GENIUS TOOLS®

INNEO That's IT.

# GENIUS TOOLS for Creo

13.0.0.0

# User manual

© 2026 INNEO Solutions GmbH

www.inneo.com

**1 Product overview 15**

**1.1 Full details of all functions 19**

**2 Ribbon menu GENIUS TOOLS 25**

**2.1 License-dependent functions 26**

**3 Ribbon menu GT MBD 30**

**3.1 Glossary 30**

**3.2 Best Practice 32**

**4 What's new in release 13 40**

**4.1 Quick Access 40**

**5 Assembly Report 41**

**5.1 Fundamentals 41**

5.1.1 Glossary 42

5.1.2 Standard behavior 43

**5.2 Usage 44**

5.2.1 User interface 45

5.2.2 Command bar 46

5.2.3 Report options 48

5.2.4 Report definitions 50

5.2.5 Model selection and view 50

5.2.6 Model list 51

5.2.7 Report table 52

5.2.8 Saving 54

**5.3 Configuration 54**

5.3.1 Assembly Report Editor 55

5.3.1.1 User interface 55

5.3.1.2 Command bar 56

5.3.1.3 General information 57

5.3.1.4 Creating table parameters 58

5.3.1.5 Configuring report table 64

5.3.1.6 Use cases 69

5.3.2 Configuring export of reports 77

5.3.3 Creating an Excel template 78

5.3.4 Automatic update of report definitions 79

5.3.5 Dynamically transferring position numbers from Windchill 80

5.3.6 Use cases 83

5.3.6.1 Removing single lines from parts list 83

5.3.6.2 Filtering out subcomponents of purchased parts 87

**6 Dimension 92**

**6.1 Starting the program 92**

**6.2 User interface ..... 94**
6.2.1 Command bar ..... 95
6.2.2 Filter and search ..... 95
6.2.3 Model selection ..... 97
**6.3 Dimensions table ..... 98**
6.3.1 Dimension symbols and context menu ..... 99
6.3.2 Input fields ..... 100
6.3.3 Colors in input fields ..... 101
6.3.4 Tolerance types ..... 102
6.3.5 Boundaries ..... 103

**7 Export TDP 104**

**7.1 Usage ..... 105**
7.1.1 User Interface ..... 105
**7.2 Configuration ..... 107**
7.2.1 Editing export templates ..... 108
7.2.1.1 Defining TDP template ..... 108
7.2.1.2 Defining chapter template ..... 114
7.2.2 Available configuration options ..... 117
7.2.3 Latex customization with Export TDP ..... 118
7.2.4 Use cases ..... 121
7.2.4.1 Create TDP template ..... 121

**8 Forms 123**

**8.1 Fundamentals ..... 124**
**8.2 Usage ..... 124**
8.2.1 User interface ..... 125
8.2.2 Command bar ..... 126
8.2.3 Model selection in assembly mode ..... 129
8.2.4 Form display ..... 129
8.2.5 Using a form ..... 131
8.2.5.1 Filling input and selection fields ..... 132
8.2.5.2 Modifying components ..... 133
8.2.5.3 Automatic setting of properties ..... 133
8.2.5.4 Setting multiple properties ..... 134
**8.3 Configuration ..... 134**
8.3.1 Forms Editor ..... 135
8.3.1.1 User interface ..... 136
8.3.1.2 Command bar ..... 136
8.3.1.3 Managing form elements ..... 137
8.3.1.4 Describing a form ..... 141
8.3.1.5 Creating form elements ..... 143
8.3.1.6 Editing form display ..... 152
8.3.1.7 JavaScript in Forms ..... 154

8.3.1.8 Automatic load of external data ... 155
8.3.2 Use cases ... 156
8.3.2.1 Creating and applying a Form ... 156
8.3.2.2 Automatically set tolerance from variant table ... 156

**9 Function Manager 167**

**9.1 Usage ... 167**
9.1.1 User interface ... 168
9.1.2 Managing functions ... 169
9.1.3 Editing metadata of a function ... 170
9.1.4 Adding functions ... 171
9.1.4.1 Combined view ... 173
9.1.4.2 Links to additional information ... 173
9.1.4.3 Predefined functions ... 174
9.1.5 Coloring functions ... 175
9.1.6 Editing combined views ... 178
9.1.7 Changing the naming scheme ... 179
**9.2 Configuration ... 180**
9.2.1 Editing function templates ... 180
9.2.2 Editing color templates ... 183

**10 Inspect 187**

**10.1 Fundamentals ... 188**
**10.2 Usage ... 189**
10.2.1 User interface ... 190
10.2.2 Command bar ... 191
10.2.3 Placing ... 192
10.2.3.1 Placing with exceptions ... 194
10.2.4 Numbering ... 196
10.2.5 Overview ... 198
10.2.6 Hide and unhide ... 201
10.2.7 Export to Excel ... 202
**10.3 Configuration ... 205**
10.3.1 Inspect Editor ... 205
10.3.1.1 User interface ... 206
10.3.1.2 Inspect configuration ... 207
10.3.1.3 General ... 208
10.3.1.4 Editing inspection symbols and tables ... 208
10.3.1.5 Editing column display ... 215
10.3.1.6 Assigning shape and position tolerances / surface finish symbols ... 215
10.3.1.7 Filtering displayed inspection symbols ... 216
10.3.1.8 Settings for 3D models ... 217
10.3.2 Creating a Creo symbol for Inspect ... 220
10.3.3 Adding parameter values to a symbol ... 224

10.3.4 Creating change symbols and tables ... 226
10.3.5 Creating a custom report table ... 229
**10.4 Inspect Revision ... 231**
10.4.1 Fundamentals ... 231
10.4.2 Starting the program ... 231
10.4.3 User interface ... 232
10.4.4 Command bar ... 232
10.4.5 Selecting a configuration file ... 233
10.4.6 Selecting the tree view ... 234
10.4.7 Creating snapshots ... 235
10.4.8 History of snapshots and symbols ... 237
10.4.9 Revision parameter ... 238
10.4.10 Export snapshot history ... 240
**11 Inspect 3D 243**
**11.1 Fundamentals ... 243**
**11.2 Usage ... 245**
11.2.1 User interface ... 246
11.2.2 Command bar ... 246
11.2.3 Placing ... 247
11.2.3.1 Placing with exceptions ... 248
11.2.4 Numbering ... 250
11.2.5 Overview ... 251
11.2.6 Hiding and unhiding elements ... 253
11.2.7 Export to Excel ... 254
**11.3 Configuration ... 257**
11.3.1 Inspect Editor ... 257
11.3.1.1 User interface ... 258
11.3.1.2 Command bar ... 259
11.3.1.3 Inspect configuration ... 259
11.3.1.4 General ... 260
11.3.1.5 Editing inspection symbols and tables ... 260
11.3.1.6 Editing column display ... 267
11.3.1.7 Assigning shape and position tolerances / surface finish symbols ... 267
11.3.1.8 Filtering displayed inspection symbols ... 268
11.3.1.9 Settings for 3D models ... 269
11.3.2 Creating a Creo symbol for Inspect ... 272
11.3.3 Adding parameter values to a symbol ... 276
11.3.4 Creating change symbols and tables ... 278
**11.4 Inspect Revision 3D ... 281**
11.4.1 Fundamentals ... 281
11.4.2 Starting the program ... 281
11.4.3 User interface ... 282
11.4.4 Command bar ... 282

11.4.5 Selecting a configuration file ... 283
11.4.6 Selecting the tree view ... 284
11.4.7 Creating snapshots ... 285
11.4.8 History of snapshots and symbols ... 287
11.4.9 Revision parameter ... 288
11.4.10 Export snapshot history ... 290

**12 Library 293**

**12.1 Fundamentals ... 294**
12.1.1 Glossary ... 294
**12.2 Usage ... 296**
12.2.1 Starting the library: in all Creo modes ... 296
12.2.2 User interface ... 298
12.2.3 Using library objects ... 299
12.2.4 Working with templates ... 303
12.2.5 Searching and filtering library objects ... 304
12.2.5.1 List of search operations ... 308
12.2.6 Displaying library objects ... 310
12.2.6.1 Detail window ... 312
12.2.7 Managing favorites ... 317
12.2.8 Searching for forms and import values from XML ... 318
**12.3 Configuration ... 319**
12.3.1 Areas in the resource directory ... 319
12.3.2 Import data from PDM/PLM ... 320
12.3.3 Multilingual use ... 320
12.3.4 Library Editor ... 322
12.3.4.1 User interface ... 323
12.3.4.2 Command bar ... 324
12.3.4.3 Category tree ... 325
12.3.4.4 Category view ... 327
12.3.4.5 Object details ... 330
12.3.4.6 Actions on library objects ... 348
12.3.4.7 Wizard: Batch mode ... 356
12.3.4.8 Wizard: Import ... 359
12.3.4.9 Wizard: MNU export ... 359
12.3.4.10 Wizard: Object collector ... 360
12.3.4.11 Cleaning up data ... 363
12.3.4.12 Customizing the search function ... 363
12.3.5 XML interface for Form values ... 364
12.3.6 Data structure ... 365
12.3.7 Setting configuration options ... 367
12.3.8 Use cases ... 367
12.3.8.1 Creating a new library ... 367
12.3.8.2 Creating a new category ... 368

12.3.8.3 Creating a new library object ... 369
12.3.8.4 Creating a library object with a form ... 370
12.3.8.5 Importing family tables ... 374
12.3.8.6 Updating objects ... 377
12.3.8.7 Importing additional object information ... 379
12.3.8.8 Importing Library Viewer libraries ... 381
12.3.8.9 Importing MUI files ... 386

**13 Material 388**

**13.1 Usage ... 389**
13.1.1 Starting the program ... 389
13.1.2 User interface ... 391
13.1.3 Selecting materials ... 392
13.1.4 Assigning materials ... 394
13.1.5 Deleting material ... 398
13.1.6 Updating materials ... 399
13.1.7 Checking materials in a model ... 399
**13.2 Configuration ... 401**
13.2.1 Material Selector Editor ... 401
13.2.1.1 Starting the program ... 402
13.2.1.2 User interface ... 403
13.2.1.3 Material directory ... 403
13.2.1.4 Defining selectable materials ... 404
13.2.1.5 Display settings of Material Selector dialog ... 406
13.2.2 Activating material check ... 408
13.2.3 Editing material files ... 410
13.2.3.1 Editing material files simultaneously ... 410
13.2.3.2 Editing material files in Windchill ... 411

**14 MBD Info 413**

**14.1 Excel to Export ... 419**

**15 MBD Tables 422**

**15.1 Size Table ... 422**
15.1.1 Usage ... 422
15.1.1.1 User interface ... 423
15.1.1.2 Formatting tables ... 424
15.1.2 Configuration ... 425
15.1.2.1 Editing templates for Size Tables ... 425
**15.2 Surface Finish Table ... 429**
15.2.1 Usage ... 429
15.2.1.1 User interface ... 429
15.2.1.2 Formatting a table ... 431
15.2.2 Configuration ... 433
15.2.2.1 MBD Tables Editor ... 433

15.2.2.2 Editing templates for Surface Finish Tables ... 435
**15.3 MBD Update Tables ... 439**
**16 Name Generator 440**
**16.1 Usage ... 440**
16.1.1 User interface ... 444
**16.2 Configuration ... 445**
16.2.1 Name Generator Editor ... 446
16.2.1.1 User interface ... 447
16.2.1.2 Available name configurations ... 447
16.2.1.3 Configuration details ... 448
16.2.2 Use cases ... 450
16.2.2.1 Global name configurations with fallback ... 450
16.2.2.2 Global name configurations with Windchill ... 452
16.2.2.3 Integrating name configurations in Quick Access ... 454
16.2.2.4 Using project numbers ... 455
**17 Orientation Manager 460**
**17.1 Usage ... 460**
17.1.1 User interface ... 461
17.1.2 Command bar ... 462
17.1.3 Create Orientation ... 463
**18 Parameter 465**
**18.1 Fundamentals ... 466**
18.1.1 Glossary ... 466
18.1.2 Parameter management concept ... 467
18.1.3 Mechanisms on starting and saving ... 468
**18.2 Usage ... 470**
18.2.1 Starting the program ... 471
18.2.2 User interface ... 472
18.2.3 Command bar ... 473
18.2.4 Reading parameter values from text files ... 476
18.2.5 Status dialog ... 476
18.2.6 Model selection ... 477
18.2.7 Model list ... 478
18.2.8 Parameter form ... 480
18.2.8.1 Calculating / assigning masses ... 483
18.2.9 Field types ... 484
**18.3 Configuration ... 486**
18.3.1 Regular Expressions ... 487
18.3.2 Parameter Editor ... 490
18.3.2.1 User interface ... 491
18.3.2.2 Command bar ... 492

18.3.2.3 Element list ... 493
18.3.2.4 Parameter definitions ... 494
18.3.2.5 Adding Parameters ... 498
18.3.2.6 Adding separators ... 514
18.3.2.7 Editing parameter definition list ... 515
18.3.3 Use cases ... 516
18.3.3.1 Creating a simple parameter definition ... 518
18.3.3.2 Creating a Parameter definition with tabs ... 529
18.3.3.3 Importing old parameter definitions ... 530
18.3.3.4 Auto-select parameter definitions ... 533
18.3.3.5 Using JavaScript with Parameter ... 535

**19 Quick Access 544**

**19.1 Usage ... 544**
19.1.1 Starting the program ... 544
19.1.2 User interface ... 545
19.1.3 Mode and selection dependence ... 547
**19.2 Configuration ... 548**
19.2.1 Different types of use ... 548
19.2.2 Quick Access Editor ... 551
19.2.2.1 User interface ... 552
19.2.2.2 Commands overview ... 552
19.2.2.3 Command details ... 553
19.2.2.4 Creating commands ... 555
19.2.2.5 Defining commands ... 555
19.2.2.6 Defining mapkeys ... 556
19.2.2.7 Export and import ... 560
19.2.3 Changing the start key ... 561
19.2.4 Inserting project images ... 562
19.2.5 Adapting zoom levels and icons ... 563
19.2.6 Use cases ... 565
19.2.6.1 Recording a mapkey ... 565

**20 Stack Note 568**

**20.1 Usage ... 568**
20.1.1 User Interface ... 568
20.1.2 Command bar ... 569
20.1.3 Creating notes ... 570
20.1.4 Placing notes automatically ... 573
20.1.5 Updating notes ... 574
20.1.5.1 Applying changes from the database ... 575
20.1.6 Splitting a note table ... 576
20.1.7 Placing symbols ... 578
**20.2 Configuration ... 578**

20.2.1 Managing note entries ... 578
20.2.2 Managing symbols ... 580

**21 Stack Note 3D 582**

**21.1 Usage ... 582**
21.1.1 User Interface ... 582
21.1.2 Command bar ... 583
21.1.3 Create note ... 584
21.1.3.1 Search note entry ... 585
21.1.3.2 Note entries with user input ... 586
21.1.3.3 Set Languages ... 586
21.1.4 Update note ... 587
21.1.4.1 Apply changes from the database ... 588
21.1.5 Placing symbols ... 589
**21.2 Configuration ... 590**
21.2.1 Create and edit note entries ... 590
21.2.2 Example Templates ... 592
21.2.3 Edit symbol database ... 592

**22 Value Transfer 594**

**22.1 Starting the program ... 594**
**22.2 User interface ... 595**
**22.3 Command bar ... 596**
**22.4 Transfer list ... 596**
**22.5 Filter list ... 598**
**22.6 Table of results ... 599**

**23 UDF Forms 601**

**23.1 Fundamentals ... 601**
**23.2 Usage ... 602**
23.2.1 User interface ... 603
23.2.2 Command bar ... 604
23.2.3 UDF selection ... 607
23.2.4 Placement ... 607
23.2.5 Form section ... 609
23.2.6 Use cases ... 610
23.2.6.1 Linking UDFs to drawing tables ... 610
23.2.6.2 Linking UDFs to drawing symbols ... 611
**23.3 Configuration ... 613**
23.3.1 UDF Forms Editor ... 614
23.3.1.1 User interface ... 615
23.3.1.2 Command bar ... 615
23.3.1.3 Details of UDF ... 617
23.3.1.4 References and variables tab ... 618

23.3.1.5 Description tab ... 627
23.3.2 Updating UDF definitions ... 629

**24 Utilities 633**

**24.1 3D Note Form ... 635**
24.1.1 User interface ... 636
24.1.2 Displaying and editing notes ... 638
24.1.3 Usage example ... 639
**24.2 Annotation Info ... 642**
24.2.1 Export to Excel ... 646
**24.3 Annotation Transfer ... 649**
**24.4 Close All Other Windows ... 652**
**24.5 Combined View Gallery ... 653**
**24.6 Command Control ... 655**
**24.7 Component Parameter ... 656**
24.7.1 User interface ... 658
24.7.2 Editing position number parameters ... 659
24.7.3 Creating component parameters ... 660
24.7.4 XML configuration ... 660
**24.8 Convert Materials ... 663**
**24.9 Copy Component Parameter To Substitution Component Parameter ... 663**
**24.10 Create Search.pro ... 665**
**24.11 Create Tolerance Table ... 669**
**24.12 Create Point and Coordinate systems ... 671**
**24.13 CS Assembler ... 673**
**24.14 Default Texts Editor ... 676**
24.14.1 Create default text ... 677
24.14.2 Select default text ... 678
**24.15 Export Points ... 679**
24.15.1 Exporting points ... 680
24.15.2 Settings for points on curves ... 681
24.15.3 Export options ... 682
24.15.4 Creating a template ... 684
24.15.5 Example template ... 685
**24.16 Export Table One-To-One to Excel ... 686**
**24.17 Export Table to CSV ... 687**
**24.18 Export Table to Excel ... 689**
24.18.1 Creating reports ... 690
24.18.2 Create template ... 693
24.18.2.1 Step-by-step guide ... 693
24.18.2.2 More options for templates ... 697
24.18.2.3 Export data from several GENIUS TOOLS extension modules ... 701

**24.19 Extend Relations ..... 702**
**24.20 Extended Dimension Functions ..... 707**
**24.21 Find Contact Surfaces ..... 709**
**24.22 Find References ..... 710**
**24.23 Full Backup ..... 712**
24.23.1 Collect objects for backup list ..... 712
24.23.2 Choose levels of dependencies ..... 714
**24.24 GenTOL References ..... 717**
**24.25 GTOL Text ..... 719**
24.25.1 Starting the program: in part mode and in assembly mode ..... 719
24.25.2 Usage ..... 720
24.25.2.1 User interface ..... 720
24.25.2.2 Edit tolerance element ..... 721
24.25.2.3 Select tolerance table ..... 722
24.25.3 Configuration ..... 722
24.25.3.1 GTOL Text Editor ..... 722
**24.26 Inspect Instance Note ..... 727**
**24.27 Load Save Converter ..... 730**
**24.28 Open Base Model ..... 731**
**24.29 Open/Create Drawing ..... 732**
**24.30 Select Contact Surfaces ..... 734**
**24.31 Select Surfaces by Color ..... 737**
**24.32 Set Component Combined View ..... 738**
**24.33 Show Information ..... 741**
**24.34 Show Pitch ..... 742**
**24.35 Show Thread Size ..... 745**
**24.36 Sort Combined Views ..... 747**
**24.37 Split Table ..... 747**
**24.38 Toggle Symbol Variants ..... 749**
**24.39 Work Dir Manager ..... 751**

**25 Configuring GENIUS TOOLS for Creo 755**

**25.1 Configuration and start ..... 755**
**25.2 Viewing the configuration ..... 763**
**25.3 Telemetry Data Transfer ..... 765**
**25.4 Configuration Utility ..... 765**
25.4.1 Starting the program ..... 765
25.4.2 User interface ..... 766
25.4.3 Configuration overview ..... 768
25.4.4 Searching configuration options ..... 769
25.4.5 Editing configuration options ..... 771
25.4.6 Saving changes ..... 775

25.4.7 Directly applying changes ... 777
25.4.8 Reseting configuration options ... 778
25.4.9 Using comments ... 778
25.4.10 Viewing configuration files and layers ... 780
25.4.11 Environment variables ... 781
25.4.12 Use cases ... 781
25.4.12.1 Set access to programm editors ... 781
**25.5 Database Version Control ... 782**
25.5.1 Usage ... 782
25.5.2 Database and product versions ... 783
**25.6 Javascript ... 783**
25.6.1 JavaScript Editor ... 784
25.6.2 List of Javascript funtions ... 786
25.6.3 Explanations and examples ... 788
**25.7 Color selector ... 796**
**25.8 Display Embedded Data ... 798**
25.8.1 User interface ... 798
**25.9 Configuration Options ... 799**
25.9.1 General Configuration Options ... 799
25.9.2 Assembly Report and Editor ... 811
25.9.3 Dimension ... 816
25.9.4 Export TDP ... 823
25.9.5 Forms and Editor ... 820
25.9.6 Function Manager ... 799
25.9.7 Inspect and Editor ... 825
25.9.8 Inspect 3D and Editor ... 799
25.9.9 Library and Editor ... 830
25.9.10 MBD Tables ... 799
25.9.11 Material and Editor ... 849
25.9.12 Name Generator and Editor ... 854
25.9.13 Orientation Manager ... 855
25.9.14 Parameter and Editor ... 855
25.9.15 Quick Access and Editor ... 863
25.9.16 Stack Note ... 867
25.9.17 Stack Note 3D ... 869
25.9.18 UDF Forms and Editor ... 871
25.9.19 Utilities ... 873
25.9.19.1 General Utilities Configuration Options ... 873
25.9.19.2 3D Note Form ... 879
25.9.19.3 Annotation Info ... 880
25.9.19.4 Annotation Transfer ... 881
25.9.19.5 Close All Other Windows ... 880
25.9.19.6 Combined View Gallery ... 799

25.9.19.7 Command Control ... 910
25.9.19.8 Component Parameters ... 890
25.9.19.9 Convert Materials ... 891
25.9.19.10 Copy Component Parameter To Substitution Component Parameter ... 799
25.9.19.11 Create Point and Coordinate Systems ... 799
25.9.19.12 Create search.pro ... 895
25.9.19.13 Create Tolerance Table ... 901
25.9.19.14 CS Assembler ... 883
25.9.19.15 Edit Standard Texts ... 896
25.9.19.16 Export Points ... 892
25.9.19.17 Export Table To Excel and Export Table One-To-One To Excel ... 899
25.9.19.18 Export Table to CSV ... 898
25.9.19.19 Extend Relations ... 882
25.9.19.20 Find Contact Surfaces ... 890
25.9.19.21 Find References ... 799
25.9.19.22 Full Backup ... 910
25.9.19.23 GTOL Text and Editor ... 886
25.9.19.24 Inspect Instance Note ... 799
25.9.19.25 Load Save Converter ... 891
25.9.19.26 Open Base Model ... 881
25.9.19.27 Open/Create Drawing ... 907
25.9.19.28 Select Contact Surfaces ... 891
25.9.19.29 Select Surfaces by Color ... 884
25.9.19.30 Set Component Combined View ... 799
25.9.19.31 Show Information ... 887
25.9.19.32 Show Pitch ... 896
25.9.19.33 Show Thread Size ... 885
25.9.19.34 Sort Combined Views ... 896
25.9.19.35 Toggle Symbol Variants ... 897
25.9.19.36 Work Dir Manager ... 911

**26 Variables 913**

**26.1 User input ... 913**
**26.2 Date and time information ... 913**
**26.3 Object information ... 914**
**26.4 Drawing information ... 916**
**26.5 Parameter information ... 916**
**26.6 Text operations for variables ... 917**
**26.7 Comment texts for Inspect ... 918**

**27 Regular Expressions 920**

**28 Available Arithmetic Operations 924**

**29 FAQ 925**

30 Copyright 928

# 1 Product overview

GENIUS TOOLS for Creo are a collection of extension modules for Creo Parametric.

These extension modules are contained in the products GENIUS TOOLS® Library, GENIUS TOOLS® Parameter, GENIUS TOOLS® MBD for ISO GPS and / or Startup TOOLS and cover different application scenarios optimally. The product package Startup TOOLS includes, in addition to the extension modules listed here, the configuration software GENIUS TOOLS Starter for managing work environments and Creo data packages.

See the following overview of the different GENIUS TOOLS products.

| Product | Startup TOOLS | GENIUS TOOLS Para-meter | GENIUS TOOLS Library | GENIUS TOOLS MBD for ISO-GPS |
|---|---|---|---|---|
| **Ribbon menu** | GENIUS TOOLS | | | GT MBD |
| **Extension modules (GENIUS TOOLS for Creo)** | | | | |
| **Parameter management** (*"Parameter"*) | ✓ | ✓ | | |
| **Creating bills of materials in assembly mode** (*"Assembly Report"*) | ✓ | ✓ | | |
| **Editing assembly parameters** | ✓ | ✓ | | |
| **Library management** (*"Library"*) | ✓ | | ✓ | |
| **Importing external model data** (*"Library Data Importer"*) | ✓ | | ✓ | |
| **Form-driven models** (*"Forms"*) | ✓ | | ✓ | ✓ |
| **Form-driven UDFs** (*"UDF Forms - Design Tools"*) | ✓ | | ✓ | |


© 2026 INNEO Solutions GmbH

| Product | Startup TOOLS | GENIUS TOOLS Para-meter | GENIUS TOOLS Library | GENIUS TOOLS MBD for ISO-GPS |
|---|---|---|---|---|
| **Ribbon menu** | GENIUS TOOLS | | | GT MBD |
| **Material selection** (*"Material"*) | ✓ | ✓ | ✓ | |
| **Transferring model properties** (*"Value Transfer"*) | ✓ | ✓ | ✓ | |
| **Inspection and change symbols for drawings** (*"Inspect"/"Inspect Revision"*) | ✓ | ✓ | ✓ | |
| **Create a table with notes in 2D** *("Stack Note")* | ✓ | | ✓ | |
| **Multiple dimension editing** (*"Dimension"*) | ✓ | ✓ | ✓ | ✓ |
| **Ring menu and Mapkey management** (*"Quick Access"*) | ✓ | ✓ | ✓ | ✓ |
| **Name Generator** | ✓ | ✓ | ✓ | ✓ |
| **Creating and managing combined views** *("Function Manager")* | | | | ✓ |
| **Create orientations** *("Orientation Manager")* | | | | ✓ |
| **Inspection and change symbols for parts and assemblies** (*"Inspect 3D"/"Inspect Revision 3D"*) | | | | ✓ |
| **Create a table with notes in 3D** *("Stack Note 3D")* | | | | ✓ |
| **Create tables in 3D** *("MBD Tables")* | | | | ✓ |
| **Export 3D models** *("Export TDP")* | | | | ✓ |


© 2026 INNEO Solutions GmbH

| Product | Startup TOOLS | GENIUS TOOLS Para-meter | GENIUS TOOLS Library | GENIUS TOOLS MBD for ISO-GPS |
|---|---|---|---|---|
| **Ribbon menu** | GENIUS TOOLS | | | GT MBD |
| **View information about annotations in combined views** *("MBD Info")* | | | | ✓ |
| **Further useful tools** *("Utilities")* | ✓ | ✓ | ✓ | ✓ |
| **Change working directory** *("Work Dir Manager")* | ✓ | ✓ | ✓ | |
| **Close all other windows** | ✓ | ✓ | ✓ | |
| **Open Base Model** | ✓ | ✓ | ✓ | |
| **Extend Relations** | ✓ | ✓ | ✓ | |
| **CS Assembler** | ✓ | ✓ | ✓ | |
| **Command Control** | ✓ | ✓ | ✓ | |
| **Extend Dimension Functions** | ✓ | ✓ | ✓ | |
| **Create offset point and offset coordinate systems** | ✓ | ✓ | ✓ | |
| **Select Surfaces by Color** | ✓ | ✓ | ✓ | ✓ |
| **Full Backup** | ✓ | ✓ | ✓ | |
| **Show Thread Size** | ✓ | ✓ | ✓ | |
| **Show Information** | ✓ | ✓ | ✓ | |
| **Javascript-Editor** | ✓ | ✓ | ✓ | |
| **Copy Component Parameter To Substitution Component Parameter** | ✓ | ✓ | ✓ | |


*© 2026 INNEO Solutions GmbH*

| Product | Startup TOOLS | GENIUS TOOLS Para-meter | GENIUS TOOLS Library | GENIUS TOOLS MBD for ISO-GPS |
|---|---|---|---|---|
| **Ribbon menu** | GENIUS TOOLS | | | GT MBD |
| **Configuration Utility** | ✓ | ✓ | ✓ | ✓ |
| **Load Save Converter** | ✓ | ✓ | ✓ | |
| **Export Points** | ✓ | ✓ | ✓ | |
| **Create Search.pro** | ✓ | ✓ | ✓ | |
| **Split Table** | ✓ | ✓ | ✓ | |
| **Default Text Editor** | ✓ | ✓ | ✓ | |
| **Show Pitch** | ✓ | ✓ | ✓ | |
| **Toggle Symbol Variants** | ✓ | ✓ | ✓ | |
| **Export Table One-To-One to Excel** | ✓ | ✓ | ✓ | |
| **Export table to CSV** | ✓ | ✓ | ✓ | |
| **Export Table to Excel** | ✓ | ✓ | ✓ | |
| **Create Tolerance Tables** | ✓ | ✓ | ✓ | |
| **Open / Create Drawing** | ✓ | ✓ | ✓ | |
| **Referencing of all available surfaces to the general tolerance** *("GenTOL References")* | | | | ✓ |
| **Annotation Info** | | | | ✓ |
| **Annotation Transfer** | | | | ✓ |
| **Combined View Gallery** | | | | ✓ |


*© 2026 INNEO Solutions GmbH*

| Product | Startup TOOLS | GENIUS TOOLS Para-meter | GENIUS TOOLS Library | GENIUS TOOLS MBD for ISO-GPS |
|---|---|---|---|---|
| **Ribbon menu** | GENIUS TOOLS | | | GT MBD |
| **Extended editing of annotation texts** *("GTOL Text")* | | | | ✓ |
| **Find Contact Surfaces** | | | | ✓ |
| **Find References** | | | | ✓ |
| **Inspect Instance Note** | | | | ✓ |
| **Select Contact Surfaces** | | | | ✓ |
| **Sort Combined Views** | | | | ✓ |
| **Set Component Combined View** | | | | ✓ |

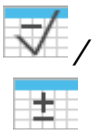
# 1.1 Full details of all functions

| Icon | Component |
|---|---|
|  | **Parameter management**[465] *("Parameter")*<br>*Parameter* creates uniform meta data for use in an automated generation of bills of materials, as well as for precalculations and integration to ERP systems. |
|  | **Creating bills of materials in assembly mode** (*"Assembly Report"*)<br>*Assembly Report* outputs freely definable bills of materials in assembly mode and generates position numbers as component parameters which can be further used in Windchill, Creo View and Creo drawing mode. |
|  | **Editing assembly parameters**[656]<br>This function generates component parameters in assemblies. Different component parameter values can be assigned for component models with the same name. |


© 2026 INNEO Solutions GmbH

| Icon | Component |
|---|---|
|  | **Library management**[293] (*"Library"*)<br>*Library* is used to make Creo objects available from a library and to allow certain actions for each object such as copy or copy-paste into a model. |
| | Importing external model data (*"Library Data Importer"*)<br>This component imports external model data, e. g. from PTC Windchill, into a library for *Library*. It is a program which has to be installed separately. |
|  | **Form-driven models**[123] (*"Forms"*)<br>*Forms* generates user-defined form masks that allow Creo users to more quickly customize the properties of parts and assemblies (PRT/ASM). |
|  | **Form-driven UDFs / Design TOOLS**[601] (*"UDF Forms"*)<br>*UDF Forms* allows you to define features once according to standards and then easily place them in the design process. |
|  | **Multi-dimensional editing**[92] (*"Dimension"*)<br>The component *Dimension* enables the simultaneous and fast editing of dimensional values and names of a feature, component, assembly or its subcomponents. |
|  | **Material selection**[388] (*"Material"*)<br>With *Material* Creo users can select materials based on various properties and assign them to a model or body. |
|  | **Ring menu and mapkey management**[544] (*"Quick Access"*)<br>*Quick Access* is a ring menu that contains shortcuts to various commands in different Creo modes as well as individually configurable mapkeys (macros). |
|  | **Transferring model properties**[594] (*"Value Transfer"*)<br>With *Value Transfer* numerous values in dimensions and parameters, as well as material definition files of subcomponents, can be edited in one step in assembly mode. |


*© 2026 INNEO Solutions GmbH*

| Icon | Component |
|---|---|
|  | **Name generator**[440]<br>This component assigns consecutive numbers to file names of parts, sheet metal parts and assemblies. |
|  | **Inspection and change symbols for drawings**[187] *("Inspect")*<br>With *Inspect*, you can place, number, and manage inspection and change symbols on drawings and create a revision history of all symbols. |
|  | **Creating and managing combined views**[167] *("Function Manager")*<br>Use *Function Manager* to create and manage combined views. You can create functions and assign the combined views directly to those functions. |
|  | **Exporting 3D models to PDFs**[104] *("Export TDP")*<br>*Export TDP* allows you to export technical data packages for viewing 3D models in PDF format. |
|  | **Inspection and change symbols for parts and assemblies**[243] *("Inspect 3D")*<br>With *Inspect 3D*, you can place, number, and manage inspection and change symbols on parts and assemblies and create a revision history of all symbols. |
|  | **Creating tables in 3D**[422] *("MBD Tables")*<br>With the submodule MBD Surface Finish Table, you can place and manage tables of surface finishes in 3D.<br>With the submodule MBD Size Table, you can place and manage tables of sizes in 3D. |
|  | **Create and detect existing orientations**[460] *("Orientation Manager")*<br>*Orientation Manager* allows you to detect and define existing model orientations. |
|  | **Create notes in a table in 3D**[582] *("Stack Note 3D")*<br>With *Stack Note 3D*, you can import comments from a database and place the selection in a table next to the model. |
|  | **Create notes in a table in 2D**[568] *("Stack Note")*<br>With *Stack Note*, you can import comments from a database and place the selection in the form of a table next to a drawing. |


© 2026 INNEO Solutions GmbH

| Icon | Component |
|---|---|
|  | **View information about annotations in combined views**[413] *("MBD Info")* Shows an overview of the annotations available in the combined views. |
| | **Further useful tools**[633] *(„Utilities")* |
|  | **3D Note Form**[635] Enables quick modification of dimension and parameter values in the notes on the model via editable form masks. |
|  | **Close all other windows**[652] Closes all Creo windows except the current and the main window. |
| | **Command Control**[655] Hides Creo Parametric ribbon menu commands (all commands). |
|  | **Configuration Utility** [765] Provides an interface for editing all configuration options and saving them to the correct locations. |
|  | **Copy Component Parameter To Substitution Component Parameter**[663] Copies component parameters of a part in the master view to substituted component parameters for a simplified part. |
|  | **Create point and coordinate systems**[671] make it possible to create a coordinate system or a point at the same location (point, vertex or coordinate system) within a part or an assembly with reference to a newly selected reference coordinate system. |
|  | **Create Search.pro**[665] Creates a project or assembly specific search path file (search.pro) based on customizable configuration settings. |
|  | **Create Tolerance Tables**[669] Creates a tolerance table at a freely selectable location on a drawing using pre-defined tolerances. |
|  | **CS Assembler**[673] Automates the assembly of components into an assembly using defined coordinate systems. |
|  | **Default Text Editor**[676] Creates a database of multilingual descriptions that can be used in multiple components to fill dialog boxes. |
|  | **Export Points**[679] Outputs reference points (single points or point fields) or dynamically generated curve points (X-Y-Z values) to a PTS or DAT file. |


© 2026 INNEO Solutions GmbH

| Icon | Component |
|---|---|
|  | **Export Table One-To-One to Excel**[686] Creates an Excel file without using a template. |
|  | **Export Table to CSV**[689] Exports an existing drawing table or fills a template table. |
|  | **Export Table to Excel**[689] Fills a file template with parameters and data from cells of report tables, e.g. of a Creo drawing table. |
|  | **Extend Relations**[702] Adds more functions to model relations that can be used to create parameters for models and bodies. |
| | **Extend Dimension Functions**[707] Opens a dialog to quickly increase or decrease a selected dimension by a defined value. |
|  | **Full Backup**[712] Quickly saves the current model with all dependent data. |
|  | **JavaScript-Editor**[784] Enables faster developing and testing of JavaScript code. Opens in the respective component. |
|  | **Load Save Converter**[730] Saves Creo objects from previous Creo-, Wildfire- or Pro/ENGINEER versions in the currently used version. |
|  | **Open Base Model**[731] Opens geometric base models that are the reference source for a feature. |
|  | **Open / Create Drawing**[732] Opens a drawing if a drawing with the name of the model already exists, or creates a drawing. |
|  | **Select Surfaces by Color**[737] Selects surfaces of the same color or all uncolored surfaces. |
| | **Show Information**[741] Creates company-specific text in the Creo Parametric main window. |
|  | **Show Pitch**[742] Extends the displayed dimensions of a metric fine thread by the pitch value of the thread. |
|  | **Show Thread Size**[745] Extends the displayed dimensions by the pipe thread size. |
|  | **Split Table**[747] Split the table so that it is placed to the right or left of the table. |


© 2026 INNEO Solutions GmbH

| Icon | Component |
|---|---|
|  | **Toggle Symbol Variants**[749] Allows you to quickly click through all the variants of a grouped symbol. |
|  | **Work Dir Manager**[751] Automatically collects all directories used during the work process and enables the current working directory to be changed quickly. |
|  | **Annotation Info**[642] Shows an overview of the annotations available in the combined views. |
|  | **Annotation Transfer**[649] Transfer annotations from one combined view to one or more other combined views. |
|  | **Combined View Gallery**[653] Displays all combined views available in a model in an overview. |
|  | **Extended editing of annotation texts**[719] (*"GTOL Text"*)<br>Edits text on existing annotations for shape and position tolerances, create annotation templates for these texts and sets links to further information. |
|  | **Find Contact Surfaces**[709] Analysis tool that searches for the adjacent surfaces (=contact surfaces) to a surface. |
|  | **Find References**[710] Find references enables the automatic assignment of references to dimensions. |
|  | **Inspect Instance Note**[727] Inspection symbols and notes can be selected to add a note. This shows which inspection symbols belong to the respective pattern. |
|  | **Extended editing of annotation texts**[717] (*"GenTOL References"*) |
|  | **Select Contact Surfaces**[734] Displays the contact surfaces found with **Find Contact Surfaces**[709]**.** |
|  | **Sort Combined Views**[747] Retroactively sorts combined views alphabetically. |
|  | **Set Component Combined View**[738] In assembly mode, sets the combined views of the components. |


© 2026 INNEO Solutions GmbH

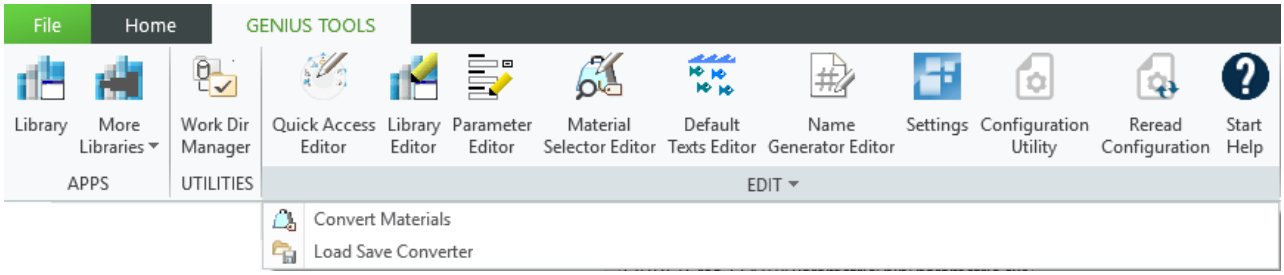
# 2 Ribbon menu GENIUS TOOLS

The ribbon menu *GENIUS TOOLS* contains various modules depending on the Creo mode you are in. If a module does not appear in the ribbon, check the steps described in FAQ 925.

