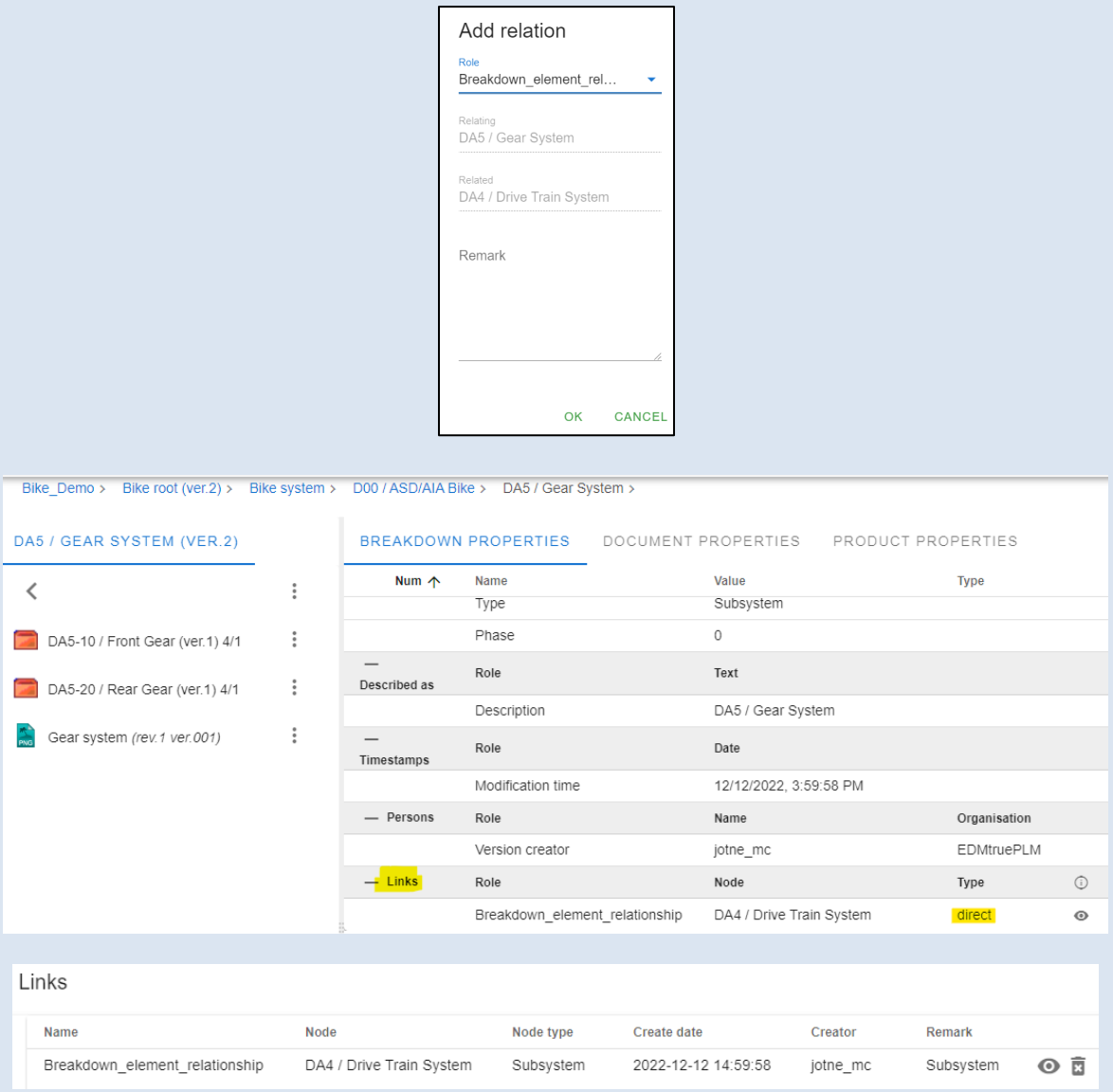


Figure 43. Establishing a direct/relating relationship

3.4.2.3.5 Add Organisation

Organizations that were created using the Catalogues icon can be added to the current breakdown element by this menu item. The role that the organization plays in this breakdown element is specified by selecting from the drop-down list of the input form; see Figure 44, below. The contents of the drop-down list are defined in the reference data section.

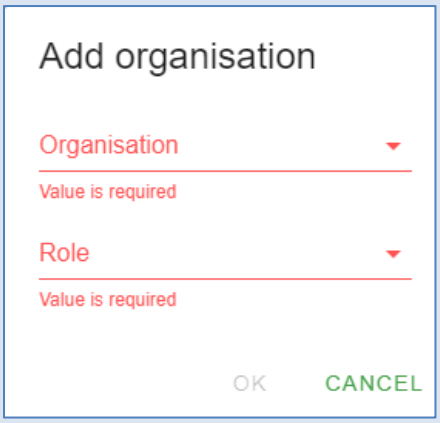


Figure 44. Add organization to a breakdown element

None, one or several organisations may be added. Details may be reviewed by clicking on the info button to the right of the "Organisations" system property; see Figure 45.

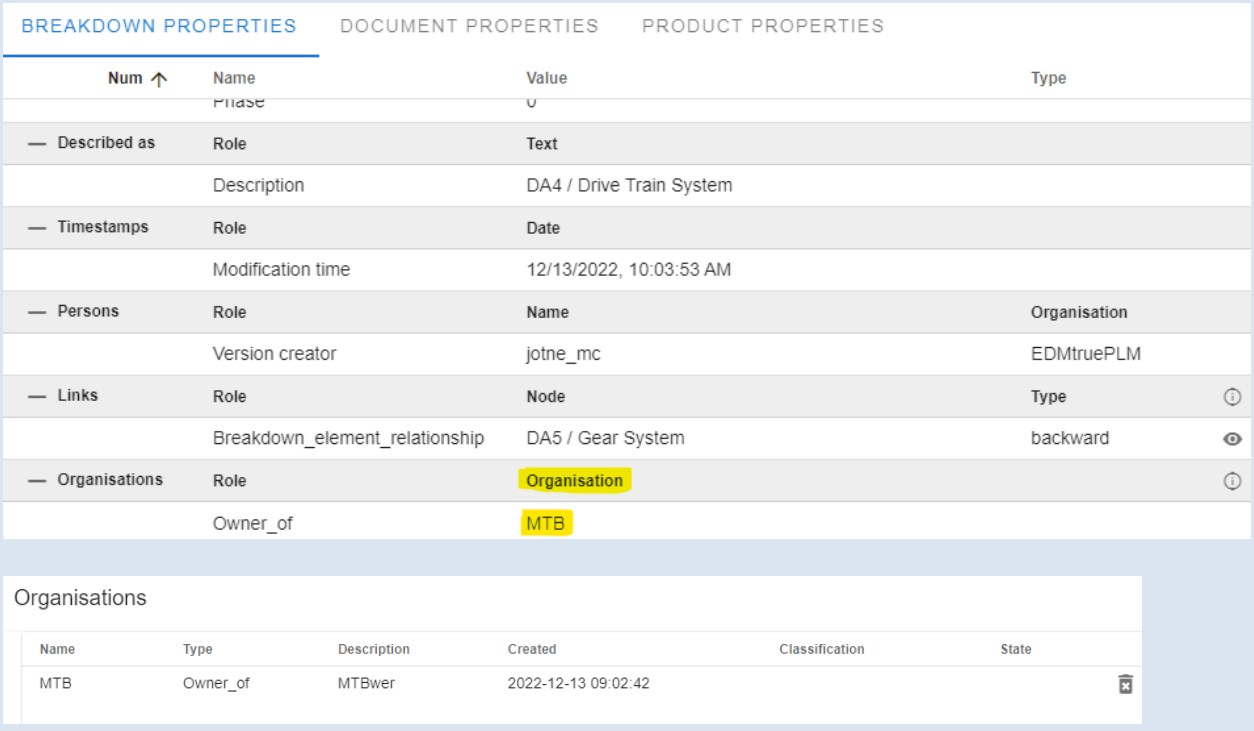
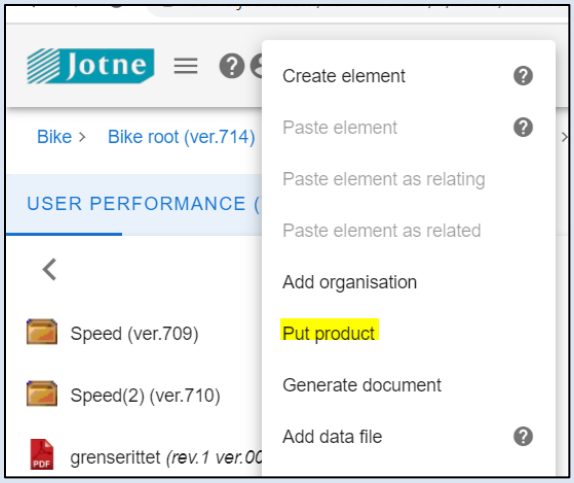


Figure 45. Organisation system property with details

3.4.2.3.6 Put Product

EDMtruePLM distinguishes products and breakdowns. Types of products are defined in the Catalogue area and may be linked to breakdown elements.

Products that were created using the Catalogues icon can be added to the current breakdown element by this menu item.



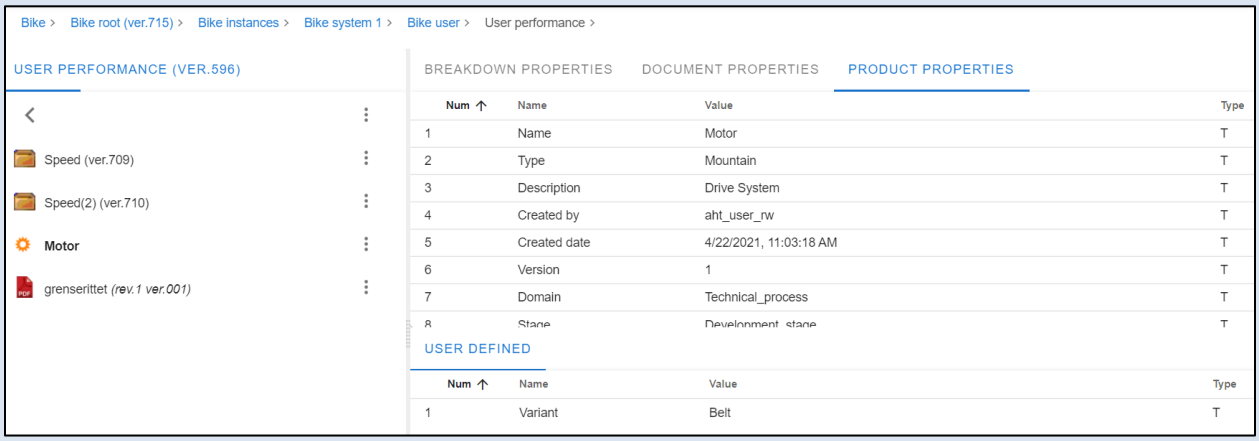


Figure 46. Product details

The product is pasted into the breakdown pane as shown on the left side in Figure 46. The product properties appear in a pane by itself, aside breakdown and document properties.

3.4.2.3.7 Generate document

This functionality is used to generate a document based on a template in Excel format. The following requirements apply for creating a template:

1) The first sheet is just the data representation page. The sheet may have any name.
2) The second sheet will be filled with the property values of the current node. The sheet may have any name.
3) The third and other sheets will be filled with data from the direct children nodes of the current node; there is no recursion below the second level. The name of each sheet must be the children node type; this type name will be used for filtering the children nodes.

Each cell of the first row of the template must include a name of the user-defined property of the current breakdown element. Additionally, the following system properties may be added:

'Name' - filled from the node name,

'Description' - filled from the node description,

'Node_ID' - filled from the node ID (GUID).

In case, not all fields were filled from by properties of the current node and the node has direct relations, related nodes will be used for filling the empty fields.

3.4.2.3.8 Add data file

Data files / documents can be added to all nodes within the breakdown structure except for the root node. Documents are listed together with child breakdown elements. To add a data file, open the parent node menu and click on the Add data file item (see Figure 47).

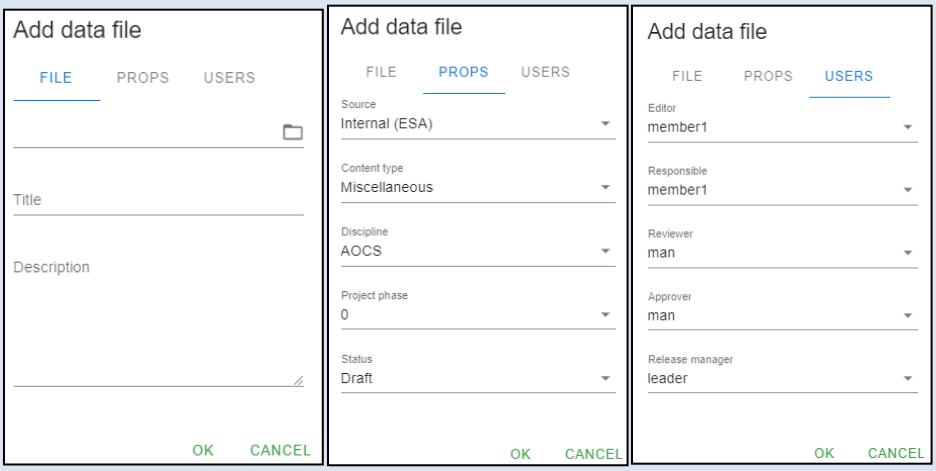


Figure 47. Add data file

To add a data file either use the browser button or just drop the file in the pop-up form. The following is a brief description of each input field.

i. File tab

- Title: The name of the document. This name will appear as document name in the breakdown structure. This field is filled automatically by the name of the selected file but can be changed.
- Description: A short description of the document. The field is mandatory.

ii. Props tab

The user may select the below information from drop-down lists. The contents of these lists are defined in the reference data section.

- Source: where does the file come from; for example, is it an internal resource or has an external origin.
- Content type: type of information in a document, e.g., CAD, Design, Requirement specification ... .
- Discipline: type of engineering, e.g., Electrical, Instrumentation, Mechanical ... .
- Project phase: project phase that the document applies to; the default value is changed by the project root menu item “Current project phase” (see chapter 3.7.1.10).
- Status: the level of progression of a document in its lifecycle workflow, e.g., Draft, Approved, Completed, Ready to review, Sent for approval ... .

iii. Users tab

The user may select the below information from drop-down lists that show the users available for the current project.

- Editor: The person who has edited the file.
- Responsible: The person who is responsible for the file.
- Reviewer: The person who is assigned to review the data file.
- Approver: The person who is assigned to approve the release of the data file.
- Release manager: The person who is responsible for the release process of the data file.

Note: The default "Responsible" user is selected based on the availability of a subdomain. If the user is adding a data file to an element that is not part of a subdomain, the default "Responsible" user is the user adding the data file. However, if the data file is added to a subdomain the "Responsible" user is the subdomain leader. In both cases, the user can use the "Responsible" dropdown control to select another user as the data file "Responsible" user.

3.4.2.3.9 Paste data file

This function pastes the latest data file that was subject to “Copy file” or “Cut file” into the current breakdown element. After “Cut file” the data file is moved into this new location and removed from its old location. After “Copy file” a link to the original data file location is added to the current breakdown element.

Thus, a single document is visible in multiple locations. Any modifications to the data file from any of the locations in the breakdown structure will be visible in all other locations.

3.4.2.3.10 Show element in context

This function displays the position of the current breakdown element with all its parents and with its direct children; that is, possibly many levels up, but only one level down (Figure 48).

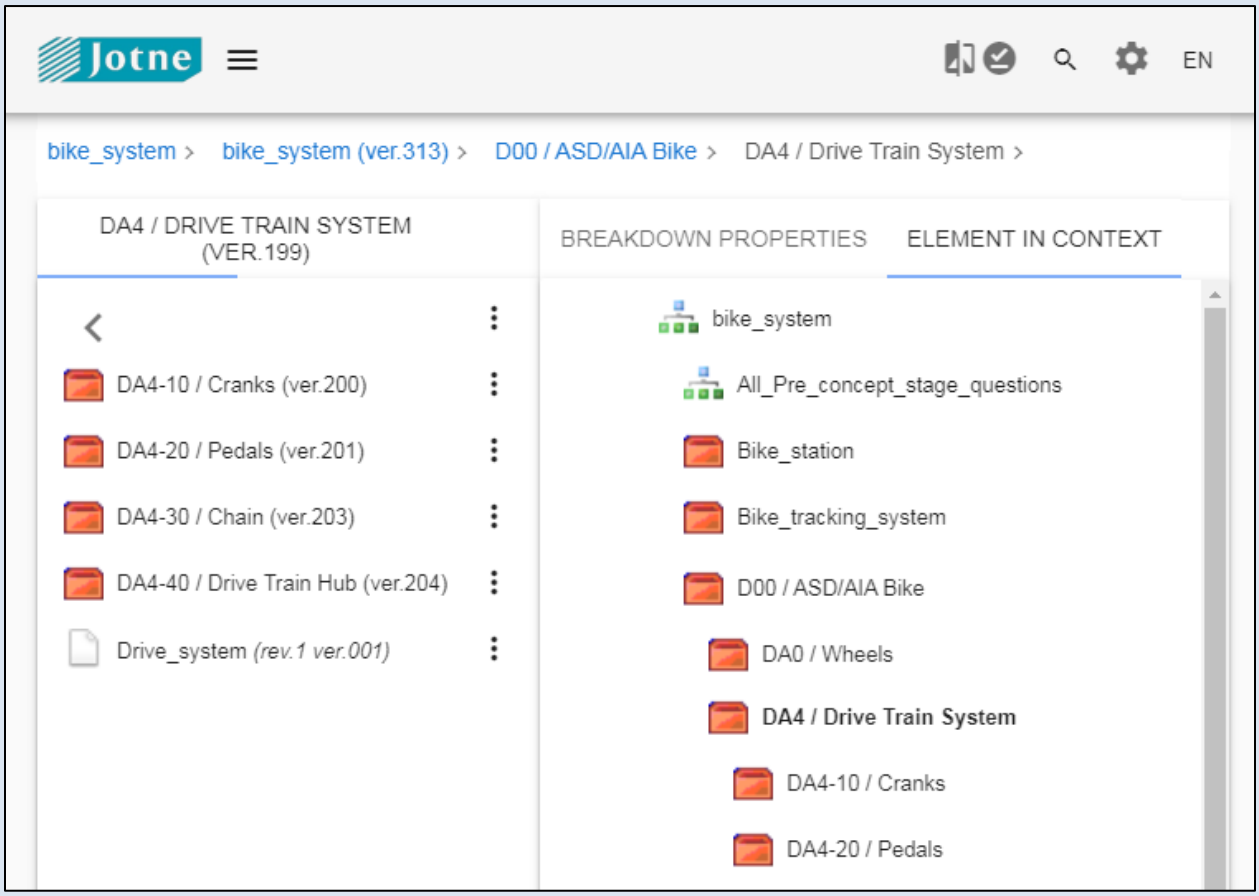


Figure 48. Show element in context

3.4.2.3.11 Show Structure Tree

This function displays the branch of the current breakdown element (Figure 49). The root of the branch is the selected parent node. Its direct children are presented; further levels of child breakdown elements may be opened interactively.

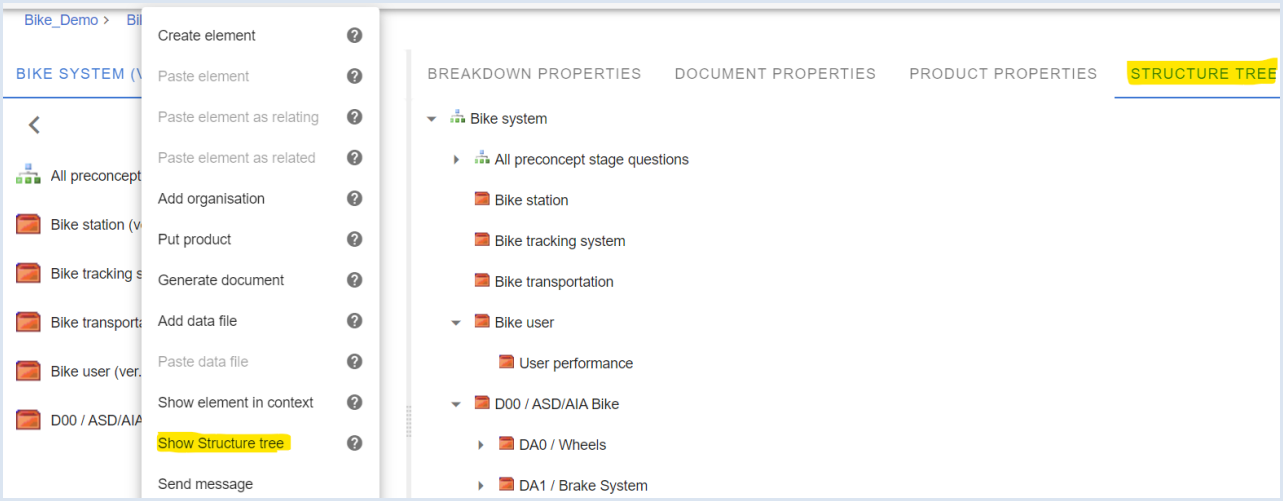


Figure 49. Show Structure Tree

3.4.2.3.12 View Change log

User changes to the properties of a breakdown element are recorded. This function displays a list of such changes to the current breakdown element; see Figure 50 for an example.

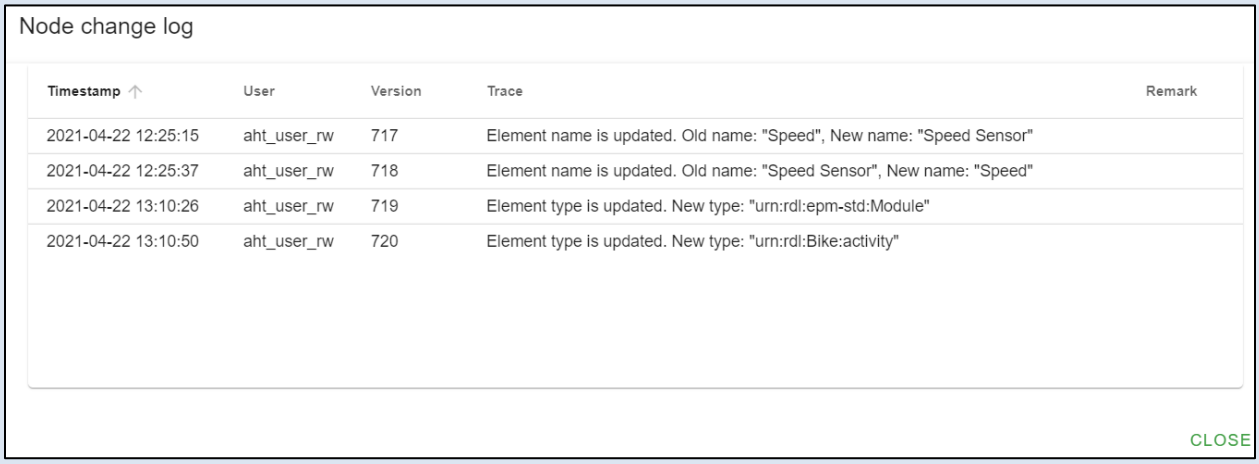


Figure 50. Breakdown element change log

3.4.2.3.13 Create baseline

A baseline is a version of the breakdown structure with a user-given name. To create a baseline, fill in the input form shown in Figure 51.

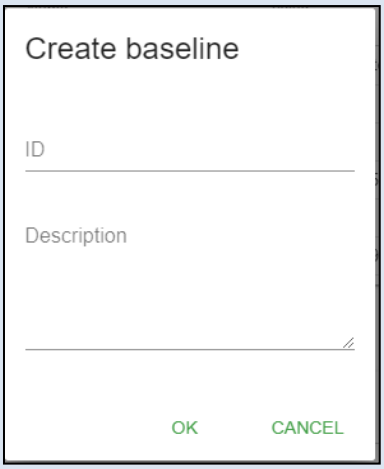


Figure 51. Create baseline

3.4.2.3.14 Subdomain info

This menu item is only visible if the current breakdown element is defined to be a subdomain. If so, details about the subdomain are displayed as shown below in Figure 52.