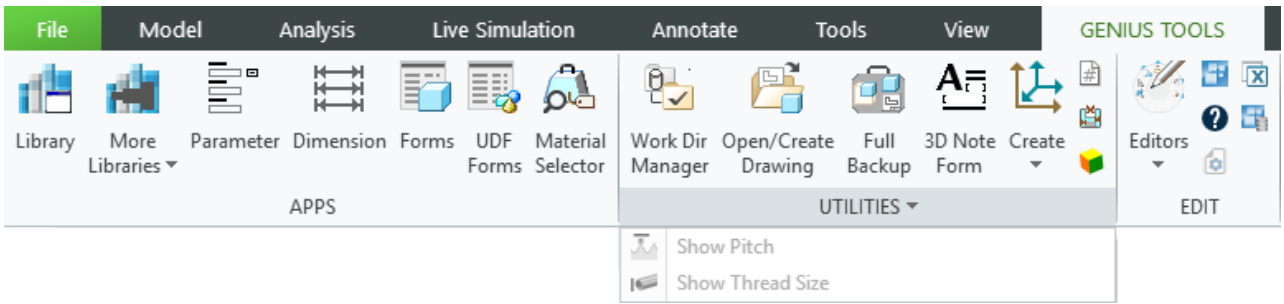
**Tip:** Click on a module in the figure to go to the function description.

Access to the Editors of *GENIUS TOOLS* modules can be restricted via configuration settings 765.

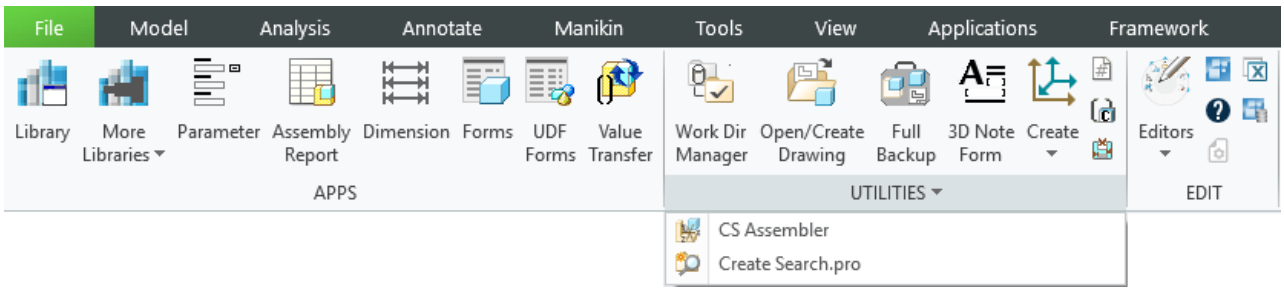
## Standby mode




## Part mode



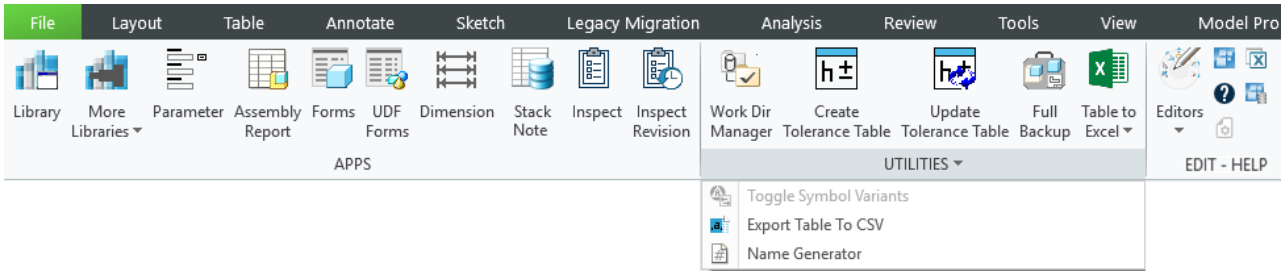

## Assembly mode





© 2026 INNEO Solutions GmbH

### Drawing mode




# 2.1 License-dependent functions

The following functions are limited to subscription licenses for either GENIUS TOOL® Library, GENIUS TOOLS® Parameter or Startup TOOLS. A complete list of all functions can be found in the product overview[15].

The following functions are available with a subscription license for *GENIUS TOOLS® Library*, *GENIUS TOOLS® Parameter* and *Startup TOOLS*.

| **Extension module (GENIUS TOOLS for Creo)** | **Description** | **Release** |
|---|---|---|
| Load Save Converter | Converts Creo objects from previous Creo-, Wildfire- or Pro/ENGINEER versions to the currently used version. | 7.0.0.0 |
| Inspect Revision | Stores all versions of inspection symbols on a drawing. Easily creates an overview of a revision history. | 7.0.0.0 |
| Open Base Model | Opens geometric base models that are reference sources for features with one click in the context menu of a feature. | 7.0.0.0 |
| Extend Relations | Adds functions to model relations that define parameters for models and – with Creo 7 – for bodies. | 7.0.0.0 - 8.0.0.0 |


© 2026 INNEO Solutions GmbH

| Extension module (GENIUS TOOLS for Creo) | Description | Release |
|---|---|---|
| Select Surfaces By Color | Surfaces of the same color can be selected with one click and be colored, measured or otherwise modified thereafter. | 7.0.1.0 |
| CS Assembler | Automates the assembly of components into an assembly using defined coordinate systems (CS). | 7.0.1.0 |
| Extended Dimension Functions | Model dimensions can be increased or decreased very quickly with the mouse in a defined increment in the graphics window. | 7.0.2.0 |
| Copy Component Parameter To Substitution Component Parameter | If a value has been assigned to a component parameter for a part in the master representation, this value can be copied to the component parameter for a simplified part. | 7.0.2.0 |
| Toggle Symbol Variants | If a grouped symbol has variants in the first level, you can switch between those variants more quickly. | 7.0.2.0 |
| Multibody to Assembly | Converts multiple bodies into assemblies (Creo Advanced Assembly Extension (AAX) is required.) | 8.0.0.0 - 12.0.2.0 |
| Work Dir Manager | Automatically collects all directories used during the work process and enables the current working directory to be changed quickly. | 8.0.0.0 |
| Full Backup | Quickly backs up the current model with all dependent data. | 8.0.0.0 |
| Command Control | Creo Parametric ribbon commands / commands can be hidden or deactivated. | 8.0.1.0 |


© 2026 INNEO Solutions GmbH

| Extension module (GENIUS TOOLS for Creo) | Description | Release |
|---|---|---|
| Feature Regeneration Profiler | Regenerates models and displays regeneration times for each feature. | 10.0.1.0 - 12.0.2.0 |
| Create offset point and offset coordinate systems | Creates a coordinate system at the same location (point, corner point, or coordinate system) within a part or assembly, but with reference to a newly selected reference coordinate system. | 12.0.0.0 |
| Stack Note | Imports notes from a multilingual database; the selection of notes is summarized in a note table and can be placed next to a drawing. | 12.0.1.0 |
| Split Table | Splits a table so that it is placed to the right or left of it and can be repositioned. | 12.0.1.0 |
| Inspect Instance Note | Inspection symbols and notes can be selected to add a note. This shows which inspection symbols belong to the respective pattern. | 12.0.1.0 |
| MBD Info | Shows an overview of the annotations available in the combined views. | 13.0.0.0 |
| Set Component Combined View | In assembly mode, sets the combined views of the components. | 13.0.0.0 |

## Deactivating functions which require a subscription license

When using a permanent license, users will receive a warning that the functions listed above cannot be opened. You can avoid receiving such a warning by deactivating these functions, i. e. by setting the corresponding configuration option ('start switch') to *0*.


© 2026 INNEO Solutions GmbH

| Configuration option (Start switch) | Function |
|---|---|
| gt_start_inspect_revision | *Inspect Revision* |
| gtu_start_csassembler | CS Assembler |
| gtu_start_copyCParamToSubsCParam | *Copy Component Parameter To Substitution Component Parameter* |
| gtu_start_extendedDimensionFunctions | *Extended Dimensions Function* |
| gtu_start_featureRegenerationProfiler | *Feature Regeneration Profiler* |
| gtu_start_fullbackup | *Full Backup* (Model and data backup) |
| gtu_start_loadSaveConverter | *Load Save Converter* |
| gtu_start_openGeomOrigin | *Open base model* |
| gtu_start_relationExtension | *Extend relations* |
| gtu_start_selectSurfacesByColor | *Select surfaces by color* |
| gtu_start_toggleSymbolGroups | *Toggle symbol variants* |
| gtu_start_work_dir_manager | *Work Dir Manager* (Managing the used working directories) |
| gtu_start_command_control | *Command Control* |
| gtu_start_datum_creation | *Create offset point and offset coordinate systems* |
| gtu_start_split_table | *Split Table* |
| gtu_start_inspect_instance_note | *Inspect Instance Note* |
| gt_start_mbdinfo | *MBD Info* |
| gtu_start_set_component_combined_view | *Kombinierte Ansichten der Komponenten setzen* |


© 2026 INNEO Solutions GmbH

# 3 Ribbon menu GT MBD

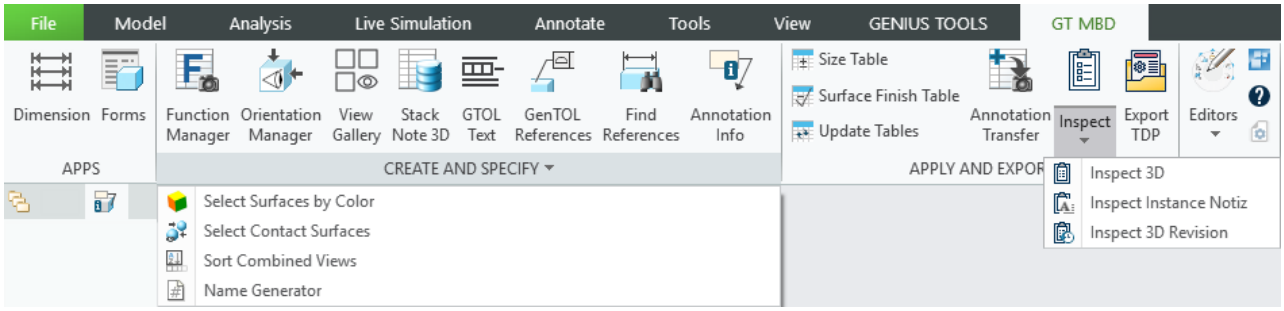
The ribbon menu *GT MBD* appears by default when you start Creo Parametric if you have installed GENIUS TOOLS® MBD for ISO GPS individually.

If you install GENIUS TOOLS® MBD for ISO GPS together with Startup TOOLS, the Startup Tools setup runs. In this case, you still need to activate the *GT MBD* ribbon: Set the configuration option `gt_start_mbd_management_tools` to 1.

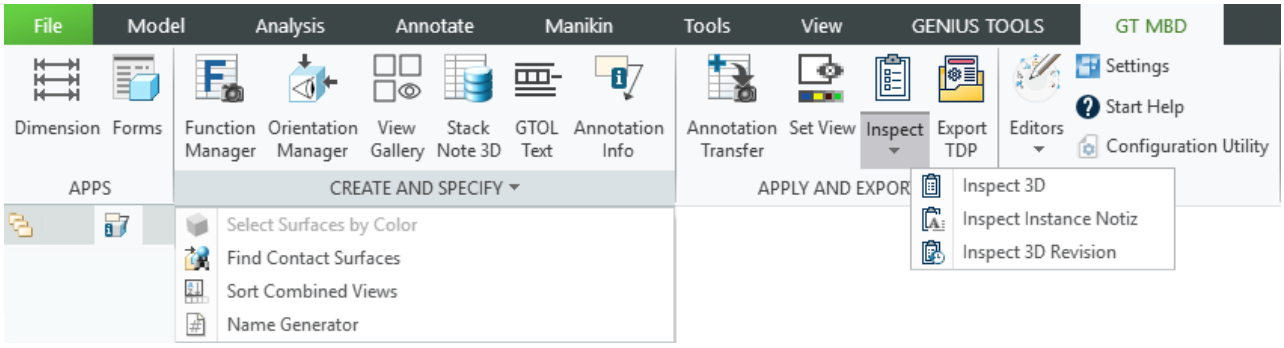
**Tip:** Click on a module in the figure to go to the function description.

Access to the Editors of *GENIUS TOOLS* modules can be restricted via configuration settings[765].

## Part mode

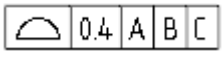



## Assembly mode




# 3.1 Glossary

**Basic frame (also: general frame)**




The basic frame is the first frame in a model and is typically specified with general tolerances according to ISO 22081. It is the basis for the geometric tolerances specified in a model. Other frames can be derived from this basic frame.


© 2026 INNEO Solutions GmbH

**Basic function (B)**
Function type that contains generalized tasks of a product. It is used to provide a general overview of the model and contains information such as metadata, welding specifications, or production details. The predefined basic functions included in the delivery are listed under Best Practice[36].

**Combined View (also: Combined State, Functional View)**
Collection of different states of a model that fulfills a specific function, see also Creo Parametric documentation, e. g. About combined views.

**Function (F); (also: Product Function)**
Description of a task a product performs. When the term *function* is used without further specification, it refers to a product function. These are functions provided by the physical structure of the product (surfaces, form elements, etc.). The geometric elements of these functions (usually surfaces) are precisely defined by ISO GPS.

In general, a function is represented by at least one combined view and one construction group. For simple parts, the product function can also be specified in the combined view of the basic function B01-Master.

*GENIUS TOOLS MBD for ISO GPS* contains several templates for function types to maintain clarity:

– **Basic Function (B)**

– **Orientation Function (O)**

– **Process Function (P)**

You can define additional function types to be used within the company.

**Functional Component (also: functional object, part of a function)**
Implementation of a function using various properties and methods. Functional objects are created and edited in Function Manager[167].

**Functional Construction and Specification in 3D**
CAD-independent workflow for the specification of MBD models (e. g. according to ISO GPS). The goal of functional construction and specification in 3D is to create models that are legible by both humans and machines.

The geometry of a 3D model is mentally "broken down" into its function-related geometry elements. These function-related geometry elements are displayed in separate views (= combined views). Colors help to quickly identify the corresponding geometries. The specification / tolerancing is then carried out in such a combined view. This procedure also complies with the ISO GPS principles of specifying geometry elements according to their function. See Best Practice[32].


© 2026 INNEO Solutions GmbH

**ISO GPS**
System of standards that *GENIUS TOOLS MBD for ISO GPS* help you to implement. Information about these standards can be found here, for example: www.wikipedia.org/wiki/Geometric_dimensioning_and_tolerancing (ASME) and www.wikipedia.org/wiki/Geometrical_Product_Specification_and_Verification (ISO).

**MBD (Model Based Definition)**
Design method where the 3D model is the source of all product information. This means that all information previously found on the drawing is contained in the 3D model. Combined views allow this information to be quickly identified.

**MBE (Model Based Enterprise)**
All product lifecycle processes are based on *one* 3D model.

**Orientation Function (O)**
Function type that focuses on the orientation of the model in the Combined View.

**Process Function (P)**
Function type that represents individual process steps, such as manufacturing or assembly steps.

**Specification Function (S)**
Function type that represents general specifications, such as installation spaces or physical constraints.

**Size (according to ISO 14405)**
Describes diameter dimensions and dimensions of opposing surfaces. Sizes can be further specified with tolerances.

**TDP (Technical Data Package)**
Set of electronic files for 3D product description for cross-departmental information retrieval. A TDP contains the 3D model of the product and can be visualized with Export TDP104. The TDP is then available to you as a 3D PDF in which you can view, rotate and zoom the 3D model.

**TED (Theoretic Exact Dimension, see DIN EN ISO 1101)**
Dimension within a rectangular frame that must not be tolerated and for which no general tolerances apply.

## 3.2 Best Practice

*GENIUS TOOLS MBD for ISO GPS* modules support you in **functional construction and specification in 3D**, making it easier to implement the ISO GPS standards. These modules and the help described here do not replace knowledge of the ISO GPS standards. However,


© 2026 INNEO Solutions GmbH

this chapter will provide you with some help to integrate these standards more easily into your workflow.

The terms introduced in the glossary[30] are used here to give you an idea of where these terms are relevant and the specific context in which they are used.

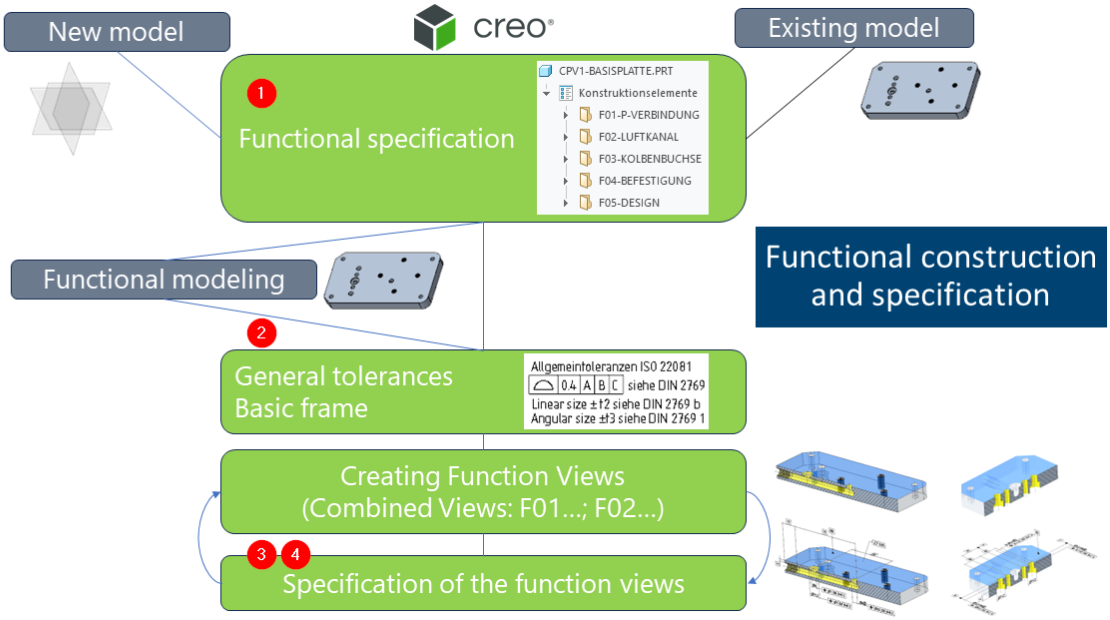
## Effective range

The standards that apply to a model are those in effect at the time of its creation. Older CAD records and drawings must be updated to current standards if a model is to be reused and integrated into a current project. This is the case, for example, if an entry was made on the drawing that refers to a new standard. In this case, the drawing must be completely updated to the new standards.

**Please note:** The presence of at least one GPS specification in a model indicates that the model was created in accordance with the ISO GPS standards.

## Workflow

The following figures illustrate the workflow and dependencies of the required steps:



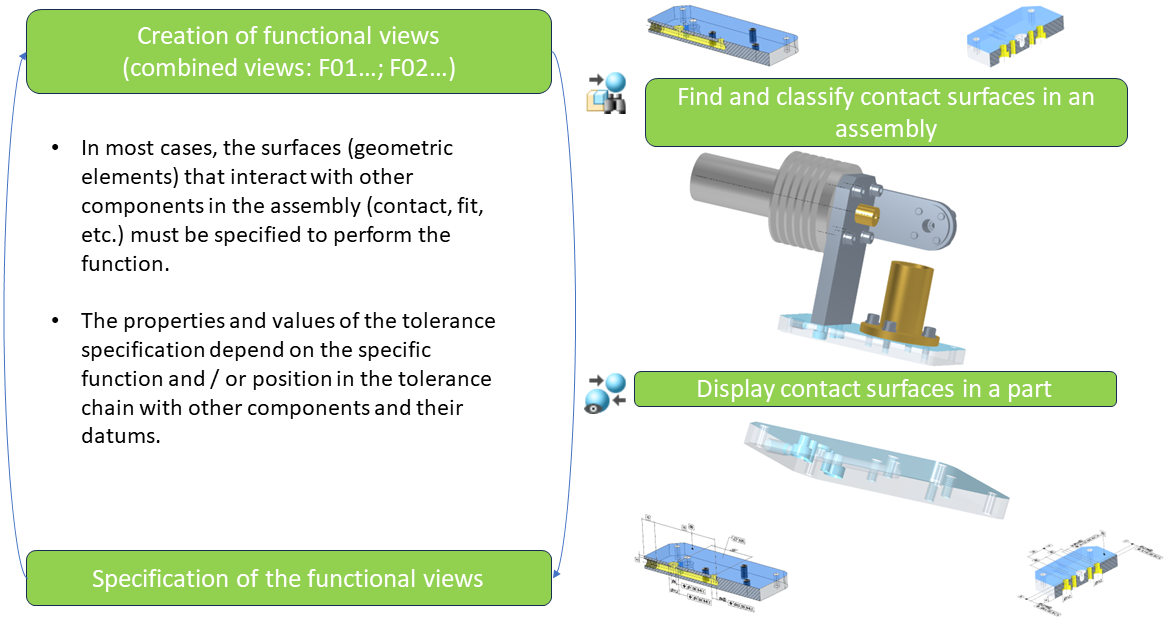

*Workflow for creating and editing an ISO-GPS compliant component*


© 2026 INNEO Solutions GmbH




*Workflow for creating MBD models*

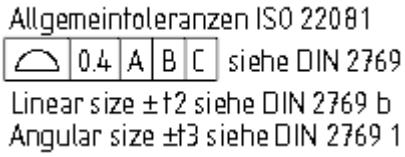

*© 2026 INNEO Solutions GmbH*




*Workflow for function specification*

The workflow applies to all models, whether new or existing.

1. The specification of the Functions[31] to be fulfilled by a model is the basis for the creation of a new model and for the evaluation of an existing model. The features of a function can be combined into construction groups.
2. The next step is modeling. The valid general tolerances must be defined and described in the Basic Frame[30]. All datums in the model must build on one another.




3. After modeling, combined views can be created as components of functions[31]. You can use the *Function Manager*[167] to create combined views in specified functions and assign features to them.

   Most functions are categorized into basic functions[31] and other product functions.[31] The specification of functions and their implementation through combined views[31] are always interlinked and interdependent.

   3.1. *Default All* is the default combined view in Creo Parametric and cannot be deleted. In this view, the model is inserted into an assembly and should always look as realistic as possible. Additional elements in this view should be hidden by default. The next combined view should be an *overview function B01-Master*, in which the model is displayed with the base frame and the general tolerances.
   Further combined views can be created for each function to be displayed.
   It is also recommended to standardize the combined views by a naming scheme.
   *GENIUS TOOLS MBD for ISO-GPS* provides some predefined combined views with


© 2026 INNEO Solutions GmbH

consecutive numbering to start with, which you can use and customize. These naming conventions are structured according to the following scheme: `TypeNumberName`, e. g. `B01-Master`.
The following table lists the predefined basic functions[31].

| Basic functions | PRT | ASM | Description |
|---|---|---|---|
| B00-Work | ✓ | ✓ | Work region |
| B01-Master | ✓ | ✓ | Model overview |
| B02-Note | ✓ | ✓ | "Title block", notes on text information |
| B03-Datum | ✓ | ✓ | All datums |
| B04-Gt-Ref | ✓ | ✓ | All GPS specifications |
| B05-Surf-Finish | ✓ | ✓ | All surface specifications |
| B06-Design-Intent | ✓ | ✓ | Design models (skeletons) |
| B07-Weld | ✓ | ✓ | Welding specification |
| B08-Ecad | ✓ | ✓ | ECAD components, wiring |
| B09-Piping | ✓ | ✓ | Piping components, piping |
| B10-Mold | ✓ | ✓ | Specific molding data |
| B11-Cast | ✓ | ✓ | Specific casting data |
| B12-MFG | ✓ | ✓ | Specific manufacturing data |
| B13-Simulation | ✓ | ✓ | Specific analysis data |
| B14-Bom | | ✓ | Bill of materials data |
| B15-Explosion | | ✓ | Exploded views |
| B16-Kinematic | | ✓ | Movable component relationships |
| B17-Assembly | | ✓ | Assembly / installation information |
| B18-Interchange | ✓ | ✓ | Interchange references |
| B19-Sheetmetal | ✓ | | Specific sheet metal information / folding |


© 2026 INNEO Solutions GmbH

3.2. Functions[170] and Geometric Dimensioning and Tolerancing[726] can be used to store information and links, e. g. instructions on how to model / specify a part in order to work according to the standard.

3.3. Use *Annotation Info*[642] to get an overview of the annotations in the combined views. With *Annotation Transfer*[649], you can transfer annotations from one or more combined views to another combined view.

**Please note:** Anything that is required to perform the function and deviates from the general tolerances must be specified more precisely. Typically, these are geometric elements (usually surfaces) that interact with other components.

4. Surfaces that are adjacent to other surfaces in an assembly must be specified. You can use Find Contact Surfaces[709] to determine which surfaces of a part in an assembly are considered as contact surfaces. This allows you to examine multiple parts of the same assembly. The determined information is stored in the examined part[734].

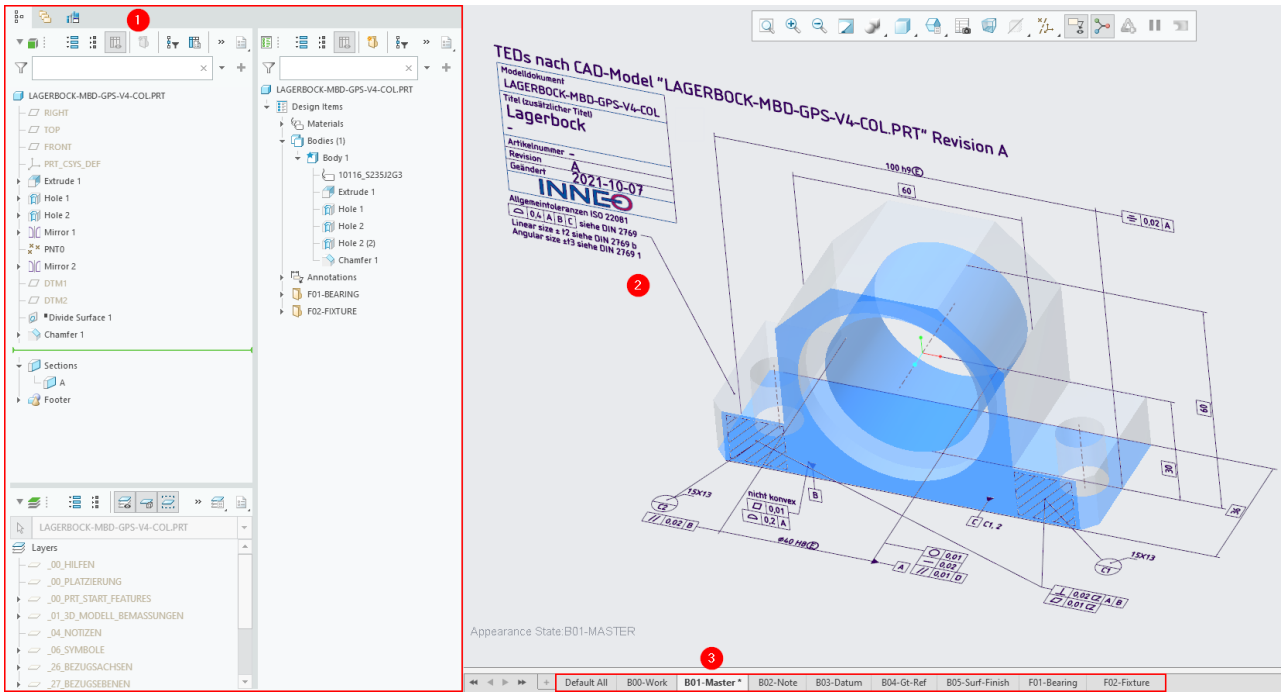
   4.1. It is recommended to work with a cross-model color scheme for surface specification. *GENIUS TOOLS MBD for ISO-GPS* provides a predefined color scheme. You can work with this scheme and customize it. For this purpose, special coloring functions[175] are available that can be used to create different color representations in the combined views with just a few mouse clicks. The following colors cover the most common application areas:

| Color | Color name | Explanation |
|---|---|---|
|  | body | Semi-transparent surface of a primitive |
|  | datum | Datum surface of the model frame |
|  | function | Functional surfaces |
|  | surface finish | Surfaces with roughness data |
|  | contact | contact surfaces |

# Work surface

The following figure shows an example of an MBD model specified according to ISO-GPS, including the tree structures that are helpful for functional construction and specification:


© 2026 INNEO Solutions GmbH

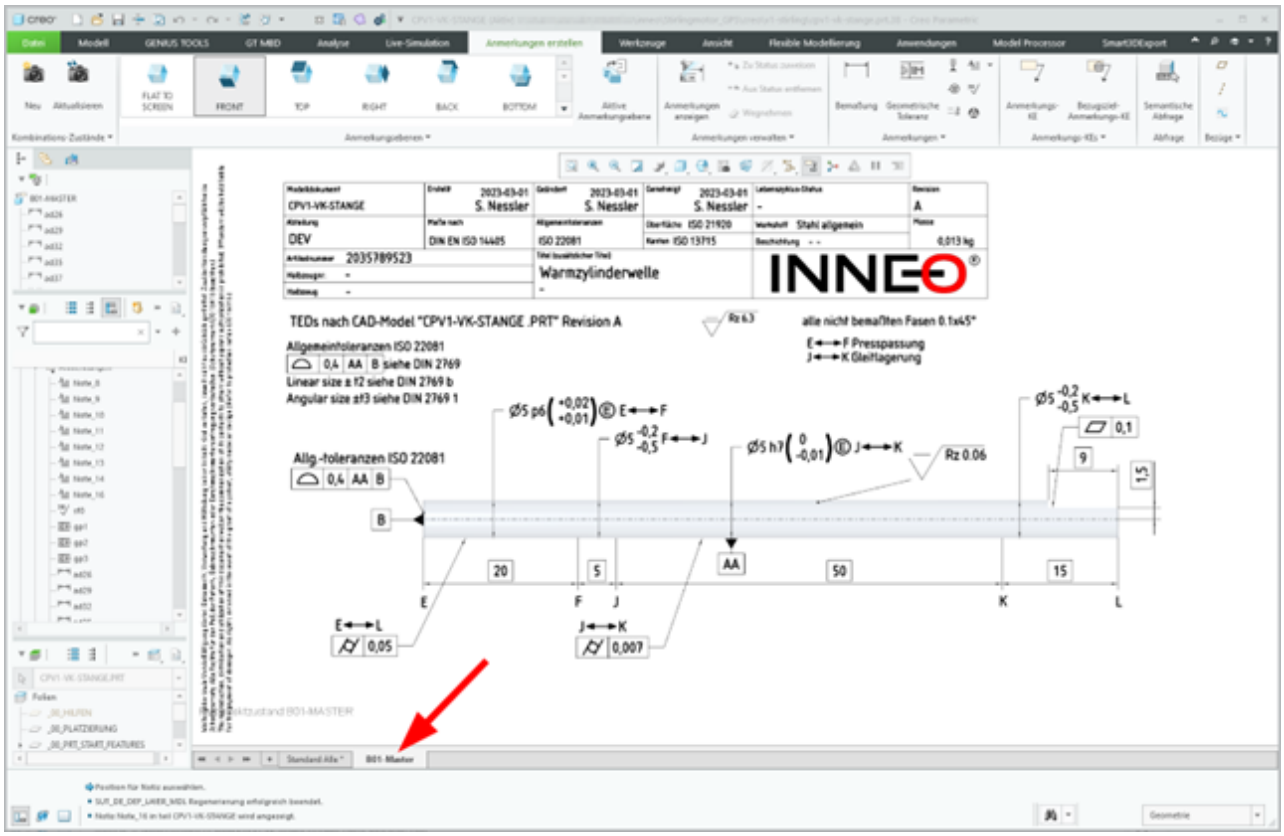



*Example of an user interface with overview function B01-Master*

1. **Management of the functions with the following tree structures**
   - Model tree
   - Construction tree for arranging functions
   - Annotation tree to view annotations and, if necessary, copy them to other combined views (since they are not displayed in the model tree)
2. **Basic frame**
3. **Combined views**

# Handling simple models

To minimize the effort of creating MBD models, it is not necessary to create multiple combined views for simple models. It is recommended that each model has a basic function *B01-Master*. All further details / specifications can be made in this view.




*Example of simple ISO GPS compliant model structure*


© 2026 INNEO Solutions GmbH

# 4 What's new in release 13

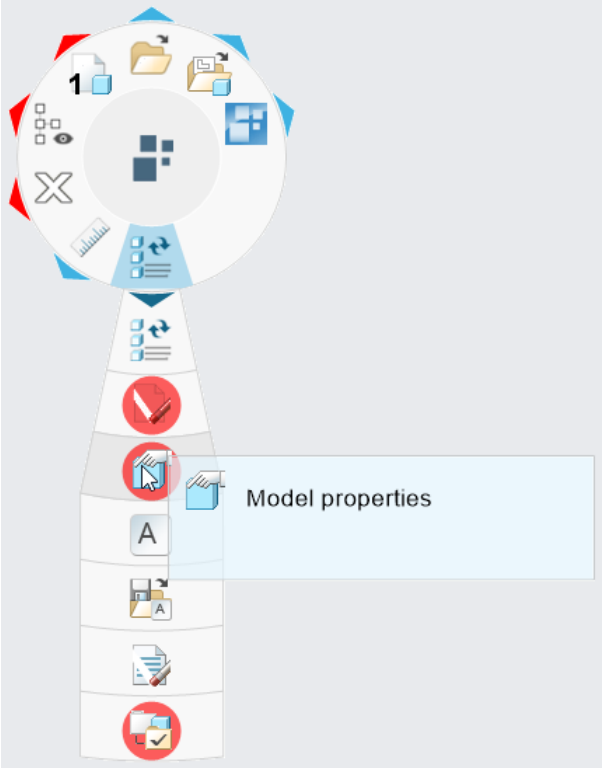
The following functions have been added to the GENIUS TOOLS add-on applications for Creo that are contained in the products GENIUS TOOLS Library and GENIUS TOOLS Parameter and GENIUS TOOLS MBD for ISO GPS.

**Please note:** If you would like to search for functions of a module across releases, enter the name of the module in the search window in the **Update Advisor** (https://updateadvisor.inneo.com). The Update Advisor also lists all changes to configuration options and files in the supplied data packages.

## 4.1 Quick Access

### Quick Access: Searching for elements (13.0.0.0)

You can search for elements in *Quick Access*. Enter the letters you are looking for, e. g. "mo", "model". Categories containing search results will be highlighted in red. Click on an arrow to view the search results it contains, which are also highlighted in red. For clarity and performance reasons, only one category can be expanded at a time.




*Results for the sample search for "mo"*


© 2026 INNEO Solutions GmbH

# 5 Assembly Report

*Assembly Report* supports you in creating reports, such as a bill of materials (BOM), in assembly mode. Reports are defined using a graphical editor. Any number of report definitions can be created as templates and applied to individual assemblies.

*Assembly Report* is available in assembly mode and for drawings of assemblies with the following features:

1. Table display with different display modes
   - single-level bill of materials
   - bulk bill of materials
   - multi-level bill of materials
   - multi-line cells
   - multiple parameters per cell
2. Usable column values
   - assembly and part parameters
   - assembly component parameters
   - report parameters (file name, assembly level, model type, quantity, mass etc.)
   - position number (as assembly component parameter)
3. Position number assignment (if used in the BOM)
   - general start and increment value
   - several number ranges in one report, definable with specifiable criteria (e.g. standard parts start at 500, foundation parts at 800)
   - manually editable
4. Multiple filtering according to all parameters
5. Multiple sorting according to all parameters
6. Export reports
   - to Microsoft Excel (with template and cell/column assignment)
   - to CSV files

## 5.1 Fundamentals

This section contains a glossary of terms and information on the standard behavior of *Assembly Report*.


© 2026 INNEO Solutions GmbH

## 5.1.1 Glossary

**Single-level bill of materials**
Single-level bill of materials that contains all elements of an assembly.

**Head parameter**
Information displayed above the report table in *Assembly Report*. Creo model parameters as well as general information such as number, file extension or level can be used as head parameters.

**Bulk bill of materials**
Unstructured bill of materials that lists each part of an assembly with the quantity with which it will be used in the final product.

**Model parameter**
Model parameters are parameters taken directly from the current Creo assembly.

**Number range**
A number range is a defined sequence in a group of elements to be ordered. Depending on defined filters in report definitions, number ranges are used to assign item numbers to assembly components.

**Position parameter**
The position parameter is created in assemblies and parts and contains a sequential number (position number).

**Report**
A report in *Assembly Report* is a bill of materials (BOM) displayed in the report table. Reports can be exported to various file formats.

**Report definition**
Template for creating a bill of materials that defines the numbering of items, the selection of parameters and the sorting. The information is stored in an XML data structure.

**Report definition, external**
Information for the report definition, which is stored in an XML file, usually in the resource directory.

**Report definition, internal**
Information for the report definition, which is stored in the data structure of an assembly as embedded data[798].

**Multi-level bill of materials**
Multi-level BOM that contains all elements of an assembly as well as all elements of sub-assemblies.


© 2026 INNEO Solutions GmbH

## 5.1.2 Standard behavior

*Assembly Report* supports you in creating various reports in Creo's assembly mode. The report definitions can be stored both internally (in the assembly data structure) and externally as XML files (e.g. in the resource directory of *GENIUS TOOLS* for Creo).

Whether internal or external report definitions are used depends on the configuration. External report definitions can be used for exchange and transfer to other assemblies.

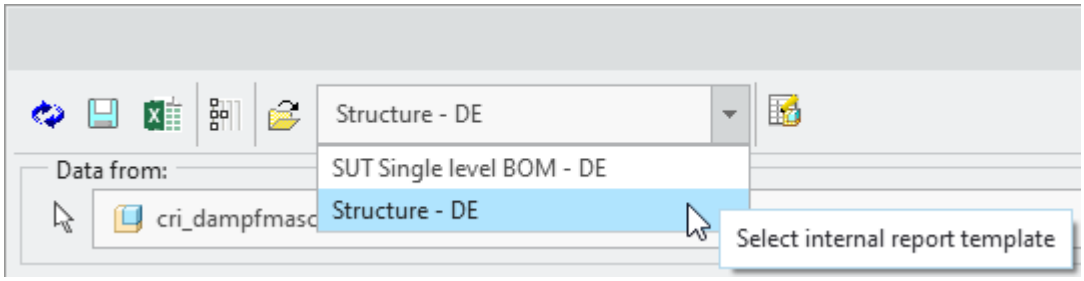
### First use in an assembly

After starting *Assembly Report* the external standard report definition is used (defined in the configuration option `gta_default_file`). The same report definition is used for each assembly.

When saving a report, the report definition is saved in the assembly. Each time Assembly Report is opened with this assembly, the report definition saved in the model is used.

### Subsequent use in an assembly

Each time Assembly Report is called, the assembly is searched for internally stored report definitions. You can select already loaded report templates via the drop-down menu. Additional report templates can be added via *Open external report template from file*. A selection of predefined templates is available there. You can use them or define your own template.




### Changing the report definition

**External usage**
External report definitions are automatically saved in an assembly as soon as a report based on them is saved. Internal definitions of the same name are overwritten.

**Warning**: The defined name in the XML file is used, not the file name.

**Internal usage**
As soon as an internal report definition is applied, the display changes. If several internal report definitions exist, you can define a definition as the internal standard using GENIUS TOOLS Assembly Report .


© 2026 INNEO Solutions GmbH

## Copy mechanism

If an external report definition is applied to an assembly, it is automatically copied and saved into the model. (See Embedded data 798.) This means that a model will be changed after changes to the report definition have been made, even without any changes on the model itself.

This behavior can be deactivated with the configuration option `gta_save_xml_in_mdl`, i. e. report definitions remain available only in the external XML files. (Default: 1=activated)

**Warning:** The configuration option has no effect on report definitions already contained in the model.

## Updating report definitions

If you make changes to external report definitions, *Assembly Report* can automatically apply these changes, see Automatic update of report definitions. 79

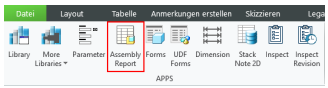
# 5.2 Usage

In this section you will find information about the use of *Assembly Report*. The general structure of the program is explained.

## Starting the program: Assembly and Drawing mode

To start *Assembly Report*  , click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section or by using *Quick Access* (key [<]).

*Assembly Report* is available in assembly mode and for drawings of assemblies.




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*



*Starting in an open assembly with existing report*

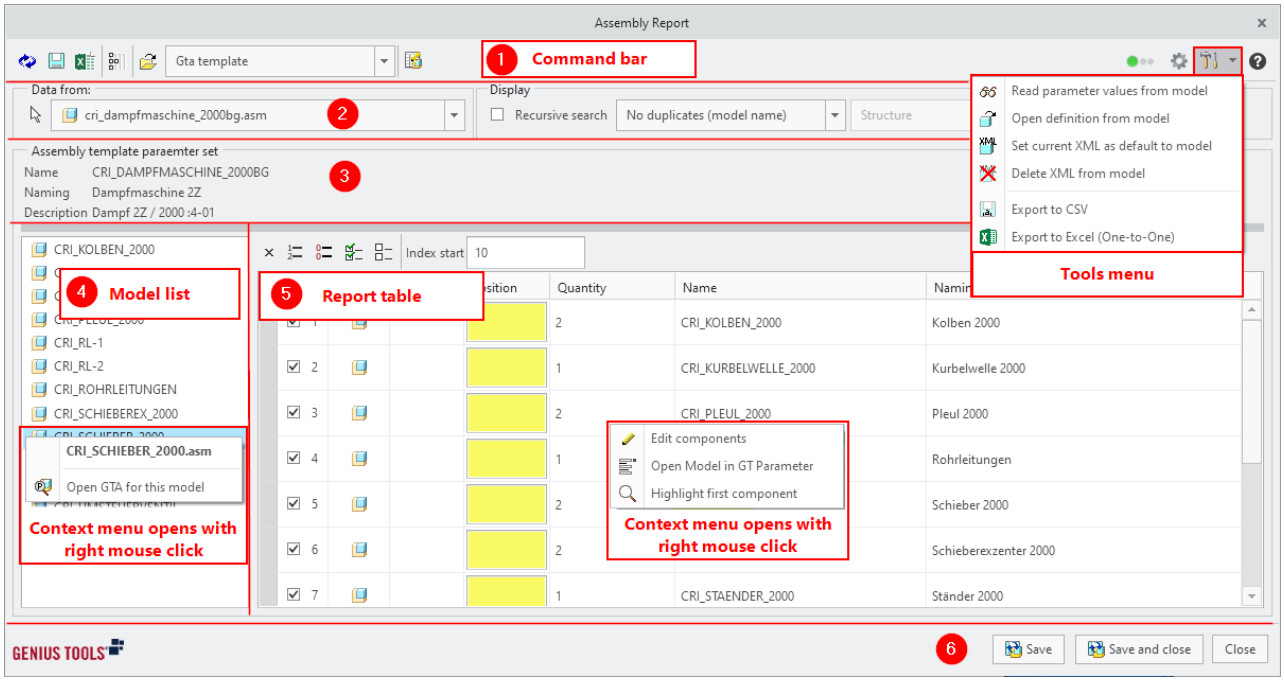

© 2026 INNEO Solutions GmbH

**Please note:** The external standard report definition is always used first in an assembly. (Configurable in the configuration option `gta_default_file`. Default: `%GT_RESOURCE_FOLDER%\assembly\gt_assembly.xml`, Startup Tools: `%GT_RESOURCE_FOLDER%\assembly\gt_assembly_single_level.xml`.)

**Warning:** Changes to report definitions must be applied explicitly to already opened assemblies!
Report definitions are stored in assemblies (depending on the configuration option `gta_save_xml_in_mdl`). Each time Assembly Report is started, the Assembly Report checks whether the current assembly contains a report definition. They are preferred for report generation.

## 5.2.1 User interface

The user interface of GENIUS TOOLS Assembly Report consists of the following elements:

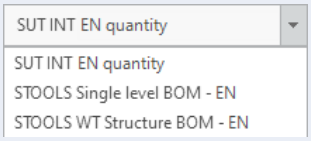



1. Command bar[46] with Report definition and Tools menu.[47]
2. Model selection and View[50]
3. Head parameters: information of the current Creo model and the description of the configured report definition.
4. Model list[51]: shows subassemblies
5. Report table[52]
6. Save:[54] The Save button turns green to indicate change in the report table.


© 2026 INNEO Solutions GmbH

## 5.2.2 Command bar

The following buttons are included in the command bar:

| Icon | Name | Description |
|---|---|---|
|  | Discard changes and update | Discards all unsaved changes and reapplies the report definition. |
|  | Save parameters | Saves changes and the report definition to the current assembly. |
|  | Export to Excel[689] | Opens *Copy Table to Excel* using a template.[78] |
|  | Create and display model tree | Regenerates the model tree and displays it in Creo Parametric.<br>Parameters marked for the model tree in the editor are also displayed in the model tree. |
|  | Open external report template from file[50] | Opens an external report definition (XML file) and applies it to the current assembly.<br>If you select an external report template for an assembly, it is automatically copied to the model, i.e. it becomes the internal report template. |
|  | Select an internal report template | Displays the internally stored report templates of the assembly.<br>If no templates are displayed, check that the `gta_save_xml_in_mdl` configuration option is set to 1. |
|  | Open Editor[55] | Opens the GENIUS TOOLS Assembly Report Editor with the current report definition. |
|  | Status indicator[476] | Shows the current status for *loading*, *working* and *saving* with traffic light colors and opens the status dialog. |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
|  | Options[48] | Opens the options dialog for the current report definition. |
|  | Tools menu[47] | Contains various supporting functions. |
|  | Help | Opens the help. |

The **tools menu**  contains the following functions:

| Symbol | Name | Description |
|---|---|---|
|  | Read parameter values from model | Parameters are read again. |
|  | Open definition from model | Opens and applies an internal report definition from the current assembly. |
|  | Set current XML as default in model | Defines the current report definition as the default of the current assembly and saves it there (if not present). |
|  | Delete XML from model | Deletes a selectable report definition from the current assembly.<br>**Please note:** Report definitions are reapplied to an assembly when Assembly Report is reopened. |
|  | Export to CSV | Exports the current report to a CSV file.<br>**Please note:** An export to CSV uses all displayed rows. |


© 2026 INNEO Solutions GmbH

| Symbol | Name | Description |
|---|---|---|
|  | Export to Excel (One-To-One) | Exports the current report without a template to an Excel file.<br>**Warning**: For Excel output an Excel version 2016 or higher must be installed!<br>Make sure that no Excel instance is running when exporting a report! |

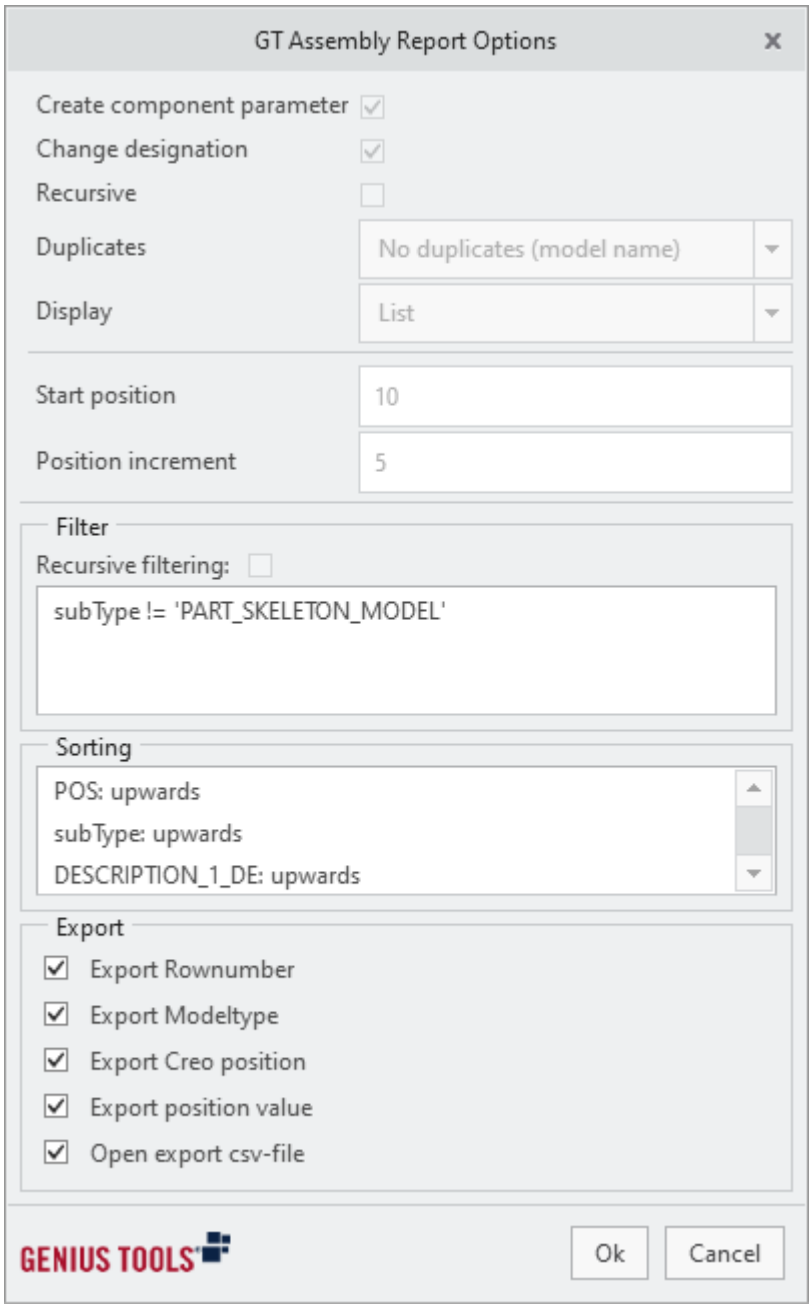
The export settings to CSV and Excel can be defined as described in Configuring export of reports 77.

## 5.2.3 Report options

In the options the settings of the selected parts list representation are displayed. These are set in the editor, see General information 57.

In the lower area export settings for this report definition can be changed quickly.


© 2026 INNEO Solutions GmbH




# Export

**Export line number, model type, position in Creo, position value**
These four options define whether the line number, model type, Creo position or position parameter (POS) are output one-to-one when exporting to a CSV file or Excel spreadsheet. The default value for the four options is defined by configuration option.

**Open CSV file after export**
Defines whether CSV files are opened after an export (default program: Excel).


© 2026 INNEO Solutions GmbH

## 5.2.4 Report definitions

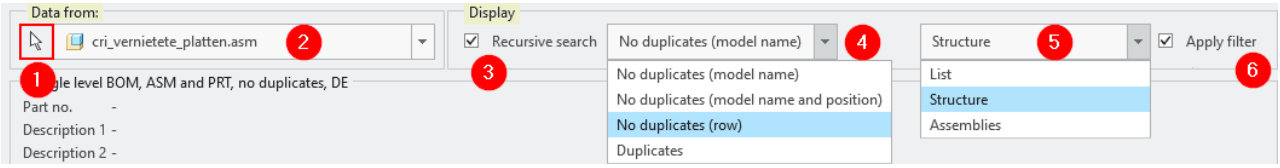
Note that you can select both external and internal report definitions (templates). If you select an external report template (XML file) for an assembly, it is automatically copied into the model, i.e. it becomes an internal report template.

You can customize any report definition supplied with *GENIUS TOOLS* for Creo or develop your own report templates, see Use cases[69].

## 5.2.5 Model selection and view

In this area you select a model and in which BOM type (single-level, bulk, multi-level) the components are displayed.

The **model selection** (Data from:) displays all models in a dropdown list that were selected by using the object selection.




*Select model and how to display the report table*

**1. Object selection:** Select subassemblies with the arrow function.

**2. Report table:** Shows the selected subassembly.

To get to the main assembly, the selected subassembly must be deselected by clicking again.

**Tip**: Assemblies selected in the model list are ignored. Only assemblies selected by object selection are displayed in the dropdown list.

In the **Display settings (BOM display)** you can order the contents of the report table by different criteria. The default settings are defined in Assembly Report Editor[57].

The following settings correspond to the common parts lists. It is recommended to specify these settings in the report definition.

| | Recursive search | Display mode | Duplicates |
|---|---|---|---|
| Bulk bill of materials | yes | List | No duplicates (model name) |
| Single-level bill of materials | no | List | No duplicates (model name) |


© 2026 INNEO Solutions GmbH

| | Recursive search | Display mode | Duplicates |
|---|---|---|---|
| Multi-level bill of materials | yes | Structure | No duplicates (model name) |

**3. Recursive search:** The report table shows sub-assemblies and parts. If activated, you can select a display mode (5).

**4. Ignore Filter:** When this option is enabled, you can control whether the filters under *Filter parameters in Assembly Report Editor*68 are applied in Assembly Report. This option is only displayed if it is enabled in the Editor.

**5. Duplicates:** This setting determines how models with the same name and different item numbers are displayed.

- No duplicates (model name): Models with the same name are displayed in one row, one row is displayed per model name.
- No duplicates (model name and position): Models with the same name but different position numbers (index) are displayed in one row per position number.
- No duplicates (row): Only rows that have the same values in all columns are considered as duplicates and combined into one row.
- Duplicates: Always shows separate rows, regardless of whether rows share values.

**6. Display mode:** The report table can be displayed using different types of layout:

- List: flat list of all models
- Structure: Contained sub-assemblies and parts are displayed in indented groups under their parent assembly. The depth of the indentation per level is defined in the configuration option `gta_struct_insert_space` and can range from 0 to 10 space characters.
- Assemblies: Sort by assembly. The name of each assembly is displayed as a heading for the contained sub-assemblies and parts.

**Please note**: Changing the view can lead to a re-reading of the report table. Unsaved changes will be lost.

## 5.2.6 Model list

The model list displays subassemblies of the current assembly. If no subassemblies exist, it is hidden.

Click on an entry in the list to open a subassembly with the current report definition.

Use the **context menu,** which opens by right-clicking on a subassembly, to open it with an embodied report definition in a new Assembly Report window.


© 2026 INNEO Solutions GmbH

## 5.2.7 Report table

The report table displays a report on the current assembly of the model selection. This report depends on the current report definition and the display of it varies accordingly.

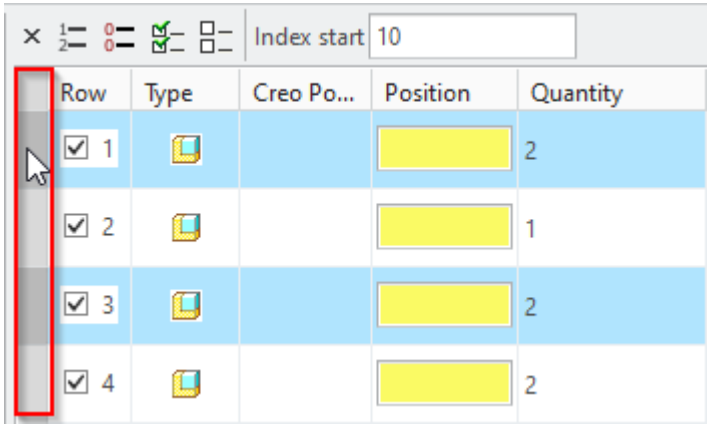
The following buttons are displayed in the **command bar** above the report table.

| Symbol | Name | Description |
|---|---|---|
|  | Deselect all lines | Deselects all models in the report table. |
|  | Use position numbers from Windchill | Adopts item numbers from Windchill. **Attention:** Only available after activating the option in the editor and with connected Windchill. See Dynamically transferring position numbers from Windchill 80. |
|  | Set position | Sets the position parameter according to the rules of the report definition for each model contained in the assembly. Only selected models are considered. |
|  | Set position values to zero | Resets the position parameter for the models contained in the assembly. Only selected models are considered. |
|  | Select All Components | Activates all rows in the report table. |
|  | Deselect all components | Deactivates all rows in the report table. |
| | Starting position | Defines the initial value for numbering. This value is used when numbering the selected rows. It is not included in the definition. |

Select the checkboxes in the *Row* column to select models for a report and automatic numbering.

**Tip**: Hold down the Shift or CTRL key and click in the gray area in front of the line number to select multiple lines. This allows values to be changed for several rows at the same time and the model activation can be changed too.


© 2026 INNEO Solutions GmbH




**Please note**: If a partial numbering already exists, missing numbers are filled according to the rules of the report definition.

The coloring of the report table columns depends on the configuration.

## Context menu

Right-click on a component in the report table to get the functions:

- Edit components
- Open Model in GT Parameter
- Highlight first component

## Editing parameters

Depending on the configuration of a report definition, parameters of components[60] can be edited by checkboxes, prefabricated lists or free input.

The cells in the column *Position* are highlighted according to their processing status:

| Color | Meaning |
|---|---|
| green | Valid once assigned value |
| red | Different values for components in the same row<br>For example, for different part numbers or for different parameter values for flexible parts. |
| orange | Component parameters not included in the model.<br>These can be filled in either manually or automatically using the button *Set position*. |


© 2026 INNEO Solutions GmbH

| Color | Meaning |
|---|---|
| yellow | Multiple assigned position numbers for different components in a report. These can be corrected either manually or automatically using the button *Set position*. |
| white | default |

The color coding is removed when multiple lines are selected.

If a parameter is of a different data type than specified in the report definition, a message is displayed in the status dialog[476].

## Sorting the report table

Click on a column name to sort the report table in ascending lexical order (1, 10, 11, 12, ..., a-z) by column. Another click sorts the table in ascending lexical order. A third click restores the original sorting.

**Please note**: If you want to use a different sort order, it must be configured in the editor.

Use Drag-and-Drop to rearrange individual lines. Click in the gray area left of the row number and drag the row to the desired position. Click the gray area left of the row number and press Shift or CTRL to select multiple rows and drag them to the desired position. Hold down Shift or CTRL while dragging.

**Tip**: Sort by row number to drag and drop reordered report tables to their initial state.

### 5.2.8 Saving

When saving, the POS parameter is written to the model.

It is possible to reset the POS parameter before saving, so that all components that were, for example, filtered manually from the parts list receive the default parameter value *-1*. To do this, activate the `gta_clear_pos_param` configuration option. The default setting is 0 (not active).

The Save button turns green when the report table has been changed.

# 5.3 Configuration

In this section you will find more detailed information about the structure of the *Assembly Report Editor* and how to create report definitions and Excel templates.


© 2026 INNEO Solutions GmbH

## 5.3.1 Assembly Report Editor

The editor is used to create and manage report definitions for *Assembly Report*, see chapter Use cases 69.

Various parameters can be displayed in a report:

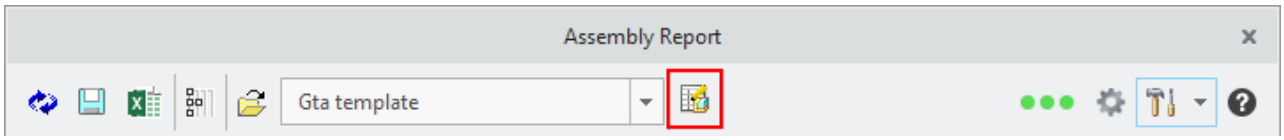
– Model parameters 60 are existing parameters and are read out.

– Component parameters 60 can be existing component parameters or can be newly created.

– Report parameters 62 are created solely for the assembly report.

### Starting the program

Start the editor in the command bar of the *Assembly Report* dialog window.




**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors 781.

### Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

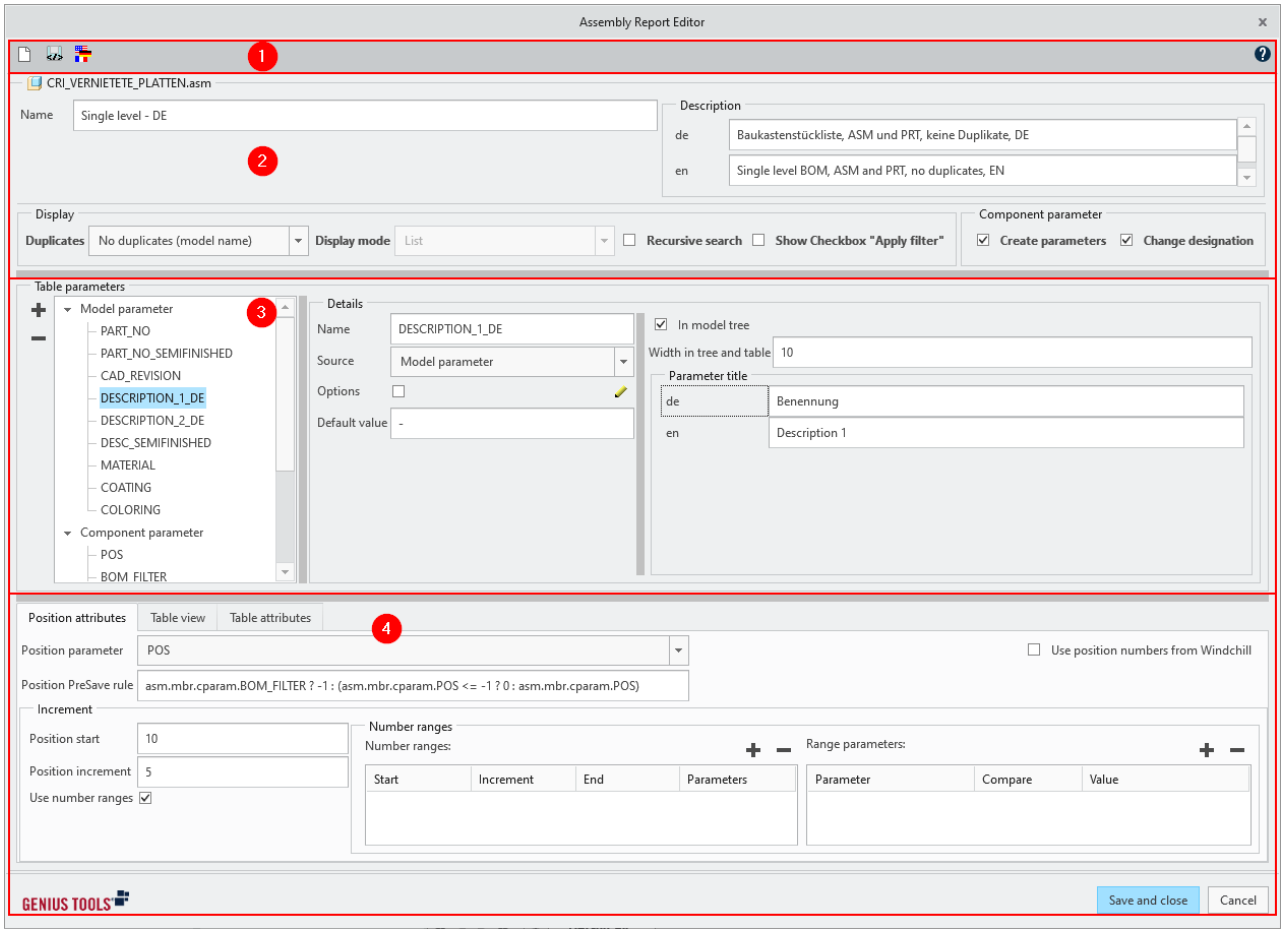
To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## 5.3.1.1 User interface

The user interface of *Assembly Report Editor* consists of the following elements:


© 2026 INNEO Solutions GmbH




1. Command bar[56]
2. General Information[57]
3. Table parameters[58]
4. Tabs: Position attributes[65] – Table view[66] – Table attributes[68]
5. Save and close:
   If this area is grayed out, changed data cannot be written into the model. Make sure that the configuration option `gta_save_xml_in_mdl` is set to 1.

## 5.3.1.2 Command bar

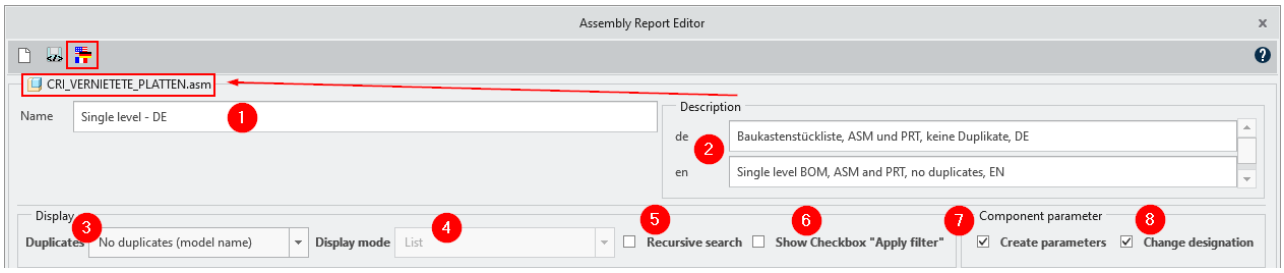
The following buttons are included in the command bar:


© 2026 INNEO Solutions GmbH

| Symbol | Name | Description |
|---|---|---|
| | Create a new empty report | Discards all entries and creates an empty report definition. |
| | Export report under a new name | Saves a report definition as an XML file under any name. |
| | Change used languages | Manages the available languages of a report definition. |
| | Help | Opens the help. |

## 5.3.1.3 General information

The general information of the opened report definition is the following. The current assembly to which the report definition is applied is displayed above the name.

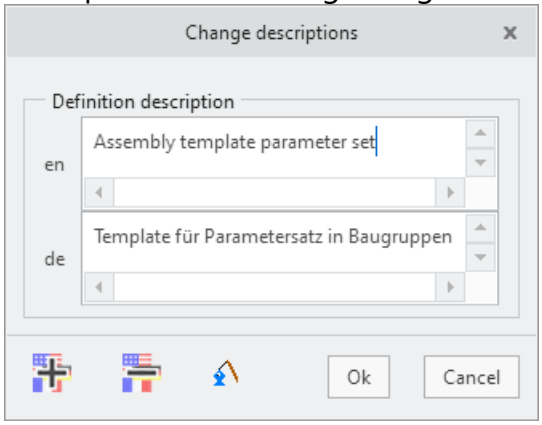



*Origin of the report definition Gta template from the assembly CRI_DAMPFMASCHINE_2000BG*

**1. Name:** Specifies the name of the report definition as it will later be saved in assemblies.

**2. Description:** Specifies a description of the report definition in different languages. By default, a description is available in German and English.

The localized descriptions are managed via the languages button in the command bar that opens the following dialog. Use the symbol for default texts[676].





© 2026 INNEO Solutions GmbH

## Display

These settings are default settings that can be changed in the Assembly Report dialog under Display.[50]

**3. Duplicates**[51]**:** Defines how models with the same name but different position numbers are displayed.

**4. Display mode**[51]**:** Defines the arrangement of the assembly components: List of all components, structure, division into the individual subassemblies. For selection, the checkbox *Recursive search (5)* must be activated.

**5. Recursive search:** Subordinate assemblies and parts are displayed in the report table. When this search is checked, the selection for *Display mode (4)* opens.

**6. Show Checkbox "Ignore Filter":** Defines the display of the checkbox "Ignore Filter" in Assembly Report[51]. When this option is enabled, you can control whether the filters under *Filter parameters*[68] are applied in Assembly Report. If this option is disabled, you cannot control the application of filter parameters because they are applied automatically.

## Component parameters

These settings apply to all component parameters[60]

**7. Create parameters:** Defines whether newly created table parameters are created as component parameters.

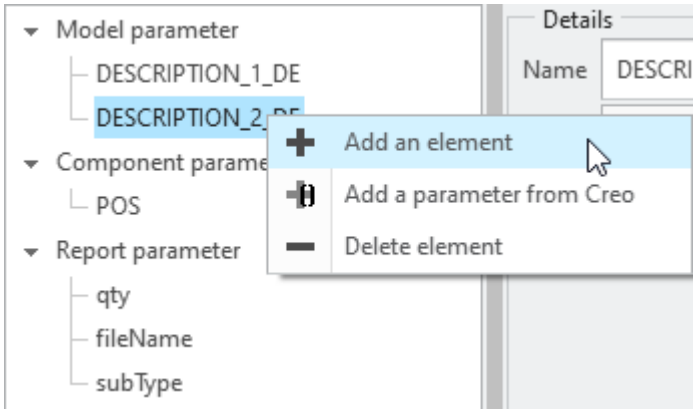
**8. Change designation:** Defines whether a component parameter is to be designated to an external system (e.g. Windchill).

## 5.3.1.4 Creating table parameters

In the section *Table parameters*, you specify which parameters are displayed in the report table and how the parameter values are output.

## Editing table parameters

The table view has its own context menu. Right-click on an item to open the context menu.





© 2026 INNEO Solutions GmbH

**Add an element**: Adds a parameter of the selected type with an arbitrary name. Parameters can be freely renamed.

**Add parameters from Creo**: Adds a parameter of the selected type from the Creo Parametric parameter selection.

**Delete element**: Deletes the selected parameter from the list.

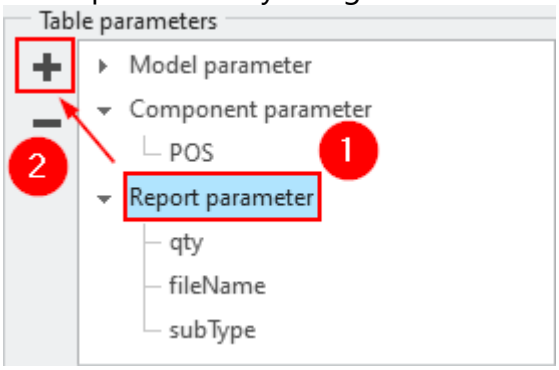
**Please note:** Be careful not to assign parameter names twice. Parameter names can be freely assigned in the table view.

# Displaying table parameters in the report table

Add all parameters that you need for the report definition.

**Procedure**

1. Select one of the three types of table parameters (1): model parameter, component parameter or report parameter.
2. Add a parameter by using the Plus button (2) or the context menu entry *Add an element*.




3. Define how the parameter will be displayed, see Defining table parameters[59].
4. Move the created parameter to the list of Head parameters in the Table view[66] tab by clicking on it.
5. Add the Head parameter to a column in the Table view[67] area.
6. Specify how the parameter values are to be filtered and sorted in the Table attributes[68] tab.

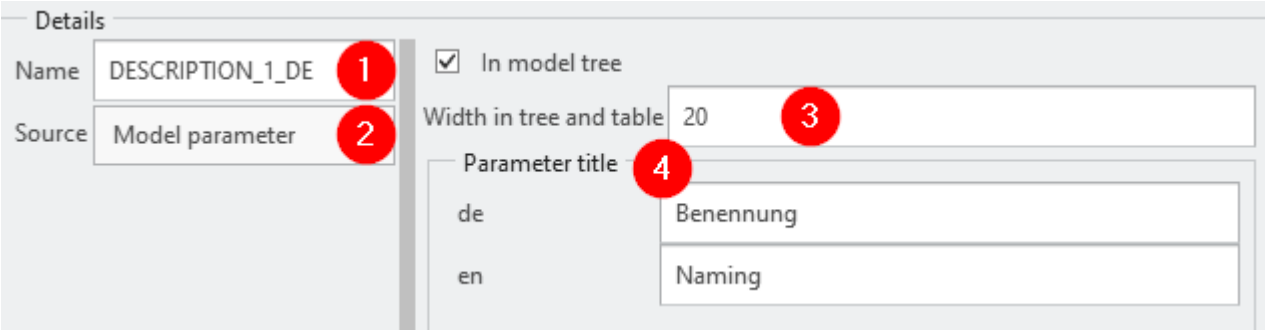
A separate section of this chapter explains the procedure using an example[924].

See the chapter *Use cases* for an example[69] of creating a report table with all table parameter types.

# Defining table parameters

In the area *Details*, the selected table parameter is configured. The following information is required for all three parameter types:


© 2026 INNEO Solutions GmbH




**Name (1):** The name of the parameter.

**Source (2):** Defines the type of parameter (model parameter[60], component parameter[60] or report parameter[62]).

**Width in tree and table (3):** Defines the width of the parameter column in characters. Affects the Creo model tree and the column in the report table.

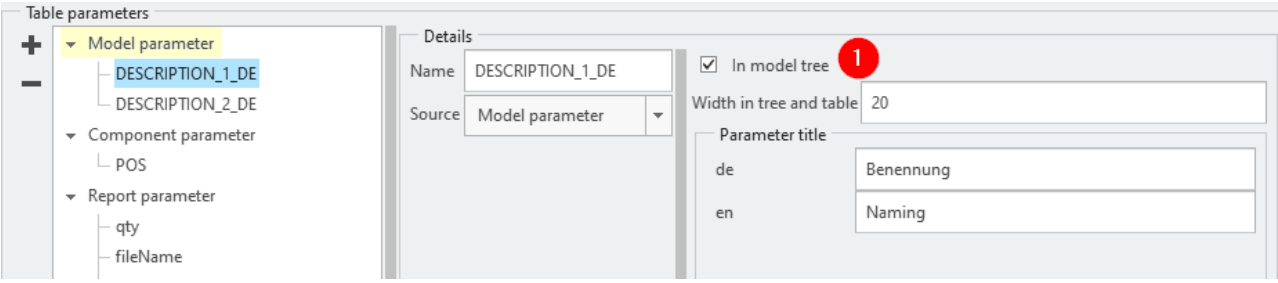
**Parameter title (4):** Defines the localized name of a parameter for the report table.

# Defining types of table parameters

**1. Model parameters**

– are existing parameters from assemblies and contained models,

– are read.




**In model tree (1):** Defines whether the parameter is also displayed in the model tree when *Create and show model tree* is clicked.

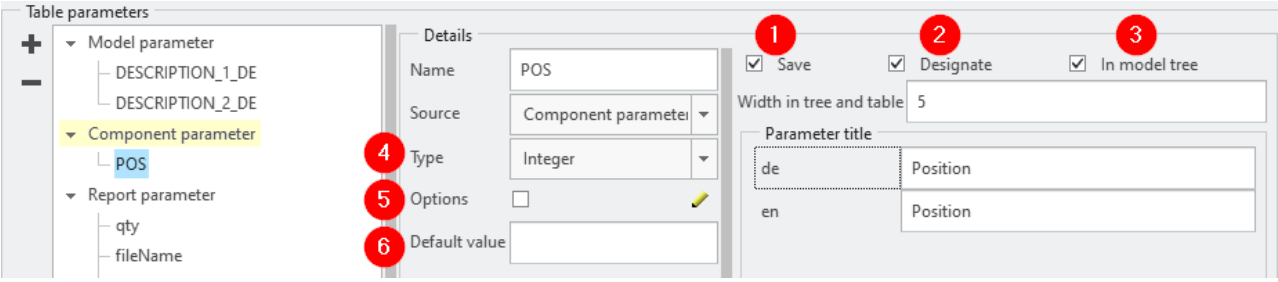
**2. Component parameters**

– are parameters that are available in assembly components in assembly mode, e.g. position numbers,

– can be created or read out in the model if the component parameter already exists,

– are filled by manual input or predefined rules.


© 2026 INNEO Solutions GmbH




**Save (1):** Specifies whether to rewrite the value. If checked, the value can be edited and saved in *Assembly Report*.

**Designate (2):** Specifies whether to designate the value when creating it. If this option is checked, the information for designation is transferred to the parameter properties. Select the checkbox if you are working with Windchill.

**In model tree (3):** Defines whether the parameter is also displayed in the model tree when *Create and show model tree* is clicked.

**Type (4):** Choose from the data types String, Integer, Double and Boolean.
The *Type* specification entered here replaces other specifications of the parameter. If this specification is not supposed to overwrite the properties of the parameter, set the configuration option `gta_update_component_parameter_type` to 0.

| Type of parameter | Description |
|---|---|
| Boolean | Displays checkboxes. |
| Double, Integer | Allows the input of numbers. |
| String | Allows the input of text. |

**Changing the type of a component parameter**
To change the data type of a component parameter, e.g. from integer to string, the configuration option `gta_update_component_parameter_type` must be set to 1 (default). This will update the data types of the component parameters as defined in the report definition.