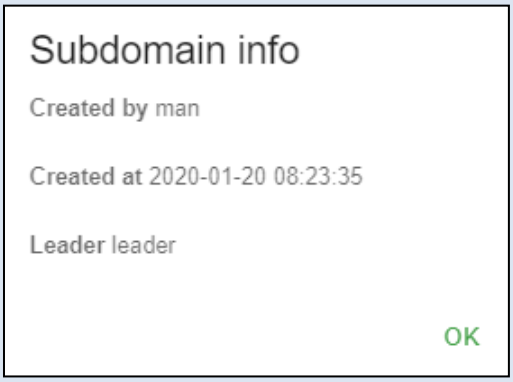


Figure 52. Subdomain info

3.4.2.3.15 Subdomain management

A subdomain is a part of a breakdown structure; it may be a branch or just an element. To create a subdomain the project manager or project administrator must select the breakdown element and click on the "Create subdomain" sub-menu to this menu item.

Notes:

- To create a subdomain the project must have a user with **Subdomain leader** as its user type. While the project manager and the project admin have higher privileges within the project, only a subdomain leader can be responsible for a subdomain.
- Project members have only access to the project's subdomains. Without a subdomain the project is only accessible to the project manager and project admin.

- The root of the breakdown structure cannot be a subdomain root.

Before a subdomain is created, the Subdomain management menu item has only the **Create subdomain** item (see Figure 53).

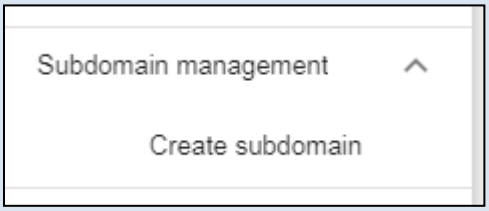


Figure 53. Subdomain management

After the subdomain is created the menu item will show three other items as shown below in Figure 54.

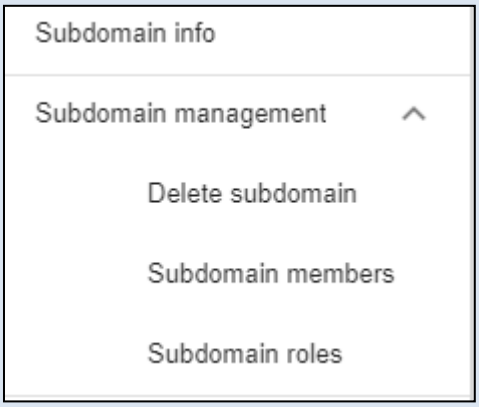


Figure 54. Subdomain options

i. Create subdomain

Click on the Create subdomain to create a subdomain (see Figure 55).

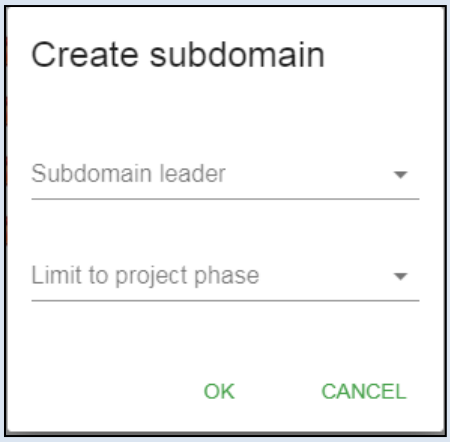


Figure 55. Create subdomain

1. Subdomain leader: the user responsible for the new subdomain. The user must have the type **Subdomain leader** associated with it.
2. Limit to project phase: the phase of the project that the subdomain is valid for.

Note: After the subdomain is created the corresponding breakdown element is assigned icon to the right of its name; for an example, see Figure 36.

ii. Delete subdomain

Will delete the subdomain (see Figure 56). Project members and subdomain leader will no longer have access to the breakdown structure if they are not part of any other subdomain.

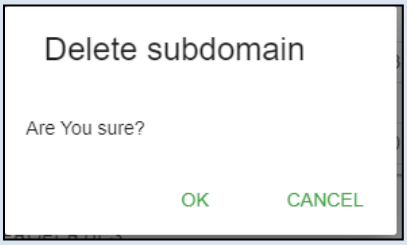


Figure 56. Delete subdomain

iii. Subdomain roles

Allows roles to be assigned to the subdomain and permissions to be attached to these roles.

Note: Roles are not permissions; a role can be assigned multiple permissions, one or none. If a role does not have any permissions assigned to it, the users that are assigned to that role will not have access to the subdomain.

As subdomain leader, select a role, defined in the reference data section, and assign permissions to the role (Figure 57).

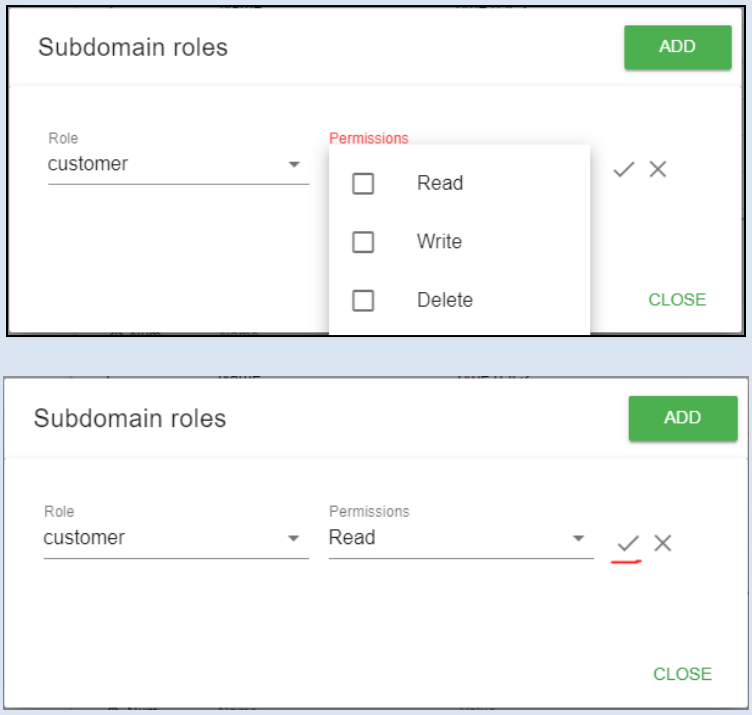


Figure 57. Subdomain roles

Each subdomain role can be given the following access rights:

- Read
- Write
- Delete

By giving the Delete access right the system will also give write and read access to the role. These access rights are only for documents and their properties within the subdomain. Subdomain roles can be granted different access rights within different subdomains.

iv. Subdomain members

Allows to add, remove, and configure the users within the subdomain (see Figure 58). Before adding users, you must have added subdomain roles to the reference data through the reference data section. To assign the defined roles to the subdomain use the Subdomain roles menu item.

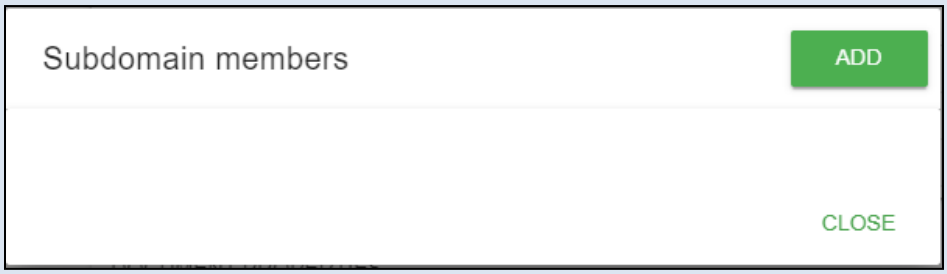


Figure 58. Subdomain members

Use the ADD button to add project members to the subdomain. Select the user and assign a role to the user as shown in Figure 59 below.

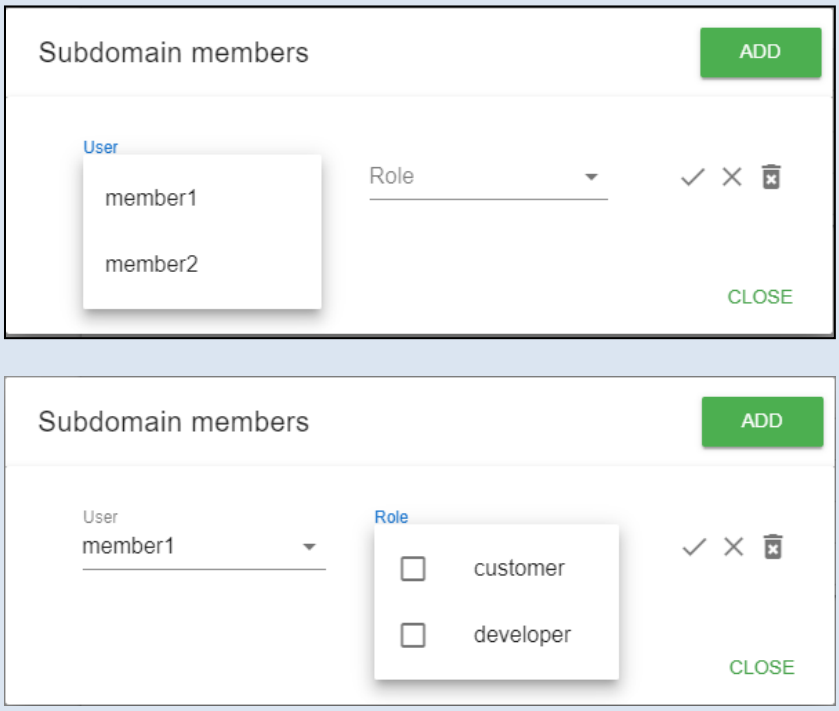


Figure 59. Add subdomain members

Description of the icons in the above form

1. Accept the values
2. Clear the values
3. Delete the user as member of this subdomain.

Multiple users can be assigned to the subdomain with different roles; see Figure 60, below.

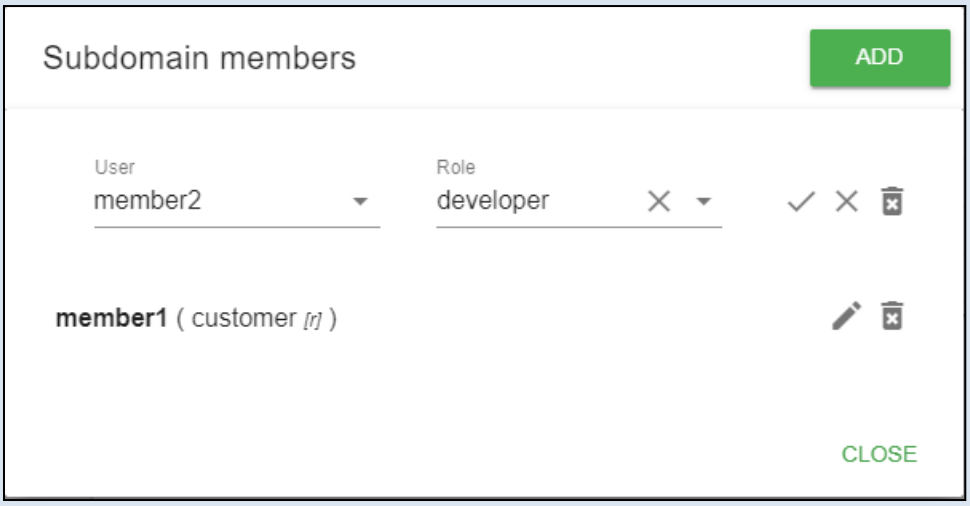


Figure 60. Subdomain members details

3.4.2.3.16 Import from STEP PDM file

The user can extend the breakdown structure by importing a PDM file (compliant to ISO 10303-242 MIM P21 or predecessors) (refer Figure 61). The system will use the data from the PDM-file to create a breakdown structure below the parent node.

The details of the input file are described in 3.7.1.5 .

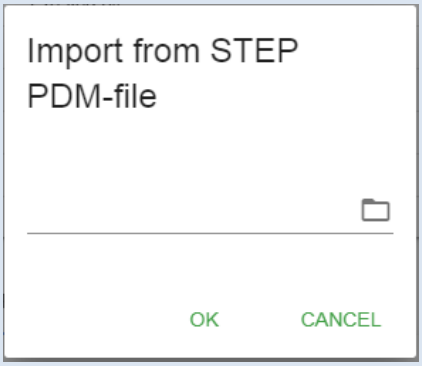


Figure 61. Import from STEP PDM

3.4.2.3.17 Import from AP242 Domain model

This menu item allows to import an AP242 Domain model in XML format (ISO 10303-4442, .stpx file type) to the breakdown structure (see Figure 62)

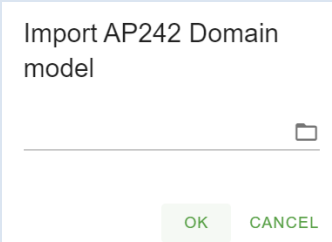


Figure 62. Import from AP242 Domain model

This import function handles the following types of data:

- Assembly structures
- Part occurrences: they are added to the breakdown structure tree
  - Limitation: attributes Id, Quantity and ShapeElement are ignored.
- Assignments of types:
  - identification
  - classification
  - description
  - date/time
  - person
- Documents.

3.4.2.3.18 Import from Req-IF file

This menu item allows to import a requirement file of type Req-IF (see Figure 63). Req-IF (Requirements Interchange Format) is an XML file format controlled by the Object Management Group (OMG) that can be used to exchange requirements, along with its associated metadata, between software tools from different vendors.

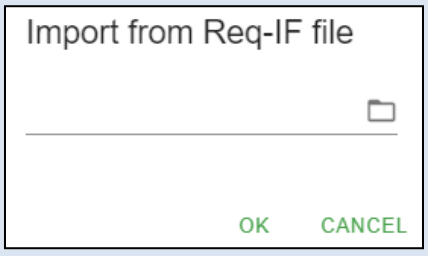


Figure 63. Import from Req-IF

3.4.2.3.19 Export to zipped folder

This command exports the current breakdown element with its child nodes and documents to a zip-file that can be unpacked into a MS Windows folder structure. The folder structure within the zip file will mirror the breakdown structure underneath the selected node.

3.4.3 Data file functionalities

EDMtruePLM is a product data management system, which collects product data into a breakdown structure that consists of breakdown elements. Each breakdown element may include sub-elements and documents. Documents are stored under each element. This allows the system to keep track of all files and their relevance for breakdown elements.

See Figure 64, below, for the menu with document / data file related functions.

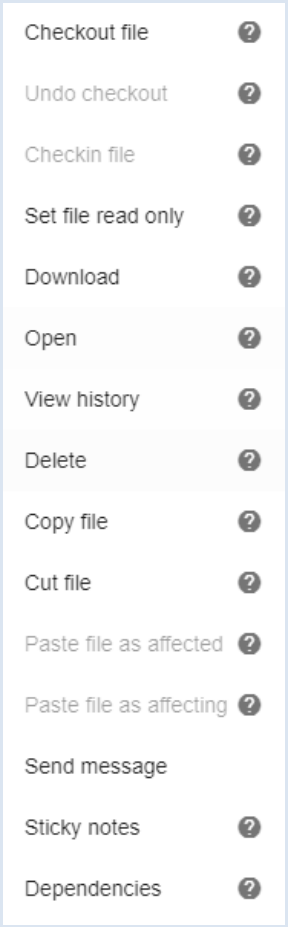


Figure 64. Data file menu

3.4.3.1 Checkout file

EDMtruePLM supports exclusive document editing that only allows one user to edit a document at a single time. To have exclusive rights to edit a specified document the user needs to check out the document, which is done through the data file context menu "Checkout file" command. To store the updated content of the document to the server use the "Checkin" command in the document's context menu. After checkout, the lock icon will be placed in front of the data file as shown below in Figure 65.

Figure 65. Checkout of a file

To unlock the file, use the *Undo checkout* command or check in the file.

3.4.3.2 Undo checkout

This feature enables the user to undo a checkout, if mistakenly the data file was checkout. Only the owner is allowed to do undo.

3.4.3.3 Checkin file

To save your updates on a previously checked out file you must check in the file (see Figure 66). During checkin the following process happens:

1. The file is copied to the database.
2. The version of the file is updated.
3. The lock on the file is removed.

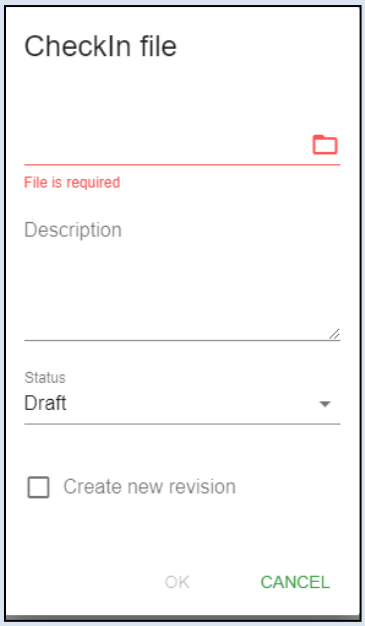


Figure 66. Checkin of a file

Status: change the status tag to one of the items in the drop-down list; status values are defined in the reference data section.

Create new revision: to track document changes, EDMtruePLM distinguishes revisions (major changes) and versions of a document. The version tag will increase automatically after each checkin. The revision tag increases when the user ticks off the box in Figure 66.

3.4.3.4 Set file read only / Clear file read only

This command changes access rights to the selected file to read only. The file name changes colour from black to grey; see Figure 67, below. The file cannot be changed until the menu item "Clear file read only" is selected. Only project manager, admin or subdomain leader are allowed to set or clear ready only.

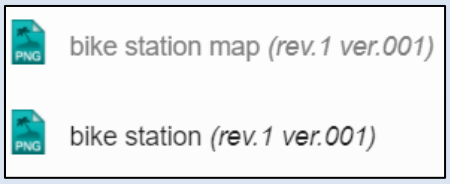


Figure 67. Read only file

3.4.3.5 Download

Downloads the current file to the client machine without any further user interaction.

3.4.3.6 Open

Opens the data file using the open file command of the web-browser. Depending on the type of web-browser the contents will become visible in a new browser tab (Firefox), or the file is downloaded (Chrome).

3.4.3.7 VCollab Visualization

This function allows to visualize the contents of CAD or similar files with geometrical data that are uploaded in STEP, ISO 10303, format. The STEP file is converted to a tessellated visualization file in .wcax format by the function “Convert to wcax”. Click “Open” for the resulting .wcax-file, as shown below in Figure 68, to visualize the geometry in the VCollab web-viewer; see Figure 69 for an example geometry.

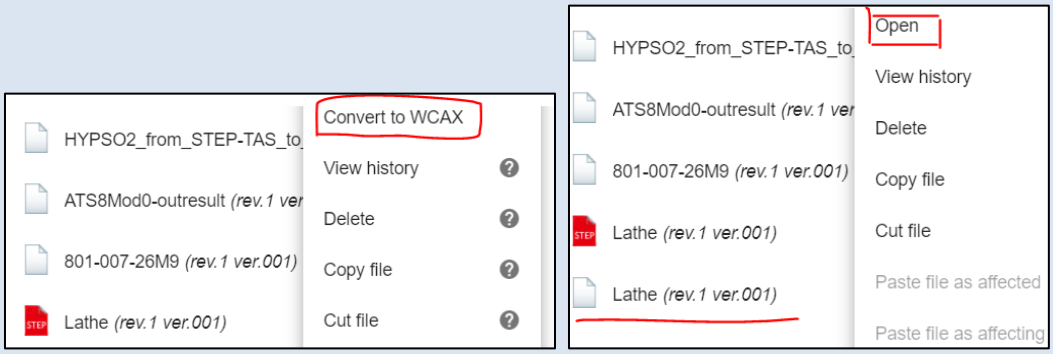


Figure 68. Convert to WCAX file

Figure 69. VCollab Web viewer

3.4.3.8 View history

Shows all the versions of the selected document in a pop-up window; see Figure 70, below.

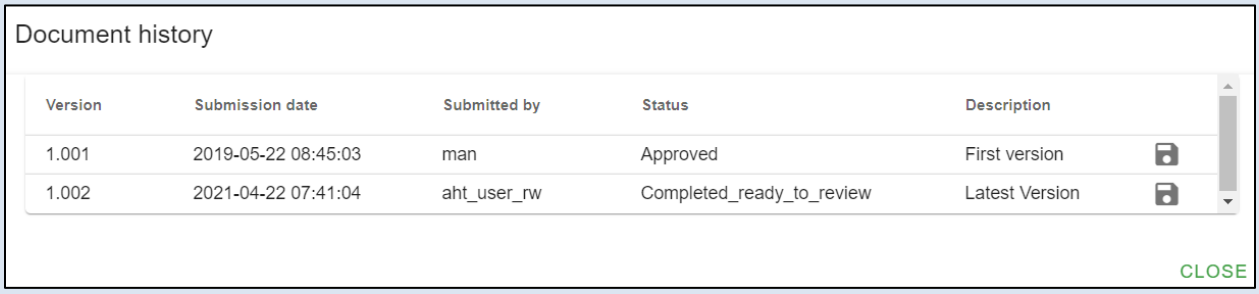


Figure 70. View History

The user can download a specific version of the document by clicking on the Save icon; there is no further user interaction involved.

3.4.3.9 Delete

Creates a new version of the breakdown and deletes the selected file from that new version. The document is still available in the previous version of the breakdown and is there still assigned to the current breakdown element.

Note: Deleting a file will prompt you a warning message about its dependencies. User can select whether to delete all or selected ones as shown in Figure 71

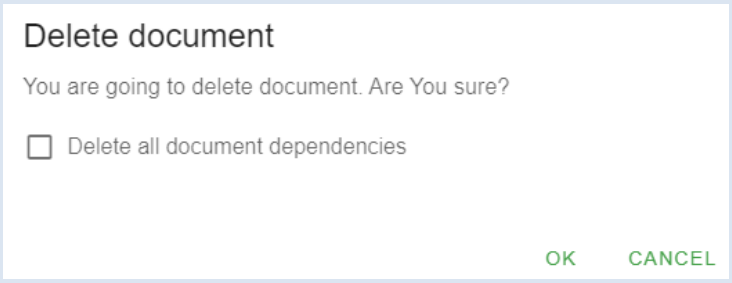


Figure 71 Delete Document

3.4.3.10 Copy file

Copies the file identifier into memory for later use by different paste commands. This function does not create an actual copy of the file but creates a link to the file.

Warning: If a file is copied from one subdomain to another, security issues must be taken into account. To avoid that different user in different subdomains have access to and potentially edit the same document, the document needs to be added individually to the different subdomains;

that is, the “Add data file” menu item needs to be applied with the same file for each subdomain in question.

3.4.3.11 Cut file

The selected file is marked for removal. It will be first removed when the “Paste data file” menu item of the parent node context menu is applied.

3.4.3.12 Paste file as affected

This function attaches a copied file to another, current file as an affected data file. This means that the pasted file is dependent on the current file. After a change to an affecting file, the affected file is marked by a red flag; see chapter 3.4.3.15 Dependencies, below.

The function does not require any other user input. The result of the paste-operation can be reviewed by the “Dependencies” menu item.

3.4.3.13 Paste file as affecting

This function attaches the copied file to another, current file as an affecting data file. This means that the current file is dependent on the pasted file. After a change to an affecting file, the affected file is marked by a red flag; see chapter 3.4.3.15 Dependencies, below.

The function does not require any other user input. The result of the paste-operation can be reviewed by the “Dependencies” menu item.

3.4.3.14 Sticky notes

Sticky notes are reminder notes for documents. The input form in Figure 72 will pop up to allow the attachment of notes to data files.

Figure 72. Sticky Notes

Use the “New”-button and then the “Submit”-button to add a note (see Figure 73).