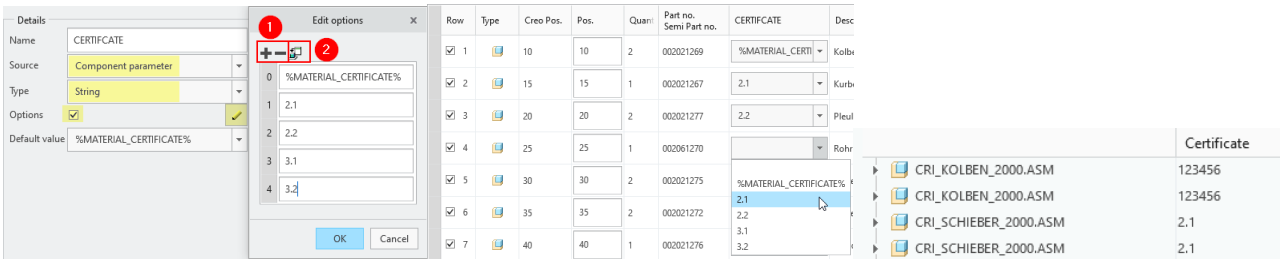
Changing data types may be necessary, to unify component parameters that were created with different data types, for example when working with Windchill.

**Please note:** If you use the configuration option `gta_autoload_folder` to automatically update the report definition[79], you must save changes in the report definition as an XML file. If this configuration option is set, changes that do not enter the XML file will not be transferred to the component parameter.

**Options (5):** Defines a selection list. Click *Edit options* to define a list of text entries, variables and model parameters available for selection by *Assembly Report* users (1). Click


© 2026 INNEO Solutions GmbH

*Add option* ➕ to add a new entry. The list can also be imported from a text file (2). To delete a row, click in the field for the row number, and then click *Delete option* ➖.




*Possibilities to fill out the options menu in Assembly Report Editor*

*Options menu in the report table in Assembly Report*

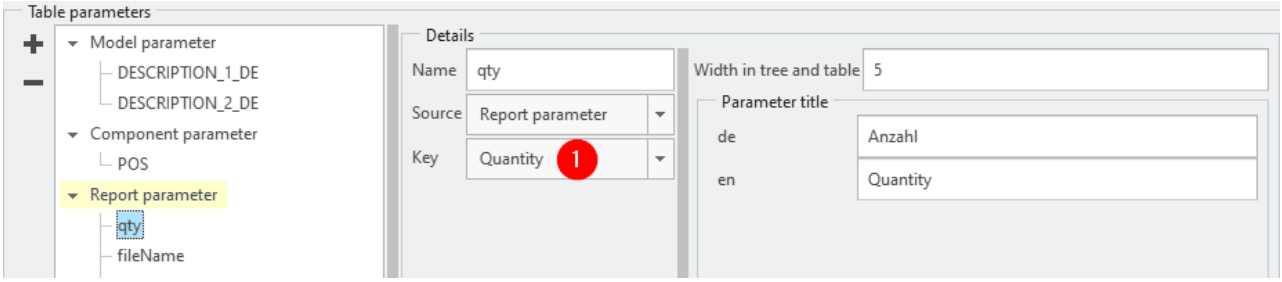
*Component parameter* `%CAD_CREATED_BY%` *is displayed as a model parameter in the model tree*

**Default value (6):** Defines a value with which a parameter is created in models. If the default value is empty, the parameter will be created empty.

### 3. Report parameters

– are properties of individual models such as number of assemblies or file name,

– are created for *Assembly Report*.




**Key (1):** Defines a property that is displayed in the parameter in *Assembly Report*. The following types of report parameters are available:

| Key | Description |
|---|---|
| Quantity | Number of times the model is assembled. |
| File name | |
| Extension | |
| Subtype | |
| Mass | |
| Total mass | Mass of all components for the model as calculated by Creo |


© 2026 INNEO Solutions GmbH

| Key | Description |
|---|---|
| Sum | This parameter type for calculating values has been replaced by the type *Relation*. It is maintained only to ensure compatibility with existing report definitions. |
| Level | Assembly level |
| ASM skeleton | Skeleton: yes/no |
| Relation | Calculation of the value to be displayed. The calculation rule is entered in one line using JavaScript syntax. For this purpose, a separate input field *Rule*63 opens. |
| Generic | Name of the generic part |
| Feature-ID | ID number of the feature<br>ID numbers are listed on one line separated by commas when duplicates are not displayed in the list, i.e. for the View settings50 *No duplicates (model name)* and *No duplicates (model name and position)*. |
| Parent model parameter | Special case64: Reads model parameters from the parent model. |
| Is embedded | Checks if a model is embedded in an assembly. |
| Structure text | Numbered, hierarchical structure with indents. Relationships are immediately apparent. |

**Filling report parameters with a rule**

The displayed value of a report parameter can be calculated with a JavaScript rule. To do this, specify the key *Relation* and enter a one-line JavaScript syntax.

- You can use arithmetic operations, the conditional operator (?) and logical operators.
- You can define whether the return value of the rule is a string, integer, number (Double) or Boolean value.
- You can use instructions in the notation `mdl.PARAMETER_NAME` to use parameters of the ROOT assembly (as in drawing tables with repeat regions).
- You can use instructions in the notation `asm.mbr.PARAMETER_NAME` to use component parameters (as in drawing tables with repeat regions).

Examples:

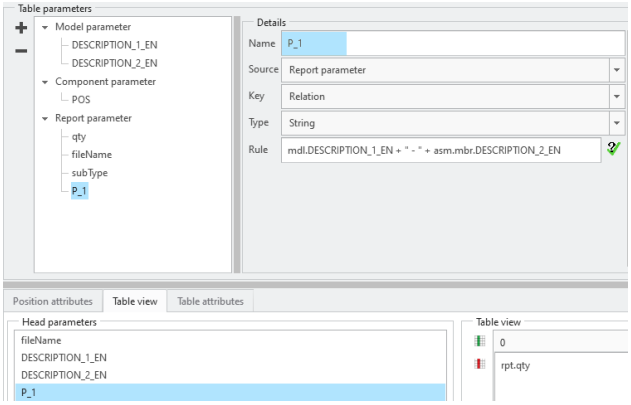

© 2026 INNEO Solutions GmbH

```
(rpt.qty > 2.0) ? 1.0 : 0.0
```
If the value for the quantity is larger than 2, the return value is 1, otherwise it is 0 (Boolean value).

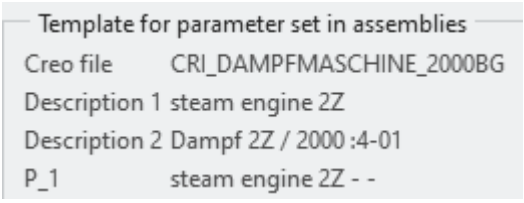
```
asm.mbr.PTC_MATERIAL_NAME.indexOf("STEEL_COMMON")>=0?"yes":"no"
```
If the material *Steel common* is used, the rule returns *yes*, otherwise *no* (string value).

```
mdl.DESCRIPTION_1_EN + " - " + mdl.DESCRIPTION_2_EN
```
Outputs the descriptions 1 and 2 from the ROOT assembly.

```
mdl.DESCRIPTION_1_EN + " - " + asm.mbr.DESCRIPTION_2_EN
```
Outputs the description 1 from the ROOT assembly and 2 from the component, as shown in the example below:




*Mixed input for report parameter P_1 in the editor*




*Output of P_1 in assembly report dialog as head parameter*

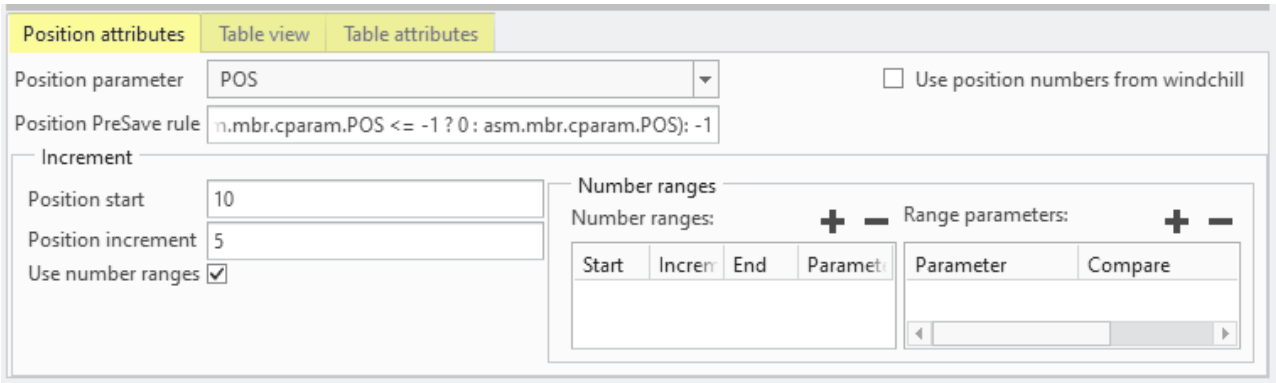
**Special case: Parent model parameter**
When a report parameter of type Parent model parameter is created, a model parameter of the parent is read. The name specified in Details must be the model parameter name.

The Parent Model Parameter type (key) is useful to filter out subcomponents from the assembly report, for example, to display purchased parts without subcomponents[87] .

## 5.3.1.5 Configuring report table

The table that makes up the assembly report is configured in the tabs Position attributes[65], Table view[66] and Table attributes[68].


© 2026 INNEO Solutions GmbH




*Assembly Report Editor: Lower segment with tabs*

# Tab: Position attributes

Define rules for assigning position numbers in the section *Position attributes*. The item number is assigned as a value to the position parameter in individual assemblies and parts.

**Position parameter:** In the dropdown list, select a parameter as the position parameter. Only parameters of type component parameter[60] are available.

**Position PreSave rule:** A Javascript PreSave function can be stored, the result of which is assigned to the position parameter. The function is executed for each line before the position parameter is written to the model. This rule is applied to each line in the report. The model remains unaffected. Component, model, and report parameters are supported.

This is opposed by the `gta_clear_pos_parameter` configuration option. If this configuration option is set to 1, the entire assembly is processed so that all components of the assembly are inheritantly set to position -1.

Example:

```
asm.mbr.cparam.POS <= -1 ? 0 : asm.mbr.cparam.POS
```

– Case 1 (POS <= -1):)
  The position parameter receives the value 0.

– Case 2 (POS > -1):
  The position parameter retains its value.

For a more detailed use case, see the example[83].

**Use position numbers from Windchill**: Shows an additional button for the transfer of position numbers from Windchill. If you have created a custom wtpart, there are additional configuration steps you need to follow, see Dynamically transferring position numbers from Windchill[80].

## Increment

**Position start**: Defines the start of the position numbers.

**Position increment**: Position numbers are incremented by the position increment.


© 2026 INNEO Solutions GmbH

**Use of number ranges**: Defines whether the rules defined under number ranges are to be used for item number assignment.

## Number ranges

In the *Number ranges* area, rules for exceptions to position number assignment are defined. If the condition of a number range is fulfilled in a model, an item number is assigned according to the number range.

Each row of the number range table corresponds to an individual number range.

Use the plus button to add a row. Configure the number range afterwards.

**Start:** Defines the start of the position numbers in the number range.

**Increment:** Defines the increment of position numbers in the number range.

**End:** Defines the highest position number to be assigned.

**Parameter:** Displays selection parameters for a number range.

Create and configure one or more parameters in the Selection parameters table. A selection parameter is automatically added to the selected number range. The selection parameters and the stored conditions are used as filters for number ranges.

**Parameter:** Specifies a parameter (table parameter) to be used as a selection parameter.

**Comparison:** Select the comparison here (greater than, less than, equal to, not equal to).

**Value:** Defines the value with which the parameter is to be compared.

To remove a line, click on the line number and press the minus button.

# Tab: Table view

Head parameters and report table are configured in the table view.

All parameters that are to be displayed in the report table must first be created in the section *Table parameters*58.

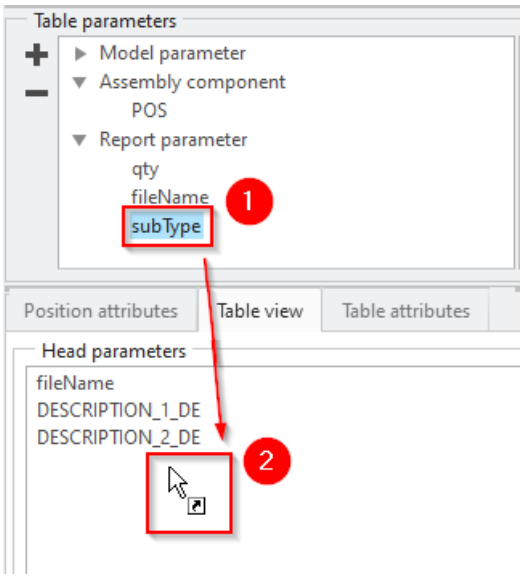
## Head parameters

Parameters defined under Head parameters are displayed in the report table 52 area of *Assembly Report*.

Click on a table parameter (1) and drag it into the head parameters (2). To change the order of the parameters also use drag and drop.

Also use the context menu to manage the parameters.


© 2026 INNEO Solutions GmbH




*Head parameters can be filled from report parameters.*

## Table view

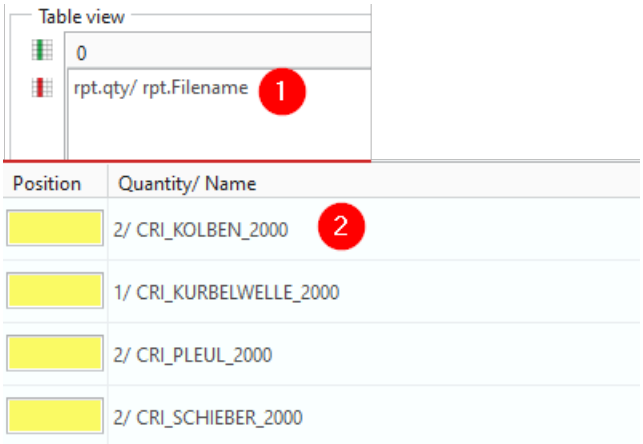
The report table is configured to the right of the head parameters.

Configure the columns of the report table using the two buttons Add Column and Delete Column.

Click *Add Column* and drag and drop a table parameter into the column field. Alternatively, use the context menu to manage the parameters.

You can also display several parameters and static texts in a column. Drag several parameters one after the other into a column field or store static texts. Make sure that there is at least one space between parameters and texts!

**Please note**: As soon as a column for assembly components contains static text, a space or several parameters, no changes can be made in the column.




*Specify several parameters per column in the editor (1). They are evaluated in the Assembly Report table displayed (2).*


© 2026 INNEO Solutions GmbH

Select a column header and click *Remove column* to remove unneeded parameters.

The checkbox *Add totals row* under the column configuration adds a row below the table that shows the sum of all values for each column that contains number parameters (Integer or Double). Each column for which you want to create a sum may contain only one Integer or Double parameter.

If the checkbox *Add totals row* is selected, you can select another option: *Write sum of mass_TOTAL in model parameter*. If you select this option, you can enter a new model parameter in the text field or overwrite an existing model parameter.




**Please note**: All displayed values in each column will be added up. In a structured list that can contain, for example, sub-assemblies and the parts used in them, parts may contribute multiple times to the value given in the totals row.

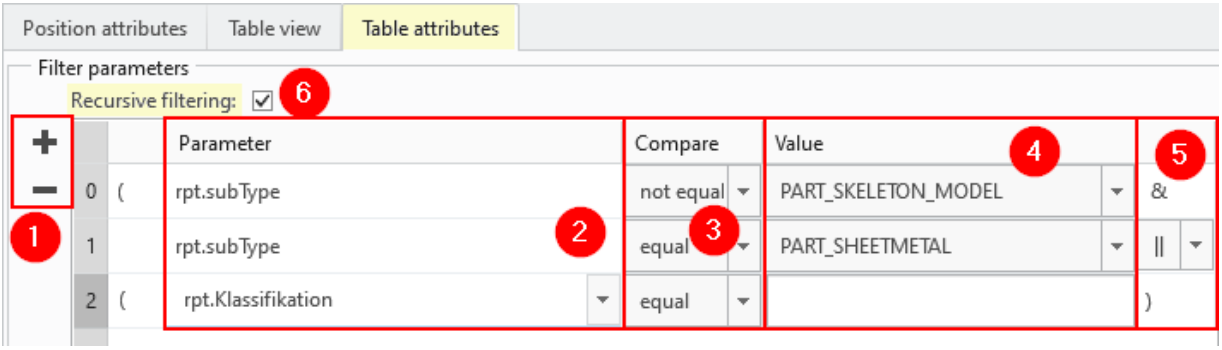
# Tab: Table attributes

The *Table Attributes* section defines general report properties and filter and sort parameters.

## Filter Parameters

The filter conditions of the report table are defined under *Filter parameters*. Only those parts and assemblies are displayed that fulfill these filter conditions.

1. Use the plus button to add a row. To delete a line, click on the line number and the minus button.




*Inputting a filter and a condition*

2. **Parameter:** Select a filter parameter that defines the filter condition.

| Modellparameter | Komponentenparameter | Reportparameter |
|---|---|---|
| asm.mbr.PARAMETERNAME | asm.mbr.cparam.PARAMETERNAME | prt.PARAMETERNAME |

3. **Comparison**: Select the comparison here (equal to, not equal, > (greater than), < (less than), contains, contains not, regular exp (expression), not regular exp (expression)).
4. **Value**: Defines the value with which the parameter value is to be compared.


© 2026 INNEO Solutions GmbH

**Tip**: Use Regular Expressions [487] in the value column!

5. **Logical operators:** If you are working with multiple filter parameters, you can use logical operators to determine whether the filter parameters are executed additively or exclusively. Linked filters are indicated by an opening parenthesis in the second column and a closing parenthesis in the last column.
   - You can set these types of links:
     & (=and): All AND-linked filters are executed.
     || (=or): A new filter group is opened, separated from the existing filters by an OR-link. Only one of the OR-linked filter groups is executed.
6. **Recursive filtering:** includes subcomponents.
   - Off (default): All components and assemblies that meet the filter conditions are filtered out from the Assembly Report.
   - On: All components and assemblies that meet the filter conditions are filtered out with their subcomponents without checking the filter conditions on the subcomponents.

**Please note:** Recursive filtering is only relevant for the *display mode List*, [50] e.g. quantity BOMs. In the *Structure* display mode, subcomponents of assembly components that have been filtered out are not displayed.

A recursive search is useful, for example, to display purchased parts without their subcomponents, see use case [87].

## Sorting Parameters

The table of sort parameters defines the parameter values according to which the displayed models in the report table are sorted.

The order of the parameters in this table determines the sort order. The above parameters are preferred.

Use the plus button to add a row. Then select a parameter from the drop-down list and select the sorting direction.

To remove a row, click on the row number and press the minus button.

# 5.3.1.6 Use cases

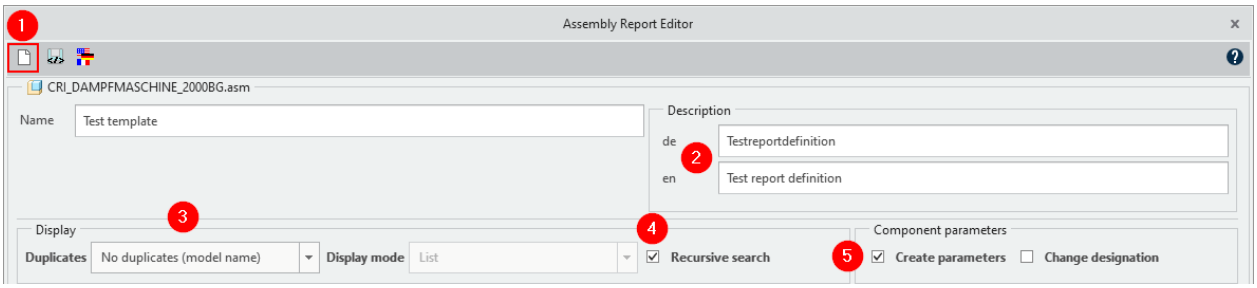
*Assembly Report* reports can be customized for use within the company. Individual report definitions supplied with GENIUS TOOLS for Creo can be customized or custom report definitions can be developed. Creating your own report definitions using the *Assembly Report Editor* is explained in this example.

1. Open an assembly.
2. Start Assembly Report and open the Editor.


© 2026 INNEO Solutions GmbH

# Creating a new report definition

1. Create a new, empty report definition.
2. Enter a name for the report definition.
3. Under *Duplicates*, select the option *No duplicates (model name)*.
   This option ensures that item numbers of multiple assembled models are different.
4. Check the box *Recursive search*.
   With this setting you ensure that all layers in a model are filtered and taken into account in the numbering.
5. Check the box *Create parameters*.
   This setting ensures that the position parameter is created in each model.




# Creating table parameters

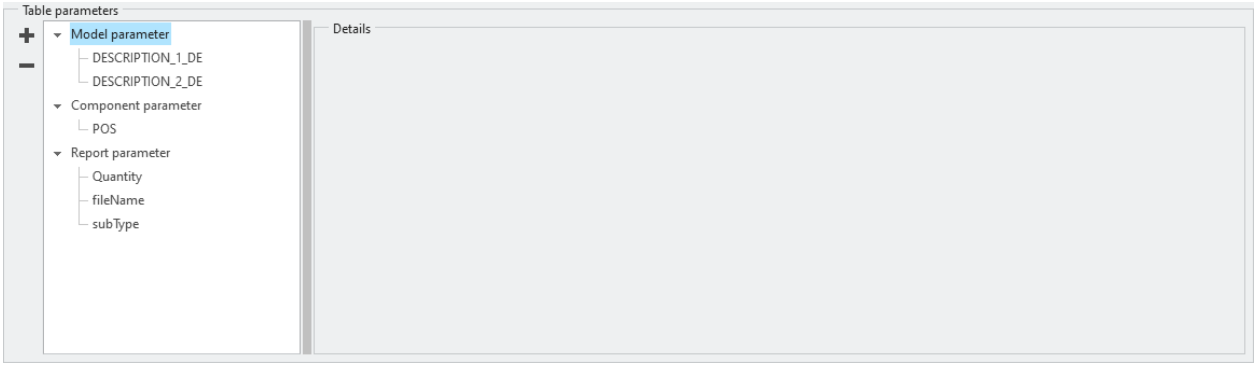
In the section *Table parameters* section of the report definition, add all the model, component, and report parameters you need for the report.

**Please note**: Any parameter to be used in other sections of the editor must be created in this section.

1. Add the model parameters::
   - DESCRIPTION_1_DE
   - DESCRIPTION_1_EN
   - DESCRIPTION_2_DE
   - DESCRIPTION_2_EN
2. Add the Component parameters::
   - POS
3. Add the Report parameters::
   - Quantity
   - fileNameDateiname
   - subType


© 2026 INNEO Solutions GmbH

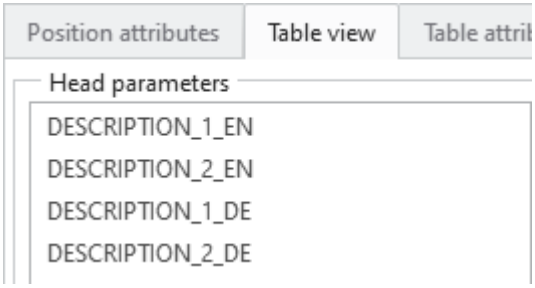
4. After all parameters have been created, the section *Table parameters* in the *Assembly Report Editor* appears as follows:




## Defining head parameters

Parameters in this list will be displayed later in the header of the report.

1. Select the model parameter DESCRIPTION_1_DE under *Table parameters*.
2. Drag and drop it into *Head parameters*.
3. Repeat the procedure with all model parameters.



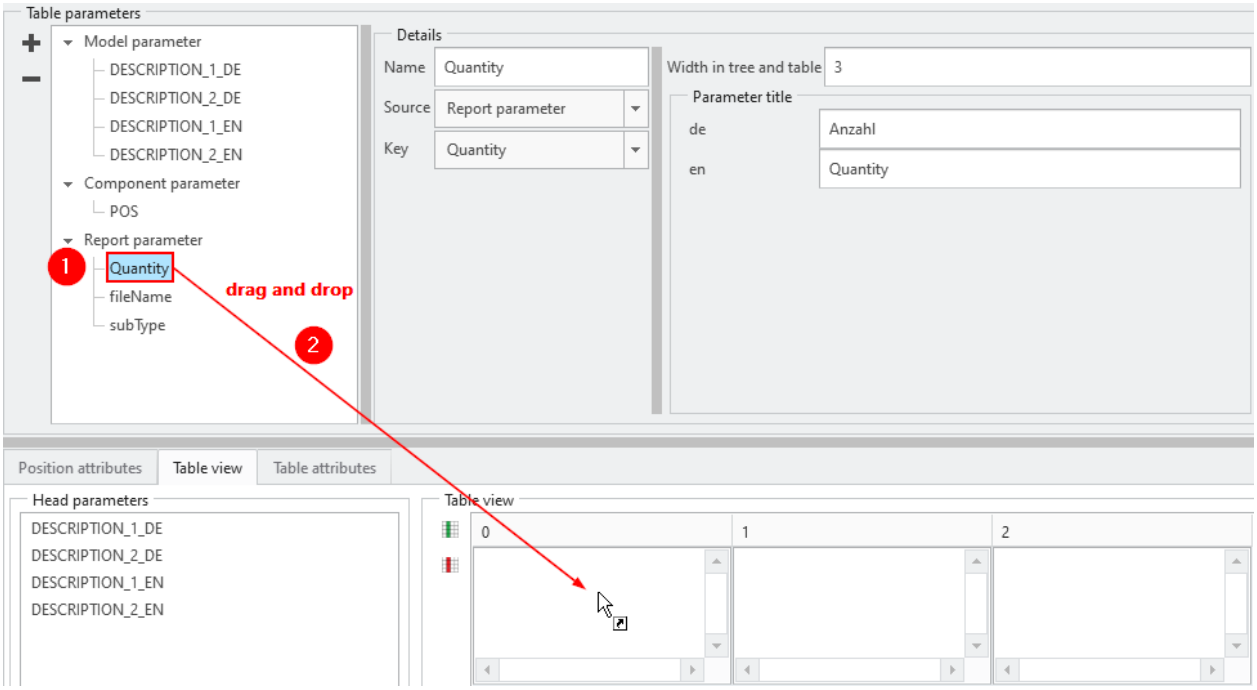

**Tip**: Parameters can be re-sorted with drag and drop.

## Configuring the table columns

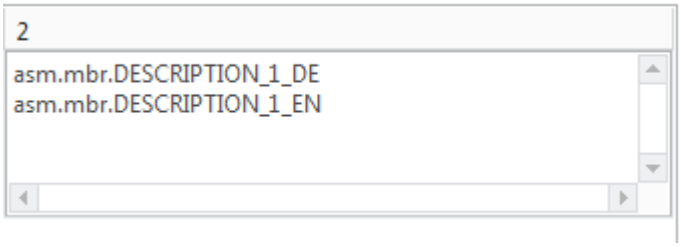
In the next step, configure the table columns of the report table. Use the two buttons on the left side of the *Table View* section to specify the number of columns displayed.

1. Create three columns. The first column is always counted as column 0.
2. Select the Report parameter *Quantity* under *Table parameters*. Drag and drop the parameter into column 0.

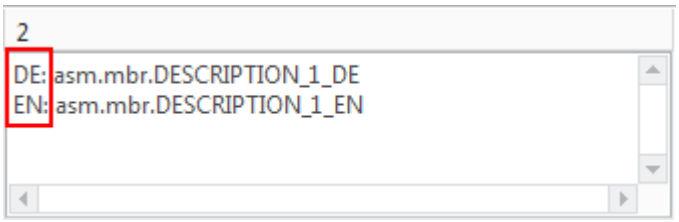

© 2026 INNEO Solutions GmbH




3. Repeat with the parameters fileName/column 1 and DESCRIPTION_1_DE/column 2. To make the name in reports bilingual, column 2 is now configured to display two parameters.
4. Drag and drop the parameter DESCRIPTION_1_EN into column 2. Separate the two parameters with a linebreak (Enter key).




**Tip**: You can also add normal text to parameters. Use a space between text and parameter!



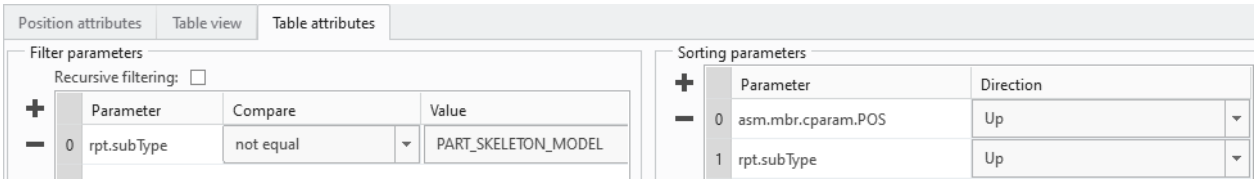

*Separate parameters from static text with a space character*

## Defining table attributes

Now, configure the table attributes. Table attributes determine which models are displayed in the report table.


© 2026 INNEO Solutions GmbH

1. Select the option *No duplicates* under *Duplicates*. This option ensures that position numbers of multiple built-in models differ.
2. Activate the options *Create parameters* and *Recursive search*. These settings ensure that the position parameter is created in each model and that models of deeper planes are filtered.
3. Add a filter parameter: Click on the Plus button and select the parameter *subType* from the dropdown list.
4. Select *diverse* under *Compare* and *PART_SKELETON_MODEL* under Value. This setting sorts out skeleton models.
5. Click on the plus button under *Range parameters* and select the *POS* parameter from the drop-down list. Select *Up* as the direction. Repeat this step with the parameter *subType*.

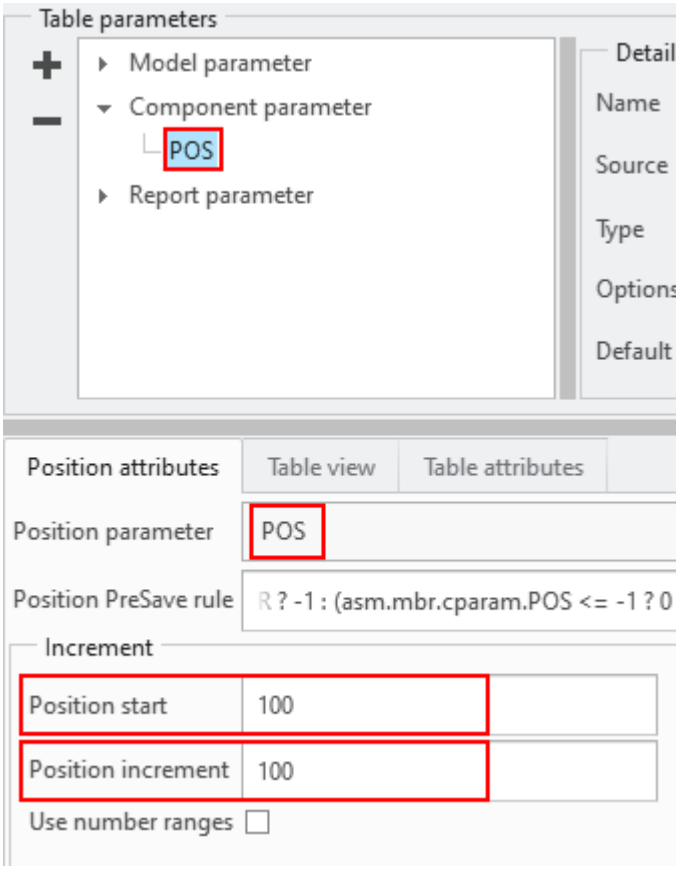



## Configuring position attributes

Configure the position attributes:

1. Select the parameter *POS* as *Position parameter*.
2. Enter a start position of *100* and a position increment of *10* under *Increment*. Elements in the report table are now numbered consecutively in steps of ten starting at 100 (100, 110, ...).


© 2026 INNEO Solutions GmbH

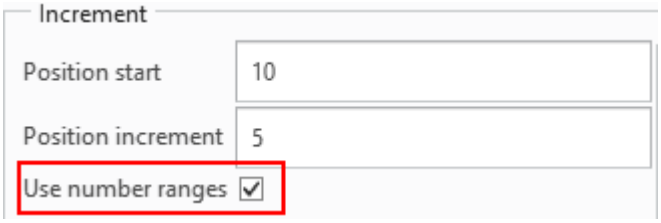



**Warning**: Make sure that the correct position parameter is set! Position parameters must be of the type *Assembly Component parameter*. Saving must be activated for the parameter.

## Defining number ranges

Create number ranges for exceptions to normal numbering:

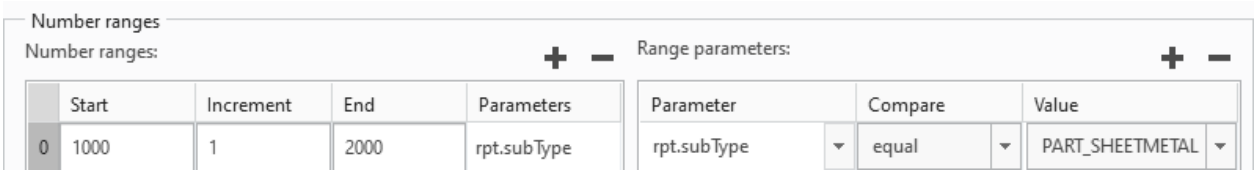
1. Under *Position attributes* set the check mark at *Use number ranges*.




2. For the definition of a number range, the tables *Number ranges* and *Range parameters* are available. With *Number ranges* you define the type of numbering. Use *Range parameters* to define the parameter you want to assign a number range to.
3. Add a new number range with the plus button under *Number ranges*.
4. Enter *Start*, *Increment* and *End*.
5. Add a comparison with the plus button under *Range parameters*.


© 2026 INNEO Solutions GmbH

6. Define the comparison.
7. After you have defined the range parameter, the defined parameter is automatically transferred to the table *Number ranges*.
8. Save the report definition.




*In this configuration, all sheet metal parts are numbered in steps of five from 1000 to 2000.*

# Result

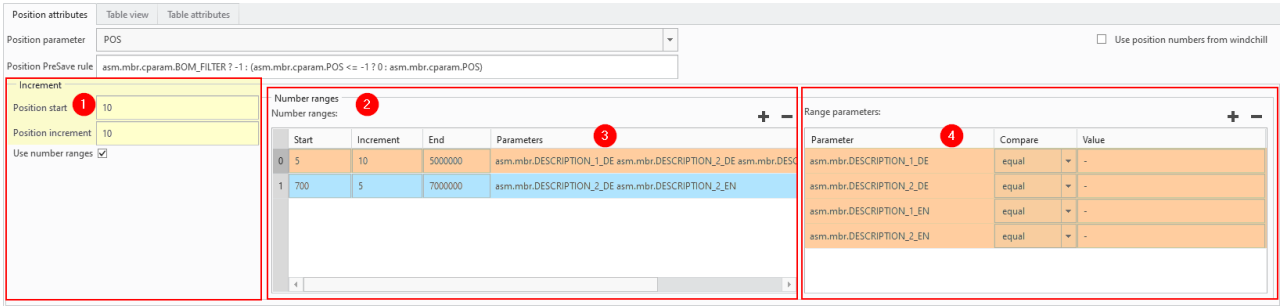
Save the report definition and close *Assembly Report Editor*. The created report definition generates the following report table:

| Row | Type | Creo Pos. | Pos. | Quantity | Creo file | DE: Description 1<br>EN: DESCRIPTION_1_EN |
|---|---|---|---|---|---|---|
| 1 | | | 10 | 1 | CRI_RL-2 | DE: Dampfleitung 2<br>EN: - |
| 2 | | | 15 | 7 | CRI_UEBERWURF_U_NIPPEL | DE: Anschlussverschraubung<br>EN: compression fitting |
| 3 | | | 20 | 1 | CRI_RL-1 | DE: Dampfleitung 1<br>EN: - |
| 4 | | | 25 | 2 | CRI_LAGERBOCK_2000 | DE: Lagerbock 2000<br>EN: bearing block 2000 |
| 5 | | | 30 | 1 | CRI_KURBELWANGE_MITTE_2000 | DE: Kurbelwange mitte<br>EN: crankweb center |
| 6 | | | 35 | 1 | RO-CRI_RL-2-0001 | DE: -<br>EN: - |
| 7 | | | 40 | 2 | CRI_GELENKSTUECK_SK_7 | DE: Gelenkstück<br>EN: joint piece |
| 8 | | | 45 | 2 | CRI_PLEULAUGE_O_2000 | DE: Pleulauge oben<br>EN: connecting-rod small end |
| 9 | | | 50 | 2 | CRI_PLEULAUGE_U_2000 | DE: Pleulauge unten<br>EN: connecting-rod big end |
| 10 | | | 55 | 2 | CRI_PLEULSTANGE_2000 | DE: Pleulstange<br>EN: conrod |
| 11 | | | 60 | 2 | CRI_PLEULVERBINDER_2000 | DE: Pleulverbinder<br>EN: piston rod connector |
| 12 | | | 65 | 1 | RO-CRI_RL-2-0002 | DE: -<br>EN: - |
| 13 | | | 70 | 7 | CRI_LOETNIPPEL | DE: Lötnippel<br>EN: tab |

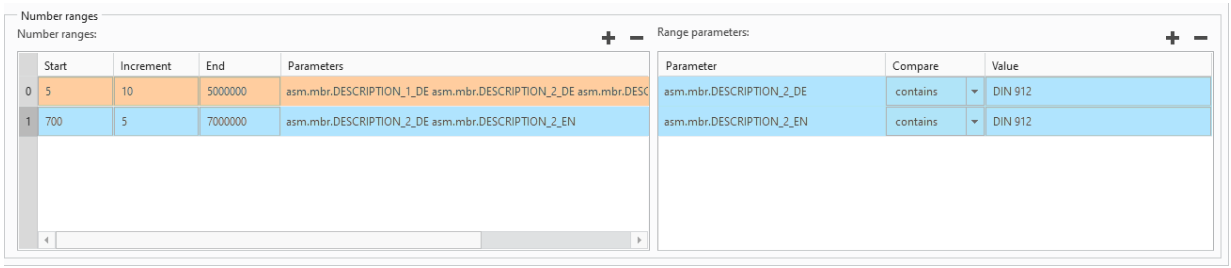
## Setting up parallel numberings

Create number ranges for exceptions to the default numbering. You can use multiple number ranges in parallel, for example, to differentiate between standard and purchased parts.


© 2026 INNEO Solutions GmbH




1. Set the base numbering under *Increment*. Select the checkbox *Use number ranges*.
2. Under *Number ranges*, create any numbering you want to use in the report table.
3. Define the number ranges with *Start*, *Increment* and *End*.