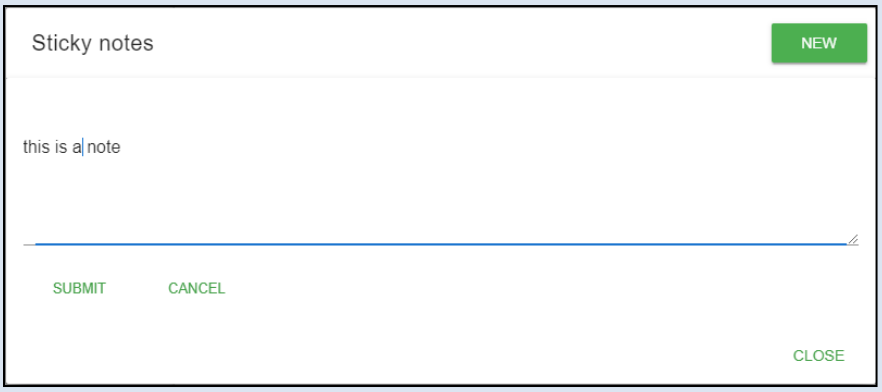


Figure 73. Create new note

All notes of the data file are collected in a single sticky note form. A yellow line icon with a number identifies documents that already have sticky notes (see Figure 74).

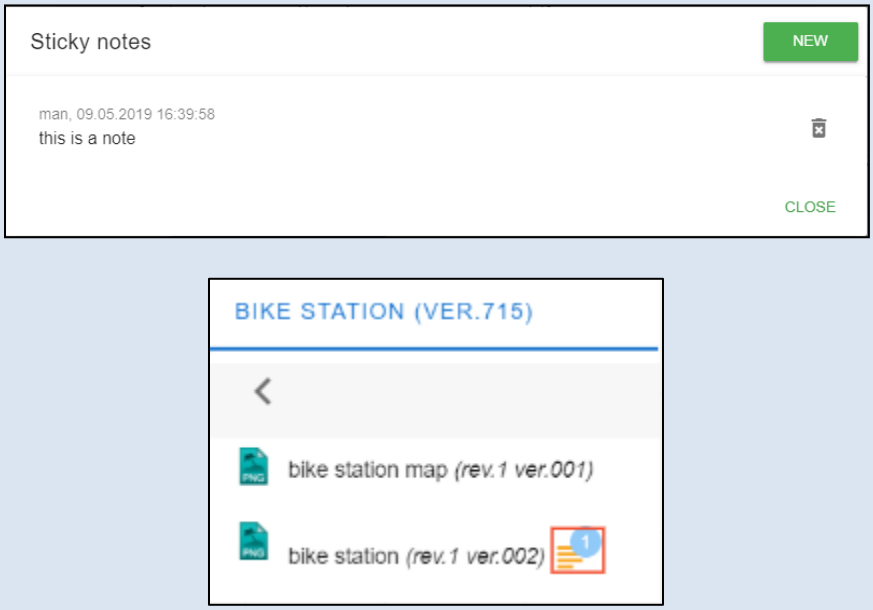


Figure 74. Sticky note form

3.4.3.15 Dependencies

"Dependencies" is a type of relationship between two files as shown below in Figure 75. An "affecting" file is the master file, an "affected" one a dependent file. When an affecting file has been edited, the user will be reminded by red flags to check all affected files whether they need to be updated, too.

Figure 75. Dependencies

By clicking on the menu item, the dependency form will appear and will allow the user to view the dependencies of the file or to remove them.

- **Paste file as affected:** Attaches the latest copied file as an affected data file. This means that the pasted file is dependent on the current document.
- **Paste file as affecting:** Attaches the latest copied file as an affecting data file. This means that the current file is dependent on the pasted file.
- **Red Flags:** Red flags are notifications to the user to check a file that may be impacted by a change done to an affecting file. When a file that is affecting other files is checked in, the dependent files are assigned red flags (see Figure 76). Subsequent changes to the affecting file will result for each affected file in one red flag per version or release that was checked in.

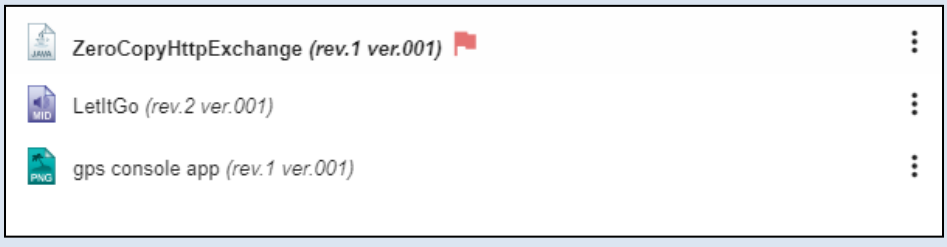


Figure 76. Red Flag

Removing red flags: To remove the red flag, open the Dependency form of the affected file; see below, Figure 77.

Figure 77. Removed red flags

Click on an affecting file with a red flag; this will open the red flag drop list as shown in Figure 78.

Figure 78. Dependencies tab menu

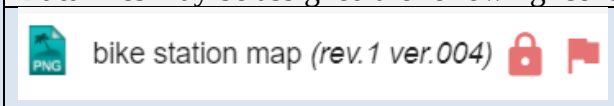
To remove a red flag, click on the tick mark of the red flag item in the drop list.

3.4.4 Icons used for breakdown elements and data files

The overview in Table 1, below, summarizes the meaning of icons used with breakdown elements and data files in the breakdown structure pane.

Table 1. Icons for breakdown elements and data files

Breakdown elements may be assigned the following icons:	
Bike (ver.601)	The brown folder icon represents one type of breakdown element; other types are assigned other icons. New types of breakdown elements are created in the reference data section; icons are assigned automatically.

Subdomain root as parent 	The icon follows the parent breakdown element name.
Subdomain root as child 	The same icon is used when the breakdown element that is a subdomain is listed as a child of its parent breakdown element.
	When the breakdown pane shows a baseline, the yellow lock is assigned to the parent breakdown element that is the root of the baseline branch. The lock icon is chosen to show that baselines can only be accessed for reading, not for writing.
Data files may be assigned the following icons:	
	Data file icons indicate the type of data file. Revision and version ids follow the data file title. The lock icon indicates that the file is checked out. The red flag icon shows that a file that this file is dependent on has been updated.
	The blue circle indicates the number of sticky notes for the data file.

3.4.5 Navigation

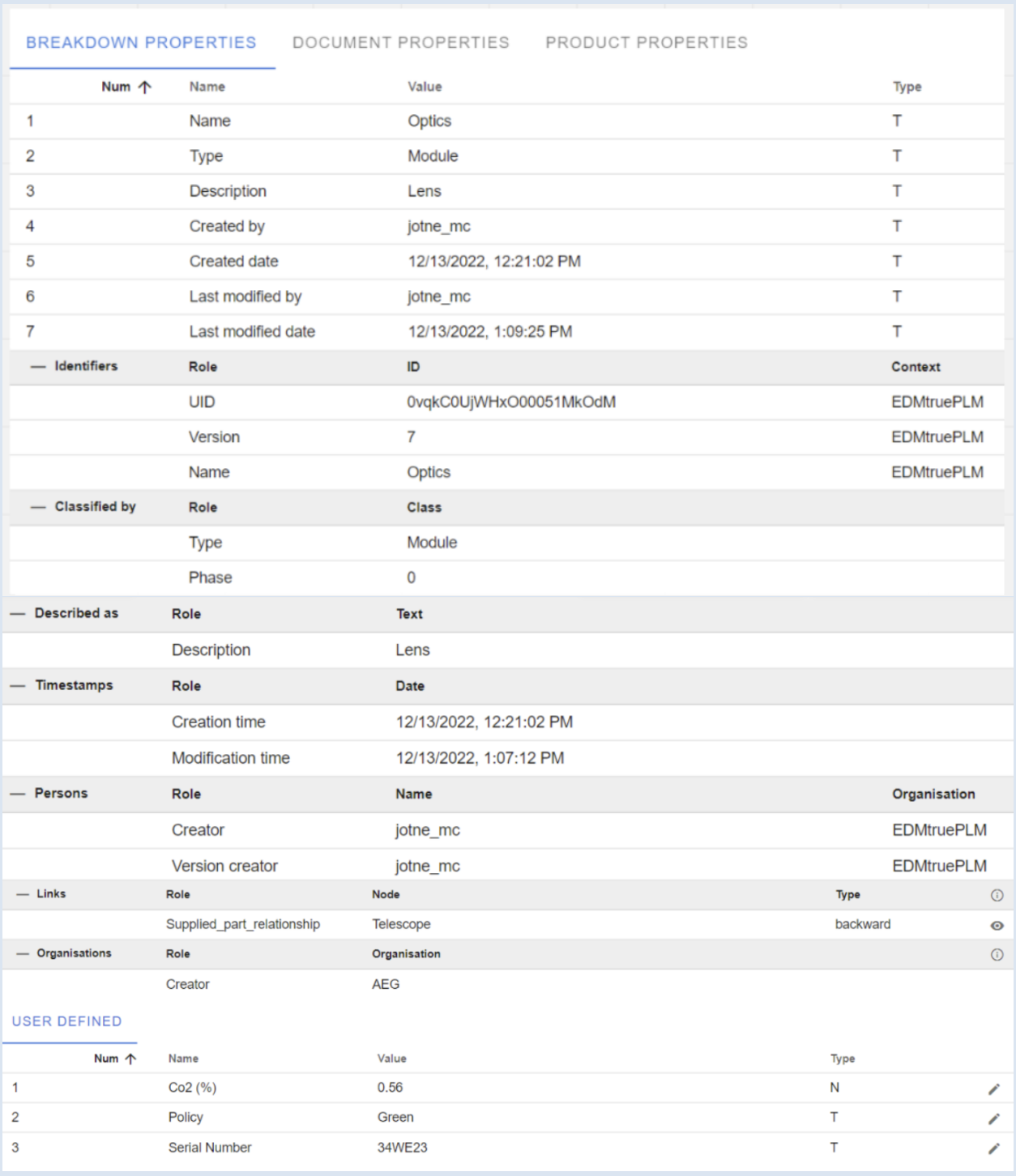
There are three methods for navigating within the breakdown structure:

1. Browsing by clicking on breakdown element nodes.
2. Searching for node or data files.
3. Following hyperlinks of the breadcrumb path.

Warning: You cannot use the browser navigation buttons to move within the breakdown structure. EDMtruePLM is a single page application and using the browser navigation buttons will take you out of the application.

3.5 Property window

The three types of property windows show metadata associated with breakdown elements, data files and products (see Figure 79 below).



BREAKDOWN PROPERTIES | DOCUMENT PROPERTIES | PRODUCT PROPERTIES

Num	Name	Value	Type
1	Name	Optics	T
2	Type	Module	T
3	Description	Lens	T
4	Created by	jotne_mc	T
5	Created date	12/13/2022, 12:21:02 PM	T
6	Last modified by	jotne_mc	T
7	Last modified date	12/13/2022, 1:09:25 PM	T

— Identifiers	Role	ID	Context
	UID	0vqkC0UjWHxO00051MkOdM	EDMtruePLM
	Version	7	EDMtruePLM
	Name	Optics	EDMtruePLM

— Classified by	Role	Class
	Type	Module
	Phase	0

— Described as	Role	Text
	Description	Lens

— Timestamps	Role	Date
	Creation time	12/13/2022, 12:21:02 PM
	Modification time	12/13/2022, 1:07:12 PM

— Persons	Role	Name	Organisation
	Creator	jotne_mc	EDMtruePLM
	Version creator	jotne_mc	EDMtruePLM

— Links	Role	Node	Type
	Supplied_part_relationship	Telescope	backward

— Organisations	Role	Organisation
	Creator	AEG

USER DEFINED

Num	Name	Value	Type
1	Co2 (%)	0.56	N
2	Policy	Green	T
3	Serial Number	34WE23	T

Figure 79. Property window

The top section shows system properties, the lower section shows user-defined properties. System properties are predefined by the system with representation as per STEP Standard for the

assignments such as Identifier, Classified by, Described as, Timestamps, Persons, Links and Organisation. User-defined properties are defined in the reference data page by the user. Right after installation of EDMtruePLM the user-defined property sections are empty.

Note: The property windows always show the properties of the last selected node, data file or product.

Icon's definition

- Allows the user to edit the property value; properties that do not have this icon are not editable. This feature does not allow you to leave the property value field empty, that is, without a value. Use the following function to unset a property value.
- Will clear a property value.

Note: Deleting a property can only be done from the reference data page.

3.5.1 System properties of breakdown elements

Breakdown elements have the following set of predefined properties (Table 6).

Table 2. Breakdown element system properties

Name	Description	Type	Editable
Name	The name of the breakdown element	string	yes
Type	The type of the breakdown element	reference data	yes
Description	The description of the breakdown element	string	yes
Created by	The name of the user who created the node	username	no
Created date	The date when the node was created	date and time	no
Last modified by	The name of the user who did the last update	username	no
Last modified date	The latest date when the node was modified	data and time	no
Identifier	Shows Context, Role and ID	STEP representation	no
Described as	Shows Role and Class	STEP representation	no

Name	Description	Type	Editable
Timestamps	Shows creation and modification time for that version	STEP representation	no
Persons	Shows details of creator	STEP representation	no
Links	The links between the breakdown element nodes	STEP representation	no
Organisation	The organisation details	STEP representation	no

User defined properties will also appear within this window.

3.5.2 System properties of data files

Documents, also called data files, have the following set of predefined properties (Table 3).

Table 3. Data file system properties

Name	Description	Type	Editable
Title	The title of the document	string	yes
Name	The file name	string	no
Description	The description of the document	string	yes
Size	The size of the document	string	no
Created by	The name of the user who added the document to the system	username	no
Created date	The date that the document was added to the system	date	no
Last modified by	The name of the user who did the last update	username	no
Last modified date	The last date when the document was checked in	date	no
Version	The current system version of the document	version id	no
Phase	The process phase that the data file was created in	reference data	no
Approver	The name of the user who shall approve the document	username	yes

Name	Description	Type	Editable
Release manager	The name of the user who manages the release process of the document	username	yes
Responsible	The name of the user responsible for the document.	username	yes
Reviewer	The name of the user who shall review the document	username	yes
Discipline	The discipline that the document is associated with	reference data	yes
External version	An external version given by the user	string	yes
Format	The file extension	string	no
RID references	A list of issue identifiers from the review process of the document	string	yes
Source	Identification of the origin of the document	reference data	yes
Status	The status of the document in its lifecycle	reference data	yes
Type	The category of document	reference data	yes

User defined properties will also appear within this window.

3.5.3 Product properties

Products have the following set of predefined properties.

Table 4. Product properties

Name	Description	Type	Editable
Name	The human understandable name of the product	string	no
Type	The category assigned to the document	reference data	no
Description	The description of the document	string	no
Created by	The name of the user who added the product to the system	username	no

Name	Description	Type	Editable
Created date	The date when the product was added to the system	date	no
Version	The current version of the product	version id	no
Domain	The discipline of the product	reference data	no
Stage	The stage that product is at in its lifecycle	reference data	no

User defined properties will also appear within this window.

3.6 Project menu

The project menu is located at the top right-hand side of the screen.

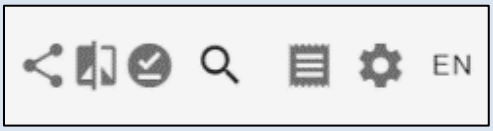


Figure 80. Project Menu

3.6.1 Blockchain

The blockchain window allows to download the block of hash values created for all the uploaded documents. Click on the red framed icon (“Open blockchain”) to open the Blockchain window as shown in Figure 81, below. Each row represents a version of a document. One can download the set of all blockchain values by selecting the download option. To copy the hash value of a single document version select the copy icon next to a hash value as shown below. The metadata properties such as Title, Version, User etc. can be viewed by clicking on the individual block.

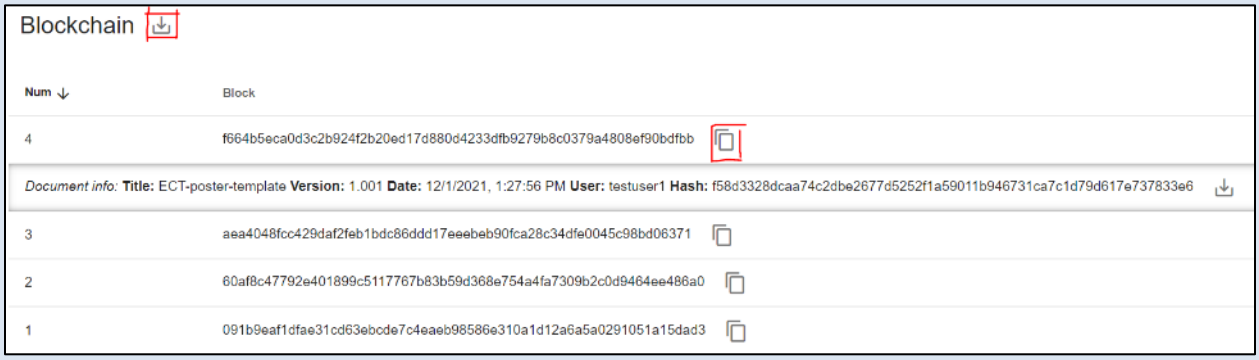


Figure 81. Blockchain

3.6.2 Make comparison

To compare breakdown versions and baselines click on the compare icon in the project menu. The opened form will allow a user to choose two versions or two baselines from the dropdown list; note that in order to compare baselines both baselines must have the same root.

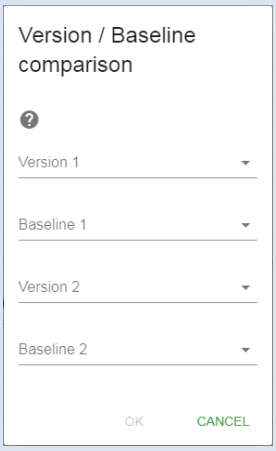


Figure 82. Version and baseline comparison tab

Select the values and click on the OK button. The Result of the comparison is displayed in a separate "Diff" window as shown below in Figure 83. This window is divided into four main sections:

- Two Product breakdown windows: Show the two baselines or versions that are compared. The breakdown identifications are placed on top of each window.
- List of differences: This summarizes the differences between the two breakdowns.
- Details: The detailed description of a selected row in the list of differences.

Note: Baselines with only a subset of the complete project breakdown can only be compared against other baselines of the same branch – one cannot compare them with the entire structure.

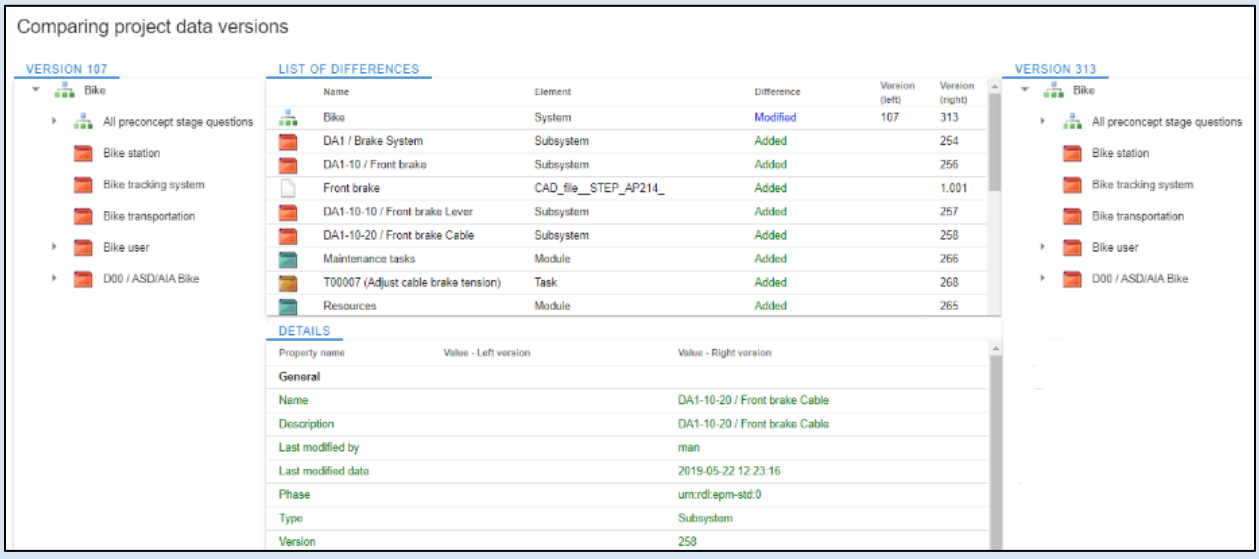


Figure 83. Version comparison window

3.6.3 Baselines

Baselines are a method of versioning the breakdown structure. A baseline is a tagged version of the entire breakdown or of a branch of the breakdown. To manage previously created baselines, click on the baseline icon in the project menu. The baselines form lists baselines and allows the user

- - to display and browse a baseline in the breakdown structure pane,
- - to export a baseline as an EDMtruePLM specific ISO 10303-239 P21 file including all files and a separate reference data P21 file (for details of the contents of the resulting zip-file, see chapter 4.1.5 “Download STEP pack”) and
- - to delete baselines.

Figure 84. Baseline tab

Note: Baseline Export depends on the access rights assigned to the user. Project manager & admin have access to the baseline export while project member or subdomain leader don’t have right to export the baseline. However, Project member & Subdomain leader can view the baseline.

3.6.4 Search

Three search methods are available for finding breakdown elements / nodes and documents in the breakdown structure; see below Figure 85. There is a “Quick search” for both nodes and documents with more detailed search specifications.. A search will be executed on the selected product breakdown version. So, if you need to search within a specific version of the breakdown, you must select the version first. Start search by the search icon in the project menu at the top right corner, as shown below. The system allows ‘wildcard’ search for all types of searches.

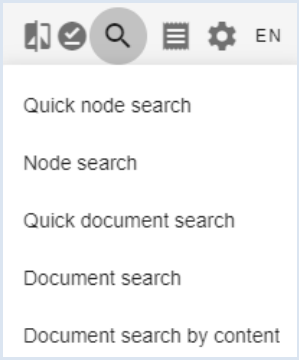


Figure 85. Search tab

3.6.4.1 Breakdown element

"Quick search" finds breakdown elements / nodes based on the following filters. As its name indicates it is quicker than "Node search", but not as sophisticated.

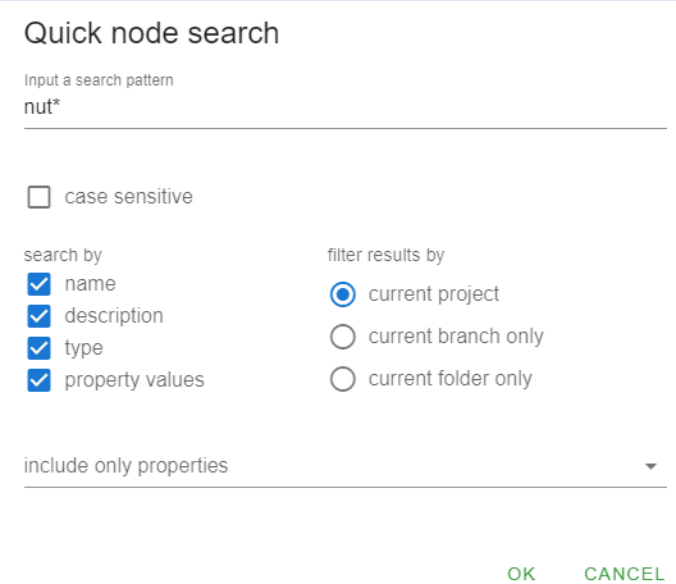


Figure 86. Quick Node Search

"Node search" offers additional filters (see Figure 87). For large data sets it is time-consuming.