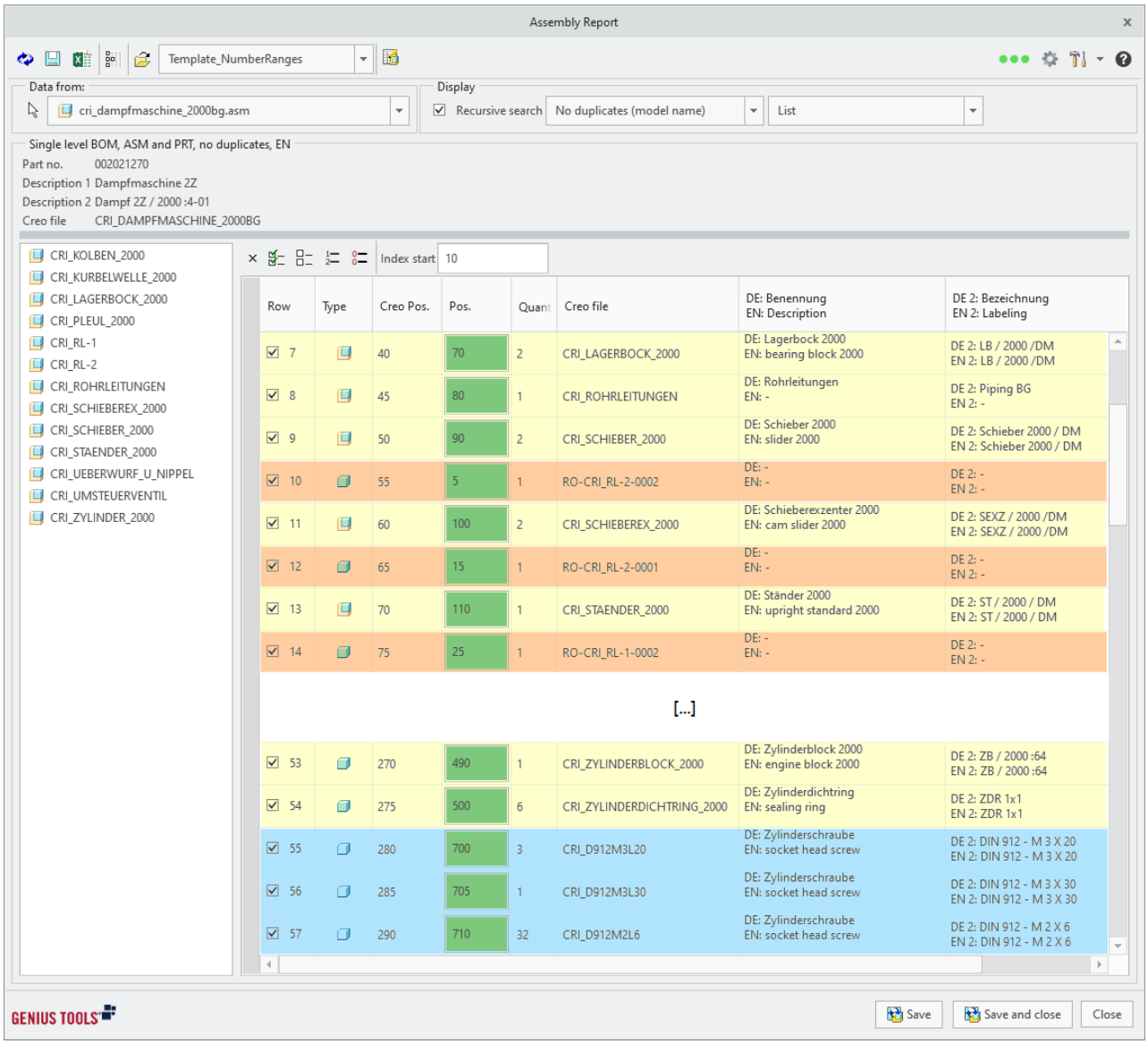

4. Comparisons must be defined for the *Parameters* entered under *Number ranges* via *Compare*. Once the number range parameters are defined, the defined parameters are automatically transferred to the table *Number ranges*.
   In this example, the following two number ranges were created:

| | | |
|---|---|---|
| | orange | Creo files that lack a description in English and / or German |
| | light blue | Standard part according to DIN 912 |

## Result

Save the report definition and close Assembly Report Editor. The created report definition generates the following report table with incremental numbering and additionally defined number ranges:


© 2026 INNEO Solutions GmbH




## 5.3.2 Configuring export of reports

When reports are exported as a CSV file and as an Excel file via the Tools menu[47], all data is exported as displayed in the user interface. All reports can be customized by the following configuration settings.

**gta_export_file**
Specifies the filename of the exported file. Take care to set the correct file name extension. If you want to export to different formats (e.g. CSV and XLSX), do not specify an extension in the configuration option. *GENIUS TOOLS* variables can be used.

**gta_export_path**
Specifies the directory where reports are saved. Default: working directory

## CSV exports

**gta_open_export_csv**
Defines whether a CSV file is opened after export (1) or not (0). Default: 1

**gtu_table_to_csv_export_sep**
Defines the separating symbol between values in CSV files. Default: ; (semicolon)

You can define the way the CSV file will be encoded with:

**gtu_table_to_csv_write_file_as_utf8**
Defines whether the CSV file is written as ASCII (0) or UTF8 (1). Start value: 1

**gtu_table_to_csv_write_file_as_utf8_with_bom**
Defines whether a CSV file written with UTF8 is additionally encoded with Byte Order Mark (BOM) (1) or not (0). Initial value: 0. For this, above option `gtu_table_to_csv_write_file_as_utf8` must be set to 1.

## Simple Excel exports

**gtu_table_to_excel_open_export**
Defines whether an Excel file is opened after export (1) or not (0). Default value: 1

**Warning:** For the output to Excel, an Excel version 2016 or higher must be installed on the executing computer!
If you want to use an Excel file with macros (XLSM), also adjust the security settings in Excel to allow macros to run.

## Reports with Excel templates

The Copy to Excel function in the command bar[46] requires a previously configured Excel template[78].

**gta_export_template**
Specifies the Excel template to be used by default for the export, i. e. entered in the export dialog. Default: *%GT_RESOURCE_FOLDER%\assembly\gt_assembly.xlsx*

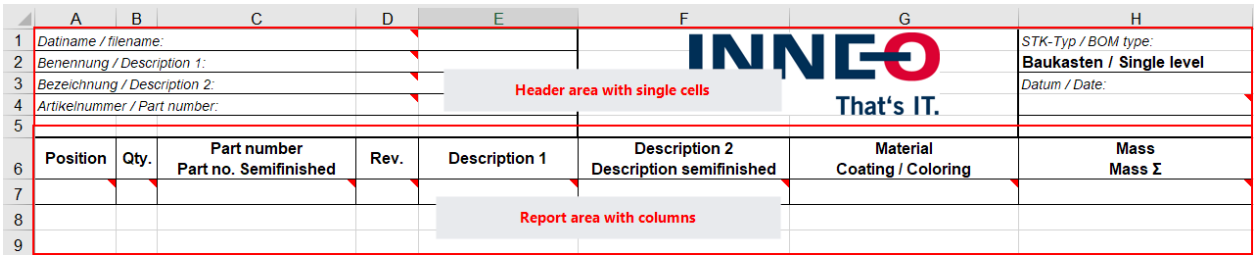
# 5.3.3 Creating an Excel template

The function Export table to Excel[689] in the command bar of Assembly Report requires a previously configured Excel template for the export process.

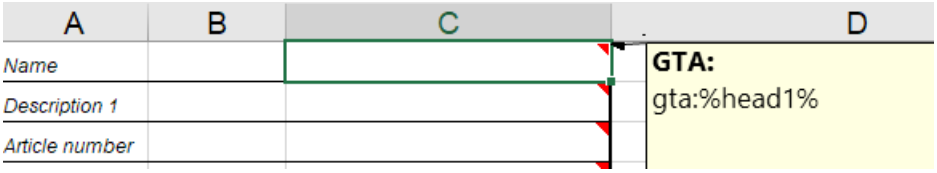
**Step-by-step guide:**
1. Use a separate Excel file for each report template.


© 2026 INNEO Solutions GmbH

2. Create a new template, following the steps in chapter Create template[693].
   Example:



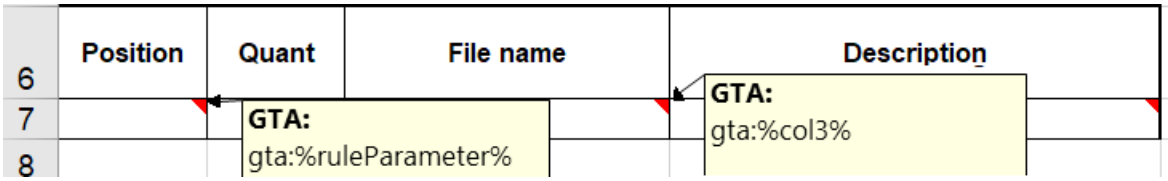

3. Take care to use the component acronym *gta* for head parameter in the header.



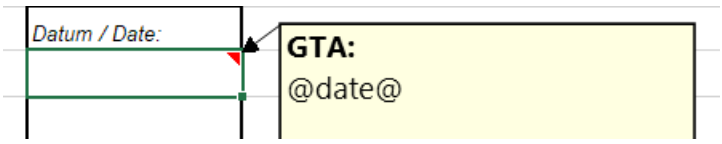

*Head parameters with numeration*

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.

4. Take care to use the component acronym *gta* for model parameter in the first row of the table columns.
   Use *gta:%ruleParameter%* to output the position parameter. Enter *gta:%col*%* to output a column. Replace * with the column number.
   Without the display of the position parameter, the counting of the columns starts at 0.




5. Use variables of GENIUS TOOLS for Creo[913] if required, e. g. date and time information as well as Creo object information.




**Please note**: The variables *@feat_id@*, *@selmdln@* and *@selmdlpath@* are not supported.

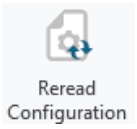
6. Save the template.

## 5.3.4 Automatic update of report definitions

Report definitions are stored in an XML file. If you make changes to it, *Assembly Report* will automatically apply the changes, if the configuration option `gta_autoload_folder` is set.


© 2026 INNEO Solutions GmbH

**Procedure**

1. Create a folder containing the current report definitions (XML files), e.g. folder update in the assembly folder of the resource folder.
2. Open *Configuration Utility*.[765]
3. Make sure, that the configuration option `gta_save_xml_in_mdl` is set to 1.
4. In the `gta_autoload_folder` configuration option, specify the path to the folder, for example, *%gt_resource_folder%assembly\update*, and save the change.
5. In standby mode, read in the configuration using the button 

in the *GENIUS TOOLS* menu ribbon.

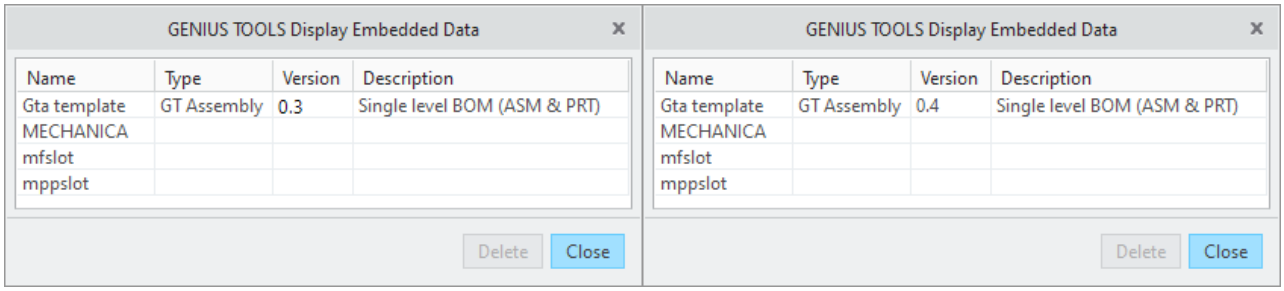
**Result**

When *Assembly Report* is opened, the externally modified XML file is loaded. This saves the changes in the model, which can be read in Embedded Data.[798]

**Tip:** It is helpful to specify a version of the report definition. The version will be visible in the embedded data[798] in the model, but not in the Assembly Report dialog.
Write the version in the XML file before the report definition name in the notation `gta version="version specification"`, for example:
`<gta version="0.2" name="Gta template">`

**Example**

Version 0.3 of report definition *Gta template* was changed and defined as version 0.4 (`gta version="0.4"`).




*before starting Assembly Report* *after starting Assembly Report*

## 5.3.5 Dynamically transferring position numbers from Windchill

When using Windchill, you can use the Windchill position numbers for numbering. WTParts are found by number or name and assigned uniquely.

This process can be made dynamic using the following configuration options:

– Searching for the WTPart: `gta_wt_part_attribute_name` (Default: *Number*)


© 2026 INNEO Solutions GmbH

– Searching for the attribute value: `gta_wt_part_attribute_value` (Default: *%PTC_WM_PART_NUMBER%*)

– Setting whether to search for the WTPart by its owner or by its attribute: `gta_wt_part_by_attribute` (0 - Owner, 1 - Attribute; Default: 0)

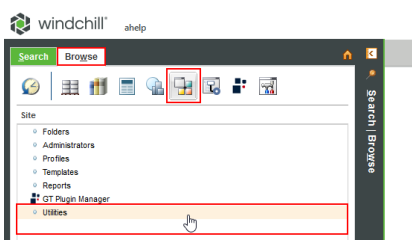
– Searching for position information: `gta_wt_part_pos_attribute_name` (Default: *Number*) Once the WTPart with the product structure is found, its position information is searched for and the position parameter set in Assembly Report is filled with it.

# Procedure

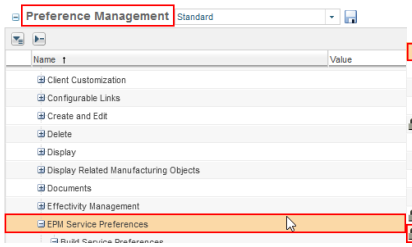
To transfer the position numbers from Windchill, proceed as follows:

1. **Set configuration options to the required values** (see Configuration Utility 765**):**
   Depending on how a wtpart is configured, adjust the configuration options accordingly.
   – `gta_wt_part_attribute_name` (Default: *Number*)
   – `gta_wt_part_attribute_value` (Default: *%PTC_WM_PART_NUMBER%*)
   – `gta_wt_part_by_attribute` (Default: 0=Owner)
   – `gta_wt_part_pos_attribute_name` (Default: *LineNumber*)

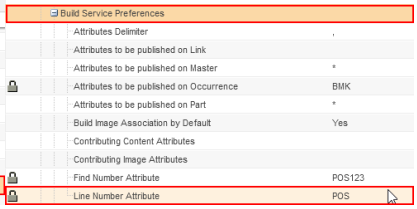
2. **Mapping WTPart**
   To accept the information during check-in, the *Line Number Attribute* must be linked to *POS* in Windchill. This setting must be defined once, after which this assignment will be made automatically for all WTParts.
   Navigate to the following setting in Windchill:
   *Browse* > *Site* > *Utilities* > Business Administration: *Preference Management* > *EPM Service Preferences* > *Build Service Preferences* > *Line Number Attribute*. Right-click to open the function *Set Preference*.
   Set *Line Number Attribute* to the desired value. Here: *POS*. Finish editing with *OK*.




*Browse* > *Site* > *Utilities*




> Business Administration: *Preference Management* > *EPM Service Preferences*

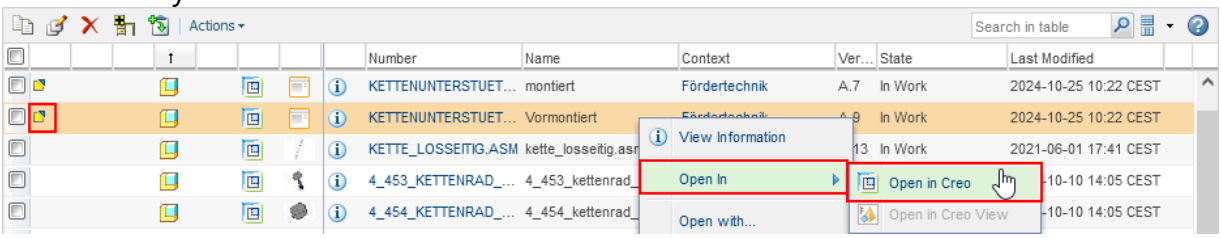



> *Build Service Preferences* > *Line Number Attribute*

3. **Checking out WTPart and opening in Creo**
   Check out the model to which you want to apply the defined line number attribute. Then open the model in Creo.
   With the default configuration settings for Assembly Report, all WTParts associated with


© 2026 INNEO Solutions GmbH

the open Creo model are searched for WTParts whose number is equal to *PTC_WM_PART_NUMBER*. *PTC_WM_PART_NUMBER* is a model parameter that is automatically filled in when the model comes from Windchill.

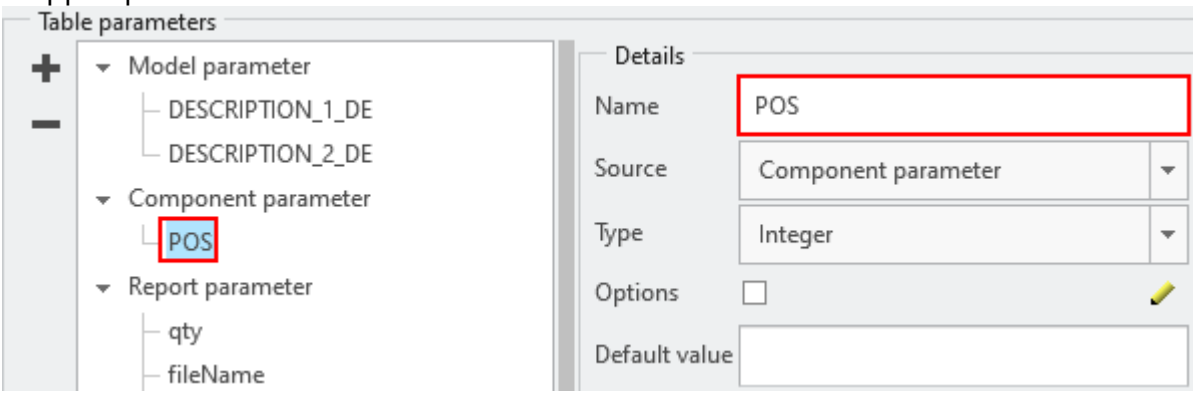



**4. Opening Assembly Report Editor**

See Assembly Report Editor[55].

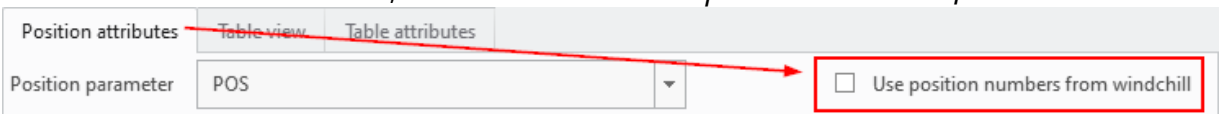
**5. Renaming position parameter**

In the section *Table Parameters*, under *Component parameter*, enter the name of the mapped parameter.




**6. Setting the use of position numbers from Windchill**

In the tab *Position attributes*, tick the checkbox *Use position numbers from Windchill*.

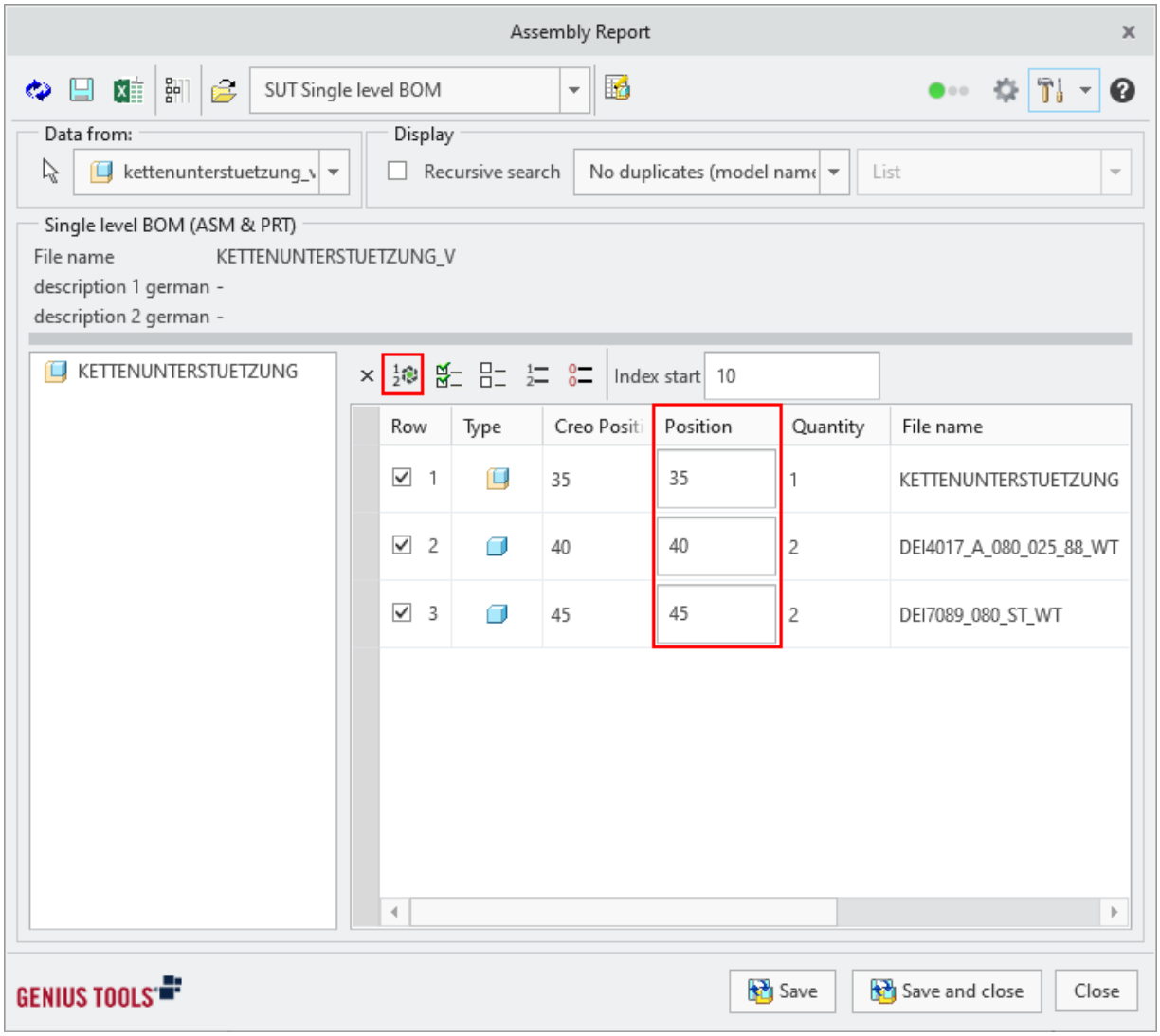



**7. Saving report definition and closing Assembly Report Editor**

In Assembly Report Editor, the numbering command *Set position from Windchill* appears. When you number the report table using this command, existing position


© 2026 INNEO Solutions GmbH

numbers from WTParts in the connected Windchill with the same name are used.




## 5.3.6 Use cases

This chapter collects some application examples for *Assembly Report*.

### 5.3.6.1 Removing single lines from parts list

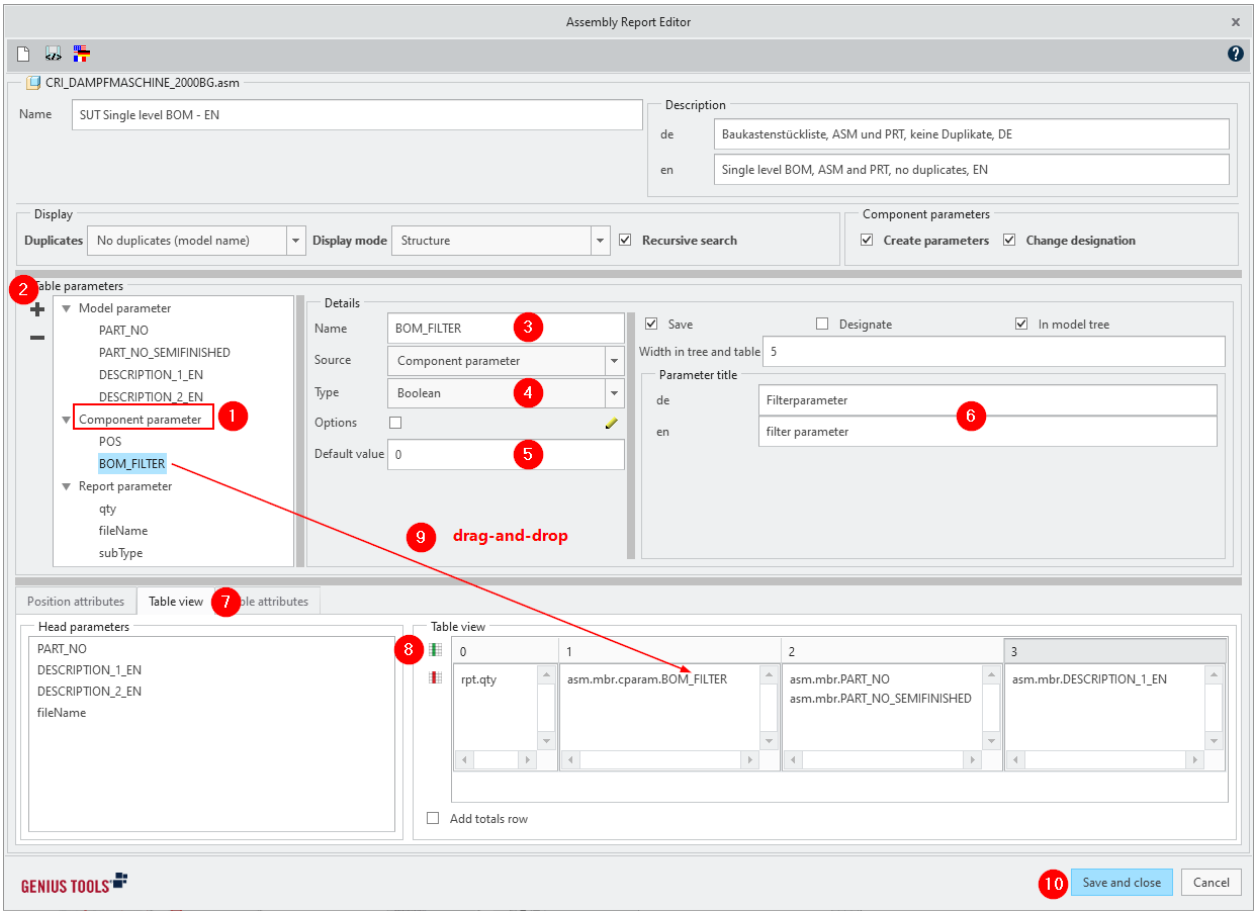
Using the component parameter BOM_FILTER, you can see an example of how a parameter is created and formatted. This example parameter adds another column to the assembly report. This column contains a checkbox in each row that is checked by default. If this checkbox is now unchecked in a row and this operation is saved, this table row is deleted. The deleted values are assigned the position number -1 in the background, see Position PreSave rule[65]. An entry removed in this way can be retrieved via component parameters[659].

The following steps are implemented in the *Assembly Report Editor*. The procedure is adapted to the operating principle in Windchill.


© 2026 INNEO Solutions GmbH

# Creating BOM_FILTER

1. In the area *Table parameters*, click *Component parameter*.
2. Click ➕ to add a new parameter to this category.
3. Enter the desired name for this new component parameter.
4. Select *Boolean* as the type. This will generate a checkbox in the user interface.
5. Set the *Default value* to 0 so that the checkbox is ticked.
6. Assign a title for this component parameter. This title will appear in the *Assembly Report* in the selected language.
7. The newly created component parameter must be added to the table. To do this, click in the tab *Table view*.
8. To add an empty column, click in the column that you want to be to the right of the new column and click the *Add a Column* button.
9. Click the newly created component parameter and drag and drop it into the new, empty table column.
10. Confirm your settings by clicking *Save and close*. The *Assembly Report Editor* closes and you are back on the start page of the *Assembly Report* user interface.

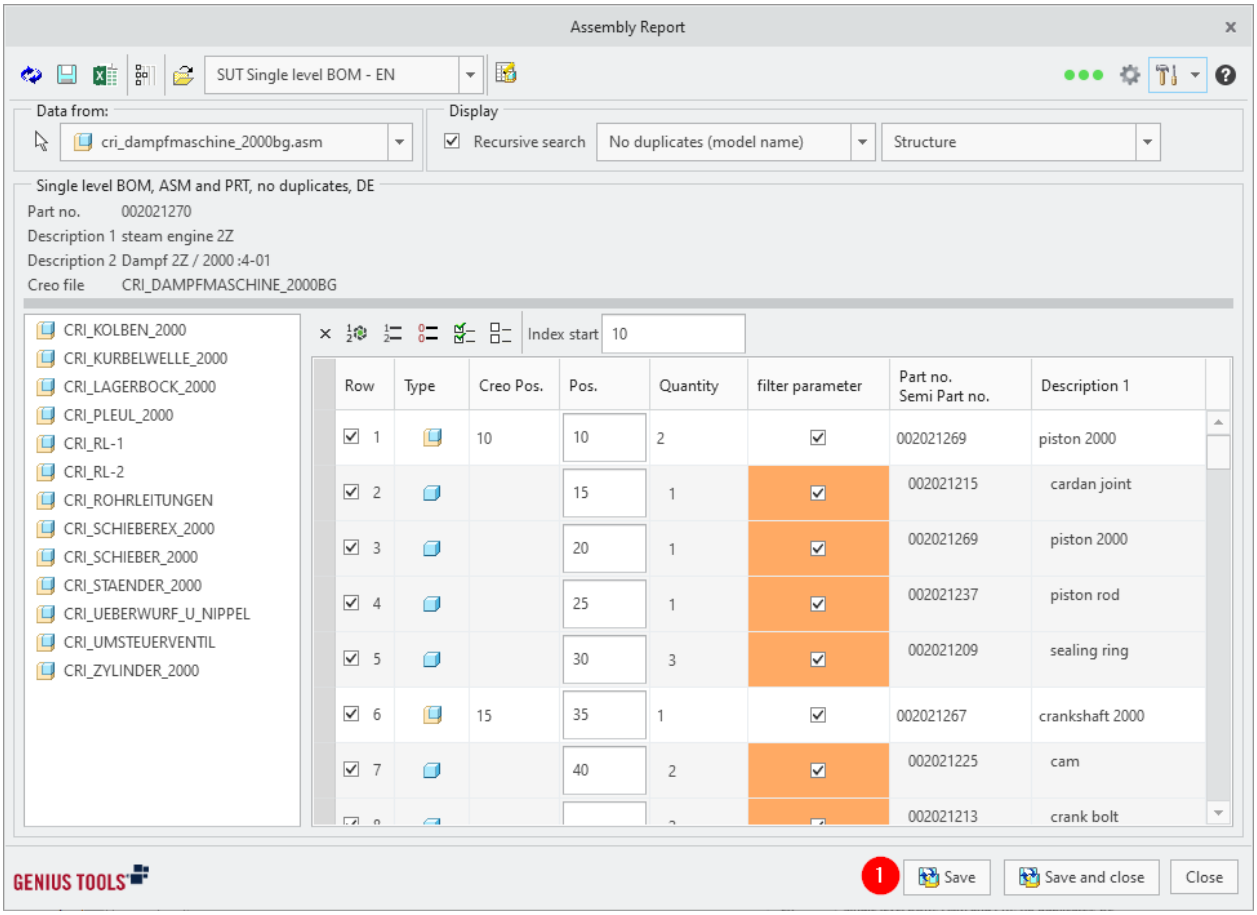



Creating a component parameter and adding it to the table


© 2026 INNEO Solutions GmbH

## Saving the preview of the table to validate it

1. You will now see a preview of the table as it will be output in the parts list. The orange column indicates that changes have been made. Confirm these changes with *Save*.



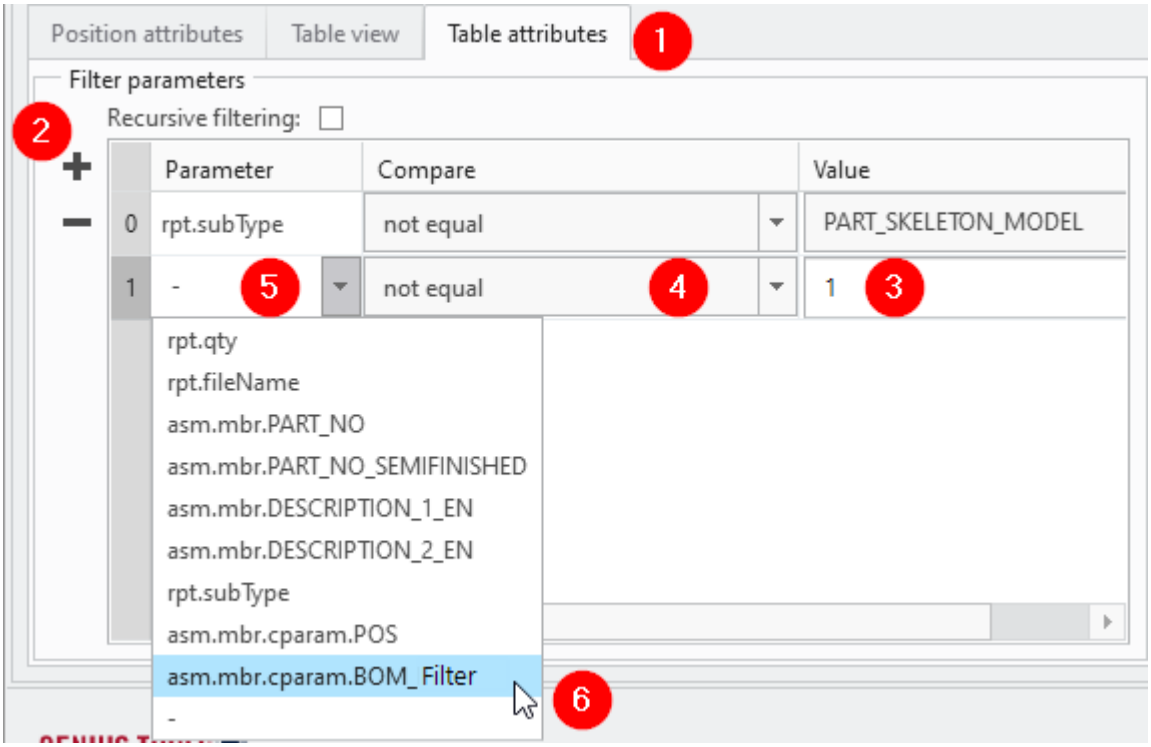

*Preview of the parts list. The appearance of the table must be confirmed via "Save".*

## Adding filter parameters

Then continue working in the *Assembly Report Editor*.

1. Under *Table attributes*, define the criteria according to which the components of the model tree are filtered. All entries that match these filter rules are displayed in the parts list.
2. Use ✚ to add further criteria.
3. Enter a value according to which filtering is performed.
4. Determine the comparison operation for this value.
5. Click in the cell to see a drop-down menu. In it you can select the component parameter you just created.


© 2026 INNEO Solutions GmbH



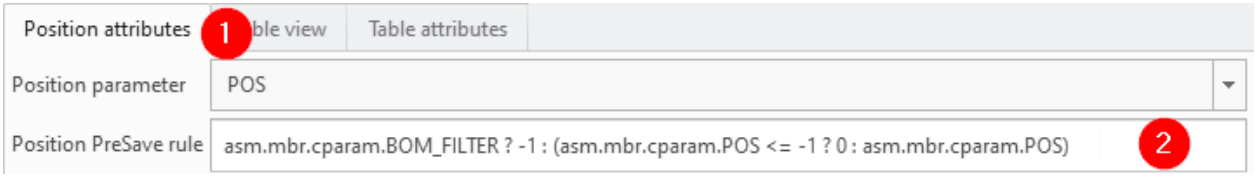
*Defining a filter parameter*

# Applying the Position PreSave rule

1. Click *Position attributes*.
2. Enter this rule:

```
asm.mbr.cparam.BOM_FILTER ? -1 : (asm.mbr.cparam.POS <= -1 ? 0 :
asm.mbr.cparam.POS)
```

The default value for the position parameter of non-displayed components is `-1`. The configuration option `gta_clear_pos_parameter` defines that all components of the model are assigned `-1`. The Position PreSave rule entered here offers you the possibility to set `-1` more selectively. For this, `gta_clear_pos_parameter` must be set to `1`.



*Set Position PreSave rule*

### Effects of the rule

This rule controls the position parameter depending on the component parameter BOM_FILTER (Boolean 0/1).

`asm.mbr.cparam.BOM_FILTER ? -1 : (asm.mbr.cparam.POS <= -1 ? 0 : asm.mbr.cparam.POS)`

`BOM_FILTER` is the check that determines whether `-1` or `(POS <= -1 ? 0 : POS)` is executed.


© 2026 INNEO Solutions GmbH

This results in the following possibilities how the value is set:

- **Case 1:** `BOM_FILTER` equals 0 (false) and `POS <= -1`
  The position parameter receives the value 0.
- **Case 2:** `BOM_FILTER` equals 0 (false) and `POS > -1`
  The position parameter keeps its value.
- **Case 3:** `BOM_FILTER` equals 1 (true)
  The position parameter receives the value `-1`. This concerns entries that are not displayed.