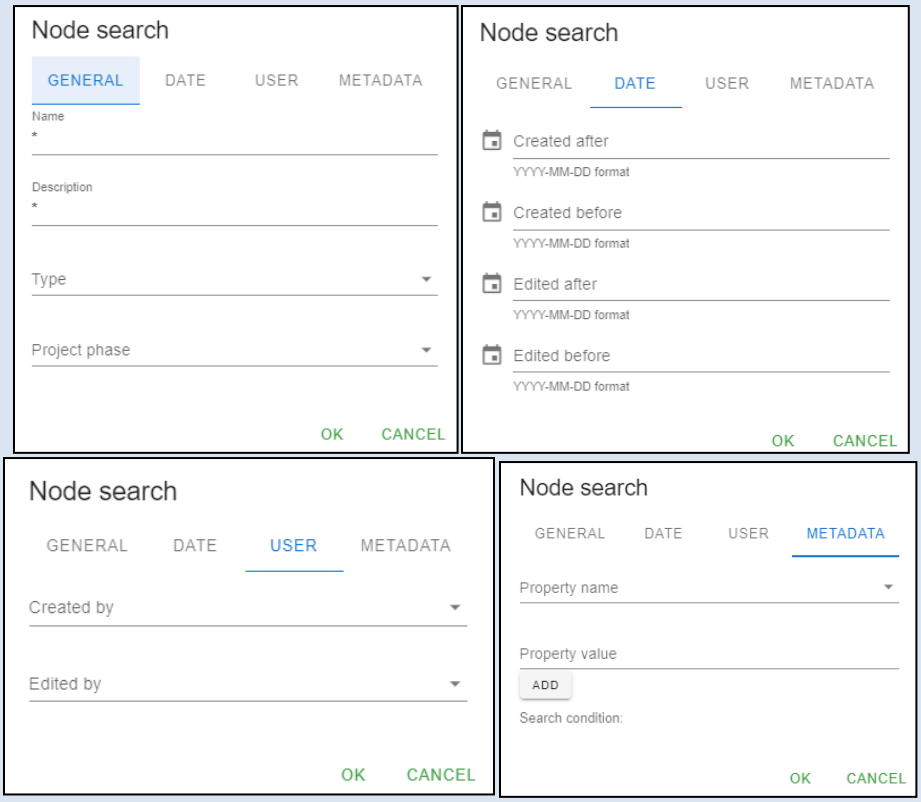


Figure 87. Node search

The different "Node search" tabs are explained in the following bullet points.

i. General tab:

All filter items in this tab are system properties of breakdown elements.

- Name: Enter breakdown element name. The following wildcards may be used in the text field.
  - @ - Matches any letter
  - ^ - Matches any upper-case letter
  - ? - Matches any character
  - & - Matches reminder of string
  - # - Matches any digit
  - $ - Matches a substring terminated by a space character or end-of-string
  - - Matches any number of characters
  - \ - Begins a pattern escape sequence
  - ! - Negation character (used with the other characters).
- Description: a string within the node property "Description".
- Type: one of an enumerated list of element type values can be selected. If no type is specified, all types of elements will be returned.
- Project phase: one of an enumerated list of phase values.

ii. Date tab:

All filter items in this tab are system properties of breakdown elements.

- Created after: Search for nodes created after the given date.
- Created before: Search for nodes created before the given date.
- Edited after: Search for nodes modified after the given date.
- Edited before: Search for nodes modified before the given date.

iii. User tab:

All filter items in this tab are system properties of breakdown elements.

- Created by: Search for nodes that were created by the selected user.
- Edited by: Search for nodes that were last modified by the selected user.

iv. Metadata tab:

The filter items in this tab are user-defined properties of breakdown elements. Several properties may be included in the same filter pattern.

- Property name: The name of a user-defined property.
- Property value: The value of the given property; for string type properties wildcards may be used.
- Add: Apply the given property name and value pair to the search. After the first property name and value pair others may be added; all such filter pairs are joined by a logical "AND".

3.6.4.2 Data files

"Quick document search" finds documents based on the following filters; see below Figure 88. As its name indicates it is quicker than "Document search", but not as sophisticated.

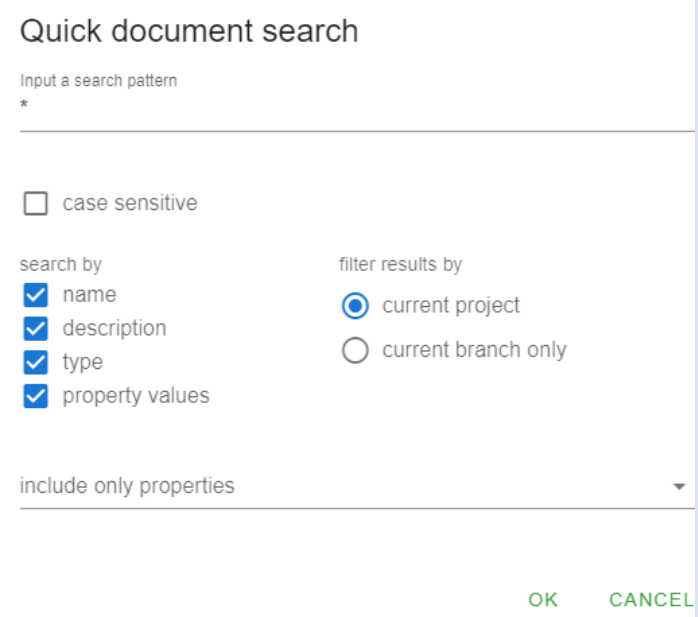
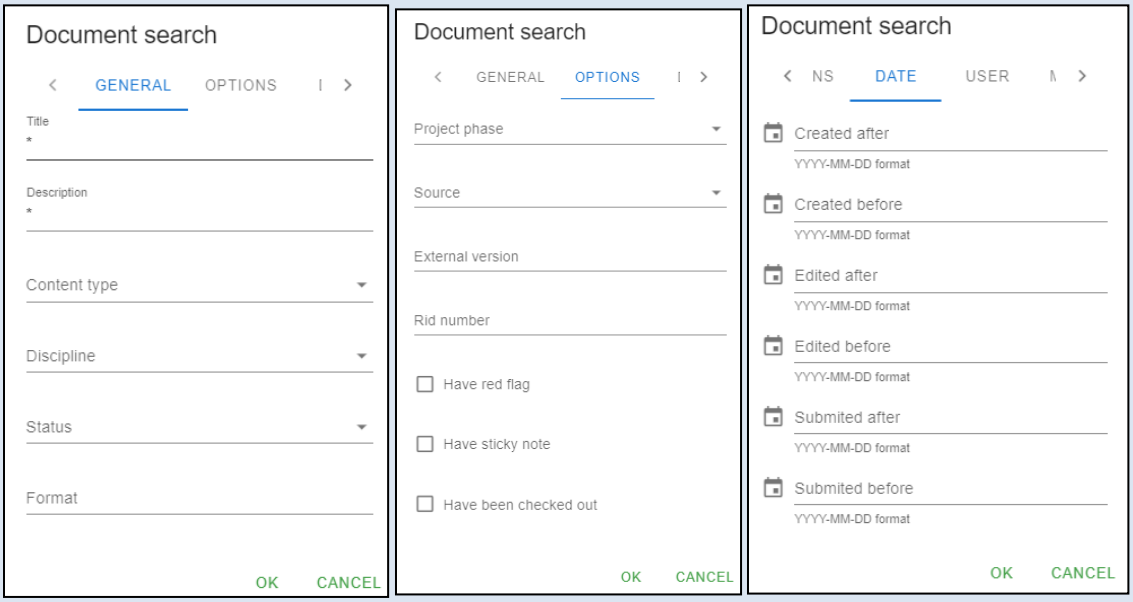


Figure 88. Quick document search

"Document search" can be used for larger set or specific content. Use the form shown in the below Figure 89, to find documents in the breakdown structure. The form consists of five tabs, which are explained in the following sub-sections.

Searches are done on the selected product breakdown version. So, if you need to search within a specific version of the breakdown, select that version first.

"Document search by content" focused on the search with available minimum details by using wildcard search.



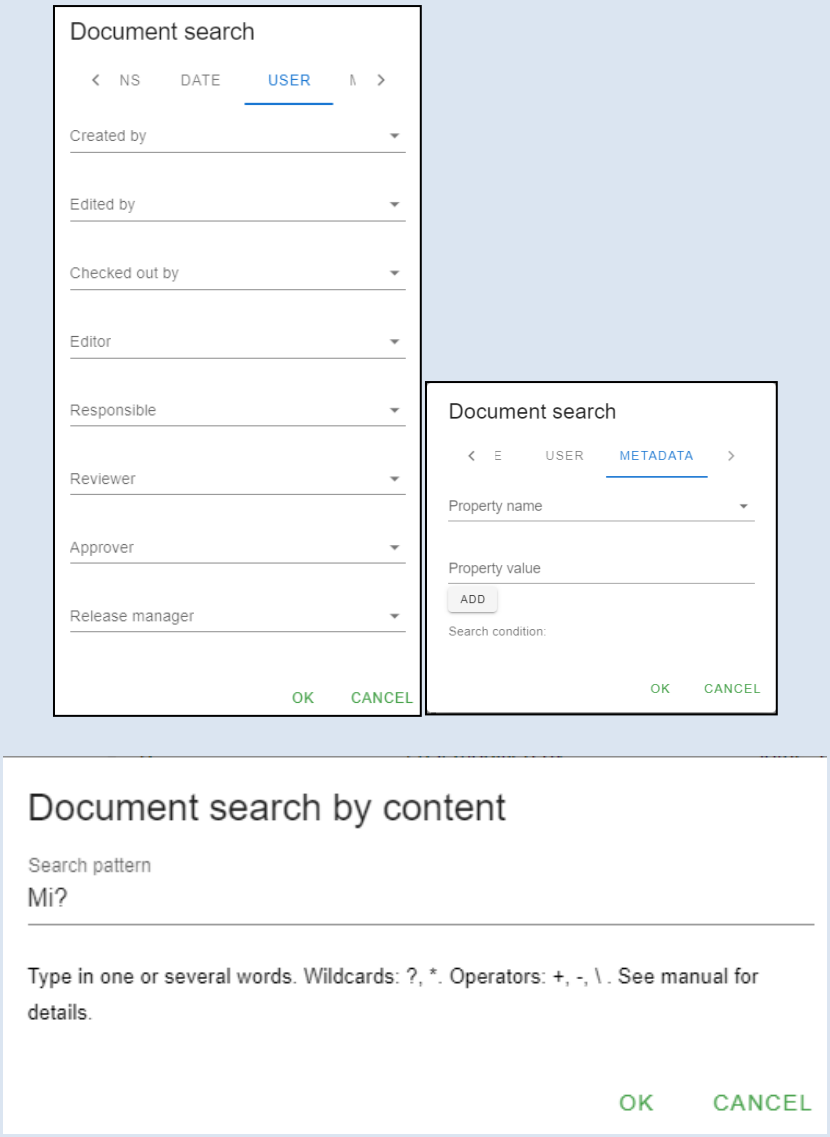


Figure 89. Document search and by content

i. General tab:

All filter items in this tab are system properties of documents.

- Title: Document title. The following wildcards may be used in this text field.
  - @ - Matches any letter
  - ^ - Matches any upper-case letter
  - ? - Matches any character
  - & - Matches reminder of string
  - # - Matches any digit
  - $ - Matches a substring terminated by a space character or end-of-string
  - - Matches any number of characters

  - \ - Begins a pattern escape sequence
  - ! - Negation character (used with the other characters).
- Description: a string within the document property "Description".
- Content type: one of an enumerated list of document "Type" values can be selected. If no value is specified, all types of documents will be returned.
- Discipline: one of an enumerated list of document "Discipline" values can be selected. If no value is specified, documents of all disciplines will be returned.
- Status of the document: one of an enumerated list of document "Status" values can be selected. If no value is specified, documents with all status values will be returned.
- Format: specify a file extension.

ii. Options tab:

All filter items in this tab are system properties of documents.

- Project phase: one of an enumerated list of project phases (defined as reference data).
- External version: the version of the document given by an external system.
- Rid number: the identifier of a Review Item Discrepancy (issues follow-up)
- Source: one of an enumerated list of origins of a document (defined as reference data).
- Have red flags: search for documents that have red flags.
- Have sticky notes: search for documents that have sticky notes.
- Have been checked out: search for documents that have been checked out

iii. Date tab:

- Created after: search for documents created after the given date.
- Created before: search for documents created before the given date.
- Edited after: search for documents modified after the given date.
- Edited before: search for documents modified before the given date.
- Submitted after: search for documents submitted after the given date.
- Submitted before: search for documents submitted before the given date.

iv. User tab:

- Created by: search for documents that were created by the selected user.
- Edited by: search for documents that were last modified by the selected user.
- Check out by: search for documents that are checked out by the selected user.
- Editor: search for documents that have the selected user as their Editor.
- Responsible: search for documents that the selected user is responsible for.
- Reviewer: search for documents that the selected user is the reviewer of.
- Approver: search for documents that the selected user is the approver of.
- Release manager: search for data files that the selected user is the release manager for.

v. Metadata tab:

- Property name: The name of the user-defined property.
- Property value: The value of the given property; for string type properties wildcards may be used.
- Add: Apply the given property name and value pair to the search. After the first property name and value pair others may be added; all such filter pairs are joined by a logical "AND".

3.6.4.3 Search result tabs

The search results are shown in table form for all types of searches. Tables can be sorted or otherwise refined based on user needs by selecting dots at each header column.

i. Document search result tab

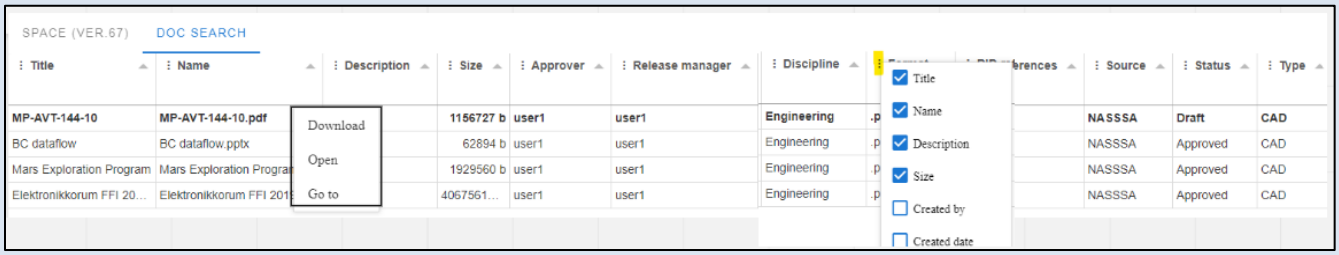


Figure 90. Document search result

The following functions are available for each document within the search result tab.

1. Download: downloads the document.
2. Open: opens the document.
3. Go to: opens the position of the document within the breakdown structure.

In addition, by clicking on the search result, the properties of that document are loaded into the document properties window.

ii. Node search result tab

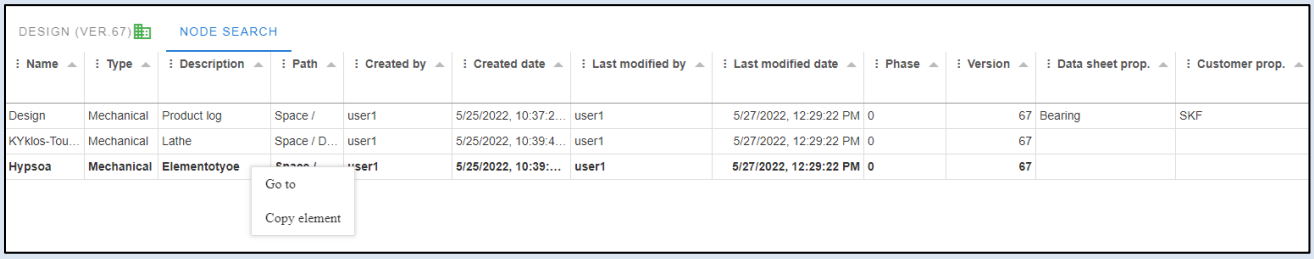


Figure 91. Node search result

By clicking on the search result menu, you can go to the position of the node or copy node element within the breakdown structure as shown in Figure 91 above. In addition, by clicking on a search result item the properties of the selected node are displayed within the breakdown properties window.

3.6.4.4 Sorting & exporting search results

Search results can be sorted alphabetically by clicking on the arrow symbol to the right of the column title. Functions to export a search result list into a format, such as, Excel, CSV or JSON, are highlighted in Figure 92, below. Search result tabs can be closed by using the "Close tab" option shown below.

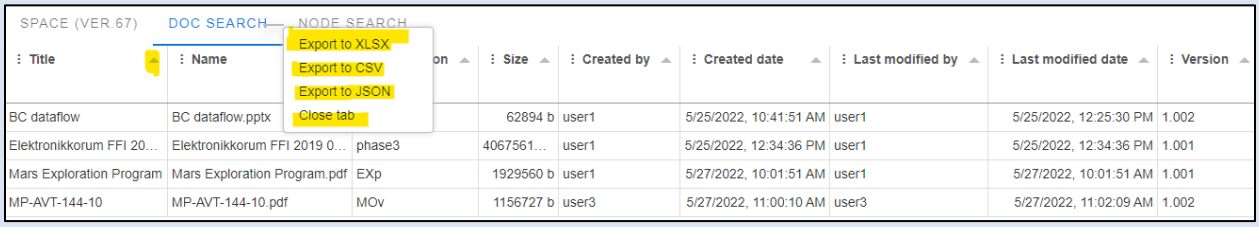


Figure 92. Sorting & Exporting search result

3.6.5 Catalogues

Notification, Products and Organizations are managed in the Catalogues menu item. Products and Organizations are considered reference data. The notification tasks are managed using Actions and Triggers tabs (Figure 93).

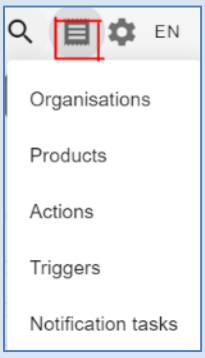


Figure 93. Catalogues

3.6.5.1 Notification

The notification feature allows to notify a user about any changes in breakdown elements. It also allows to send the notification to an individual user about the specific task or more info relevant to the node. This feature provides the capabilities to respond to a linked notification by simply replying to the received message.

The notification is divided into the following categories:

1. For a Task or info type
2. For the change in the Breakdown element properties

The below sections guide you towards that.

3.6.5.1.1 For a Task or info type

To send messages about the information about the general task, one can opt for the info type. In this a user can share more details about a task which is not directly related to any specific node. To create such a message, the user needs to select the send message icon as shown below. Then select the type of message to 'info' from the drop-down list and complete the form with necessary details. Please note, to receive a message user must be added to the trigger list as explained in the 'Step 3' in the below section. While for the message other than 'info' relates to the specific task in the breakdown tree and also assigned with the status and deadline to track the changes; see below Figure 94, **Error! Reference source not found.**, Figure 96 & Figure 97.

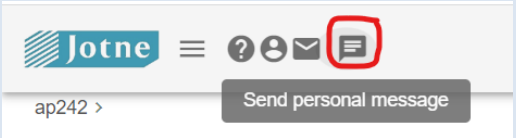


Figure 94. Send personal message

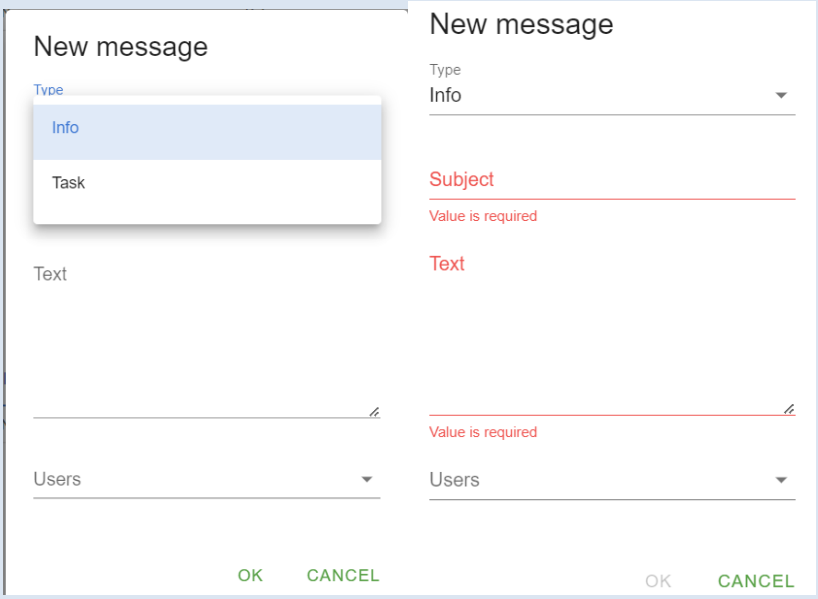


Figure 95. Message Form Info Type

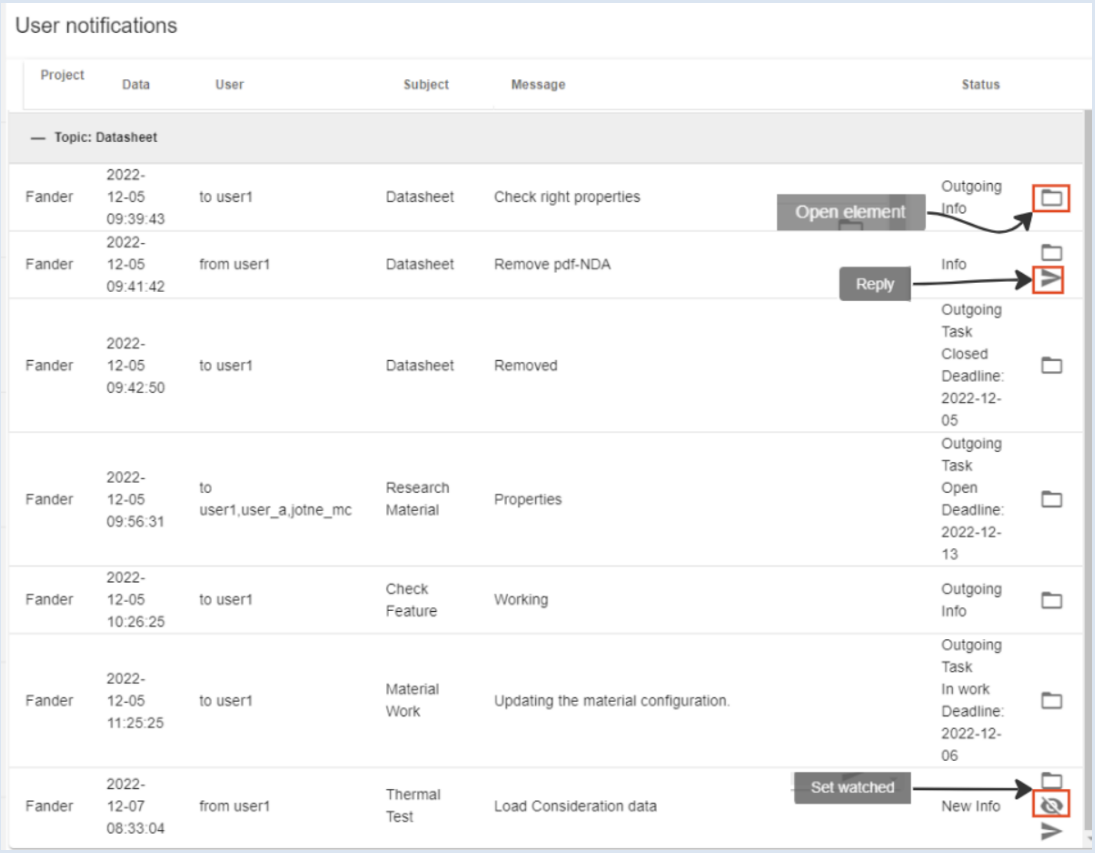


Figure 96. User Notification

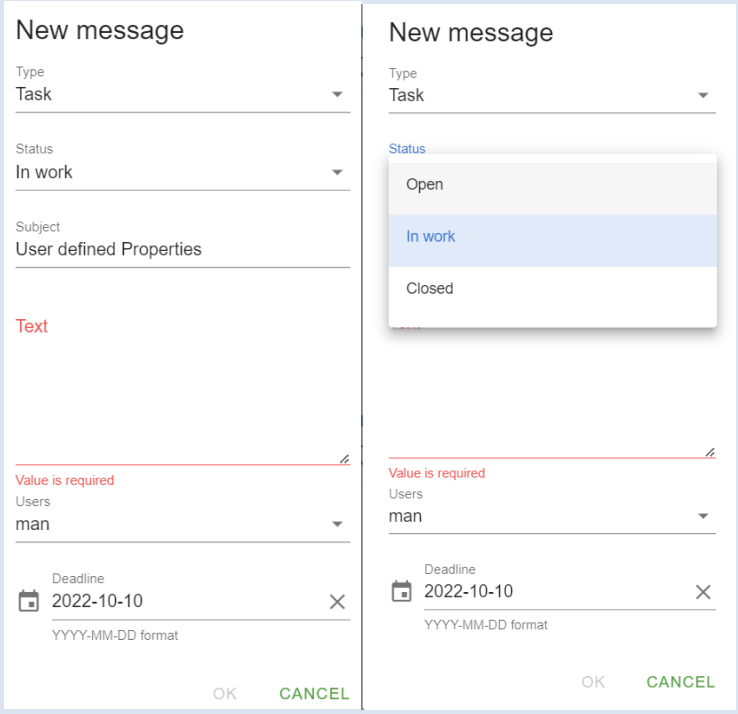


Figure 97. Message Form Task type

3.6.5.1.2 For the change in the Breakdown element properties

If a user needs or wants to be notified of any changes or updates of a breakdown element properties, they can select this category. Follow the steps, below, to create a notification based on this:

Step 1. First link a user with a valid e-mail address to the project using the admin login.
Step 2. Reference data - go to the 'breakdown element property' that requires a notification (see below Figure 98 e.g., HWitem - FinancialState).