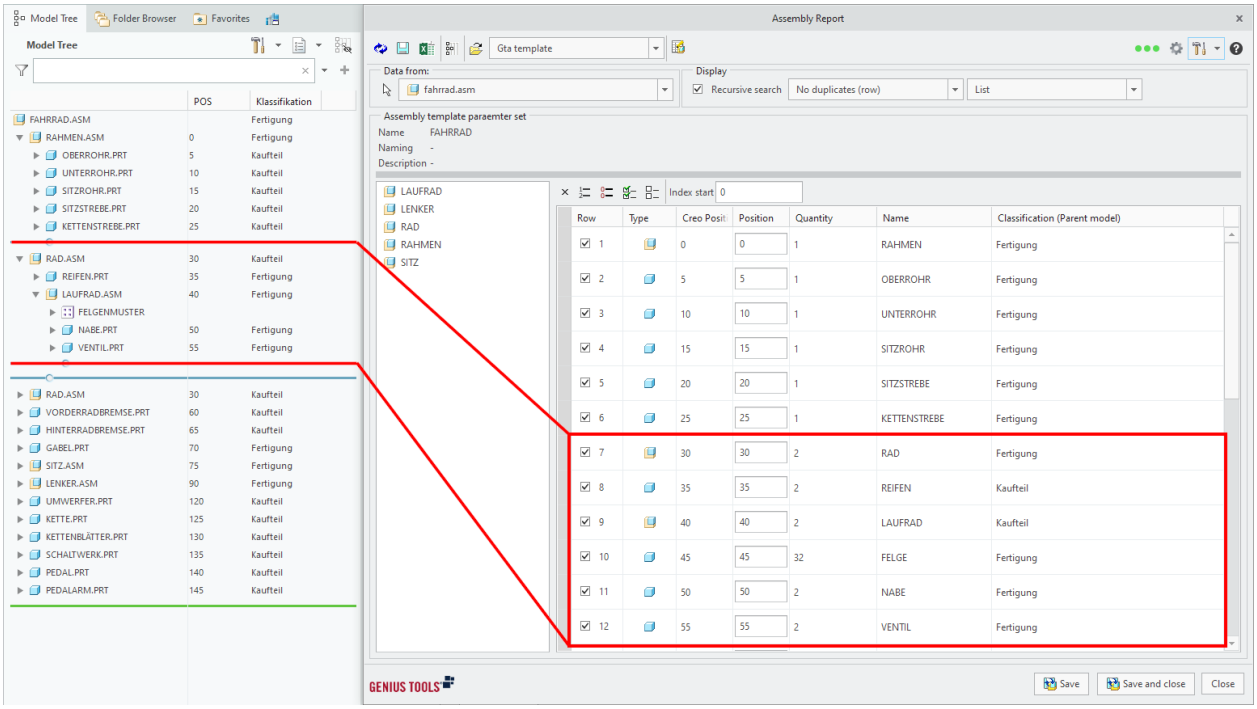
## 5.3.6.2 Filtering out subcomponents of purchased parts

The following application example illustrates how purchased parts are displayed in a BOM with submodels (recursive BOM), but the submodels are sorted out.

### Starting state

You see the model tree for the opened model and the *Assembly Report*. In both user interfaces, the parts that belong together are highlighted. These parts are considered in the example. The following steps are performed in the *Assembly Report Editor*.

1. Open the *Assembly Report Editor*.




*Bicycle with specification of position number and classification in the model tree and the corresponding lines in the Assembly Report*


© 2026 INNEO Solutions GmbH

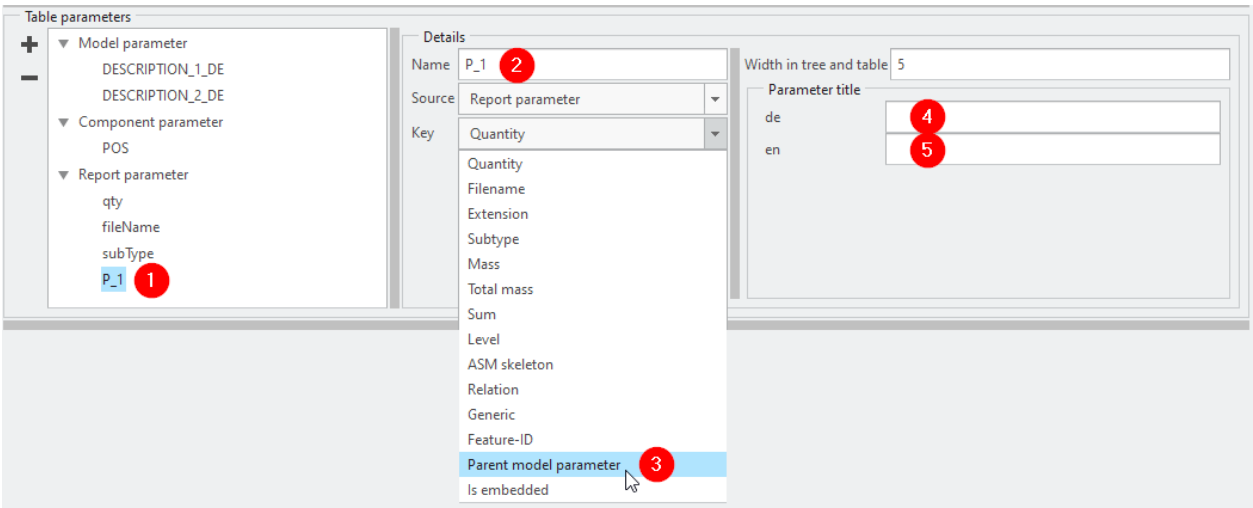
## Creating a report parameter

1. Click *Report parameter*.
2. Create a new parameter by clicking *Add a parameter* ✚.



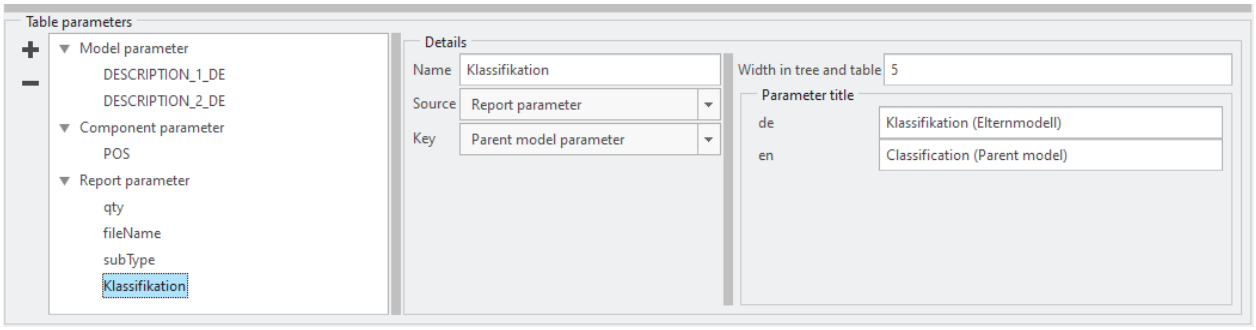

## Defining a new report parameter as a parent model parameter

1. Select the created report parameter with a click.
2. Set a name.
3. Use the drop-down menu to select a key, in this case *Parent model parameter*.
4. Specify under which column title the new report parameter will be listed in the parts list - in German
5. and English.




*Creating a report parameter in the Assembly Report Editor, with intermediate steps.*

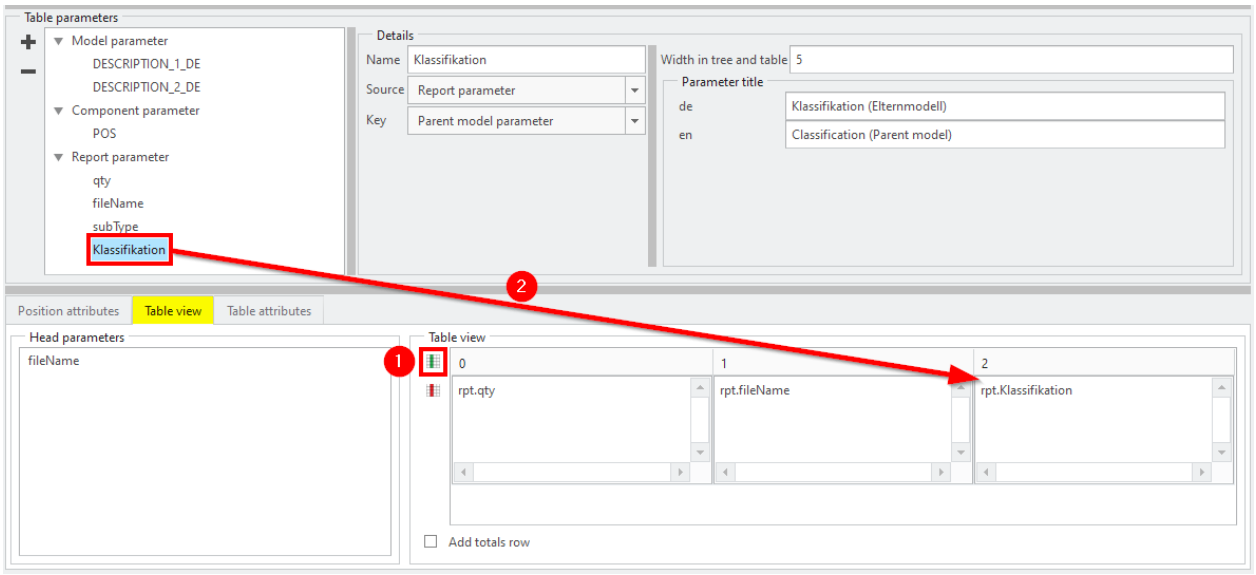

© 2026 INNEO Solutions GmbH




*Report parameter filled in completely*

# Adding a new report parameter to the report table

1. To add a new empty column, click in the column you want to be to the right of the new column and click the *Add a column* button.
2. Click the newly created component parameter and drag and drop it into the new empty table column.

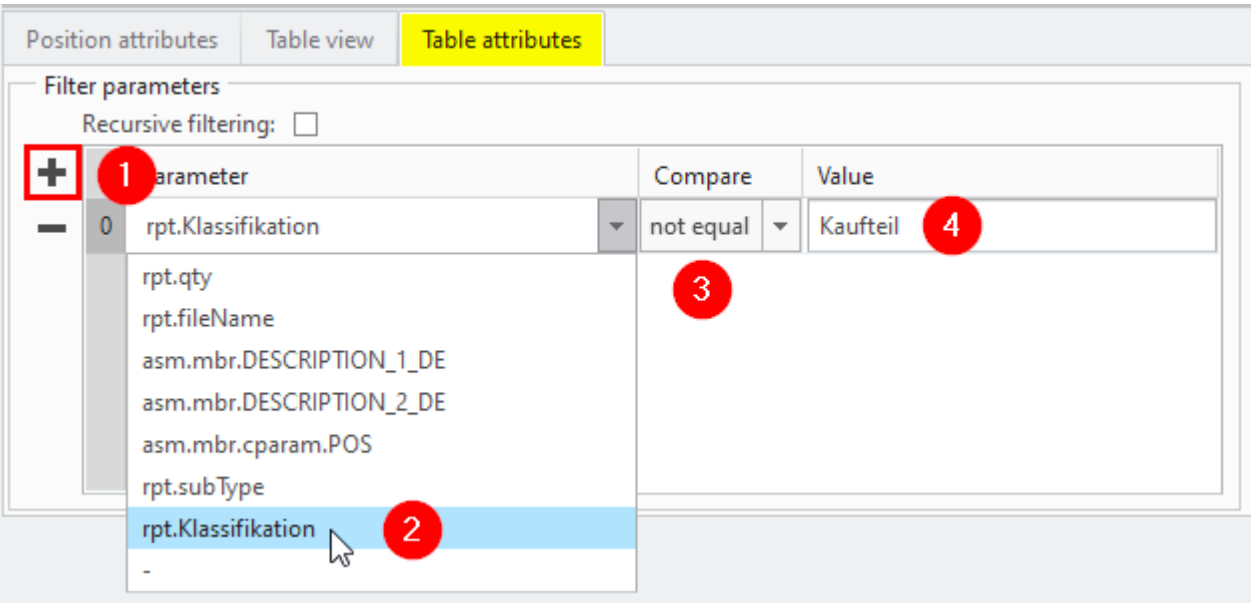



# Adjusting the filter

1. Under *Table attributes*, use ✚ to add a new filter.
2. to 4. Define the filter that will be applied for classification.

The filter defined in this case specifies that all models whose parent models are not purchased parts are included in the parts list.

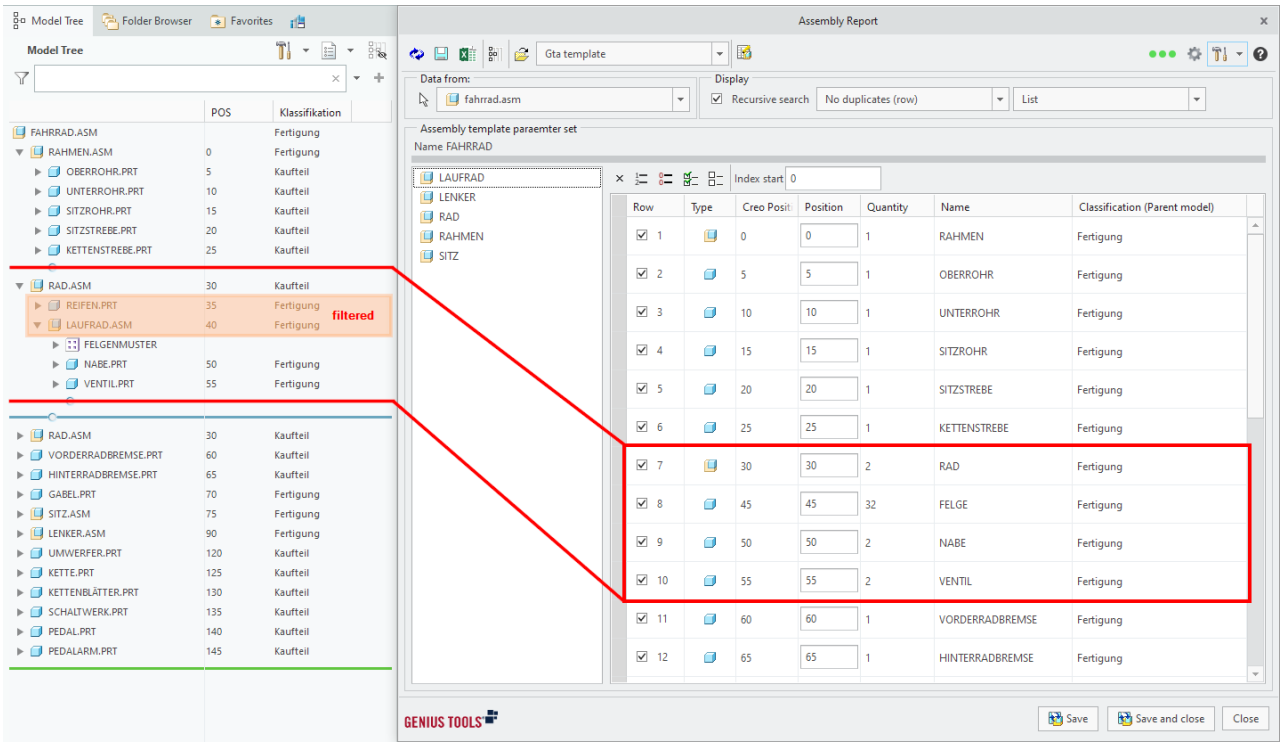

© 2026 INNEO Solutions GmbH




*Creating a filter rule for classification*

# Interim result

Comparing Model Tree and the table in the Assembly Report, you can see that two entries have been filtered out. However, the models on the subsequent lower layers were not filtered out, although NABE.PRT and VENTIL.PART are not purchased parts either. Therefore, the filter must be adjusted.

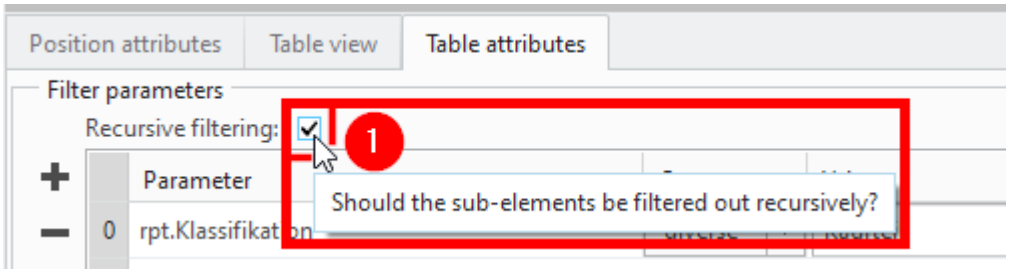



*Interim result of filtering via classification*


© 2026 INNEO Solutions GmbH

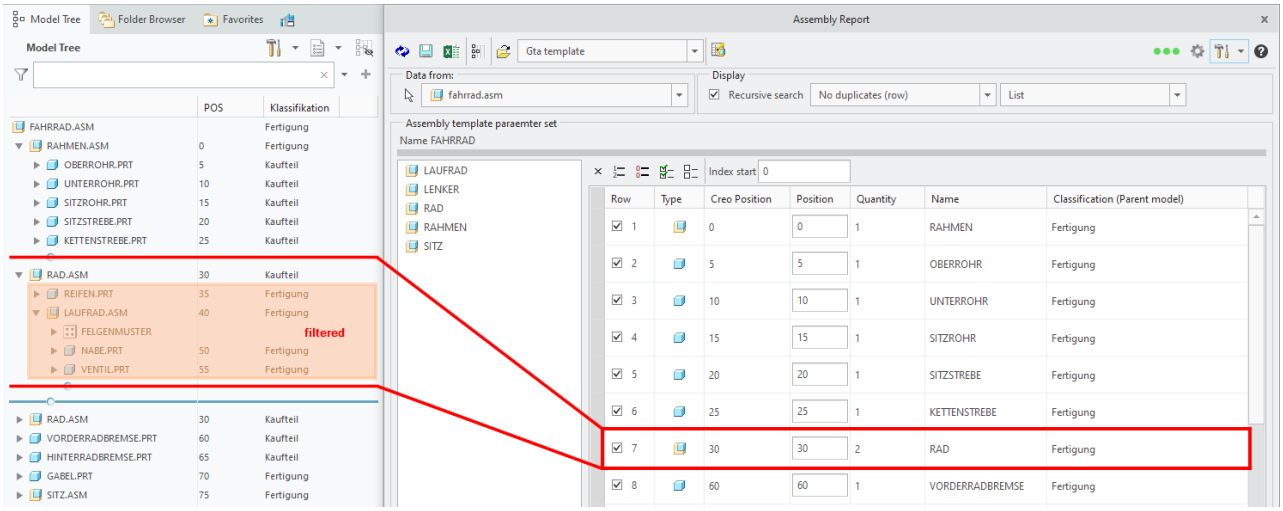
## Setting filter to Recursive

In the *Assembly Report Editor*, under *Table attributes*, activate the *Recursive filtering* function by ticking the checkbox. This will apply the filter rules to all sub-elements as well.




## Result

Now compare the model tree and the assembly report. You will now see that all models in the orange area have been filtered out. Note that *Classification* and *Classification (parent model)* are different parameters. *Classification (parent model)* is a report parameter that is useful for filtering.




*Final result with all filtered out models*


© 2026 INNEO Solutions GmbH

# 6 Dimension

Available in the ribbon menus GENIUS TOOLS | GT MBD

With *Dimension* you can quickly, clearly and comfortably edit dimension values and names of parts and assemblies, features, feature groups and variable UDF dimensions.

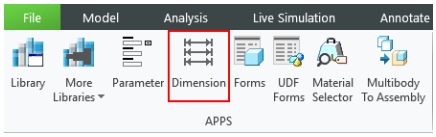
*Dimension* is available in assembly, part, sheetmetal and drawing mode, providing the following features:

1. Display dimensions with following properties:
   - Dimension type (linear, angle, diameter, radius)
   - Dimension name
   - Dimension status (colored background in relations, family tables etc.)
   - Dimension value
   - Tolerance type
   - Boundary
2. Filter displayed dimensions by name, dimension type and tolerance type
3. Dynamic search for dimensions
4. Highlighting dimensions in the graphics window when selecting a value in the *Dimension* window
5. Easy renaming of dimensions
6. Assigning new dimension values (including mathematical expressions)
7. Quickly displaying the original Creo dialog “Dimension properties”
8. Quick call-up of the relations dialog (for relation-driven dimensions)
9. Quickly assigning dimensions to family tables
10. Saving the values as CSV-file

## 6.1 Starting the program

To start *Dimension*  , click *GENIUS TOOLS* or *GT MBD* in the ribbon menu and go to *APPS* section or via *Quick Access* (key [<]).


© 2026 INNEO Solutions GmbH




*Starting via the GT MBD in the ribbon menu*



*Starting via Quick Access*

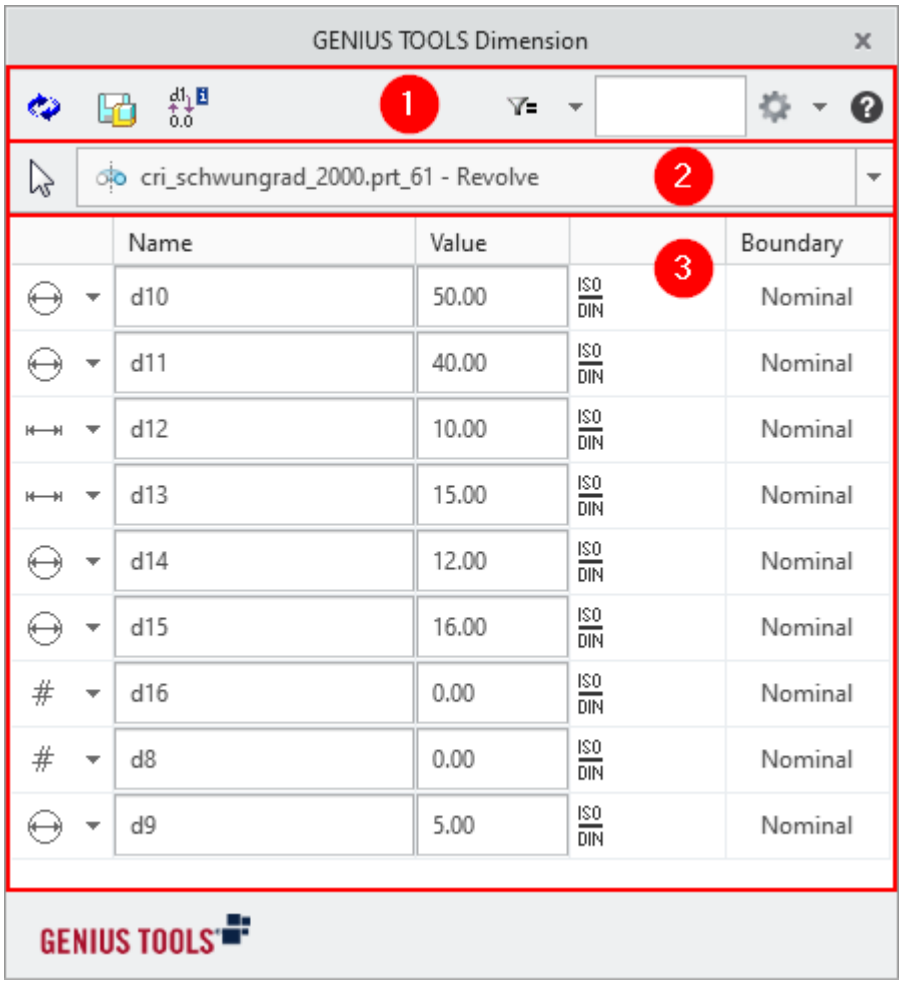
Dimension starts only after selecting an element in the Creo main window or in the model tree. Alternatively, an element can be selected first.

**Please note**: When exiting *Dimension*, only changes with valid inputs will be save<<d.


© 2026 INNEO Solutions GmbH

# 6.2 User interface

The user interface of *Dimension* consists of the following elements:



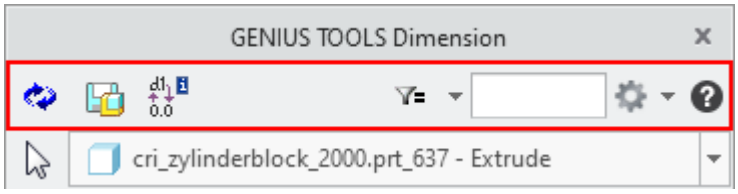

1. Command bar[95] with filter[95]
2. Model selection[97]
3. Dimensions table[98]

The user interface height is calculated by the amount of shown dimensions. Maximum 1/2 of screen size. Minimum 4 lines. The width is calculated by the combined width of the shown columns.


© 2026 INNEO Solutions GmbH

## 6.2.1 Command bar

The command bar consists of general control elements and the filter area with search field.




The following elements are available in the command bar:

| Icon | Name | Description |
|---|---|---|
| | Reload | Reloads the currently selected model or feature. Unsaved changes will be discarded. |
| | Save and regenerate | Saves all changes made and regenerates the whole model.<br>**Note:** It is not possible to save with invalid (highlighted red) input fields or not performed calculations. |
| | Change dimensioning display | Changes the display style of dimensions in Creo Parametric between dimension value and dimension name. |
| / | Filter inactive / active | Opens the list of available filter options[95]. If filter options are already active, the icon is displayed green. |
| | Search | Search field to search for names and values. |
| | Tool menu | Selection to open a family table from current model or save as CSV file with comma or semicolon. |
| | Open Help | Opens the Help. |

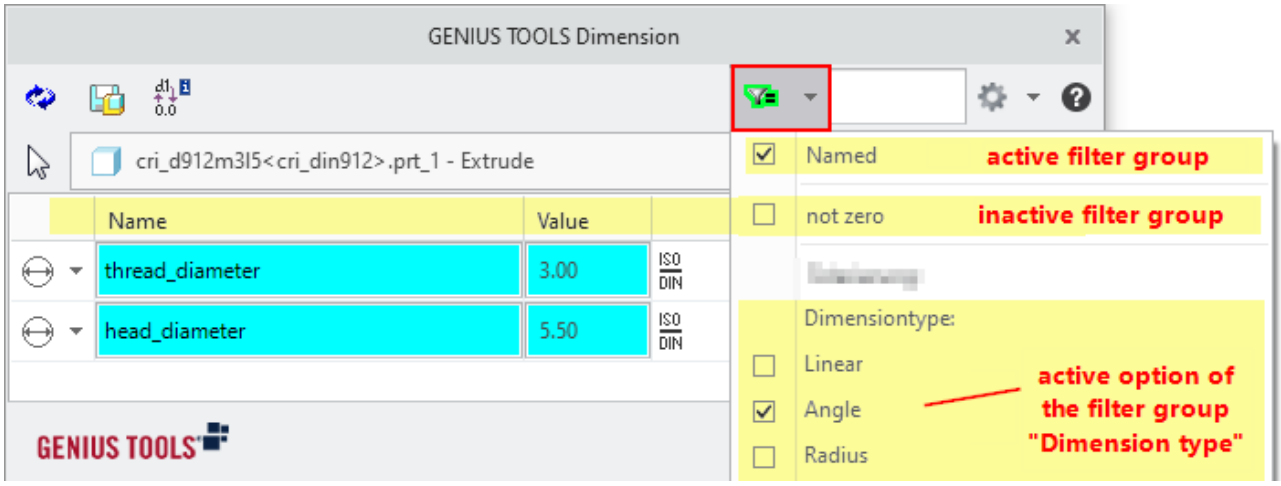
## 6.2.2 Filter and search

The filter controls the display of dimensions with predefined options. Click on the filter icon and select the appropriate filter options.


© 2026 INNEO Solutions GmbH

# Filter options

The filter controls the display of dimensions with predefined options. Click the filter icon and select the appropriate filter options.




**Filter behavior**

The selected filter options are linked with each other ("AND operation"), i.e. in the example, all named measures that correspond to the measure type "Angle" are displayed.

Within a group, all dimensions that match the active filter options are displayed.

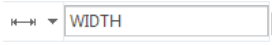
The configuration option `gtd_filter_additive=1` can be used to change the filter behavior so that the results of all active filters are displayed individually. That means, in the example above, all named dimensions as well as all angle dimensions would be displayed ("OR-operation").

**Filter groups**

– Named dimensions / Unnamed dimensions / name filter: Shows all dimensions with modified names.



Unchanged name of a dimension



Modified name of a dimension, a named dimension

– not zero: Displays all dimensions that have a value different from zero.

– Tolerance: Displays all dimensions that correspond to the selected tolerances.

– Dimension type: Displays all dimensions that correspond to the selected dimension types.

– Boundaries: Displays all dimensions that correspond to the selected limits.

– Value `(0.001)` and Value `(5.000)`: All dimensions are displayed that are equal to, smaller than or larger than the selected value. It is possible to search for two different values, e. g. to search for a range of values. The values that can be filtered by are adjustable via `<<Change Value>>` in the user interface. Alternatively, the values can be set with the

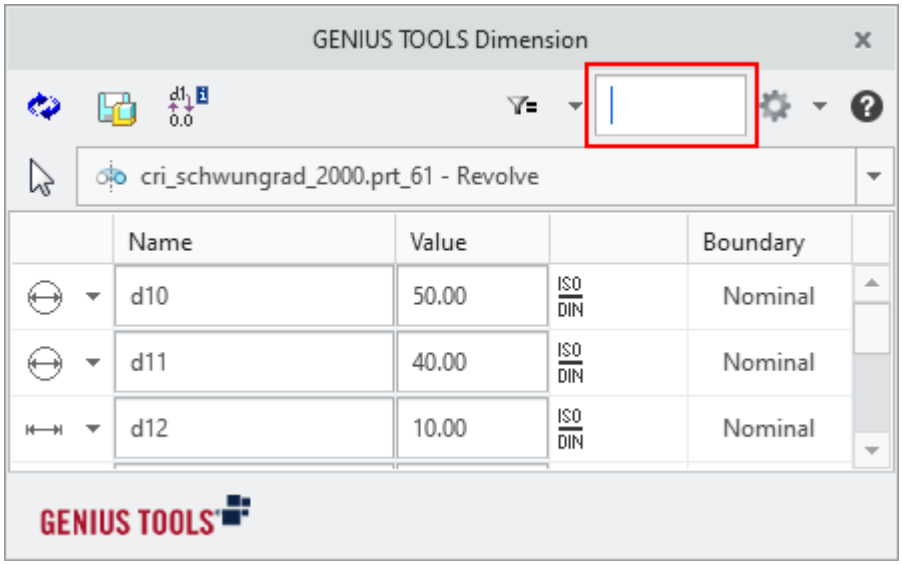

© 2026 INNEO Solutions GmbH

configuration options `gtd_filter_value` and `gtd_filter_value2`. The filter options are only active if the configuration option `gtd_filter_additive=0` is set (default).

**Default filter selection for named dimensions**
With the configuration option `gtd_filter=1` you can set that the name filter *(Named dimensions)* is activated when *Dimension* is started.

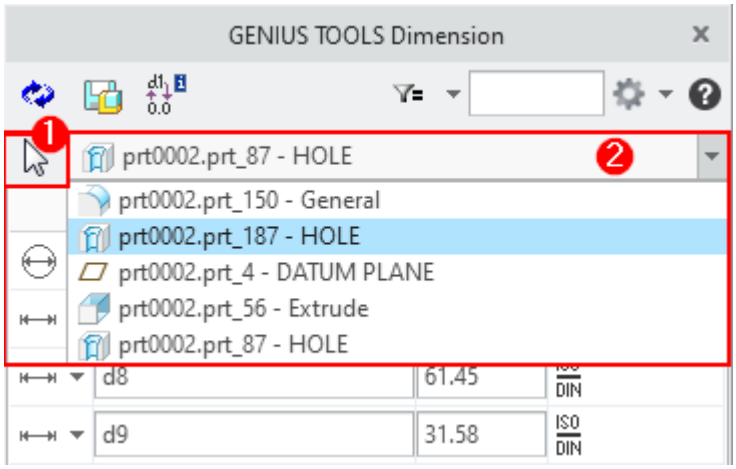
## Search field

Use the search to search for names and values of dimensions. Only the currently displayed dimensions will be searched.




## 6.2.3 Model selection

The model selection consists of the two elements: object selection and drop-down list.



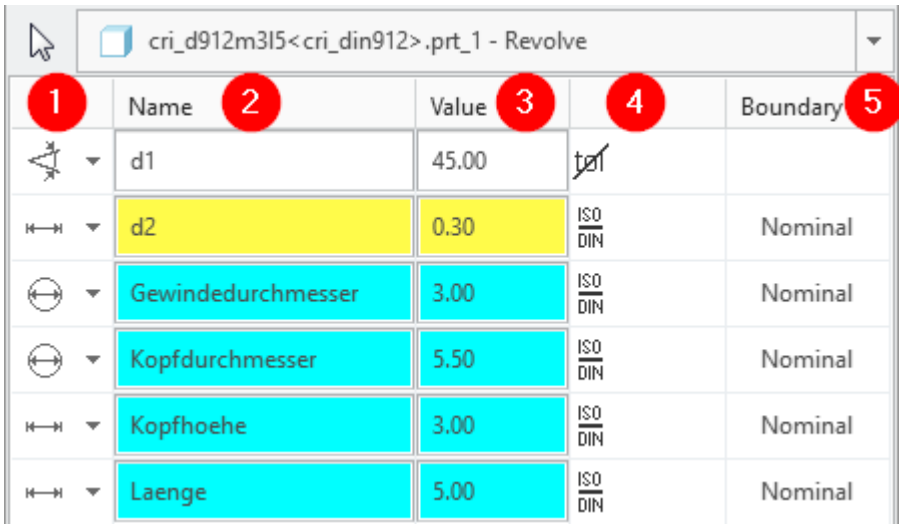

Select models, features, groups and UDFs in the model or in the model tree using the object selection (1).


© 2026 INNEO Solutions GmbH

The drop-down list (2) shows the current selection. Open the list and select from the recently selected objects.

## 6.3 Dimensions table

The dimensions table displays each dimension in a separate row:




1. Dimension symbol with context menu[99]
2. Dimension name with color highlighting[101]
3. Dimension value[100] with color highlighting[101]
4. Tolerance type[102]
5. Boundaries[103]

Before each dimension, the *symbol* corresponding to the dimension type is displayed (1). Click on the symbol to open the context menu for the dimension. The menu shows suitable options for each dimension type.

Click into input fields (columns *Name* (2) and *Value* (3)) to highlight the selected dimension in the Creo model.

**Please note**: In drawing mode, only the displayed dimensions are highlighted.

Behind each dimension, the *tolerance type* (4) is displayed. Click on the icon to open the ribbon *Dimension* for the selected dimension. The applied dimension boundary is displayed under *Boundary* (5). *Nominal* is the default setting that is displayed when no dimension boundary is explicitly defined.


© 2026 INNEO Solutions GmbH

## 6.3.1 Dimension symbols and context menu

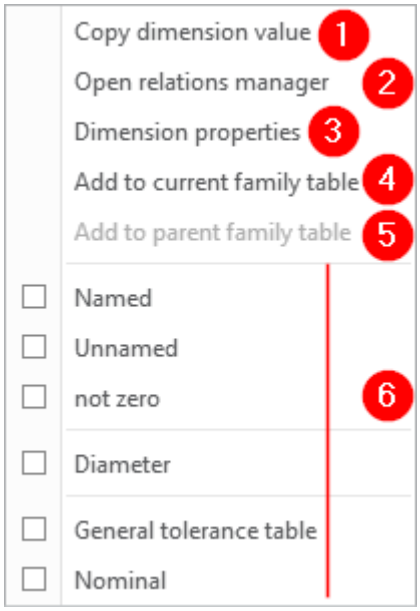
The following symbols are displayed in front of the dimensions:

| Icon | Description |
|---|---|
| | Linear dimensioning |
| | Radial dimensioning |
| | Diameter dimensioning |
| | Angular dimensioning |
| # | Pattern count |
| | Thread |

Click on a dimension symbol to open the context menu for the dimension.

## Context menu

The following options are available in the context menu:





© 2026 INNEO Solutions GmbH

1. **Copy dimension value**
   Pastes the value of a dimension to be selected in the model into the current dimension.
2. **Open relations manager**
   Opens the Creo relations dialog.
3. **Dimension properties**
   Opens the Creo dimensions dialog.
4. **Add to current family table**
   The selected dimension will be added to the family table of the current model.
5. **Add to parent family table**
   The selected dimension will be added to the family table of the generic part.

**Warning:** The two commands **Add to current family table** and **Add to parent family table** will be executed immediately; no saving or regenerating is required.

6. Specific filter options depending on the dimension type can be found below the general options.

## 6.3.2 Input fields

In input fields, dimensions are displayed with names and associated values. They are sorted alphabetically by name.

Click into an input field to modify the name or the value of a dimension. Dimensions are highlighted in the Creo main window when you click in an input field.

**Warning**: Do not rename dimensions according to the scheme ad[Number] (e.g. ad34)! ad dimensions are used internally by Creo Parametric and are filtered out automatically.

Changes can be made in white input fields. The meaning of colored input fields can be found in the table in section Colors in input fields 101.

Dimensions without a specific dimension type are treated as count and will be automatically rounded to integers.

The width of the table columns **Name** and **Value** are controlled with the configuration options `gtd_label_col_size` and `gtd_value_col_size`. The number of decimal places in the **Value** column is controlled with the configuration option `gtd_dec_places`.

**Warning**: Check the Creo configuration option `show_dim_sign`, as *Dimension* respects it.
`show_dim_sign=no`: Entering a negative value changes the direction vector of a dimension. The value will be positive.
`show_dim_sign=yes`: When a negative value is entered, the value remains negative.


© 2026 INNEO Solutions GmbH

## Calculations

Input fields for values allow calculations as in Creo. Input up to 250 characters is supported.

Enter a mathematical function into a value field and press Enter. The result is copied into the field. After saving and regenerating, the result will be applied to the model.

**Please note**: When calculations create unusable values (e.g. -1 in a count field), the last valid value will be re-entered. An information will be prompted in the Creo Parametric message area.

Also use more complex mathematical operations such as root, powers, or rounding in value fields.

**Examples:**

| | |
|---|---|
| sqrt(d27) | square root of the dimension d27 |
| pow(d23,3) | third power of the dimension d23 |

## 6.3.3 Colors in input fields

Input fields in the dimensions table have a color coding to show how the dimension is controlled. Move the mouse over an input field, a tooltip shows the meaning.

| Field color | Description |
|---|---|
|  White | Unchanged name or value. |
|  Red | Invalid content or the dimension name already exists in the model; it is not possible to save. |
|  Green | Modified content, the dimensions table needs to be saved. |
|  Yellow | Locked dimension, e.g. by a relation. Value and Name cannot be modified.<br>**Tip:** The dimension can be edited in the Creo relations dialog / ribbon (accessible via the Dimension context menu). |
|  Blue | The dimension is determined by the family table of the current model. |


© 2026 INNEO Solutions GmbH

| Field color | Description |
|---|---|
|  Cyan | The dimension is determined by the family table of the parent model. Value and name cannot be modified using the dimension table. |
|  Magenta | The dimension is determined by a pattern table. Value and name cannot be modified using the dimension table. |

## 6.3.4 Tolerance types

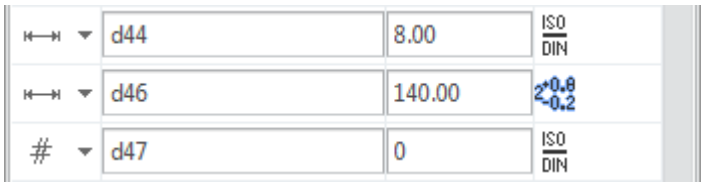
The tolerance type is displayed behind dimensions. Click the icon to open the *Dimension* ribbon.

| Icon | Description |
|---|---|
|  | No tolerance table. |
|  | Tolerance table “General according to DIN/ISO”. |
|  | Tolerance table “Shaft with tolerance class”. |
|  | Tolerance table “Hole with tolerance class”. |
|  | Tolerance table “Broken edge”. |
|  | Tolerance mode: “Plus-Minus” (The symbol does not display the values.)<br>No tolerance table, but free values. |
|  | Tolerance mode: “Limits”. (The symbol does not display the values.)<br>No tolerance table, but free values. |
|  | Tolerance mode: “+-symmetrical” (The symbol does not display the values.)<br>No tolerance table, but free values. |
|  | Tolerance mode: “+- symmetrical (superscript)” (The symbol does not display the values.)<br>No tolerance table, but free values. |

**Please note**: Tolerances are not displayed for threads. The entire dimension text is displayed instead.

Tolerance symbols change color (blue or cyan) when tolerances come from a family table.