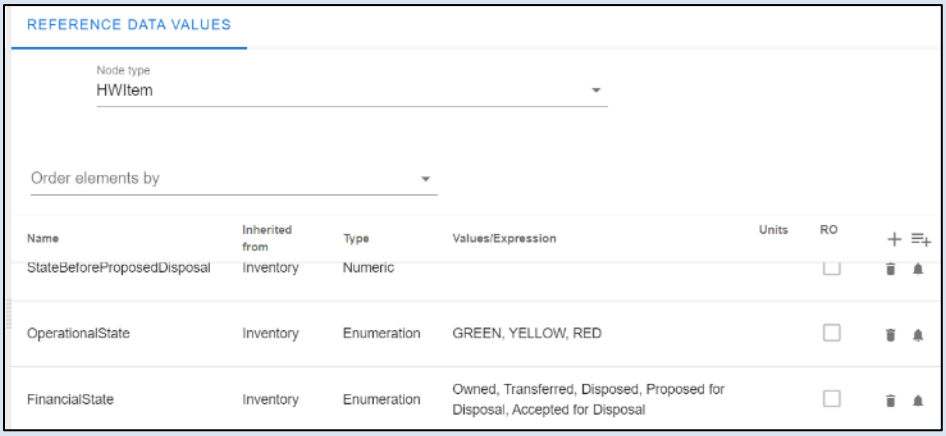


Figure 98. Notification

Step 3. Click on the bell icon (see Figure 99) to open the input form to add a notification.

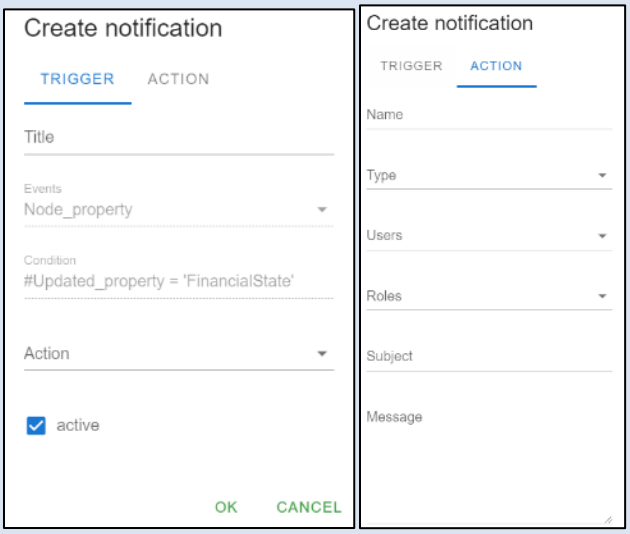


Figure 99. Notification Menu

a. The form contains a trigger-tab and an action-tab.

b. If there are no suitable actions that may be reused, fill first the action-tab with all details such as user, subject and notification text. The text may include the following DB parameters, which need to be prefixed by the character "#":
   i. Timestamp (in milliseconds), Date (creation date), Actor, Revision, Type, Node_path, Name, Description, Updated_property, Property_value_before, Property_value_after.
   ii. Example notification text:
   Subject - Change of property value for #Node_path"
   Message - "Property #Updated_property was updated from #Property_value_before to #Property_value_after."
   iii. The valid parameters are listed in the reference data section, as shown below:

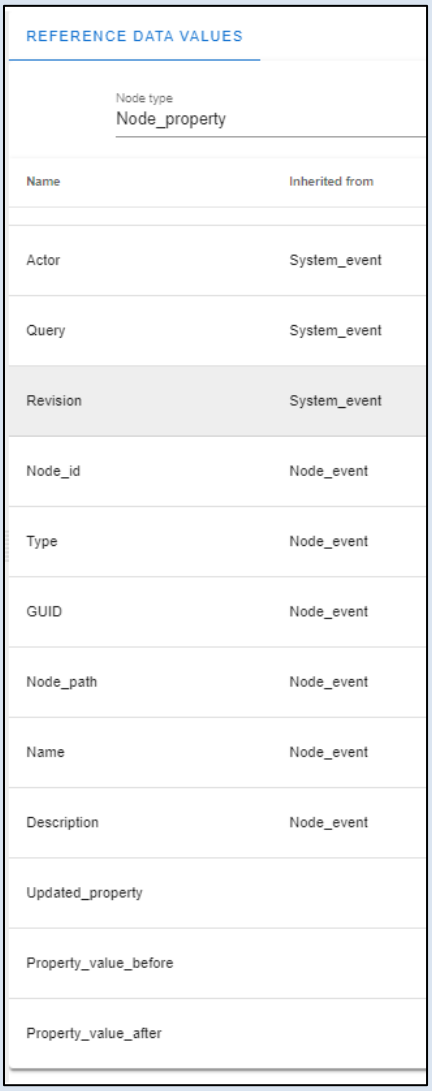
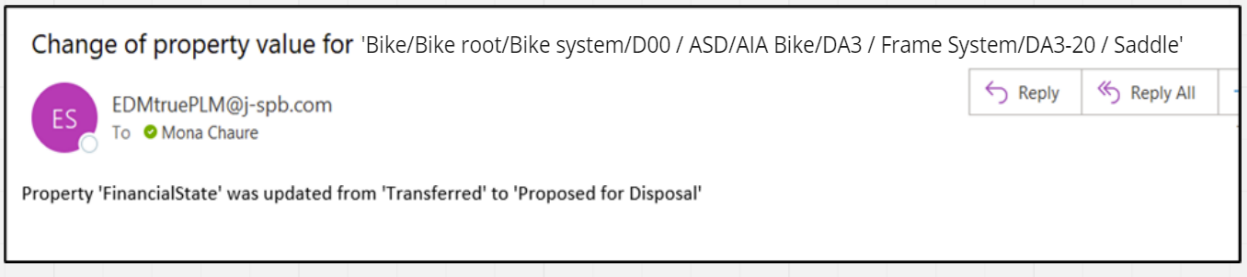


Figure 100. Reference Data for Notification

c. Fill in the Trigger-tab: To include the newly defined action from the Action-tab choose 'New action' from the Action dropdown list. Else select any previously defined action.

Step 4. A change of value in the Inventory HWItem property “Financial state” (from e.g., “Transferred” to “Proposed for Disposal”) will now trigger a notification to the assigned users.



3.6.5.2 Product

This version of EDMtruePLM offers support for representing products and their relations to breakdown elements. Types of products may be created and may then be pasted into one or several breakdown elements.

Before creating products, confirm that reference data have been defined for “Product element type”, “Product stages” and “Product domains” in the reference data GUI. To add a product, click on the +-sign as shown in Figure 101.

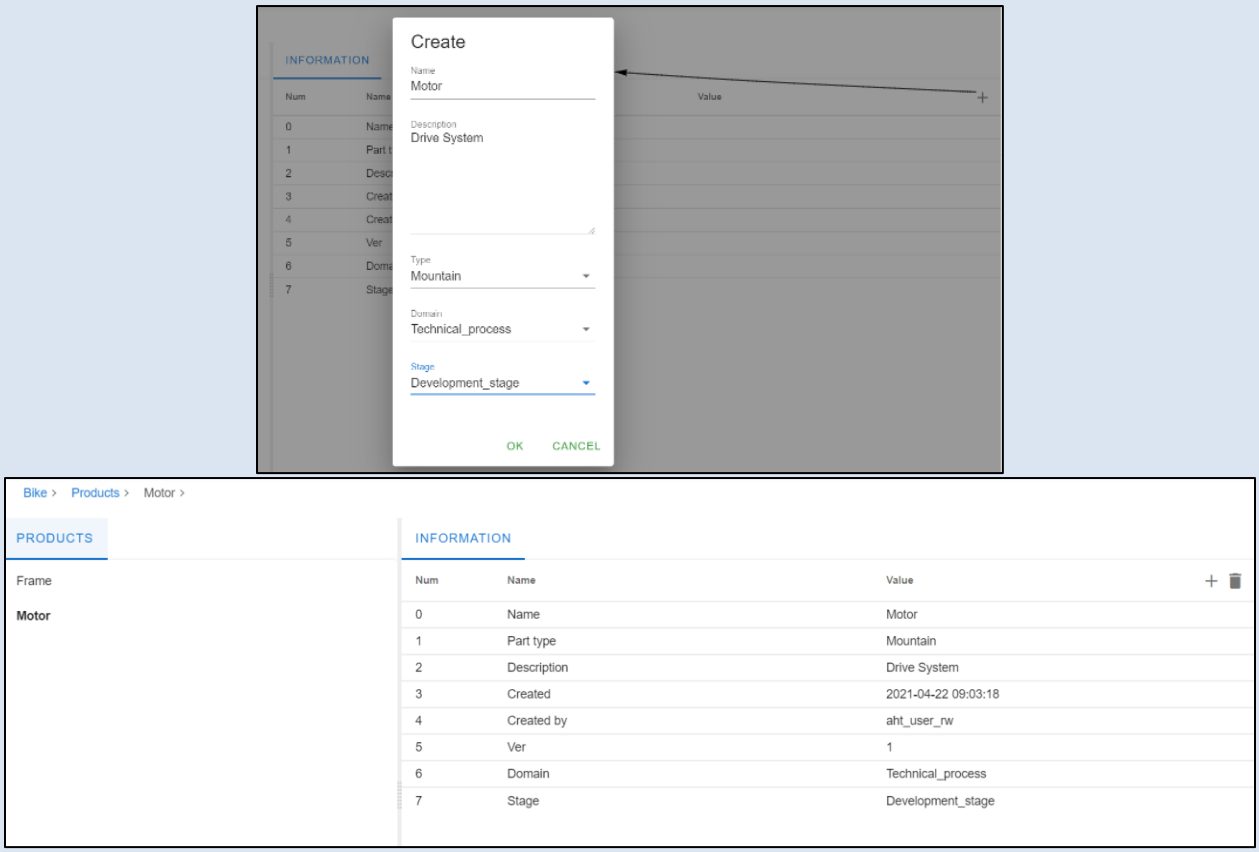


Figure 101. Product menu

Property values cannot be changed after product creation; delete and re-create instead.
For relating products to breakdown elements, see in chapter 3.4.2.3.6 Put Product.

3.6.5.3 Organisation

To create an organisation, select “Organisation” from the Catalogue menu, click on the +-sign and fill in the form.

The form consists of information such as Organisation name, Unique id (Uid) and Address as shown below in Figure 102. Property values may be changed after creation.

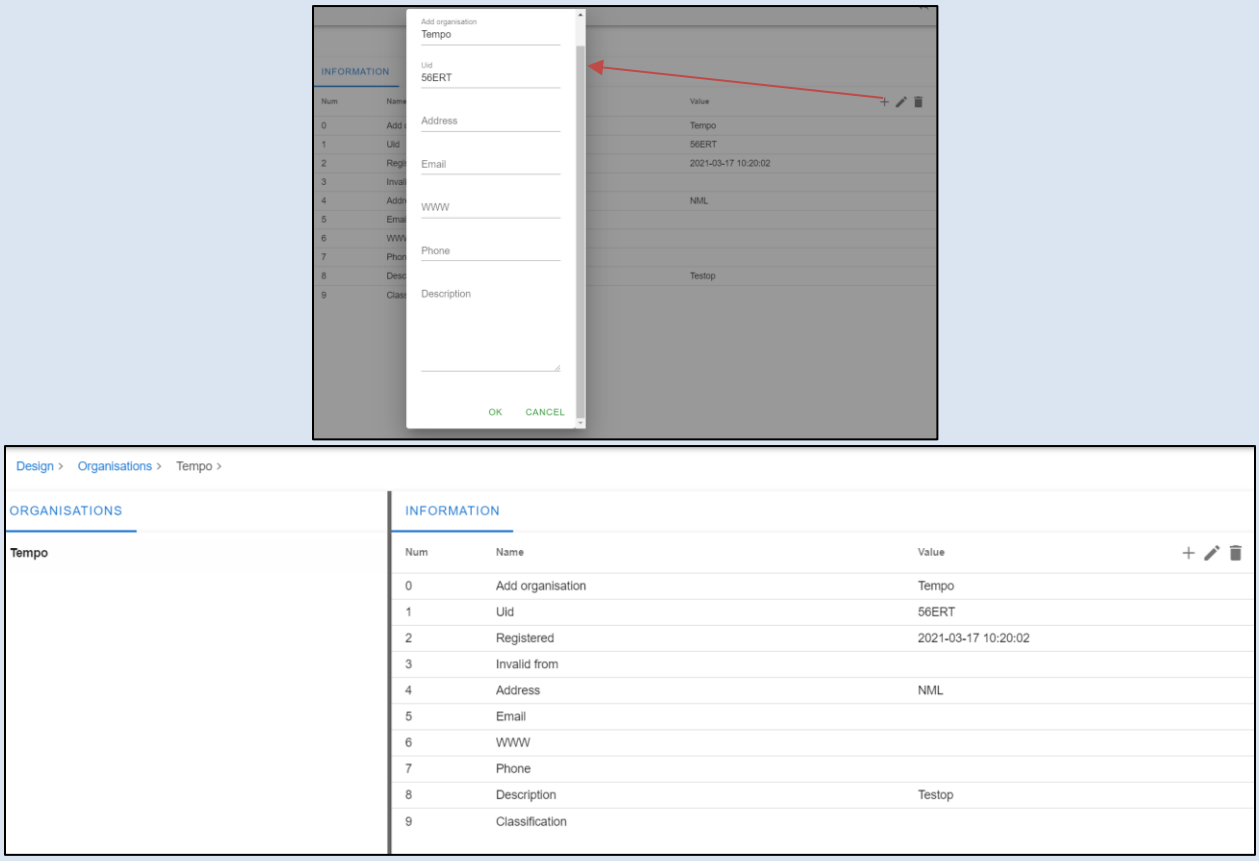


Figure 102- Organisation menu

For relating organizations to breakdown elements, see in chapter 3.4.2.3.5 Add Organisation.

3.6.6 Reference data definitions

The "Reference data" page allows authorized users to manage the metadata of the project. See column “RDL manipulation” in Table 8 for authorized types of users.

3.6.6.1 Opening the RDL page

To manage project reference data, click on the reference data icon in the project menu.

3.6.6.2 Reference data GUI

The "Reference data" page allows the user to manage project metadata. These metadata are a powerful method to configure the system for different uses and processes.

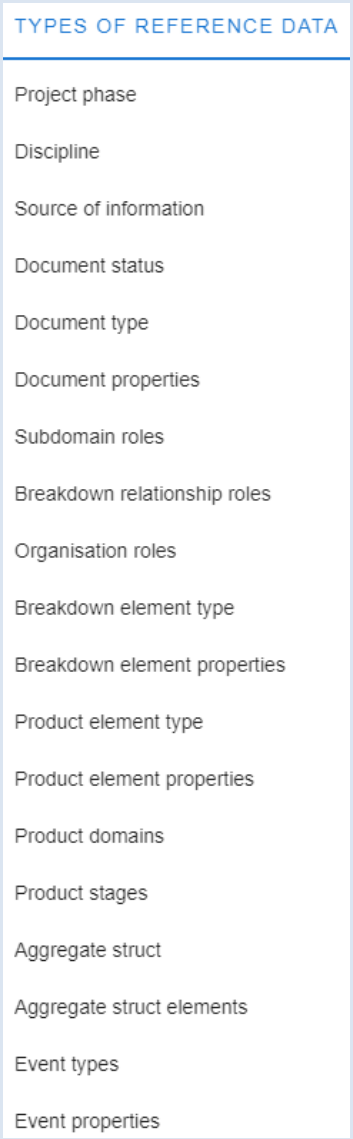


Figure 103. Reference data

The system allows the user to define values for the following metadata types (refer Figure 103):

- Project phase: stage in an activity, e.g., project lifecycle stages
- Discipline: type of engineering, e.g., Electrical, Instrumentation, Mechanical ... .
- Source of Information: origin of a document, for example, external and internal
- Document type: type of information in a document, e.g., CAD, Design, Requirement specification ...
- Document status: the level of progression of a document in its lifecycle workflow, e.g., Draft, Approved, Completed, Ready to review, Sent for approval ...
- Subdomain roles: level of access of a project member to information in a subdomain
- Breakdown relationship roles: types of usages of one breakdown element by another breakdown element, such as, Supplied part, Requirement and Subordinate

- Organisation roles: types of usages of organisations by breakdown elements, e.g., Contractor, Security classifier, Creator, Operator…
- Breakdown element types: classes of nodes in a tree structure, e.g., subsystem, system, unit, module, product definition, requirement ...
- Breakdown element properties: user-defined attributes of a breakdown element and its subtypes, e.g., length, breadth and height
- Document properties: attributes that are added to documents
- Product element type: category of a product, e.g., aircraft, car, ship and pump
- Product element properties: attributes that are added to products
- Product domains: engineering discipline; a product may combine several disciplines, such as, electrical and mechanical
- Product stages: lifecycle phase of a product, e.g., design, analysis and manufacturing
- Aggregate struct: a type of structure that is an array of sensor values
- Aggregate struct elements: the members of an “Aggregate struct” and their metadata.
- Event types: used to define type of notification
- Event properties: attributes of notification types

3.6.6.3 Adding reference data

To change or add metadata to the system, use the icon in a selected reference data type menu. All metadata except for Breakdown properties, Document properties, Product properties, Requirement properties and Aggregate struct elements define new metadata by adding a new value to the form below.

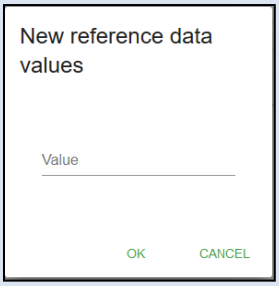


Figure 104. Add reference data

As an example, the project manager can define as "Source of information" different types of source categories, e.g., internal data, customer data, public data etc. .

3.6.6.4 Deleting reference data

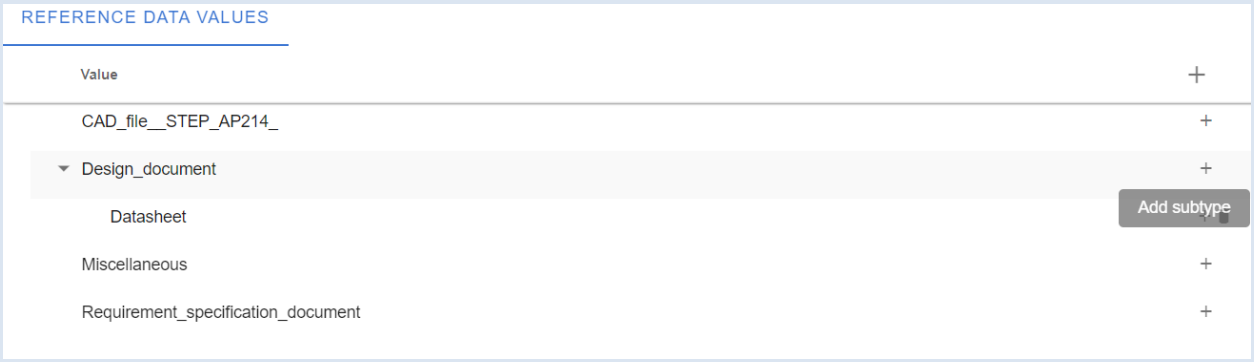
The system will allow metadata to be removed only if it is not used in a project. Metadata that can be removed have an icon in front of them. The remove icon can be used to remove the metadata (see Figure 105).

Figure 105. Delete reference data

Note: The system will not allow a metadata to be deleted if it is in use, this will guarantee that the data will not become corrupted by removing the used metadata.

3.6.6.5 Adding document property

This item allows the user to extend the EDMtruePLM system by creating additional properties (metadata) for documents. To add a new document property, click on the icon and fill in the form; this is similar as for breakdown properties, see Figure 106 below. The smaller +-signs enable the definition of subtypes of existing document types, see below Figure.



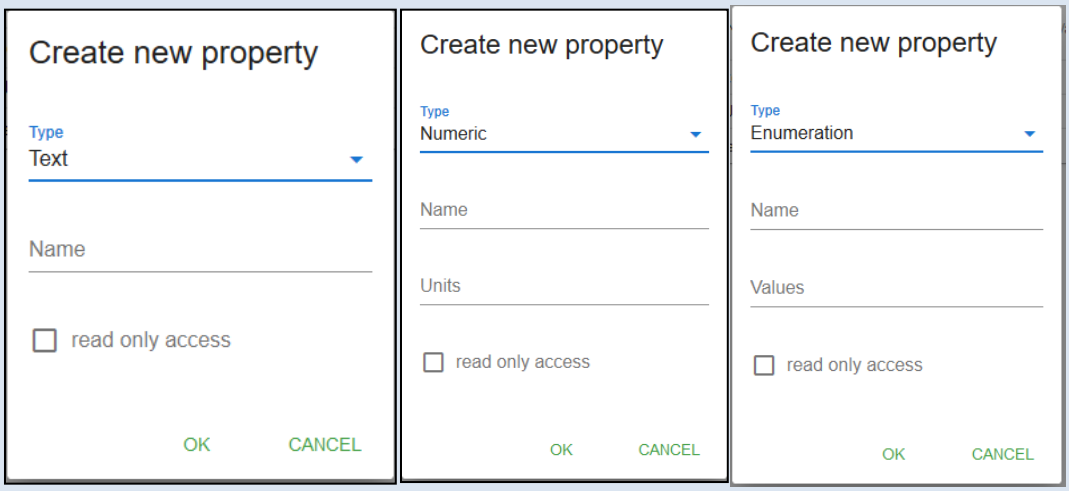


Figure 106. Add document property

Document properties may be of different data types, such as text and numbers. The input forms vary depending on the requested data type; see Figure 106, above.

The property types are explained in Table 5, below.

The "read only access" option allows a user to create properties the values of which can only be changed by the project manager.

Table 5. Document property types

Type	Definition
Text	String value property
Numeric	Numeric value property. Provide a name and a unit of measure
Date	Date time values
Enumeration	An enumeration type property consists of a list of pre-defined values. Provide the property name and the values of the enumeration; values are separated by comma.
Boolean	True/false property

3.6.6.6 Adding breakdown element type

Breakdown element types may be defined as a class hierarchy where subtypes inherit the properties of their parents. The process starts with the interaction in Figure 107. Selecting the upper-most +-sign will trigger the user interaction in Figure 108 to add a new top level breakdown element type. The smaller +-signs enable the definition of subtypes of existing element types.

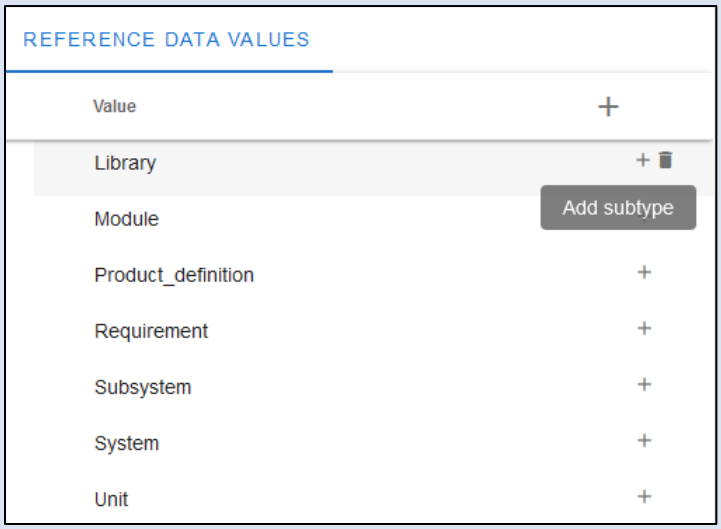


Figure 107. Add breakdown element subtype

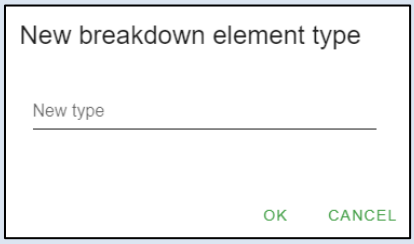


Figure 108. Add breakdown element type

3.6.6.7 Adding breakdown element property

Breakdown properties are defined per type of breakdown element; subtypes inherit from their parents.

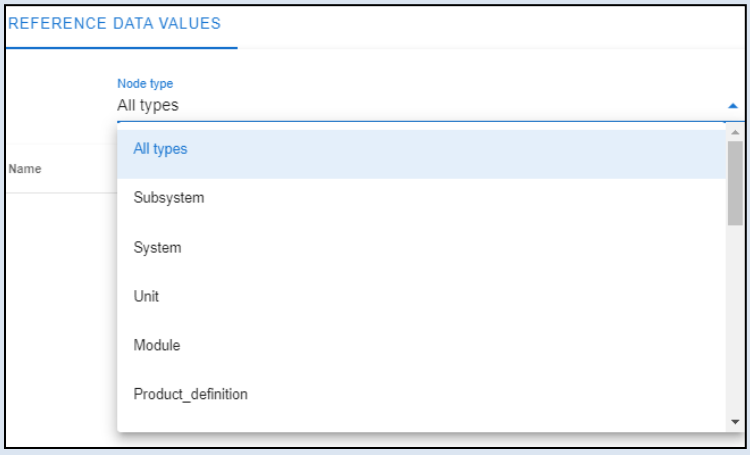
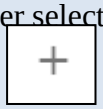


Figure 109. Add property for specific breakdown element type

After selecting the breakdown element type that the property shall become a part of click on the  icon to create a breakdown property; see Figure 110, below.

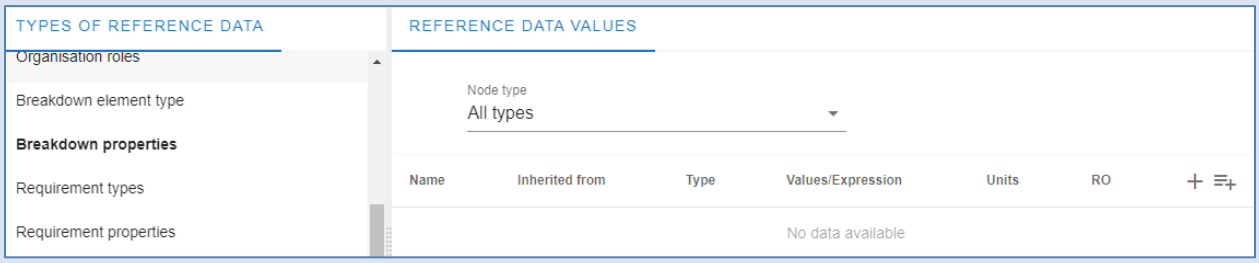


Figure 110. Breakdown property input form

Breakdown properties may be of different data types, such as text and numbers, similar to the document properties described above. The input forms vary depending on the requested data type; see Figure 111, below.

The property types are explained in Table 6, below.

The “read only access” option allows a user to create properties the values of which can only be changed by the project manager.

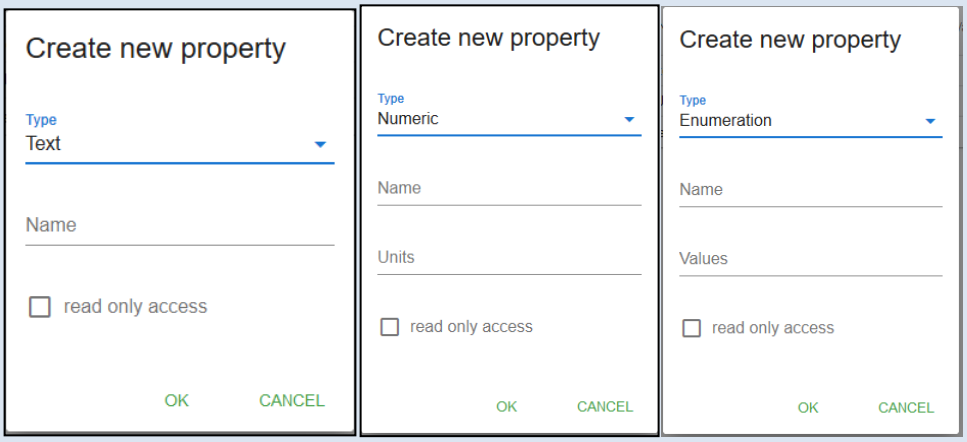


Figure 111. Create new breakdown element property

Table 6. Breakdown element property types

Type	Definition
Text	String value property
Numeric	Numeric value property. Provide a name and a unit of measure
Date	Date time values

Type	Definition
Enumeration	An enumeration type property consists of a list of pre-defined values. Provide the property name and the values of the enumeration; values are separated by comma.
Boolean	True/false property
Serial	This is a sequence number. When a node with such a type of property is created, the system checks for the last given sequence number, increases it by 1 and assigns it to the newly created node. If the node is the first occurrence of its type, the value 1 is assigned.
Formatted serial	This is a text property that represents a sequence number with some textual prefix/suffix. An example is 'SN:00012345', where 12345 is a value of a property of type sequence (see above) or an implicit sequential number calculated like the sequence number, but used only in the context of the property typed formatted_serial. The format of the property representation must be stored in a parent property named "<name_of_child_property>-format". The sequence number must follow the formatting code of the C programming language; the default formatting value is '%010d' (to produce strings like '0000000012' or '0000065412'). E.g. a child breakdown element has a property by name 'XXX' and of type formatted_serial. Its parent is given the text property 'XXX-format'. In case XXX-format = 'S/N:%06d' the 123rd instance of the child node receives the property name and value pair XXX = 'S/N:000123'. Note: formatted_serial as well as sequence type properties use node type specific counters that are valid over the whole target model and not only in the domain of a single parent.
Counter	This is a numerical counter of the children of a parent node. Child nodes with a property typed as Counter will at creation time be assigned a numeric value that is the number of children of their parent node.

Type	Definition
Formatted Counter	This is a textual representation of the Counter property type, alike Formatted_serial described above. The format of the representation must be stored as a parent node property. The format is composed of two pieces: prefix (value of parent property with the same name) and format of counter representation (should be stored in parent property named '<child_property_name>-format'). The property can be used to generate hierarchical counters, like 'XXX.012.001.01' for child of node 'XXX.012.001'... E.g. a parent has the property 'ID' of type formatted_counter with value 'XXX.012' and textual property 'ID-format' = '%s.%02d'. A newly created child node (7th one) receives 'ID' = 'XXX.012.07' as well as a copy of 'ID-format' = '%s.%02d' of its parent. Note 1: <child_property_name>-format may be undefined. Note 2: The default format string is '%s.%3d'.
Extended Counter	This is a reserved type of counter for future use. It works exactly like formatted_counter with the only difference that the default format value is '%s.%03d'

3.6.6.8 Adding derived property

A derived property calculates its value from other properties.

The icon is for the creation of the derived property and is located in the breakdown element and document property creation area; see Figure 110, above.

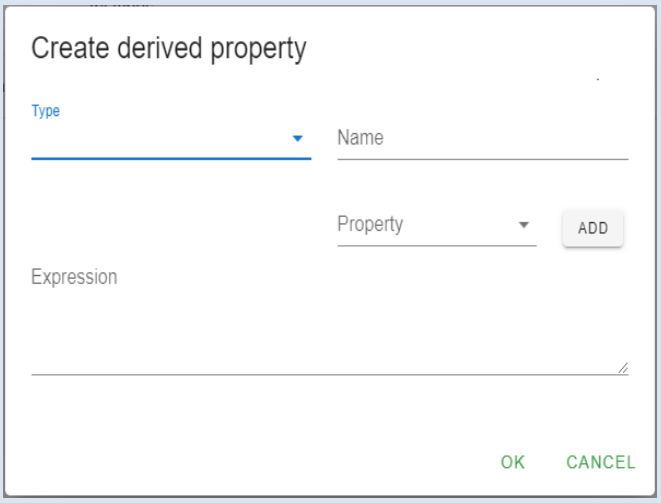


Figure 112. Create derived property

Select the property data type from the drop-down list and give a name to the new derived property (see Figure 112). To create the derived property value pattern, add the properties from the *Property* drop-down list and click on the *Add* button. The drop-down list contains all local properties, that is, only the properties of the current document or breakdown element type.

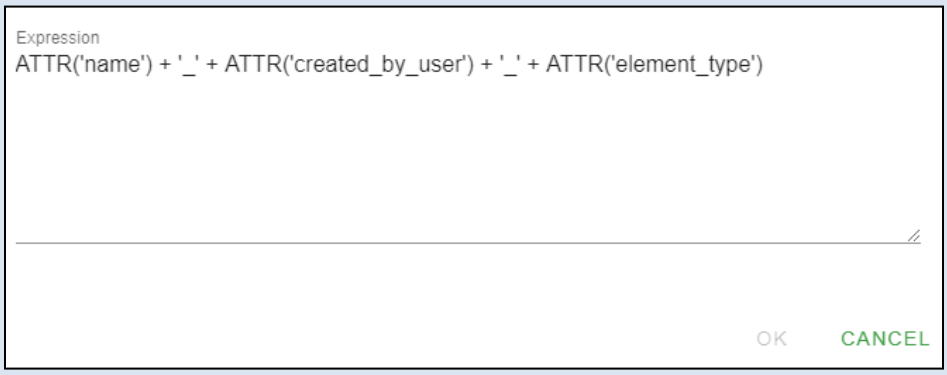


Figure 113. Derived property example

Each selected property will be shown as an argument to the *ATTR ()* method. To add static text or a delimiter between the values, place the text between single quotes, see Figure 113 above. To create the derived property, click the OK button.

3.6.6.9 Adding an aggregate structure

An aggregate structure is an array of value sets of the same type. It is intended to store streamed sensor values.

To add an aggregate structure, first create an aggregate structure type and afterwards add types of elements to it. To create a new structure type, click on the **"Aggregate struct"** in the reference data menu and create a new structure type by clicking on the icon. After having created the structure, go to **"Aggregate struct elements"** from the reference menu and select the new structure type from the drop-down list.

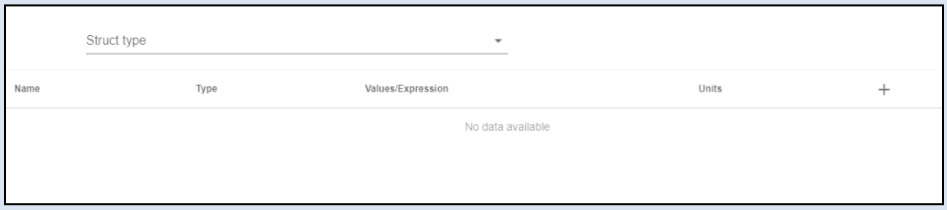


Figure 114. Aggregate structure

Use the icon to add new elements to the structure.

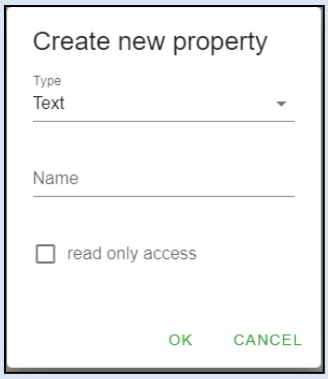


Figure 115. Create new aggregate property type

You can add the following types of elements to the structure:

Table 7. Aggregate property types

Type	Definition
Text	String value property
Numeric	Numeric value property. Provide a name and a unit of measure
Date	Date time values
Enumeration	An enumeration type property consists of a list of pre-defined values. Provide the property name and the values of the enumeration; values are separated by comma.
Boolean	True/false property

3.6.6.10 Adding a filter key to aggregate data

To be able to filter the aggregate data, EDMtruePLM provide the functionality to assign an element in the aggregate structure as primary key. To assign an element as the key to an aggregate type, click on the icon in the same row (see Figure 116).

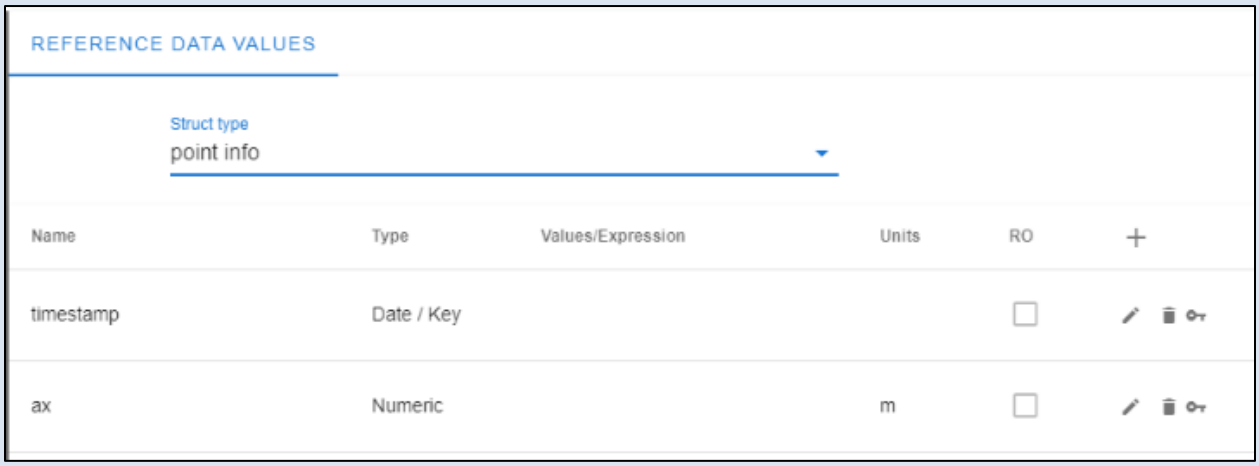


Figure 116. Aggregate type

After selecting an element as the key to the aggregate type, the word "**Key"** is added to the "Type" string of the element.

3.6.6.11 How to upload existing sensor data in TruePLM

The service
POST **/api/bkd/aggr/{repository}/{model}/{node}/{prop}/{token}**
should be used for the uploading of the sensor data with the following parameters

Name	**Description**
file * **file** *(formData)*	Data file
model * **string** *(path)*	Model name - the project name should be used here
node * **integer** *(path)*	Breakdown element instance ID - can be seen in the URL for the currenly opened node in the GUI
prop * **string** *(path)*	Aggregated property name (URN) - usually has the format - "urn:rdl:" + project name + ":" + name of the property
repository * **string** *(path)*	Repository name - "TruePLMprojectsRep" should be used here
token * **string** *(path)*	Server connection token

Data file should contain data in the JSON format.
The example: [{"timestamp":"2020-11-23T15:24:21.533","Bool value":"False "}, {"timestamp":"2020-11-23T15:24:28.100","Bool value":"True"}, {"timestamp":"2020-11-23T15:24:28.600","Bool value":"True"}]
So, the user can upload several records in one service call but only for the one sensor node. In case of the several sensor nodes, the operation has to be repeated for each sensor node.
The simplest way to get information about all sensor nodes in the project is the quick search by the property 'serial number':
GET /api/bkd/q_search/TruePLMprojectsRep/{Project name}/{token}/?pattern=*&domains=PROPERTY&props=urn:rdl:{Project name}:serial number
The response will contain the list of the sensor nodes in the project. The node ID will be specified in the field `bkdn_elem_info/instance_id`.

3.6.6.12 Reference Data Uniqueness

The names of user defined properties are unique across a project; one cannot create already existing properties. If one tries to create a property that already exists, the system will return the following error message.

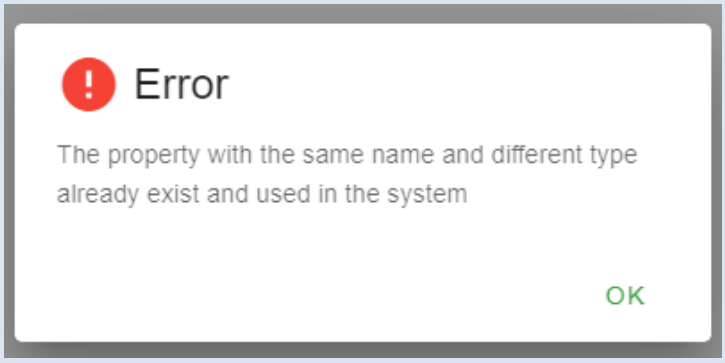


Figure 117. Reference Data Uniqueness

3.6.6.13 User Defined and System Properties sequencing (Order Elements by)

The user defined property sequence can be defined based on the needs of the specific project in other words the most important properties can be set first. With this, users can focus on the essential properties according to their requirements.
For example: In the figure, below, the user defined property can be set to appear first by drag and drop. The Plating property is moved down and Shell Material moved up.

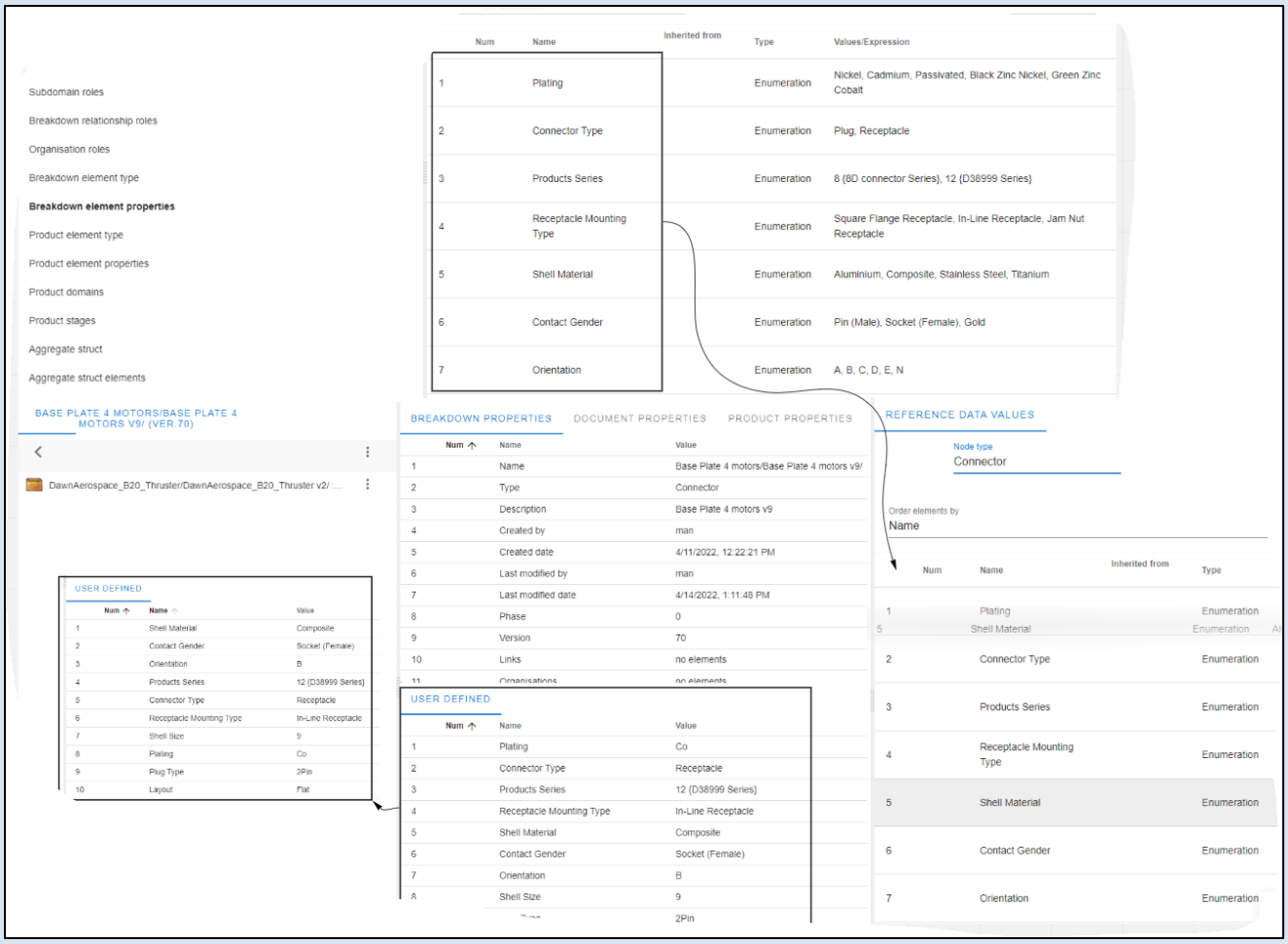


Figure 118. Drag & Drop property order

3.6.7 Language selector

The language selector allows the user to change the user interface language as shown below between English and Norwegian.

Figure 119. GUI language selection

3.7 Project root context menus

After creation of a project a root node is established by the application, which represents the top of the breakdown structure. This project root has different features compared to its child breakdown elements; these features are reflected by two context menus associated with it:

1. When the project is opened and the root node has not yet been selected, the menu in Figure 120 is available.
2. When the root node has been selected to show or to be able to create children, its context menu changes to the one in Figure 124.

The menu in Figure 120 becomes also available by selecting the left-most constituent of the breadcrumb path, which represents the project root.

3.7.1 Project root menu

This chapter describes the menu items of the project root as shown in Figure 120. Some of the items are applicable to an ordinary breakdown element when it plays the role of a child node.

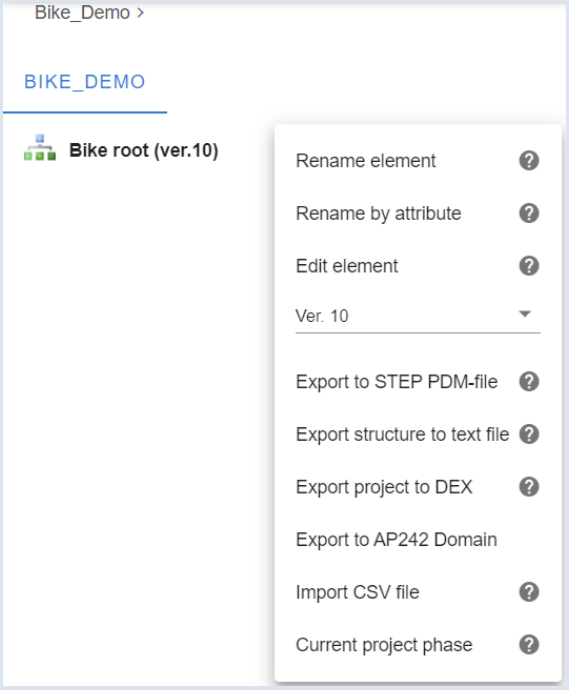


Figure 120. Initial project root context menu

3.7.1.1 Rename element

See description in chapter 3.4.2.2.1 .