© 2026 INNEO Solutions GmbH




## 6.3.5 Boundaries

Dimension boundaries are displayed in the column *Boundaries* (Upper, Middle, Lower, Nominal):

| Name | Value | | Boundary |
|---|---|---|---|
| d10 | 50.00 | | Middle |
| d11 | 40.00 | | Upper |
| d12 | 10.00 | | Lower |
| d13 | 15.00 | | Upper |
| d14 | 12.00 | | Upper |
| d15 | 16.00 | | Nominal |

**Tip:** The boundaries can be changed in the Creo ribbon menu *Analysis* > *Dimension Boundaries* by selecting the dimension and setting the desired dimension boundary.


© 2026 INNEO Solutions GmbH

# 7 Export TDP

Available in the ribbon menu GT MBD

The module *Export TDP* allows you to create technical data packages for transferring model specifications and other information. The TDP replaces the technical drawing and provides the working basis for the work processes that follow the design.

Technical data packages are generated on the basis of chapter-oriented templates (TDP templates). The chapters can be inserted and sorted in any order. Chapters are also defined in templates (chapter templates). Only the chapters define which information and data are inserted into a TDP. The most important elements are the combined views from Creo Parametric.

The exported PDF files can be created in the following ways:

– Structuring PDF files by chapters

– Optional content: Title page, table of contents, chapters with one or more combined views

– Format: classic drawing format with drawing frame or book format without drawing frame

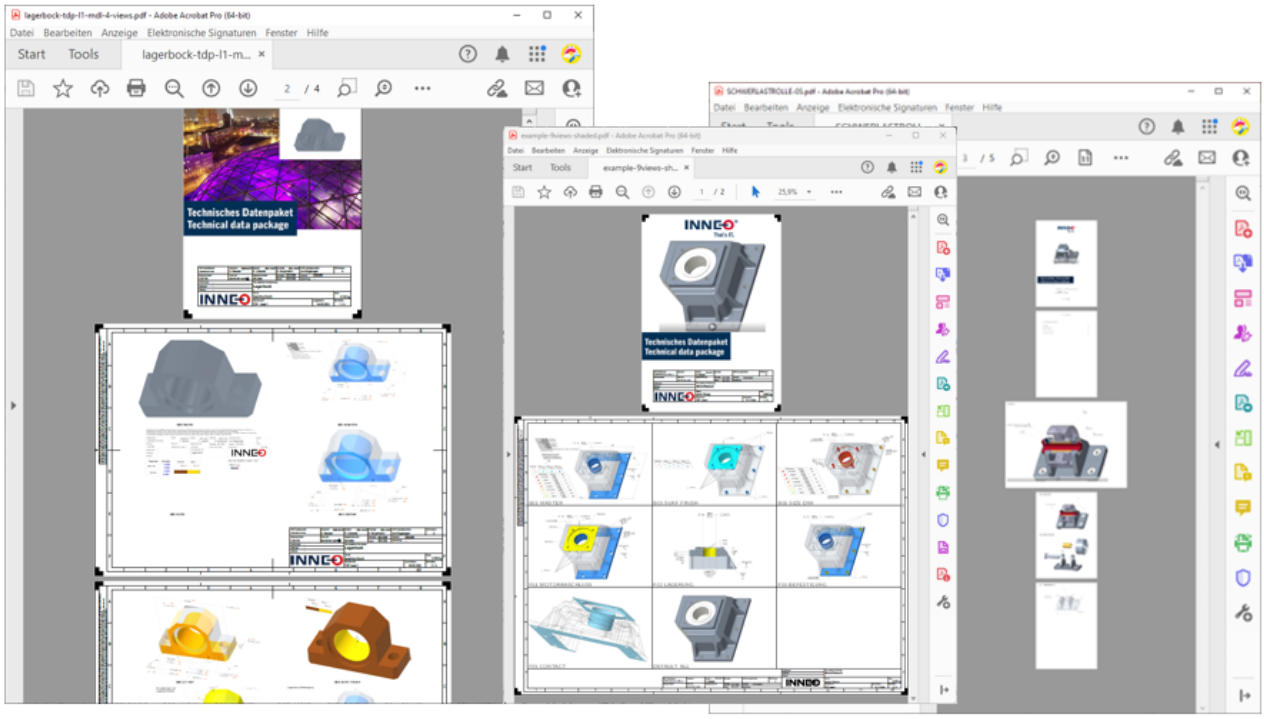
– Embedding files, e.g. XMP data with model parameters

Both PDF files and PDF/A archive files can be generated.

The export of PDF files can be automated with Model Processor Worker Extension.

Examples of various output options:


© 2026 INNEO Solutions GmbH

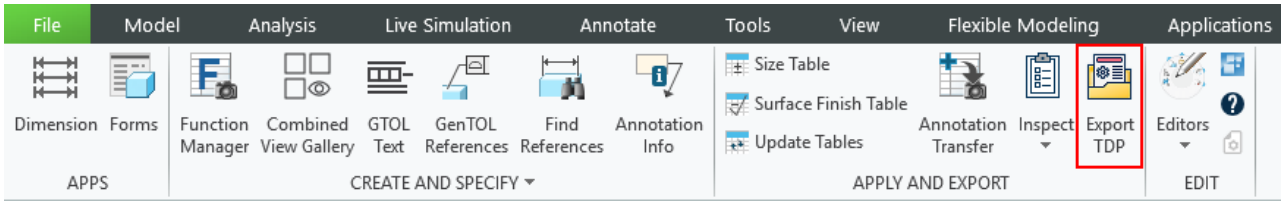


# 7.1 Usage

In this section you will find information on the use of *Export TDP*.

## Starting the program

To start *Export TDP* , click *GT MBD* in the ribbon menu and go to *APPLY AND USE* section.

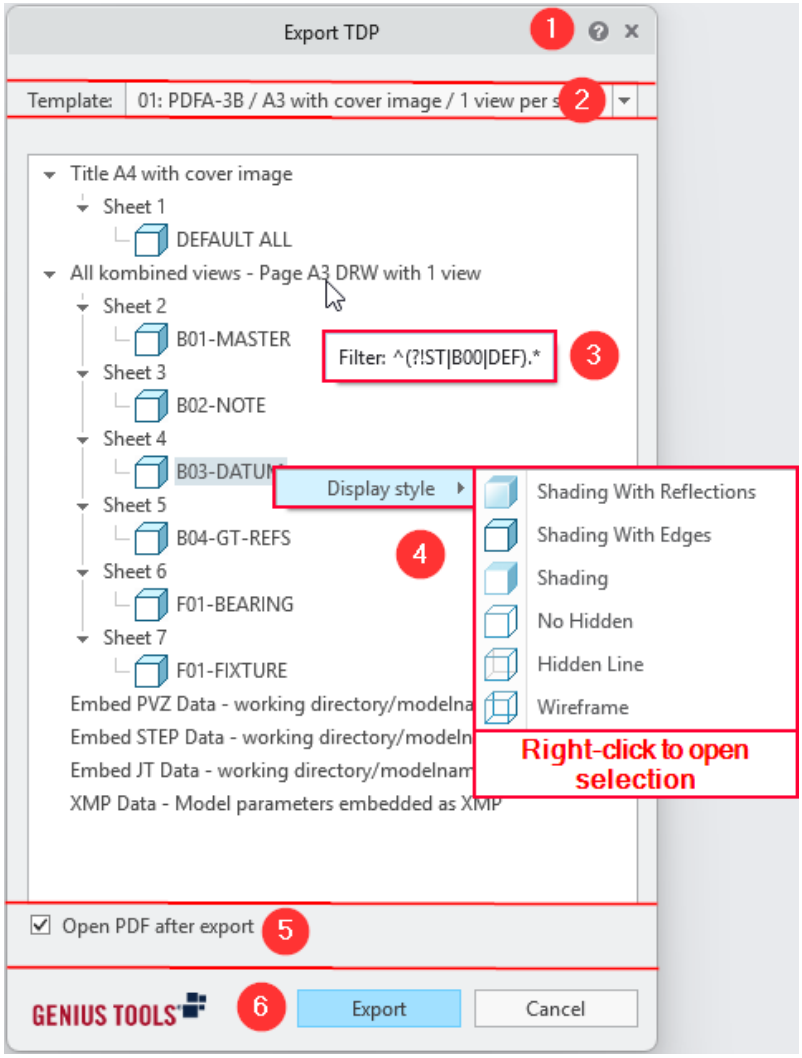



*Starting via the GT MBD in the ribbon menu*

## 7.1.1 User Interface

The *Export TDP* user interface consists of the following elements:


*© 2026 INNEO Solutions GmbH*

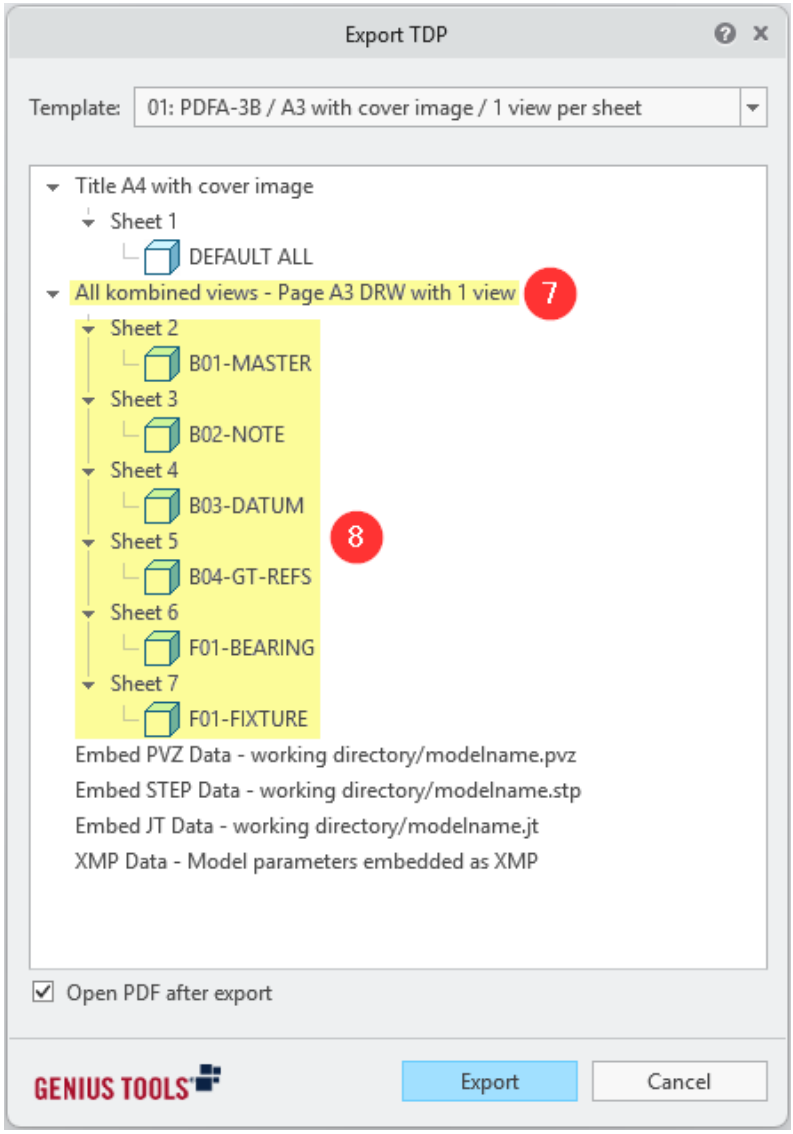



1. Opens this help.
2. TDP templates[108]
   TDP templates: Selection of the TDP templates to export the 3D model.

**Tip:** Shows the name filter (3), which of the combined views should be displayed in the chapter.

4. Chapter overview with the selection menu to define the display style of a view.
5. Option to open the PDF after export
   You can use the configuration option `gttdp_open_after_export` to activate this option as the default.
6. Export and cancel


© 2026 INNEO Solutions GmbH




7. Language-dependent chapter heading consists of the title of the TDP template and the title of the chapter template.
8. Combined views are displayed for the respective sheet.

## 7.2 Configuration

The Latex text typesetting system is used to export the 3D models. Latex is a freely available software for document creation. It is particularly suitable for technical documents, as complex representations can be clearly integrated into PDFs. Latex does not need to be installed on the application computers.

In this section you will find more information on how to use the module Export TDP and how to create export templates.


© 2026 INNEO Solutions GmbH

## 7.2.1 Editing export templates

The Latex templates (XML files) are structured according to a fixed scheme. *Export TDP* provides a ready-made Latex package `<latexUsepackage>`. This package contains all of the Latex commands that are needed to export a 3D model. The templates are predefined for displaying PDFs in German and English.

The supplied templates are located in the folders *template, chapters, components*:

– *templates*
  TDP templates to define the structure of the PDF[108]

– *chapters*
  chapter templates to define individual chapters[114]

– *components*
  Storage location for drawing formats that are used in chapter templates, e.g. templates for drawing frames and company logos

Each export requires one chapter template from the *chapters* folder and one TDP template from the *templates* folder. The drawing formats under *components* are used for each export regardless of the template selected.

**Please note:** When editing Latex templates, make sure that you only adjust information. Do not make any structural changes. Missing elements, e.g. brackets and backslashes, mean that a template cannot be executed. In this case, an error message is displayed in the message log.

The export templates and storage locations can be defined with configuration options, see Available configuration options[117].

### 7.2.1.1 Defining TDP template

The structure of the PDF files is defined by the templates in the folder *template*. The file path is set in the configuration option `gttdp_template_folder`.
(Default: *%gt_resource_folder%export_tdp\templates*)

The following example explains the input needed to build a TDP template.

Example: TDP template *01-drw-a3-titlemodel-1view.xml*


© 2026 INNEO Solutions GmbH

```
<?xml version="1.0" encoding="UTF-8"?>
<gttdp version="0.1" name="FIRST_DEFINITION">
    <info>
        <title>
            <de>01: A3 mit Titelmodel / 1 Ansicht pro Blatt</de>
            <en>01: A3 with title sheet / 1 view per sheet</en>
        </title>
        <description>
            <de></de>
            <en></en>
        </description>
        <icon></icon>
        <function></function>
        <help></help>
    </info>

    <elements autoAddNewline="true" runLatexTwice="false" displayCS="false">
        <chapter component="title-01-a4-drw-titlemodel.xml" combstatefilter="^STAND.*|^DEFAULT.*"[...]
        <chapter component="page-01-drw-a3-1-view.xml" combstatefilter="^B(?!00).*" maxpage="99">
            <title>
            <de>Basisfunktionen</de>
            <en>Basic functions</en>
            </title>
        </chapter>
        <chapter component="page-01-drw-a3-1-view.xml" combstatefilter="^F.*" maxpage="99">
        <title>
            <de>Produktfunktionen</de>
            <en>Product functions</en>
            </title>
        </chapter>
        <chapter component="page-01-drw-a3-1-view.xml" combstatefilter="^P.*" maxpage="99">
        <title>
            <de>Prozesse</de>
            <en>Processes</en>
            </title>
        </chapter>
        <chapter component="embed-01-pvz.xml">
        <title>
            <de>PVZ Daten einbinden</de>
            <en>Embed PVZ Data</en>
            </title>
        </chapter>
        <chapter component="embed-02-step.xml">
        <title>
            <de>STEP Daten einbinden</de>
            <en>Embed STEP Data</en>
            </title>
        </chapter>
    </elements>
```


© 2026 INNEO Solutions GmbH

```
<latex>
    \documentclass[a4paper, parskip=half ,12pt]{scrartcl}
    \usepackage[german]{babel}
    \babelprovide[hyphenrules=german-x-latest]{german}
    \usepackage[utf8]{inputenc}
    \usepackage[T1]{fontenc}
    \usepackage[pdftex,
    colorlinks=true,    %Bei false wird ein roter Rahmen,um die Links gesetzt.
    linkcolor=black,    %Linkfarbe im PDF schwarz machen. Für elektronische Version ändern.
    citecolor=black,    %Linkfarbe im PDF schwarz machen. Für elektronische Version ändern.
    linktocpage=true,
    urlcolor=blue,
    bookmarks=true,      %Inhaltsverzeichnis auf der linken Seite beim öffnen der PDF
    bookmarksopen=true,     %Das Verzeichnis öffnen
    bookmarksopenlevel=1,   %Öffnungstiefe (chapter, section, subsection)
    pagebackref=false,     %Aus Literaturverzeichnissen etc. auf die entsprechende Seite springen
    bookmarksnumbered=true, %Einträge im Verzeichnis nummerieren
    pdfstartpage=1,         %Startseite für die Anzeige festlegen
    pdfpagelabels=true,
    pdfpagemode=UseOutlines,
    pdfauthor={$username$},
    pdftitle={TDP Level 1 @mdln@}
    ]{hyperref}

    % Mandatory Metadata Header Creation for PDF/A-3b
    \begin{filecontents*}{\jobname.xmpdata}
        \Title{@mdln@}
        \Author{$username$}
        \Language{de-DE}
        \Subject{PDF/A-3b compliant document}
        \Keywords{PDF/A, LaTeX, TDP}
    \end{filecontents*}
    \usepackage[a-3b,mathxmp]{pdfx}[2018/12/22]

    @usepackages@
    \setcounter{secnumdepth}{0}
    \pagestyle{headings}

    \begin{document}

    @chapters@

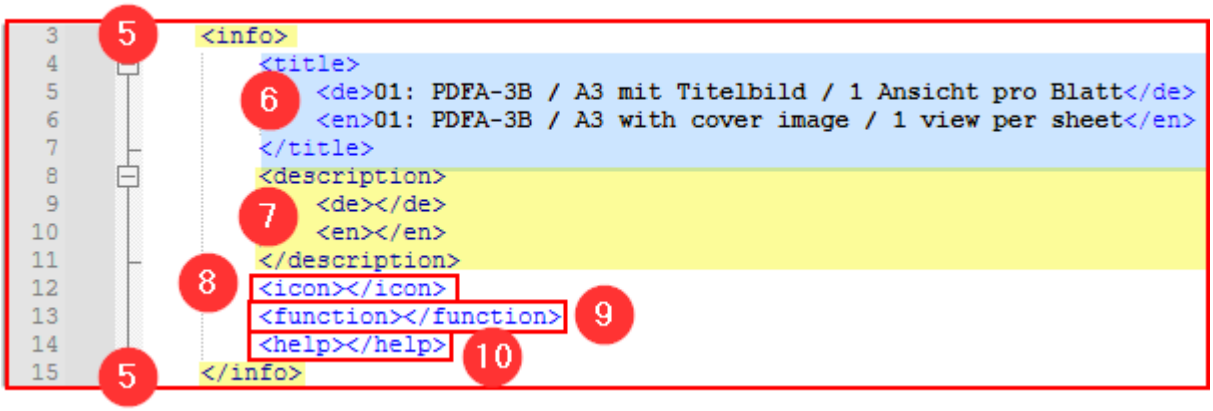
    \end{document}
</latex>
</gttdp>
```

1. `<gttdp>...</gttdp>`
   Tags as frame for the template
   - `version` Specification of a number for versioning the TDP template
2. Information about the template[110]
3. Each of the chapters[111]
4. Composition of the Latex document[112]

# Information about the template

Each template requires the following tags:


© 2026 INNEO Solutions GmbH




5. `<info>... </info>`
   Tags to frame the information section

6. `<title>... </title>`
   Display name of the template in *Export TDP* in English `<en>` and German `<de>`

7. `<description>... </description>`
   Description text in English `<en>` and German `<de>` (optional)

8. *Currently not used*

9. *Currently not used*

10. *Currently not used*

## Each of the chapters

The chapters are inserted into the exported PDF document in the following form:




11. `<elements>... </elements>`

– `autoAddNewline` Do you want to add a row between chapter components? yes/no (`true/false`)

– `runLatexTwice` Will Latex be executed twice? yes/no (`true/false`)
The command `runLatexTwice` must be set to `true` if the PDF contains a table of contents. If a table of contents is to be created, Latex must be executed twice, once to obtain the page numbers and a second time to create the table of contents.


*© 2026 INNEO Solutions GmbH*

– `displayAxes` Should datum axes be displayed? Yes (`true`)/ No (`false`). Default: `true`

– `displayPoints` Should datum points be displayed? Yes (`true`)/ No (`false`). Default: `true`

– `displayCS` Should datum coordinate systems be displayed? Yes (`true`)/ No (`false`). Default: `true`

– `displayPlanes` Should datum planes be displayed? Yes (`true`)/ No (`false`). Default: `true`

– `displaySpinCenter` Should the 3D spin center be displayed in the default position? Yes (`true`)/No (`false`). Default value: `false`

12. `<chapter.../>`

Chapter definition with one additional node:

– `definition` template from the folder *chapters*[114] inserted here as a chapter: *title-01-a4-drw-titleimage.xml* as title page

– `combstatefilter` Specify combined views to be exported using Regular Expressions[487], here: combined view *Default* (de: *Standard*)

– `defaultDisplayStyle` Display Style, here: Shading with edges

– `maxpage` Maximum number of pages for the chapter, preset to 99 pages

– `onlyPublished` Export only combined views that are set to *Publish* in Creo View Manager? yes/no (`true/false`). Default: `false`

13. `<chapter>... </chapter>`

Chapter definition with one additional node:

– `definition` XML template from the folder *chapters*[114] inserted here as a chapter: *page-01-drw-a3-1-view.xml*

– `combstatefilter` Specify combined views to be exported using Regular Expressions[487], here: *B01* with ascending numbering

– `maxpage` Maximum number of pages for the chapter, preset to 99 pages

– `onlyPublished` Export only combined views that are set to *Publish* in Creo View Manager? yes/no (`true/false`). Default: `false`
`<title>... </title>` Additional node, with a heading for all combined views in *page-01-drw-a3-1-view.xml*, here: *Basic functions*

14. *Same principle as in point 13*

15. *Same principle as in point 13*

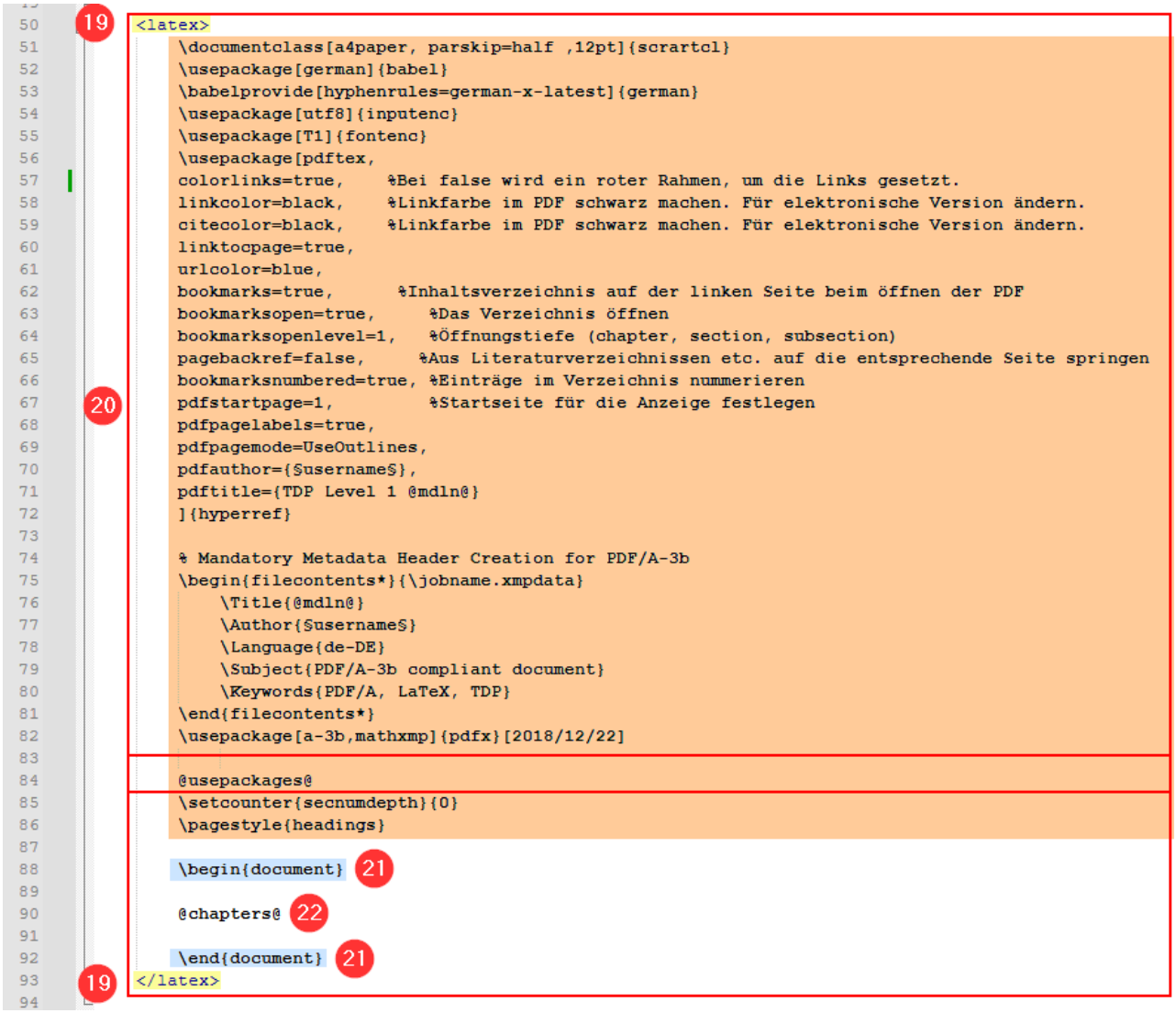
16. *Same principle as in point 13*

## Composition of the Latex document

After all chapters are listed, the Latex commands required for the export are listed.


© 2026 INNEO Solutions GmbH

**Please note:** This part applies to every template. To avoid errors, do not change these commands.



```
<latex>
    \documentclass[a4paper, parskip=half ,12pt]{scrartcl}
    \usepackage[german]{babel}
    \babelprovide[hyphenrules=german-x-latest]{german}
    \usepackage[utf8]{inputenc}
    \usepackage[T1]{fontenc}
    \usepackage[pdftex,
    colorlinks=true,    %Bei false wird ein roter Rahmen, um die Links gesetzt.
    linkcolor=black,    %Linkfarbe im PDF schwarz machen. Für elektronische Version ändern.
    citecolor=black,    %Linkfarbe im PDF schwarz machen. Für elektronische Version ändern.
    linktocpage=true,
    urlcolor=blue,
    bookmarks=true,      %Inhaltsverzeichnis auf der linken Seite beim öffnen der PDF
    bookmarksopen=true,     %Das Verzeichnis öffnen
    bookmarksopenlevel=1,   %Öffnungstiefe (chapter, section, subsection)
    pagebackref=false,     %Aus Literaturverzeichnissen etc. auf die entsprechende Seite springen
    bookmarksnumbered=true, %Einträge im Verzeichnis nummerieren
    pdfstartpage=1,         %Startseite für die Anzeige festlegen
    pdfpagelabels=true,
    pdfpagemode=UseOutlines,
    pdfauthor={$username$},
    pdftitle={TDP Level 1 @mdln@}
    ]{hyperref}

    % Mandatory Metadata Header Creation for PDF/A-3b
    \begin{filecontents*}{\jobname.xmpdata}
        \Title{@mdln@}
        \Author{$username$}
        \Language{de-DE}
        \Subject{PDF/A-3b compliant document}
        \Keywords{PDF/A, LaTeX, TDP}
    \end{filecontents*}
    \usepackage[a-3b,mathxmp]{pdfx}[2018/12/22]

    @usepackages@
    \setcounter{secnumdepth}{0}
    \pagestyle{headings}

    \begin{document}

    @chapters@

    \end{document}
</latex>
```

17. `<latex>... </latex>`
    Command to execute the template with Latex. This template is applied to the model from which the module *Export TDP* is opened.

18. Header of the Latex definition with formatting specifications

**Please note:** The command `@usepackages@` is a placeholder that must be placed in the header of the Latex definition. This is where all the Latex packages (function libraries) are defined. All Latex packages of the subchapters[116] are inserted here.

19. `\begin{document} ... \end{document}`
    Document section with tags for the beginning and the end of the PDF file

20. `@chapter@`
    The contents of the document are listed between `\begin{document} ... \end{document}`. This formatting links all the chapters listed above[111].


© 2026 INNEO Solutions GmbH

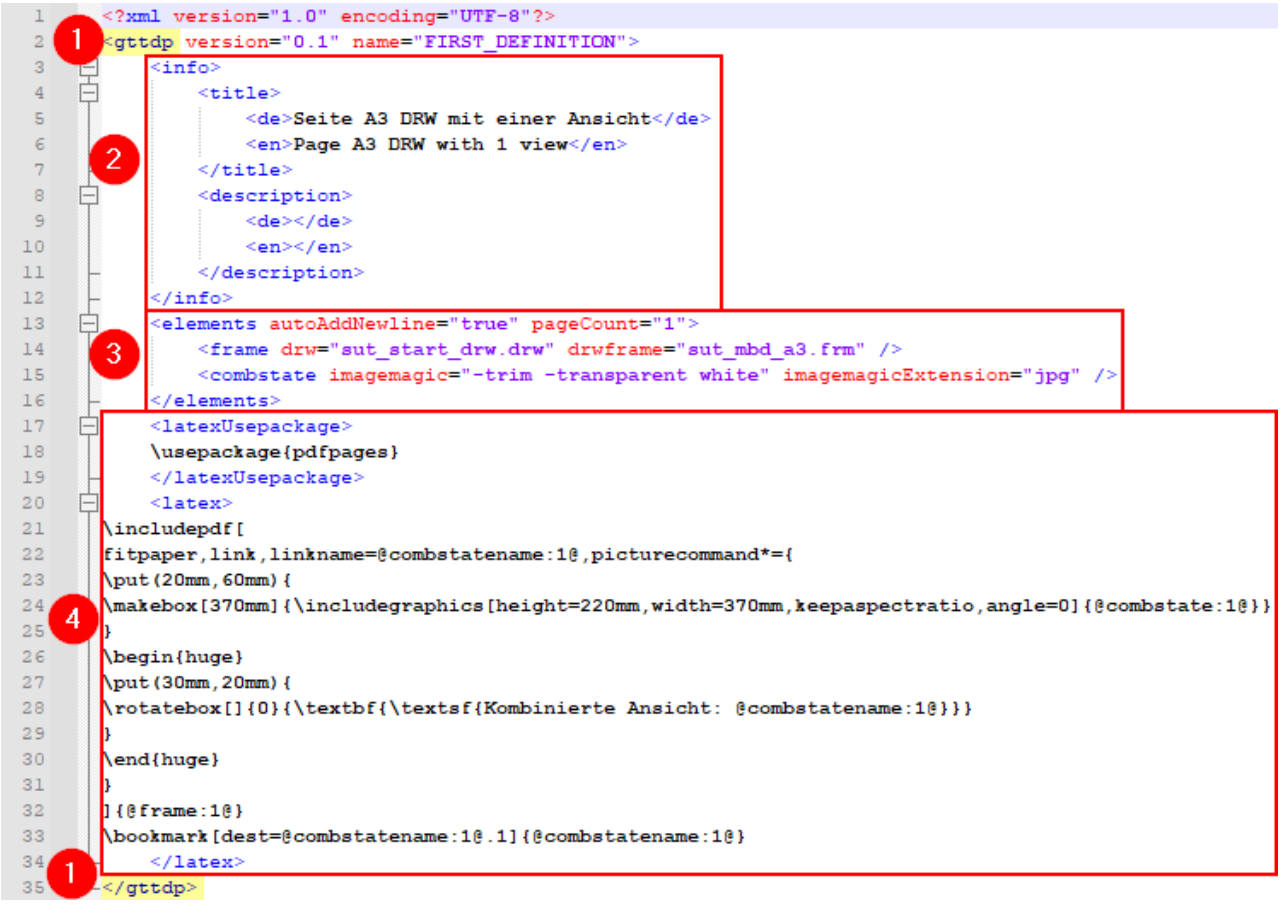
**Please note:** The command `@chapters@` is a placeholder that must be placed in the document section of the Latex definition. All Latex content for the subchapters[115] is inserted here.

## 7.2.1.2 Defining chapter template

Each chapter of the PDF document to be exported is defined in its own chapter template (XML file). These templates are located in the folder *chapters*. The corresponding file path is set in the configuration option `gttdp_chapter_folder`.
Default: *%gt_resource_folder%export_tdp\chapters*

The following figure shows an example of such a template: *page-01-drw-a3-1-view.xml*. This example explains the different sections of a template.

**Please note:** If you rename a *chapters* template, you must also change the name in all TDP template files[108] that contain the renamed *chapters* template.




```
<?xml version="1.0" encoding="UTF-8"?>
<gttdp version="0.1" name="FIRST_DEFINITION">
    <info>
        <title>
            <de>Seite A3 DRW mit einer Ansicht</de>
            <en>Page A3 DRW with 1 view</en>
        </title>
        <description>
            <de></de>
            <en></en>
        </description>
    </info>
    <elements autoAddNewline="true" pageCount="1">
        <frame drw="sut_start_drw.drw" drwframe="sut_mbd_a3.frm" />
        <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
    </elements>
    <latexUsepackage>
    \usepackage{pdfpages}
    </latexUsepackage>
    <latex>
\includepdf[
fitpaper,link,linkname=@combstatename:1@,picturecommand*={
\put(20mm,60mm){
\makebox[370mm]{\includegraphics[height=220mm,width=370mm,keepaspectratio,angle=0]{@combstate:1@}}
}
\begin{huge}
\put(30mm,20mm){
\rotatebox[]{0}{\textbf{\textsf{Kombinierte Ansicht: @combstatename:1@}}}
}
\end{huge}
}
]{@frame:1@}
\bookmark[dest=@combstatename:1@.1]{@combstatename:1@}
    </latex>
</gttdp>
```

1. `<gttdp>...</gttdp>`

Tags as frame for the template

- `version` Specification of a number for versioning the template

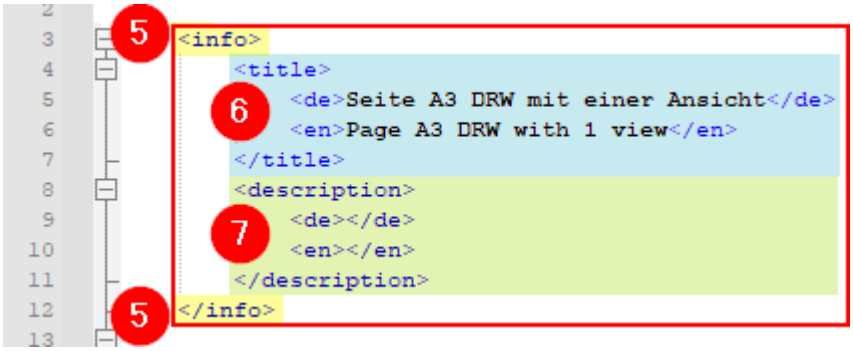
2. Information about the template[115]


© 2026 INNEO Solutions GmbH

3. Defining elements[115]
4. Latex Formatting Specifications[116]

# Information about the template

Each template requires the following tags:




5. `<info>... </info>`

Tags to frame the information section

6. `<title>... </title>`

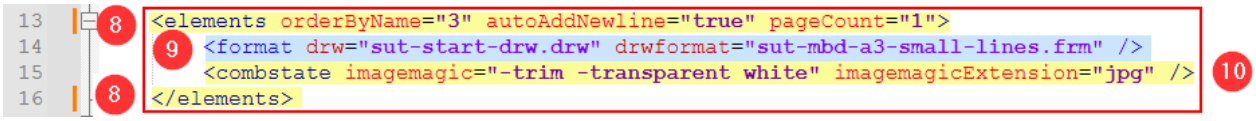
Display name of the template in *Export TDP* in English `<en>` and German `<de>`

7. `<description>... </description>`

Description text in English `<en>` and German `<de>` (optional)

# Defining elements

The chapters are inserted into the exported PDF document in the following form:




8. `<elements>... </elements>`

Definition of the elements to be included in this chapter:

– `autoAddNewline` Do you want to add a row between chapter components? yes/no (`true/false`)

– `pageCount` The page number for this element. By default, this attribute is set to 1 and hidden. If you want to set it to a different integer ≥ 0, write the attribute in the start tag, here on line 13.

**Tip:** `pageCount=0` is suitable for elements that embed data so that no additional, empty page is inserted for the embedded data, e.g. *embed-01-pvz.xml*. In this case, do not insert any `elements` attributes in the template.


© 2026 INNEO Solutions GmbH

– `orderByName` Attempts to collect the Combined Views logically on individual PDF pages. `orderByName=3`, for example, sorts the Combined Views based on the first three characters of the names. If it is the case that all of the affected Combined Views can fill a page, they are not split, i.e. `orderByName` is ignored.

Depending on which elements you want to export, specify one or more of the following attributes:

9. `<format... />`

Creates a tiff file of a drawing. The current 3D model is used as the drawing model. The following attributes must be defined for the drawing to be exported:

– `drw` Defines the drawing name so that a drawing can be associated with the open 3D model, here: supplied template *sut_start_drw.drw* from the template folder model. Alternatively, you can enter your own template.

– `drwformat` Drawing frame to be used from the template folder *components*, here: *sut_mbd_a3.frm*. If no drawing frame is specified (`drwformat`), the drawing will appear in the PDF without a drawing frame.

10. `<combstate... />`

Creates a screenshot of a selected combined view:

– `imagemagic` Attribute for editing the screenshot, here: `-trim -transparent white`. The result is that the screenshot is cropped to the visible elements and the white areas are displayed transparently.

– `imagemagicExtension` Drawings are exported to this file format, here: `jpg`

Another possible tag for `elements` definition (not shown in the screenshot):

– `<u3d .../>`

Creates a U3D model, see e.g. *title-03-a4-book-titlemodel.xml*:

– `imagemagic` Attribute for editing the screenshot.

– `imagemagicExtension` Drawings are exported to this file format.

**Please note:** Repeat steps 8 through 10 for each chapter you want to use. You can enter as many chapters as you like.

## Latex Formatting Specifications

The following commands are formatting specifications for the images to be exported.

**Please note:** This part applies to every template. To avoid errors, do not change these commands.


© 2026 INNEO Solutions GmbH




11. `<latexUsepackage>... </latexUsepackage>`

Formatting specifications

12. `<latex>... </latex>`

Command to execute the XML template with Latex. This template is applied to the model from which the module *Export TDP* is opened.