3.7.1.2 Rename by attribute

See description in chapter 3.4.2.2.2 .

3.7.1.3 Edit element

See description in chapter 3.4.2.2.3 .

3.7.1.4 Versions

See description in chapter 3.4.2.1 .

3.7.1.5 Export to STEP PDM-file

The current model will be exported in the ISO 10303-242 MIM/P21-format.
AP242 contains the following two data models:

1. The schema “Ap242_managed_model_based_3d_engineering_mim_lf”, which is also referred to as the MIM-schema. Data according to this schema are usually in ISO 10303-21 (P21) formatted files.
2. The schema “managed_model_based_3d_engineering_domain_model”, which is also referred to as domain model. In edition 1 of AP242 this schema was called the business object model (BOM); this name is now deprecated.

This sub-section concerns the first item, the MIM-schema. The domain model import and export are described in 3.4.2.3.17 (import) and 3.7.1.8 (export).

The function is executed immediately after selection, without any further user interaction.
It creates a zip-file with the name of the current project postfixed by “_pdm”. The zip-package contains the following files:

1) A STEP P21-file according to the EXPRESS-schema “PDM_SCHEMA” with the file name <project_name>.stp . It contains the complete current version of the breakdown structure, but in the PDM_SCHEMA representation; that is, interpreted as an assembly structure using next_assembly_usage_occurrence. Documents are referenced to be in the folder where the file will be unzipped. Files of type PDM_SCHEMA are compliant with ISO 10303-242 and its predecessors (AP203 and AP214).
2) A STEP P21-file according to the EXPRESS-schema “ISO_12006_3_VERSION_14” with the name <project_name>_RDL.stp, which contains the Reference Data Library (RDL) associated with the breakdown structure.
3) A file with the name “TRUEplm_project_name.txt”. It contains only one line with the name of the project.
4) A file with the name “PLMexport_file_descriptions.txt”. It contains the mappings of the EDMtruePLM internal names of files referenced by entity instances in the PDM_SCHEMA P21-file to the original user-defined names.

5) All the digital files referred to by the breakdown structure. All REPRESENTING_DOCUMENT instances of the AP239/PLCS database are converted to PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS instances. PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS.id represents EDMtruePLM names of files in the package. The files are stored in the zip-file with these internal names. See "PLMexport_file_descriptions.txt" for the original user-defined names.

Such exported PDM-files may be used for local project creation (chapter 3.2.4.3), as initial population of new globally defined projects (chapter 4.1.4) and for import as a branch into an existing breakdown (chapter 3.4.2.3.16).

3.7.1.6 Export structure to text file

The function is executed immediately after selection, without any further user interaction. It creates a zip-file with the name of the current project postfixed by "_txt". The zip-package contains the following files:

1) A text file that is structured according to the description in chapter 5 "Annex Guide – Text file structure". It contains the complete current version of the breakdown structure; documents are referenced to be in the folder where the file will be unzipped.
2) A STEP P21-file according to the EXPRESS-schema "ISO_12006_3_VERSION_14" with the name <project_name>_RDL.stp, which contains the Reference Data Library (RDL) associated with the breakdown structure.
3) A file with name "project_name.txt". It contains only one line with the name of the project.
4) All the digital files referred to by the breakdown structure. The files are stored in the zip-file with the original user-defined names.

Such structured text files may be used for local project creation; see chapter 3.2.4.4 .

3.7.1.7 Export project to DEX

The current model will be exported in the ISO 10303-239 (PLCS) format. When selecting the DEX export menu item a pop-up window appears; see Figure 121.

If the checkbox is empty, the model will be exported as a PLCS breakdown structure. Documents will be included in the zip-file in the same way as for PDM-files; see items #4 and #5 in chapter 3.7.1.5, above.

If the checkbox is selected, the breakdown structure will be interpreted as a product assembly and will correspondingly be represented using the Next_assembly_usage concept of the AP239 ARM. Document files will in this case not be included in the zip-file.

Click "OK" to continue.

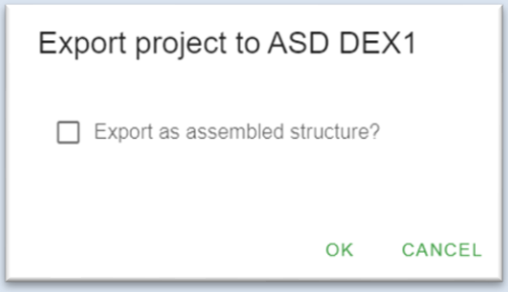


Figure 121. Export to DEX input form

The function creates a zip-file with the name of the current project postfixed by "_dex". The zip-package contains the following files:

1) A STEP P21-file according to the EXPRESS-schema "AD_PROD_BREAKDOWN_DEX" with name <project_name>.stp . It contains the complete breakdown structure, but documents are not included. Files of type AD_PROD_BREAKDOWN_DEX are compliant with ISO 10303-239, PLCS.
2) A STEP file with name <project_name>_RDL.stp, which contains the RDL associated with breakdown structure.
3) A file with name "TRUEplm_project_name.txt". It contains only one line with the name of the project.

Such exported DEX-files may be used for local project creation (chapter 3.2.4.6) and as initial population of new globally defined projects (chapter 4.1.4).

Note: DEX export does not include associated documents.

3.7.1.8 Export to AP242 domain

This feature allows you to export a project into AP242 format.

3.7.1.9 Import CSV file

EDMtruePLM offers an import file format based on comma separated values (csv). This format is especially useful to add data to branches of an existing breakdown. The function is launched from the breakdown root menu even though it may cause changes to a branch far down in the breakdown.

The format is specified as follows:

- /// - starts comments (text to the right of the three slashes is not interpreted)
- CONFIG - key word to introduce the optional configuration section. A character directly following the keyword specifies the delimiter used in the entire CSV file.
  In the CONFIG-section each line is a property/value pair with global applicability to all records in the DATA-section. The following properties are interpreted:

  - Operation: type of manipulation with the data listed below; the only available is currently:
    - ADD: to create new BE according to specified property values;
  - Folder: path to the breakdown element that is the parent folder of the data; this is the location where to store the imported data;
  - Type: class of breakdown element of the children that shall be instantiated based on the data;
  - Date_format: specifies the date/time format for properties of type date/time, for example 'YYYY-MM-DD hh:mm:ss'.
- DATA - keyword to introduce the mandatory data section. A character directly following the keyword specifies the delimiter used in the entire CSV file; this is only used if the CONFIG section is omitted.
  The data section starts with a single header line. The line is a list of comma separated property names of the "folder" specified in the CONFIG-section. Names of system properties and names of user-defined properties may be listed. User-defined properties may use the full URN, such as, "urn:rdl:TruePLM:Comments", or just the property name, such as, "Comments".
  The subsequent data records list values in the order specified by the header line of the DATA section. In case a value includes a comma, it must be quoted like this "Comma,,,<containing value>". The character " must be doubled inside quoted strings. Line breaks (CR/LF) are ignored.

Figure 122 and Figure 123, below, show an example csv import file and the population that resulted from its import. The right-hand side properties in Figure 123 belong to breakdown element "Sensor1".

```
CONFIG,
operation, ADD
folder,Sensor part/USB GPS
type,Sensor part
DATA
name, description,Link,manufacturer,model,serial number,Tellucloud ID
Sensor1,Wheel F sensor,,ABB,A56,145822,T67
Sensor2,Time sensor,,ABB,A57,14568,T68
Sensor3,Acceleration sensor,,ABB,A58,116686,T69
Sensor4,Speed sensor,,MTU,MTU478,451,T70
Sensor5,Wheel R sensor,,MTU,MTU479,546,T71
Sensor6,TimeR sensor,,MTU,MTU480,776,T72
Sensor7,AccelerationR sensor,,MTU,MTU481,445,T73
Sensor8,SpeedR sensor,,MTU,MTU482,7875,T74
Sensor9,Pressure sensor,,MTU,MTU483,4554,T75
```

Figure 122. Example of an import file in csv-format

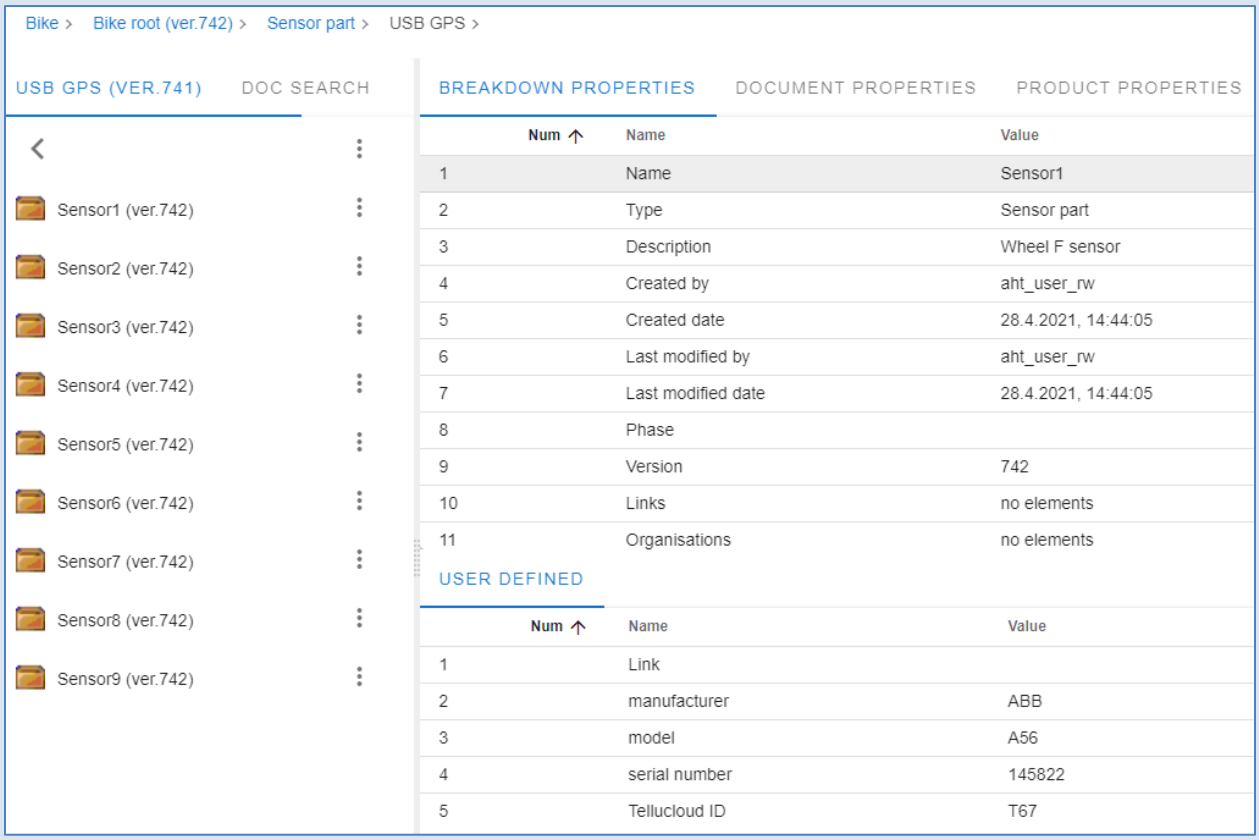


Figure 123. Bike structure after csv import

3.7.1.10 Current project phase

The function allows to set a project phase value as default. This value is then applied to new documents (see chapter 3.4.2.3.8), new document versions and new breakdown elements (see chapter 3.4.2.3.1). For documents the value may be changed during upload; for breakdown elements it is fixed.

Select from the drop-down list the desired project phase value. The drop-down list shows all project phase values that were defined in the reference data section.

3.7.2 Project root selected menu

This chapter describes the menu items of the project root after it has been selected; see Figure 124. The menu contains a subset of the context menu items of an ordinary breakdown element when it plays the role of a parent node. Therefore, this chapter just references the corresponding descriptions.

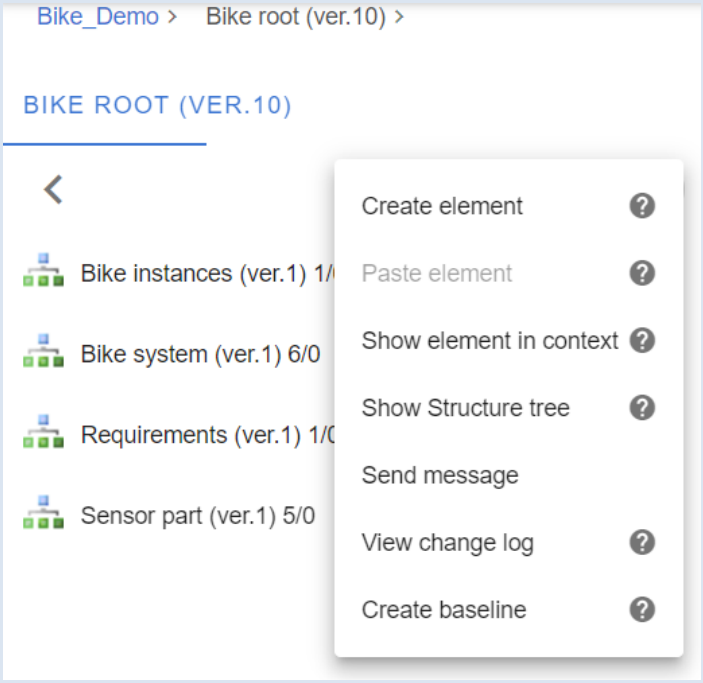


Figure 124. Project root context menu after selection

3.7.2.1 Create element

See description in chapter 3.4.2.3.1 .

3.7.2.2 Paste element

See description in chapter 3.4.2.3.2 .

3.7.2.3 Show element in context

See description in chapter 3.4.2.3.10 .

3.7.2.4 Show structure tree

See description in chapter 3.4.2.3.11 .

3.7.2.5 Send message

See description in chapter 3.6.5.1.1

3.7.2.6 View change log

See description in chapter 3.4.2.3.12 .

3.7.2.7 Create baseline

See description in chapter 3.4.2.3.13 .

4 EDMtruePLM Administration GUI

This chapter describes the system administration functionality. This is only available for the EDMtruePLM ***system administrator***, also called superuser.

Table 8 lists user types and their default access to application functionality.

Table 8. List of User Types

	User Type	Project and System Administration	Role in breakdown structure	Role in Subdomain Breakdown structure	RDL manipulation	Document Role
1	System Administrator (superuser)	Full (Read, Write, Delete)	None	None	None	None
2	Project manager	None	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)
3	Project admin	None	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)
4	Document manager	None	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)	Full (Read, Write, Delete)
5	RDL manager	None	None	None	Full (Read, Write, Delete)	None
6	Subdomain leader	None	None	Full (Read, Write, Delete)	None	Full in subdomain, none in other areas
7	Project member	None	None	None	None	Based on the given roles in a subdomain

After login into the system as EDMtruePLM ***system administrator***, the following user interface will be presented.

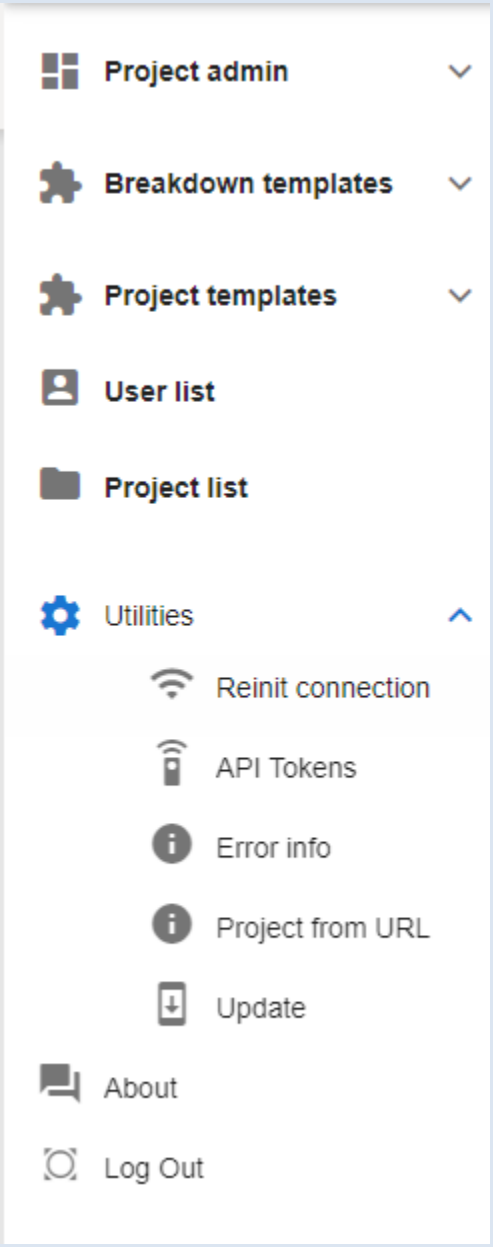


Figure 125. User Interface

The administrator can create, delete, and manage projects within the system using the left-hand side menu (see 4.1,4.2, 4.3, 4.4 and 4.5).
Under **Utilities tab**, admin can work on the below points.

- Reinit connection - reinitialize the connection to the EDM server.
- API Tokens - check the list of API tokens for the superuser.
- Error Info - get error description by the error number. Very useful for the REST API errors.
- Project from URL - New Utilities element 'Project from URL' provides info about a project by project URL. You can try open any project and copy URL from the address

line of the browser. Then use the URL in the admin area to get project info - name, model and description.

- Update - choose here update package to update the server.

4.1 Projects Administration

All the **active** projects available within the system are presented under the heading of "Project admin". The superuser can use this area to add and remove users from a selected project.

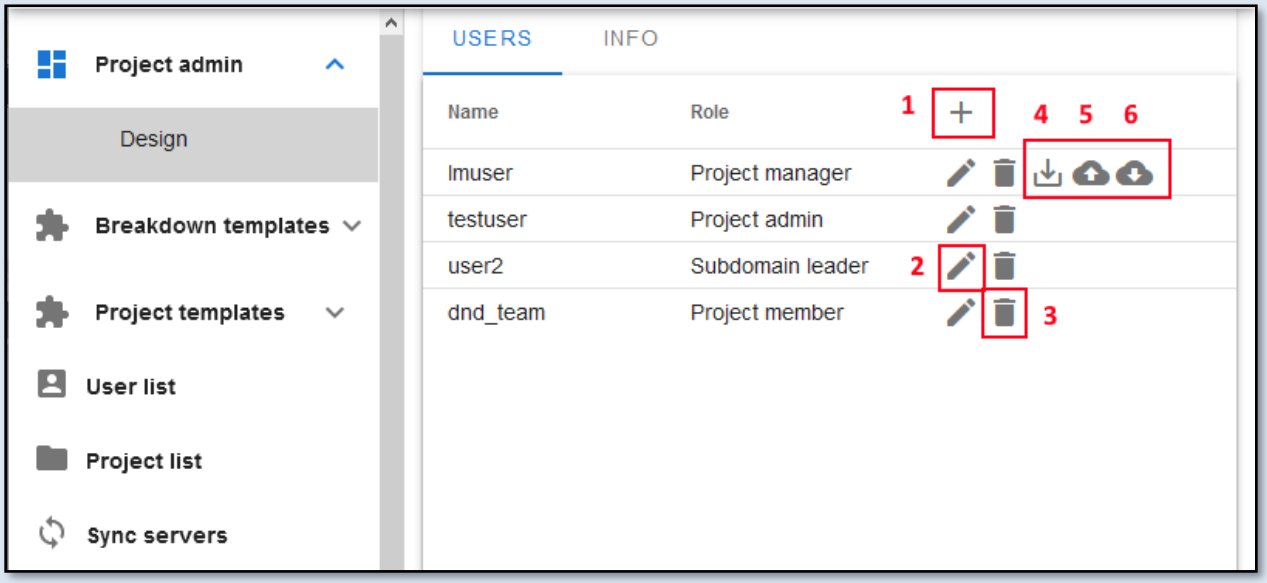


Figure 126. Project Administration

The list of "Users" in the right-hand pane of Figure 126 will be empty if no user is assigned to the project. Items 1 through 6 highlighted in Figure 126 are described below; the third digits in the chapter numbers refer to the digits in Figure 126.

4.1.1 Add user

Use the plus sign to add a username and select a role from the given list. The roles define the access rights of the user to the selected project; see Table 8 for available roles. The plus symbol is not visible, if there is no user created in the system.

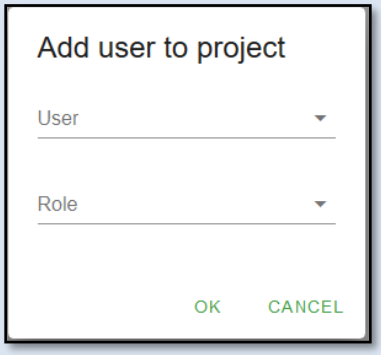


Figure 127. Add User

4.1.2 Edit user

The pen icon is used to edit the role of a user within the project (see Figure 128, below).

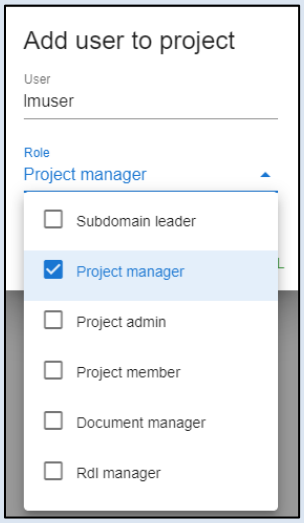


Figure 128. Edit User

4.1.3 Remove user

The bin icon is used to remove a user from the project (see Figure 129, below).

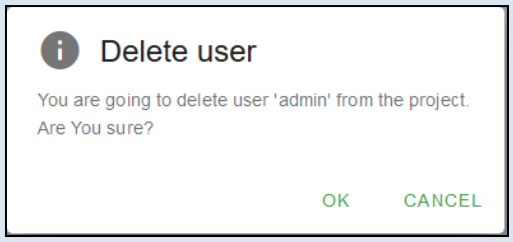


Figure 129. Delete User

4.1.4 Populate new project by import

To populate the new project with externally defined data, use the import icon . When the input form appears (see Figure 130), select a zip-file of your preferred contents and specify the corresponding source types.

The imported data will replace any existing data in the project. A corresponding message will appear before the input form in Figure 130.

Note: The import function does not produce a log-file. Errors during execution will stop the import. In such a case the project will be left empty, even though it may have been populated before the import started.

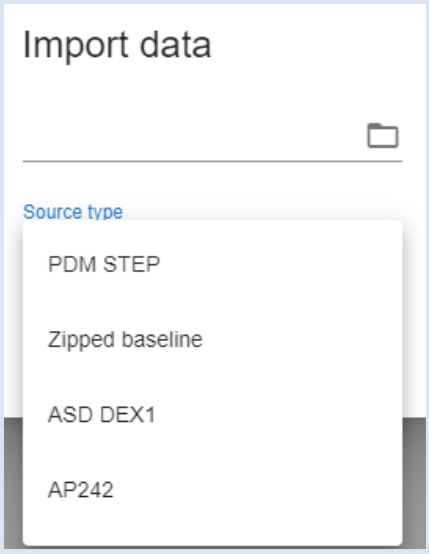


Figure 130. Import data

The source types are all zip-files, but the data for import may use the following different formats:

- AP242/PDM STEP: The breakdown structure needs to be represented according to the ISO 10303-242 MIM/P21-format. The content of this type of zip-file is described in

chapter 3.7.1.5 . PDM STEP files may originate from ISO 10303-242 compliant CAD and PLM applications, for example, or from EDMtruePLM (see chapter 3.7.1.5).

- Zipped baseline: The breakdown structure must be represented as an EDMtruePLM specific ISO 10303-239 P21 file including all files and a separate reference data P21 file (for details of the contents of the resulting zip-file, see chapter 4.1.5 “Download STEP pack”); for the creation of such baseline packages, see chapter 3.6.3 .

  Baselines may also be imported by individual users, and not only the Superuser, to create user-specific projects; this is described in chapter 3.2.4.5 .

  The import described here adds a baseline package into an existing project; thus, project name and "Project Manager" are given already. The import function will, thus, replace the "Project Manager" that is in the exported baseline. The previous "Project Manager" will be given the role of “Project Admin”.

  In principle, the import function assumes that all users that are part of the exported baseline also exist in the current EDMtruePLM server. These users will then get the same access to the imported project that they had to the exported project. However, the import function checks which users in fact exist in the database. Only users from the baseline file that already exist in the database will be assigned to the project. Only those users will have access to the project – unless the Superuser changes user access after import.

  For all other users, no action is taken during import. This may cause undesired behavior if, for example, a "Subdomain Leader" or "RDL Manager" is not available anymore. To have a fully functional imported baseline it is, therefore, important that the superuser ensures that all required user roles are covered by respective users.

- ASD DEX1: The breakdown structure needs to be represented according to the ISO 10303-239 DEX 1 “Product breakdown for support”. The content of this type of zip-file is else described in chapter 3.7.1.7 . DEX 1 STEP/PLCS files may originate from ISO 10303-239 compliant PLM applications, for example, or from EDMtruePLM (see chapter 3.7.1.7).

Note: To further populate the new project after data import, check and complete the assignment of users (chapter 4.1.1) after the data import process.

4.1.5 Download STEP pack

The upward arrow icon is used to download a so called “STEP pack”. All data of the project will be exported as a zipped STEP package file. The zip-package includes two files:

1) the product data as an extended ISO 10303-239 (PLCS) P21-file. The extended schema includes all concepts used by EDMtruePLM, not only the pure AP239 concepts. The file also includes all files uploaded by the user; they are appended to the end of the P21-file, which, thus, also follows an extended EDMtruePLM specification compared to the P21-standard.

Therefore, the STEP pack capability is useful as a backup function. It is not suited for data exchange with other applications than EDMtruePLM. For such data exchange options, see the PDM and PLCS DEX export features in chapter 3.7.1 .

2) the corresponding reference data as an ISO 12006-3 file.

The package can be used for project backup and restore or for the synchronization of project data on different EDMtruePLM servers.

4.1.6 Upload STEP pack

The downward arrow icon is used to upload a STEP pack. This will populate an initially empty project with the imported data.

4.2 Breakdown template list

A breakdown template is a type of project with a snippet of a product breakdown including attached documents. This type of project can be used to store and reuse common breakdown structures as small templates.

The below figure shows where available breakdown templates are listed. The EDMtruePLM system administrator can use this to add a project manager to the breakdown template project.

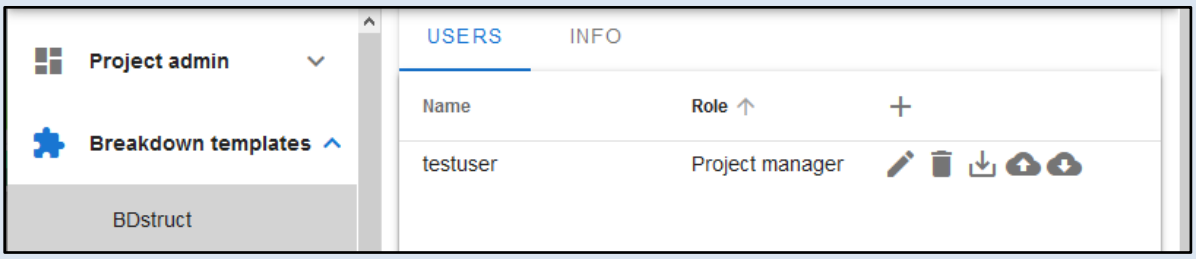


Figure 131. Breakdown Template

A breakdown template is created within the “Project list” menu item to the left in Figure 131 and explained later in this section, and by selecting the value “Breakdown template” for the “Type” entry, as shown in Figure 132, below.

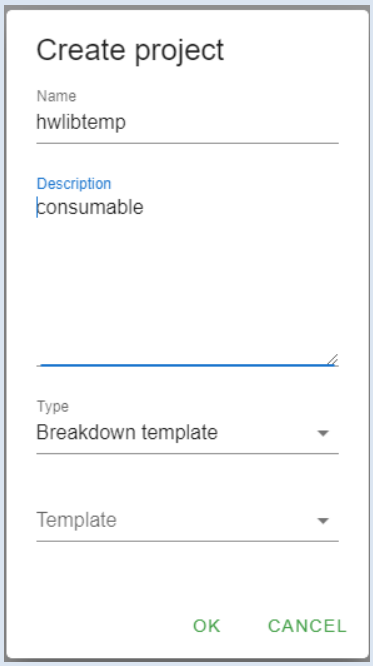


Figure 132. Create breakdown template type

A breakdown template may be inserted into an existing project by the “Create element” menu item (see Figure 38).

4.3 Project template list

A project template is a type of pre-defined reusable project. Figure 133 shows where available project templates are listed. Here the system administrator can assign a project manager to a selected project template.

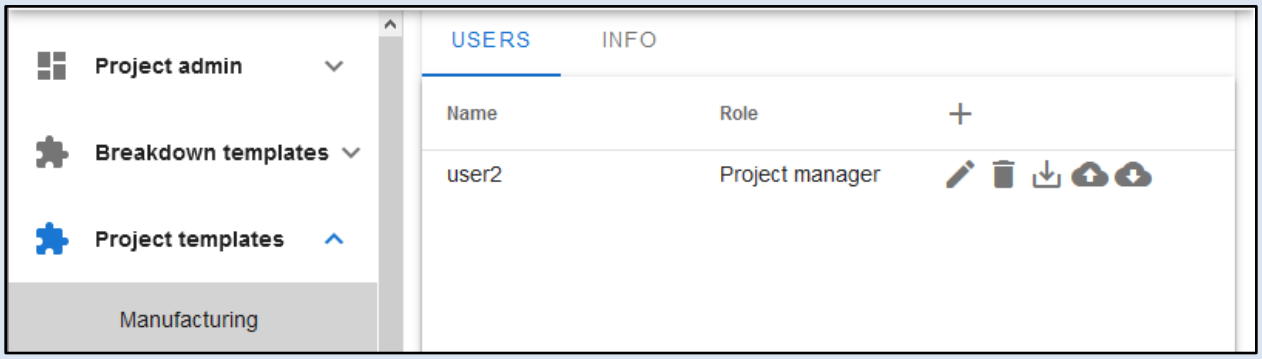


Figure 133. Project Template

A project template is created within the “Project list” menu item to the left in Figure 126 and by selecting the value “Project template” for the “Type” entry, as shown in Figure 134, below.

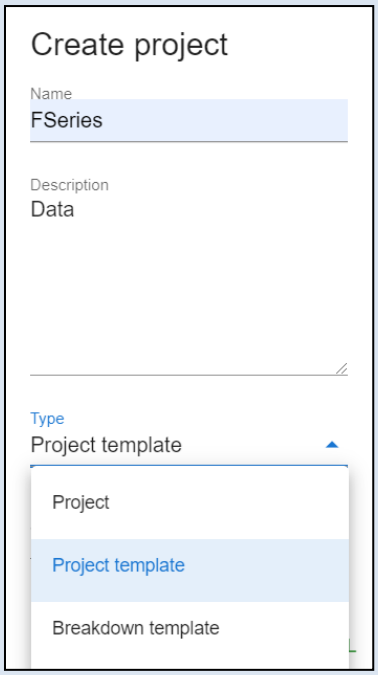


Figure 134. Create project template type

A project template may be reused as the initial definition of a project by the “Create project” menu items (see Figure 141). In contrary to breakdown templates, project templates cannot be inserted into existing breakdown structures.

4.4 User list

The user list allows the EDMtruePLM system administrator to add users to the system.

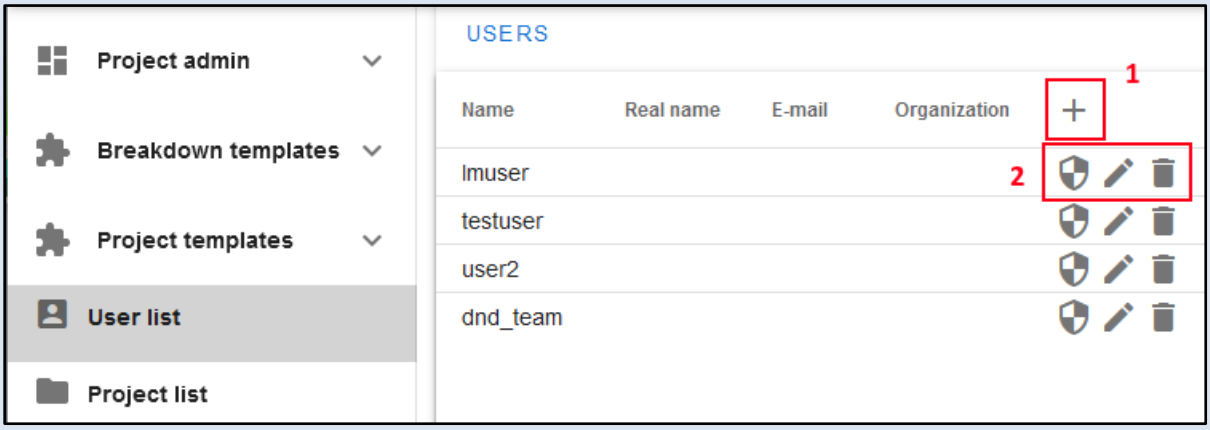


Figure 135. User list

Note: Usernames can only be stated in lowercase letters; the system automatically replaces uppercase letters.

4.4.1 Create user

To add a new user to the system, click on the plus sign (see “1” in above Figure 135). Fill in the form and click on 'OK' to finish. Password must be minimum 7 characters and maximum 24 characters and consist of at least one lowercase and one uppercase letters, one digit and one non alphanumeric character.

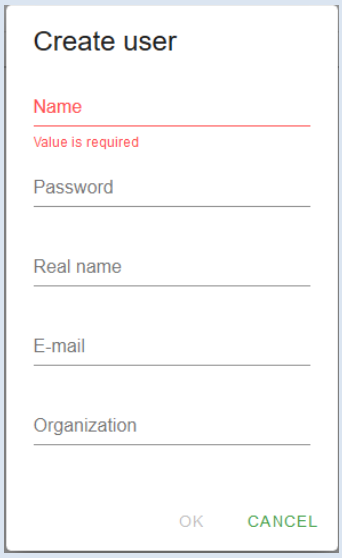


Figure 136. Create User

4.4.2 Deleting user

To delete a user from the system, use the bin icon (see “2” in Figure 135).

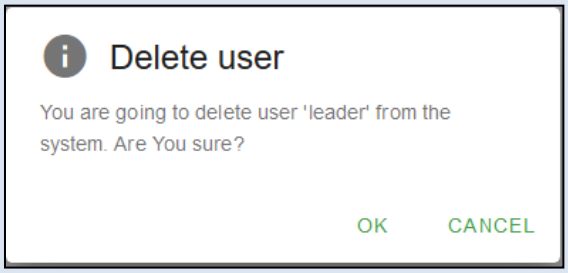


Figure 137. Delete user

4.4.3 Editing user

To edit the user's info within the system, use the pen icon (see “2” in Figure 135).

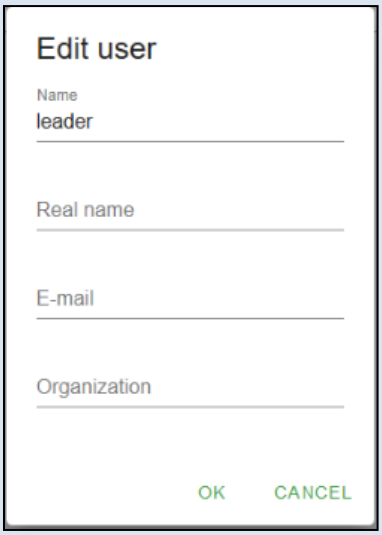


Figure 138. Edit user

4.4.4 Changing user password

To change the user's password, use the shield icon (see “2” in Figure 135).

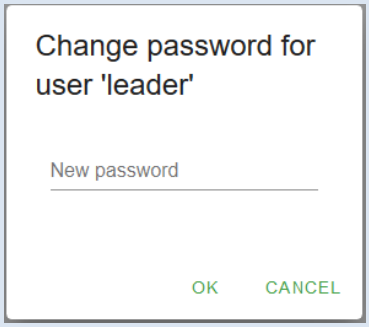


Figure 139. Change Password

4.4.5 Two Factor Authentication (2FA)

The admin can enable the 2FA by authenticating user credentials in the Google or Microsoft authentication apps or similar.

4.5 Project list

The project list shows all the projects within the EDMtruePLM server (see Figure 140, below).

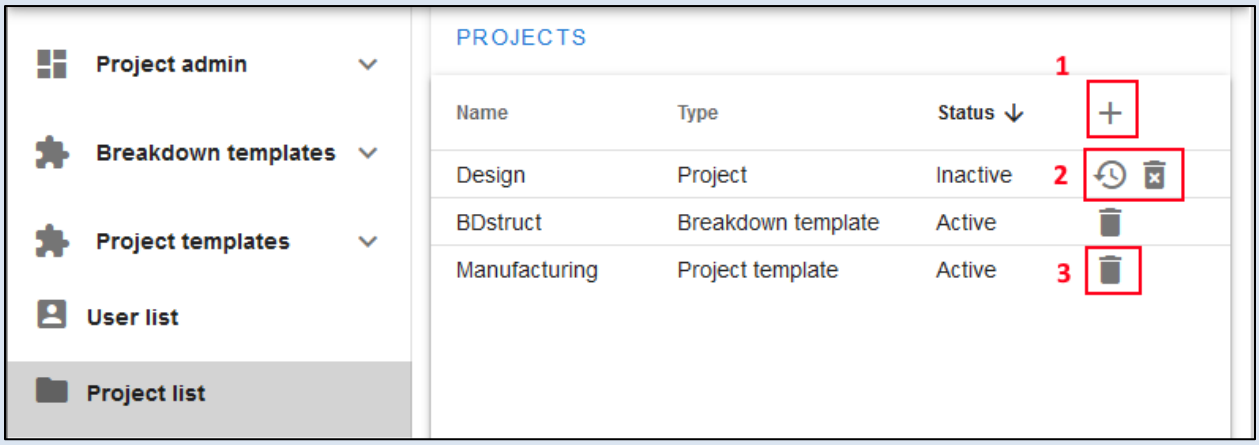


Figure 140. Project list

4.5.1 Add project

Use the plus icon (see "1" in Figure 140) to create a project. The system administrator may create the following three types of projects (as shown in Figure 141):

Type 1. Project

Type 2. Project template

Type 3. Breakdown template

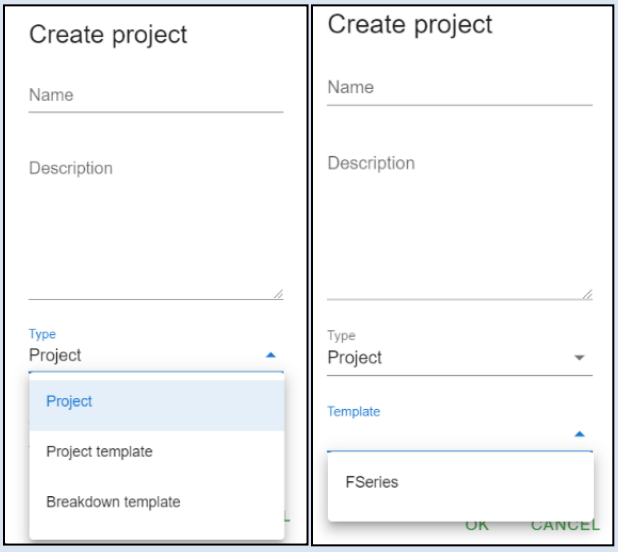


Figure 141. Create project

Select a project template from the template list (see above Figure 141) to create the project from a template.

Warning: A project name must be unique and may contain Latin letters, underscore symbols and numbers. No spaces or special characters are allowed.

After project creation the empty project will become selectable by the assigned users. They may now use the context menus described in chapter 3.7 to add data.

4.5.2 Deactivate project

Deactivating a project will remove the project from the available project list. For archival reasons, the project will remain in the database. Ordinary users cannot access a deactivated project. To deactivate a project, use the bin icon, see “3” in Figure 140.

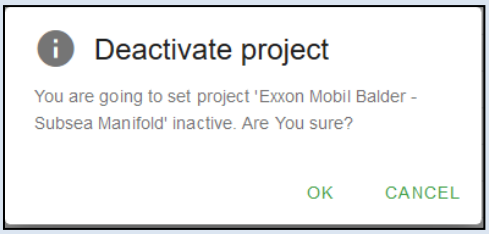


Figure 142. Deactivate project

4.5.3 Reactivate project

To reactivate a deactivated project, use the clock icon, see “2” in Figure 140.

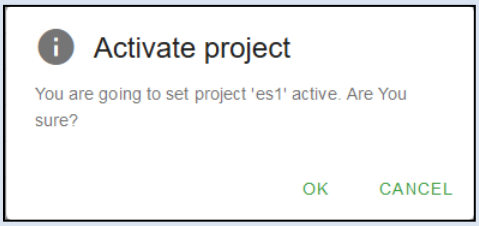


Figure 143. Activate project

4.5.4 Delete project

To delete a project from the system, use the bin icon with the ‘x’ within it, see “2” in Figure 140. This operation will remove the project physically from the database; this cannot be undone.

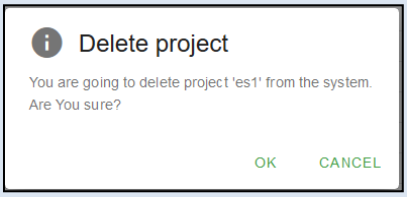


Figure 144. Delete project

4.6 About (for an admin)

The form shows the type of user who is logged on, the email address for system support and details of the current EDMtruePLM server version.

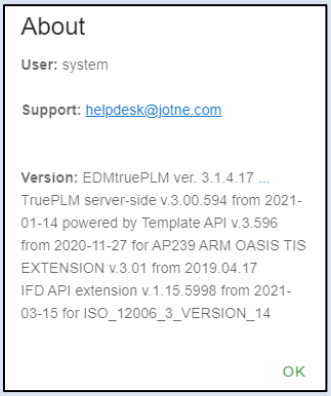


Figure 145. About (admin)

4.7 Log out

Will log out the current user.

Note: A user session is automatically closed after a certain period of time, if the system is idle without any activity. This period is default set to be 15 minutes. But can be configured by the system administrator in the following configuration file.

C:\ProgramData\Jotne\TruePLM_base\conf\EDMtruePLM\application.properties

5 Annex Guide – Text file structure

This section gives the brief explanation about ‘Text file structure format’.

This is a plain text file, where the levels in the product breakdown structure are defined by the indentation level on the file. The system allows the user to create a project based on a text file that describes the structure of the product tree. The following annex describes the structure of the text file and the zip file that includes this file. The given file can be either a plain text file, if there are no file references or else it must be a zipped package containing all files.

Zip file Package

If the file (named breakdown_structure.txt for instance) does not contain any references to attached files, the user can simply import this plain file. Otherwise, one must prepare a zip package containing breakdown_structure.txt along with all the referred files. In addition, the zip package must contain a file with name TRUEplm_project_name.txt, which contains the name of the file described in the next paragraph. Please note that the name of the file should be without the filer type/suffix (.txt)

Text file structure

The main rule is that each new line in the text file represents an element in the breakdown structure. Lines that are on the same level of indentation represent elements on the same level in the breakdown structure. There are, however, some exceptions from the main rule. If a line starts with a prefix, there is one specific rule for each prefix.

The prefixes are:

- **E:** For breakdown elements, the name of the node is followed by the colon "**:**"
  - TYPE: type of the node, e.g. urn:rdl:epm-std:System
  - PHASE: The phase of the node in the project
  - DESCR: the description of the node
- **F:** file/document, the name of the file is followed after the colon "**:**"
  - TYPE: type of the node, e.g. urn:rdl:epm-std:System
  - PHASE: The phase of the node in the project
  - DESCR: the description of the node
- **P(s):** string property prefix, the name of the property is followed after the colon "**:**"
- **P(n):** numerical property prefix, the name of the property is followed after the colon "**:**"
  - UNIT= the name of the unit
- **P(e):** enumerate property prefix, the name of the property is followed after the colon "**:**"

Structure of the file: See example in the next page 115.

To create the file, follow below mentioned rules

1. The structure is created by indentation
2. The indent is given in the file as – indent = "number of spaces"
3. Child nodes (element or documents) are listed under the parent node one by one
4. Properties of a node are listed one by one directly under the node
5. RDL files added to zipped pack
6. For classes and types that belong to project's domain can be specified without urn prefix, external types - with full urn name

Example

Code Block

```
-- TruePLM text export v.1.0
-- for urn:rdl:test_project
-- "E:" - node/folder prefix
-- "F:" - file/document prefix
-- "P(s):" - string property prefix
-- "P(n):" - numerical property prefix
-- "P(e):" - enumeric property prefix
-- indent = 4
E:carstens muinck, TYPE= urn:rdl:epm-std:System, PHASE= 0, DESCR= haley
lafever
    E:dinnie oswell, TYPE= urn:rdl:epm-std:Product_definition, PHASE= 0,
DESCR= swinkels wiggins
        P(s):date1 = 2019-09-12
        P(e):enum1 = mo
        P(n):num1 = 123, UNIT= metre per hour
        P(s):text1 = 100 km
        F:document.txt, TYPE= urn:rdl:epm-std:Design_document, DESCR=
clendeni lemyre
            P(s):text2 = $result = User::register($name, $email, $password);
    E:oscar, TYPE= urn:rdl:epm-std:Module, PHASE= 0, DESCR= truesdal vrouwerf
        P(s):date1 = 2019-09-27
        P(e):enum1 = tu
        E:zuranato cupido, TYPE= urn:rdl:epm-std:Subsystem, PHASE= 0, DESCR=
amando abdel-az
            P(s):date1 = 2019-09-06
            P(e):enum1 = tu
            F:AdminController.php TYPE= urn:rdl:epm-std:Design_document,
DESCR= sdfsdgdfg
                P(s):text2 = sample of textual property of a document
            F:SiteController.php TYPE= urn:rdl:epm-std:Design_document,
DESCR= qqqqqqqqqq
            E:hopkin, TYPE= urn:rdl:epm-std:System, PHASE= 0, DESCR= nimish
smrke-su
                E:wolfgang, TYPE= urn:rdl:epm-std:Unit, PHASE= 0, DESCR=
woei-pen
                    P(s):date1 = 2019-09-26
                    P(s):text1 = $categories = Category::getCategoriesList();
                    E:toby, TYPE= jobe pautenis, PHASE= 0, DESCR= tjahjadi
                        P(s):date1 = 2019-09-14
                        P(n):jobe = 11, UNIT= abbai
```

```
                        F:CatalogController.php TYPE= urn:rdl:epm-
std:Design_document, DESCR= rosenber wigderso
                            P(s):text2 = require_once
    E:another, root, TYPE= urn:rdl:epm-std:Module, PHASE= 0, DESCR= Test
element to show description
        P(s):date1 = 2019-09-12
        P(e):enum1 = mo
        P(s):text1 = sample of text "with quote" and 'apostrophy'
        P(n):num1 = 12345.7, UNIT = metre per hour
```

6 Reference Material

https://jotne.atlassian.net/wiki/spaces/EDM/pages/3402104834/EDMtruePLM+Reference+Material