13. Formatting specifications

## 7.2.2 Available configuration options

The following configuration options are available for configuring export templates and storage locations:

**gttdp_can_change_workdir**
Defines whether the working directory may be changed to find required models (1) or not (0). Default setting: 1; if the option is set to 0, a search path must be defined that points to the directory. By default, the model directory *%gt_resource_folder% export_tdp\components*

**gttdp_chapter_folder**
Defines the folder for the chapter templates. Default: *%gt_resource%export_tdp\chapters*

**gttdp_component_folder**

Defines the folder for the drawing formats. Default: *%gt_resource_folder% export_tdp\components*

**Warning:** The configurations *gttdp_chapter_folder* and *gttdp_component_folder* have been renamed.

**gttdp_delete_template_folder**
Defines whether the temporary folder that is created for each export should be deleted after the export (1) or not (0). Default setting: 1


© 2026 INNEO Solutions GmbH

**gttdp_file_name**
Defines the default setting for the name of the PDF file to be exported. Default: @mdln@

**gttdp_imagemagick**
Defines the installation site of an existing ImageMagick installation. Default @unzip
@unzip - Unzips the included ImageMagick installation into the directory *gttdp_tools_unzip_folder*.

**gttdp_lang**
Defines the display language of *Export TDP*. Default: de

**gttdp_miktex**
Defines the installation location of an existing MikTex installation. Default: @unzip
@unzip - Unpacks the included MikTex installation into the directory *gttdp_tools_unzip_folder*.

**gttdp_open_after_export**
Defines whether the checkbox for opening the PDF after export should be activated (1) or not (0). Default setting: 0

**gttdp_show_cmd_commands**
Defines whether executed commands are displayed for reporting purposes (1) or not (0). Default: 1

**gttdp_template_folder**
Defines the folder for the templates. Default: *%gt_resource_folder%export_tdp\templates*

**gttdp_tools_unzip_folder**
Defines the folder in which tools are to be unzipped with @unzip. Default: *%appdata%\INNEO\GENIUS_TOOLS\for_Creo*

## 7.2.3 Latex customization with Export TDP

The TAG `<latex>` contains the Latex syntax, which generates a Latex document at the end from which the PDF document is then generated. The *Export TDP* module supports additional placeholders to ensure the following functionalities.

| Within the TDP template | Enter value |
|---|---|
| @usepackages@ | Collects the latex syntax of all chapters included in the TDP template, which is contained in the XML node latexUsepackage. |


© 2026 INNEO Solutions GmbH

| | |
|---|---|
| @chapters@ | Collects the latex syntax of all chapters included in the TDP template, which is contained in the XML node latex. |

| Within the chapter template | Enter value |
|---|---|
| X | index (see the example below) |
| #IF / #ENDIF | If the requirement is fulfilled, the Latex syntax between IF and ENDIF is taken into account and executed. |
| @combstatename:1@ | Names of the combined view (string) |
| @combstate:1@ | Path of the exported combined view (JPEG) |
| @combstateannotations:X: : @ | String of all information from annotations Info<br>Example: `$$repllatex$@combstateannotations:3: : @$$` |
| @format:1@ | The exported drawing frame (PDF) |
| @u3dimage:1@ | Preview image of the U3D (JPEG) |
| @u3dname:1@ | Name of the combined state (string) |
| @u3d:1@ | Path of the U3D file (U3D) |

| Within both chapter templates | Enter value |
|---|---|
| #FOR / #ENDFOR | Executes the Latex syntax repeatedly as long as a defined number has been run through. |
| #FOR PARAM | Loop runs through all model parameters |
| #FOR DPARAM | Loop runs through all designated model parameters |


© 2026 INNEO Solutions GmbH

Example for #IF / #ENDIF and X

`#IF @combstate:1@`: If a first combstate is defined, the Latex syntax is taken into account within the If statement (yellow marking).

`#IF @combstate:2@`: If a second combstate is defined, the Latex syntax is taken into account within the If statement (blue marking).

```
    <elements autoAddNewline="true">
        <format drw="sut-start-drw.drw" drwformat="sut-mbd-a3-small-lines.frm" />
        <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
        <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
        <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
        <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
    </elements>
    <latexUsepackage>
    \usepackage{pdfpages}
    </latexUsepackage>
    <latex>
\includepdf[
fitpaper,link,linkname=@combstatename:1@,picturecommand*={
#IF @combstate:1@
\put(20mm,174mm){
\makebox[185mm]{\includegraphics[height=110mm,width=185mm,keepaspectratio,angle=0]{@combstate:1@}}
}
\begin{LARGE}
\put(20mm,165mm){
\rotatebox[]{0}{\textbf{\textsf{$$repllatex$@combstatename:1@$$}}}
}
\end{LARGE}
\bookmark[dest=@combstatename:1@.1]{$$repllatex$@combstatename:1@$$}
#ENDIF @combstate:1@
#IF @combstate:2@
\put(210mm,174mm){
\makebox[190mm]{\includegraphics[height=110mm,width=190mm,keepaspectratio,angle=0]{@combstate:2@}}
}
\begin{LARGE}
\put(210mm,165mm){
\rotatebox[]{0}{\textbf{\textsf{$$repllatex$@combstatename:2@$$}}}
}
\end{LARGE}
\bookmark[dest=@combstatename:1@.1]{$$repllatex$@combstatename:2@$$}
#ENDIF @combstate:2@
```

Example for #FOR PARAM and #FOR DPARAM

The code is executed for all assigned parameters:

```
Hello,


here comes a table
\begin{tabular}{|c|c|}
  \hline
  Parameter & Value \\
  \hline
#FOR DPARAM DESCR.*
  $$repllatex$@para@$$ & $$repllatex$%@para@%$$ \\
#ENDFOR DPARAM
  \hline
\end{tabular}
```


© 2026 INNEO Solutions GmbH

This is the result:

Hello,

here comes a table

| Parameter | Value |
|---|---|
| DESCRIPTION_1_DE | Buchse |
| DESCRIPTION_2_DE | WN 96 |
| DESCRIPTION_1_EN | |
| DESCRIPTION_2_EN | |

## 7.2.4 Use cases

### 7.2.4.1 Create TDP template

With this use case, you can create a TDP template with a title page and add a company logo.

**Procedure**

Create and rename copies

1. In the folder *templates*, create a copy of one of the TDP templates and rename it, for *example 03-drw-a3-titlemodel-4view.xml*.
2. In the folder *chapters*, create a copy of one of the chapter templates and rename it, for example *title-01-a4-drw-titleimage.xml*.
3. In the folder *components*, create a copy of one of the drawing formats and rename it, for example *title-01-a4p-logo.frm*.

Customize TDP template

4. Change the title in the TDP template in German `<de>` and English `<en>`. Enter the file name of the chapter template after the attribute `definition`. Check in Creo whether the renamed title is displayed.

```
4       <title>
5           <de>Firmenname: A3 mit Titelmodel / 4 Ansichten pro Blatt</de>
6           <en>Company name: A3 with title sheet / 4 views per sheet</en>
7       </title>
8       <description>
9           <de></de>
10          <en></en>
11      </description>
12      <icon></icon>
13      <function></function>
14      <help></help>
15  </info>
16
17  <elements autoAddNewline="true" runLatexTwice="false">
18      <chapter definition="title-Firmenname-a4-drw-titleimage.xml" [...]
19      <chapter definition="page-02-drw-a3-4-view.xml" combstatefilter="^B(
```

Customizing the chapter template


© 2026 INNEO Solutions GmbH

5. Change the title in the chapter template in german `<de>` and english `<en>`. Enter the file name of the drawing format after the attribute `drwformat`. Check in Creo whether the renamed title is displayed.

```
4        <title>
5            <de>Titel Firmenname A4 mit Titelbild</de>
6            <en>Title company name A4 with cover image</en>
7        </title>
8        <description>
9            <de></de>
10           <en></en>
11       </description>
12       <icon></icon>
13       <function></function>
14       <help></help>
15   </info>
16   <elements autoAddNewline="true">
17       <format drw="sut-start-drw" drwformat="Firmenname-title-01-a4p-logo" />
18       <combstate imagemagic="-trim -transparent white" imagemagicExtension="jpg" />
19   </elements>
```

Customize drawing format

6. Open the Export TDP in Creo, select the TDP template under *Template* and click on *Export*.
7. Open the FRM file in Creo.
8. Click on *Layout* in the ribbon and then on *Object*.
9. Under *Object type*, select *Paintbrush Picture* from the drop-down menu and confirm with *OK*.
10. Insert the graphic and close Paintbrush Picture.
11. Left-click on the empty drawing frame.
12. Remove the INNEO logo and move the new graphic to the appropriate position.

**Tip:** Optionally, the drawing header table can be replaced.

13. Save


© 2026 INNEO Solutions GmbH

# 8 Forms

Available in the ribbon menus GENIUS TOOLS | GT MBD

With *Forms* you create forms and apply them. These are user-defined form masks with which you can assign different properties to models. Fill these form masks with various information. You can select and edit dimensions and parameters of parts, features and assemblies, suppress features or components and replace components with others. Depending on which properties are relevant for a model, select the corresponding information in the form. Use this module to call different variants of a model or to configure frequently used start objects.

You can save related properties as a template in a value table so that you can quickly switch between different configurations. Self-defined mapkeys can also be stored in form elements, which are then executed by clicking on the associated buttons in *Forms*.

*Forms* is available in assembly, parts, sheet metal, skeleton, layout and drawing mode with the following features:

1. Systematically observe and manipulate model properties in form dialogs:
   - Dimension values
   - Dimension tolerance
   - Model-parameter values
   - Feature-parameter values
   - Suppress features
   - Material
   - Replace components
   - Variant dimensions tables
2. Executing saved mapkeys
3. Usage of form- / auxiliary parameters
4. The forms are generated via the graphical editor.
5. The forms are saved directly in the models.
6. Rules between properties can be defined with JavaScript.
7. Loading of external data into the mask is possible (e.g. EXCEL, CSV).


© 2026 INNEO Solutions GmbH

# 8.1 Fundamentals

With *Forms* you can use forms to control properties and metadata of Creo parts and assemblies.

*Forms* is also integrated into *Library*. If a library model contains a form, it will be displayed directly in the Library details window and can be used for editing the model.

## Glossary

**Form**
A form is a collection of model properties. A form is either stored directly in the model (default) or as an XML definition outside a model.

**Form definition**
A form definition contains the configuration of a form. This form definition is stored either externally (as an XML file) or internally (at the Creo model).

**Form element**
A form element describes a model property that is stored in a form with different values.

**Form element list**
A form element list is the overview for managing the form elements in *Forms Editor*.

**Webcode**
The webcode is a linking property between model and form. The webcode is required by *Library*[293] to establish a connection between an external form and a model through a library object.

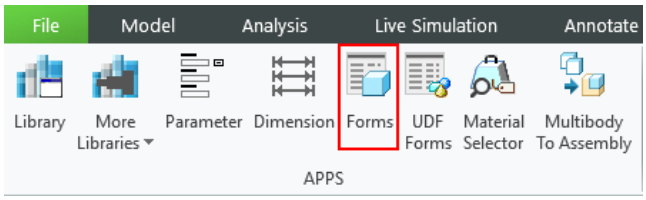
# 8.2 Usage

This section contains information on using *Forms*. It describes the general structure of the program.

## Starting the program

To start *Forms*  , click *GENIUS TOOLS* or *GT MBD* in the ribbon menu and go to *APPS* section or by using *Quick Access* (key [<]).


© 2026 INNEO Solutions GmbH




*Starting via the GT MBD in the ribbon menu*



*Starting via Quick Access*

## Forms and UDF Forms in the model

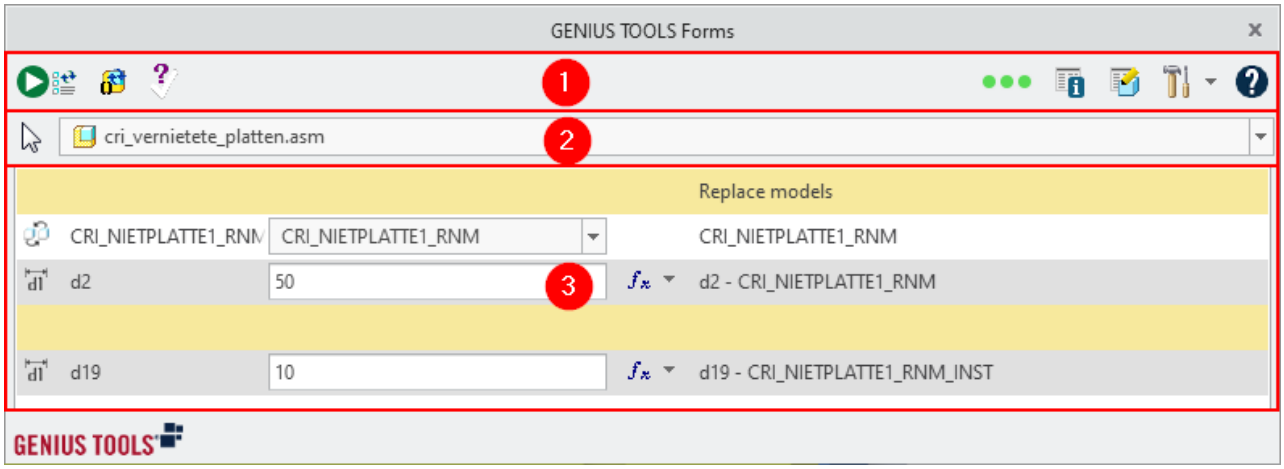
If a model contains a form or UDF form, the corresponding forms icon is displayed in the Creo Parametric main window. Click on the icon to open the form.

 Form in the model

 UDF Form in the model

## 8.2.1 User interface

The user interface consists of the following elements:

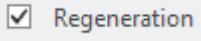



1. Command bar[136] with tools menu
2. Model selection[129]
3. Form section[129]


© 2026 INNEO Solutions GmbH

## 8.2.2 Command bar

The Command bar displays general controls. The following controls are available:

| Icon | Name | Description |
|---|---|---|
|  | Apply changes and regenerate | Applies changes in the form to the model and regenerates it. The quantity of regenerations can be defined with the configuration option `gt_regen_times`. This can be useful for complex models.<br>If the form includes a JavaScript *PreSave* function, this function is run before the values are applied. If there is an error, you will be asked whether you want to save anyway. Click *No* to open the status dialog with the error message. |
|  | Open Value Transfer[594] | Opens the module *Value Transfer*. The values are taken from the form.<br>**Note:** The function is only available in assembly mode. |
|  | Execute Check function | Starts the JavaScript function `CheckUI`.<br>**Note:** The function must be saved in the Form. |
|  | Regeneration | If the checkmark is set, the model is regenerated after the values are transferred to the model. By default, the checkbox is set, but the function is not displayed. The configuration option `gtf_show_regen_btn` controls whether the checkbox is visible. Set the checkbox to visible and clear the checkbox if you want the function  to *Apply changes* but not *to regenerate*, in order to avoid long regeneration times for large components and to be able to continue working directly without waiting. In this case, note that values and image do not match. |

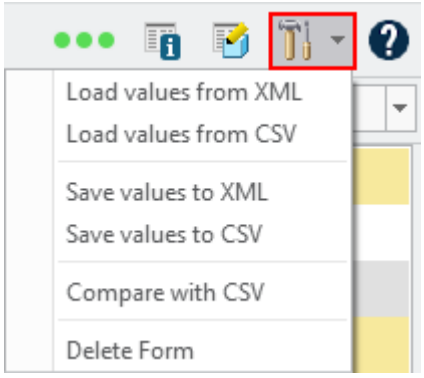

© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
| | Status indicator[476] | Shows the current status for *loading*, *working* and *saving* with traffic light colors and opens the status dialog. |
| | Forms information | The button *Forms information* displays an info area with text and a preview image. Click on the preview image to enlarge it.<br>Text and image are stored in the *Forms Editor*[135]. |
| | Open Forms Editor[135] | Opens the *Forms Editor* to edit a form. |
| | Tools[127] | The Tools menu contains various supporting functions. |
| | Help | Opens the help. |

**Warning**: To apply changes, use the button *Apply changes and regenerate* ! Modifications in a form will not be adopted by models on closing.

## Tools menu

The following functions are called up via the Tools menu.




| Function | Description |
|---|---|
| **Load values from CSV/XML file** | Values from a CSV or XML file are read in and applied to the form.<br>When reading from an XML file, the values are assigned to the form elements using IDs. If no IDs are available, the assignment is made using the names of the form elements. |


© 2026 INNEO Solutions GmbH

| Function | Description |
|---|---|
| | When reading from a CSV file, the values are always assigned using the name of the form element.<br>Values of dimensions or parameters without correspondence in the current form are ignored when loading.<br>Line names of value tables may be included. These must be entered manually, since the tag is not created when saving. The ID attribute will be omitted.<br>No JavaScript OnChange functions are executed when reading from a CSV or XML file.<br>`Example: <element name="ValueTable"  value="Square" />` |
| **Save values to CSV/XML file** | Saves the current values in the form to a CSV or XML file.<br>If you want to save a CSV file, make sure that all form elements have unique names. If multiple form elements have the same name, only the first of these elements is written to the CSV file.<br>Units of real parameters are written to the file as well.<br>Value table settings are not saved. |
| **Compare with CSV file** | Compares the values of the chosen form with the values from a CSV file.<br>Select the desired CSV file. The values from the current form and from the CSV file are displayed in a Compare dialog box. Changes are highlighted in color. |
| **Delete Form** | Irrevocably deletes the form from the model. The form remains intact, so you can create further models, see Creating a library object with a form 370. |

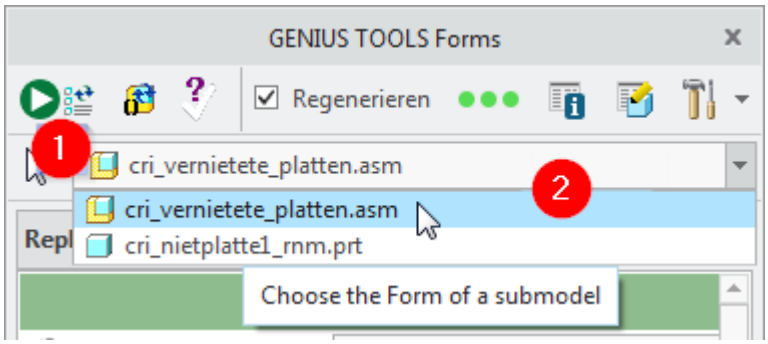
The default folder for CSV and XML imports and exports is defined in the configuration option `gtf_default_folder`.

**Please note**: Make sure that all form elements have unique names.
With the configuration option `gtf_import_xml_use_name_if_no_id_defined =0` all entries without ID will be ignored.


© 2026 INNEO Solutions GmbH

## 8.2.3 Model selection in assembly mode

Use the model selection to switch between forms of different assemblies and parts in the current assembly. The model selection is only displayed in assembly mode.




*The model selection with object selection (1) and drop-down list (2).*

Select the models in an assembly using the object selection (1) directly in Creo or in the model tree. The drop-down list (2) shows the current selection. Open the list and select from the models. Models that have already been selected by object selection are additionally displayed in the list.

## 8.2.4 Form display

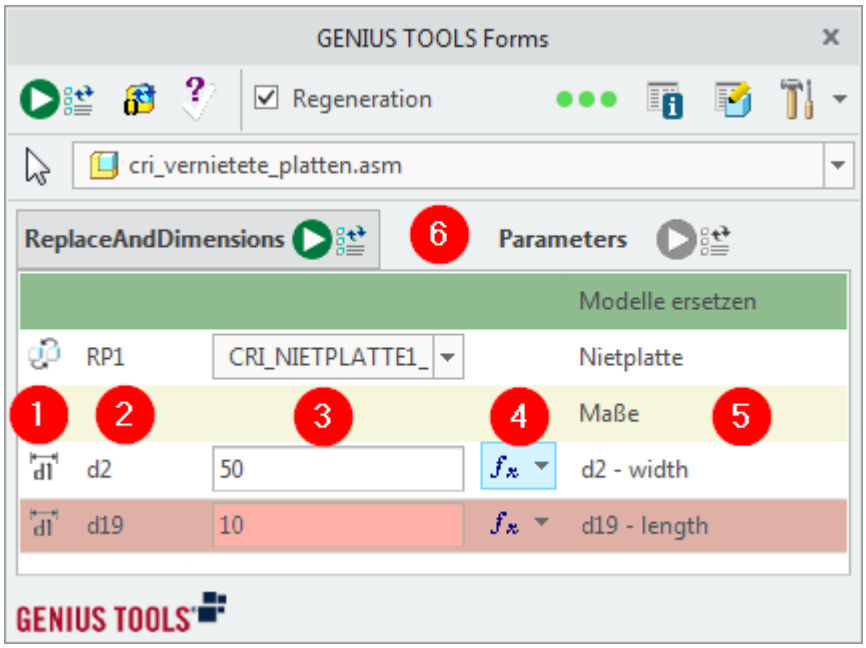
The form area displays the form of the selected model. In each line a property is listed, whose value can be changed in the third column **Values with input fields**. Click on *Apply changes and regenerate* to apply the changed values to the model.

Different values are stored in the value tables for several properties, so that when a value table is selected, the values of several rows are set simultaneously. This reduces the need to set individual properties per line and offers the possibility to set the model to a predefined state.

The content of the form is customized in *Forms Editor*[137]. Visibility and order of the individual form columns are adjusted in *Forms Configuration*[152].

**Tip**: After opening *Forms*, the selection that corresponds to the current model is displayed. If none of the selection options meet the condition, the element icon turns yellow. The first stored option is displayed.


© 2026 INNEO Solutions GmbH



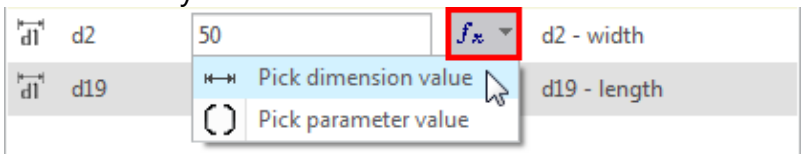

*Form area with highlighting of a form element (red marked line) and division into five columns*

1. **Type**
   The type of a form element is displayed as an icon in the first column. Click an icon to highlight the associated model element in Creo.
   Color highlighting indicates whether a form element was changed:

| Icons of type column | Meaning |
|---|---|
| | Change to element value |
| | The current model value is not intended as an element value. Outside the *Forms* user interface, a value was set that is not stored in *Forms*. |
| | The element addresses a model property that does not exist, e.g. if a parameter was deleted. |


© 2026 INNEO Solutions GmbH

2. **Name / ID**
3. **Values with input fields**
   Depending on how the form is defined, there are differently limited choices for setting properties[131].
4. **Function**
   Behind input fields that provide for direct input, there is an additional button for the following functions:
   – *Pick dimension value:*
   Allows the transfer of a value from any dimension into the input field.
   – *Pick parameter value:*
   Allows a value from a parameter with the same name to be transferred to the field. If there is no parameter with the same name, a dimension with the same name is searched for and its value is taken over. The parameter value transfer is only available in assembly mode.

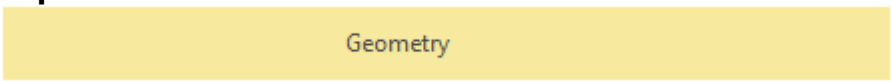



*Transfer values from dimensions and parameters into a Form*

5. **Description**
   The fifth column displays a language and configuration dependent description of the form elements.
6. **Tabs (groups in the editor)**
   Logical sectioning of larger forms. Click on a tab to open that section of the form.

## Structuring a form

**Separator**




Separators provide a logical structure for form elements.

## 8.2.5 Using a form

Changes to the values are applied to the model and visible after clicking the button *Apply changes and regenerate* . Below you will find an overview of the input types used to set properties. Depending on the settings in the *Forms Editor*[135], you have different and possibly limited options for setting a value in the individual lines. You can change the input types and values that can be set in the *Forms Editor*.


© 2026 INNEO Solutions GmbH

**Warning**: Check the Creo configuration option `show_dim_sign`, as *Forms* respects it.
`show_dim_sign=no`: Entering a negative value changes the direction vector of a dimension. The value becomes positive.
`show_dim_sign=yes`: If a negative value is entered, the value remains negative.

# Filling input and selection fields

### Input field




Input fields accept any string for input.

### Binary checkbox



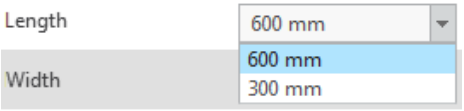

Checkboxes allow you to choose between two options. This can be Yes/No decisions, for example. Always two values are deposited in the configuration for checkboxes. With the checkmark set, the first value is used. Checkmark not set applies the second value.

### Radio button



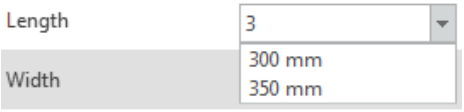

Radio buttons display predefined selectable values. Only one of the values can be selected.

### Dropdownbox




Dropdownboxes contain a selection of possible inputs.

### Extended dropdownbox



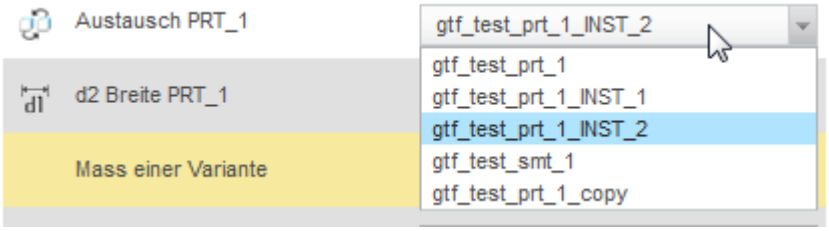

Extended dropdownboxes accept free input of values as well as selection from a list. Each input filters the displayed list items.


© 2026 INNEO Solutions GmbH

# Modifying components

### Replacing components

The function *Replace components* exchanges predefined components for one another using selection fields.




*Replacements are selected via lists*

Select a model from the drop-down list to replace a component. Click the icon *Replace component* in the first column to highlight the component to be replaced in Creo.

**Please note**: Replacements fail when assemblies have missing components.

**Warning**: Using the *GENIUS TOOLS for Creo* together with the *Startup TOOLS*: When replacing with an interchange assembly (instead of using a family table), a new feature ID is assigned to the part. Check the Creo configuration option `remember_replaced_components`. This option is set in Creo Parametric, but is disabled by *Startup TOOLS*.

If new feature IDs are created by an interchange assembly, the new IDs are automatically written to the model.

### Suppress components / features

The function *Feature suppression* works using checkboxes. If a checkbox is active (checkmark is set), the component / feature is displayed. With a deactivated checkbox, the component / feature will be suppressed after regenerating.

**Please note**: On resuming, dependent suppressed elements are not resumed.

Click on the *Suppress components / features* icon to highlight the associated component / feature in Creo.

# Automatic setting of properties

### Run Mapkey / JavaScript



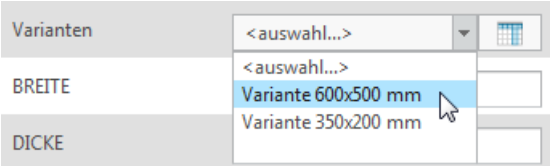

*Mapkey with configuration-dependent "Start" label*

Mapkeys / JavaScript elements stored in the form are started by clicking the corresponding button.

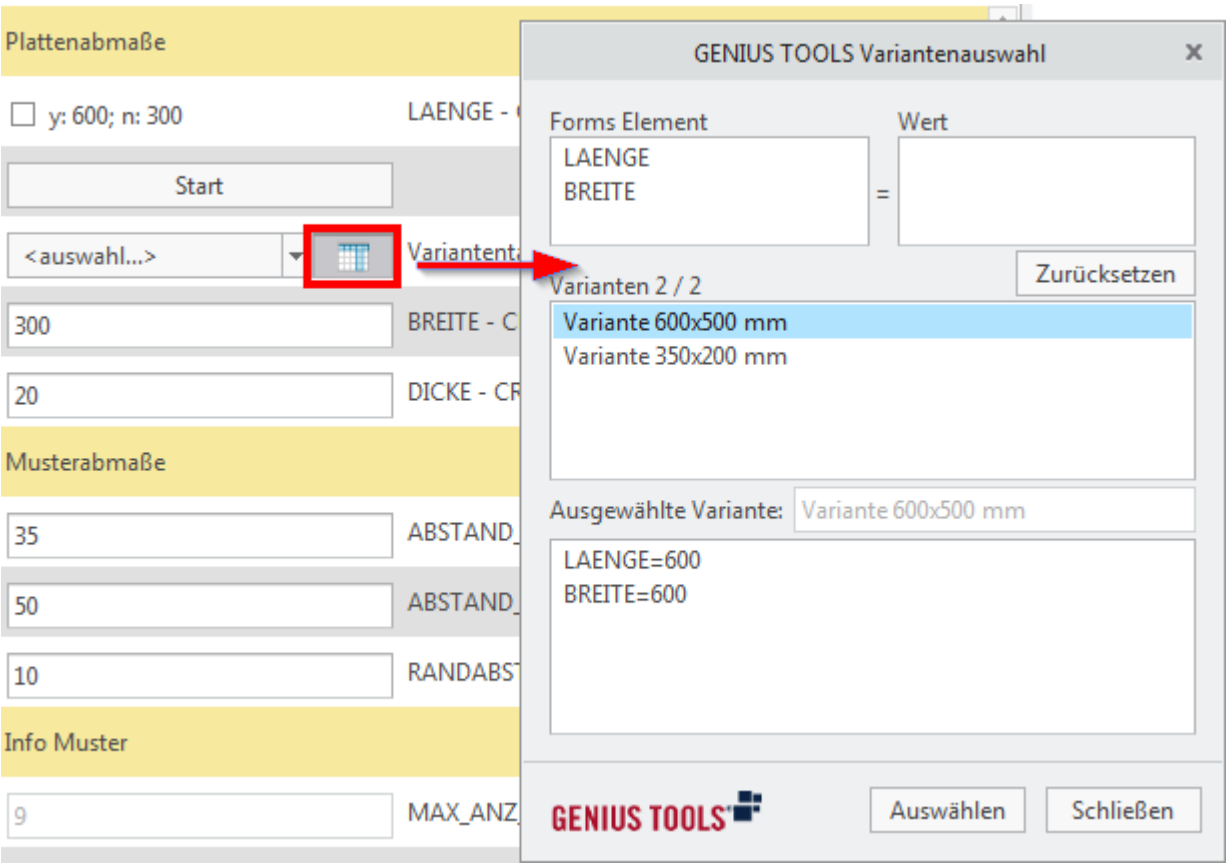

© 2026 INNEO Solutions GmbH

## Setting multiple properties

### Value table




Variant tables display preconfigured variants of a model. In most cases, each variant affects multiple properties of a model at the same time. Select the variants from the list or alternatively use the button *Variant selection* to filter variants.




*Variant selection for filters by individual properties*

**Please note**: If the value table symbol turns yellow or red, please check whether the externally read value tables are correct.

**Warning**: Value tables cannot affect each other!

# 8.3 Configuration

In this section you will find more information about the *Forms Editor*, which allows you to create and manage forms. Suggestions for creating and configuring forms can be found in


© 2026 INNEO Solutions GmbH

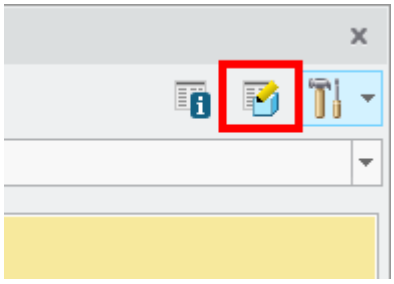
Use cases[156].

## 8.3.1 Forms Editor

You create forms with the *Forms Editor*.

### Starting the program

Start the Editor via the command bar in the *Forms* user interface.



**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].

### Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

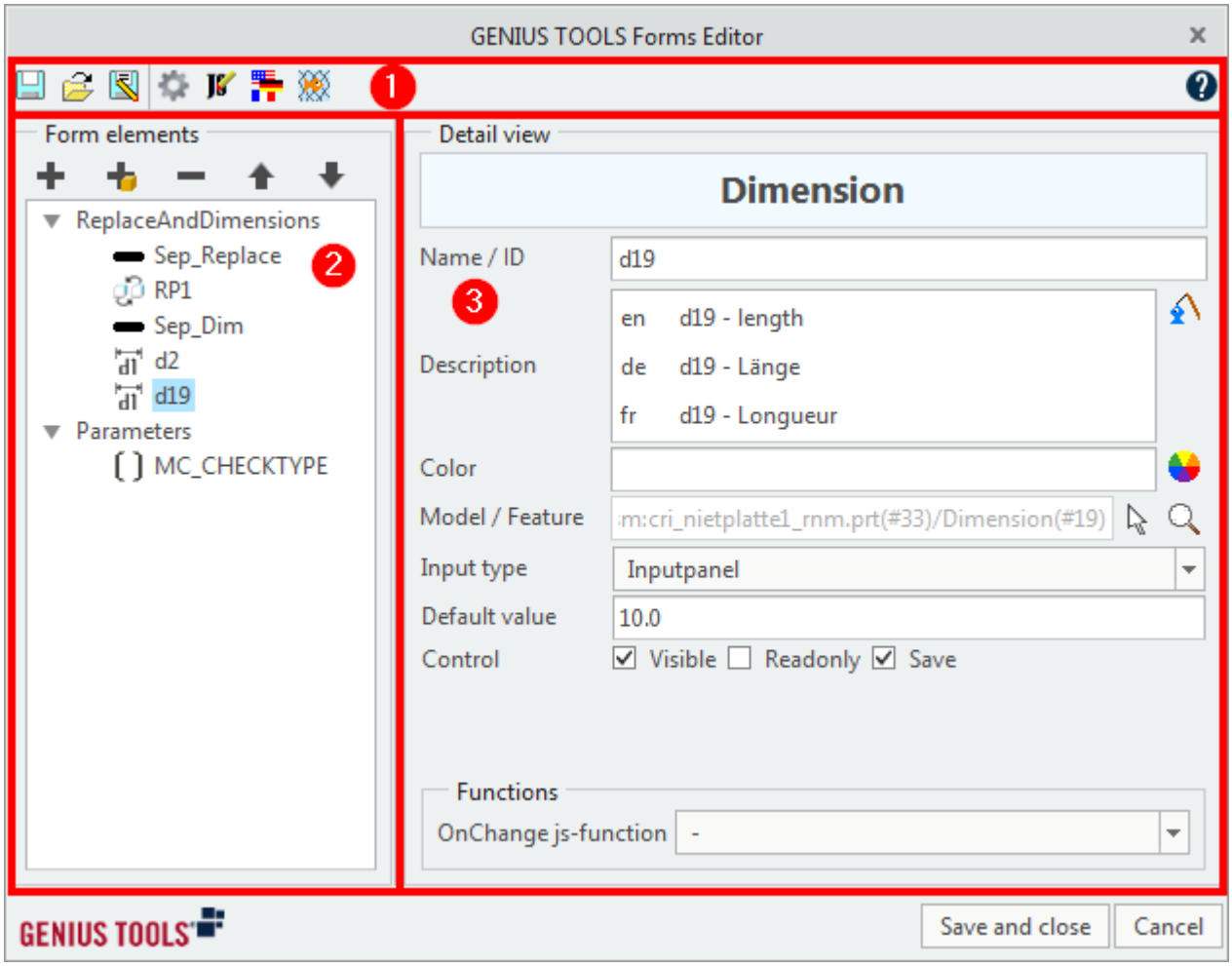
To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.


© 2026 INNEO Solutions GmbH

## 8.3.1.1 User interface

The user interface of *GENIUS TOOLS Forms Editor* consists of the following elements:




1. Command bar[136]
2. Form elements list[137]
3. Detail view[141]

## 8.3.1.2 Command bar

The following buttons are contained in the command bar:

| Icon | Name | Description |
|---|---|---|
| | Save | Saves the current form and closes *GENIUS TOOLS Forms Editor*. |
| | Open | Opens a forms definition (XML file). |

| Icon | Name | Description |
|---|---|---|
|  | Save as | Saves the current form as a forms definition (XML file). |
|  | Edit Forms configuration[152] | Opens the dialog for configuring the *Forms* user interface. |
|  | Edit JavaScript[784] | Opens the *JavaScript Editor*. |
|  | Manage languages | Opens the dialog for editing the included languages. |
|  | Automatic default texts | Opens the dialog for automatically setting standard texts for all used form elements. The names of the form elements are used as keys for the translation, see Set standard texts for multiple elements[678]. |
|  | Help | Opens the Help. |

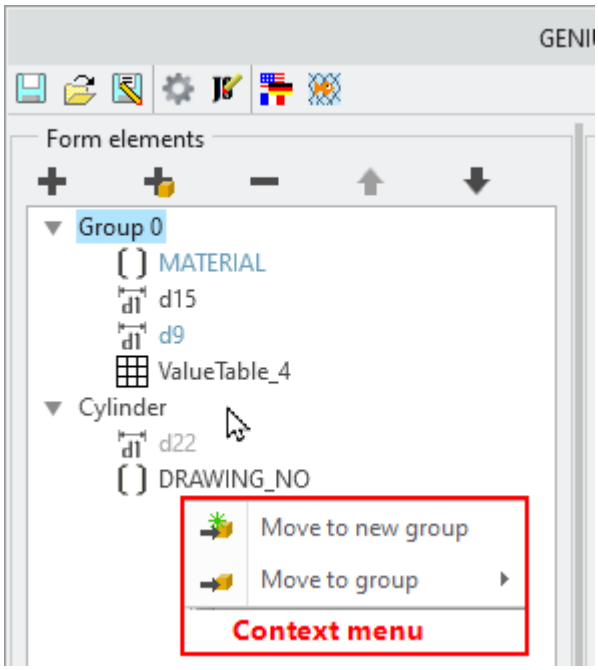
## 8.3.1.3 Managing form elements

The list control is used to manage form elements. The following buttons are contained in the list control:

| Icon | Name | Description |
|---|---|---|
|  | Add new item to list[138] | Opens selection dialog for creating a new form element. |
|  | Add new group | Creates a new group, displayed as tab in *Forms*. The groups serve as separators in the table. |
|  | Delete selected item from list | Deletes the selected form element or group.<br>**Warning:** Groups are deleted along with their contents. Move any form elements you still need before deleting a group. |
|  | Move selected item | Moves the selected form element or group up or down in the list. |

The form element list is displayed under the list control. All created form elements are displayed here in the order in which they are displayed in the *Forms* panel. Read-only elements are displayed in blue font. Non-visible, read-only elements are displayed in gray font. A value with a blue background is currently selected and opened in *Detail view*.


© 2026 INNEO Solutions GmbH



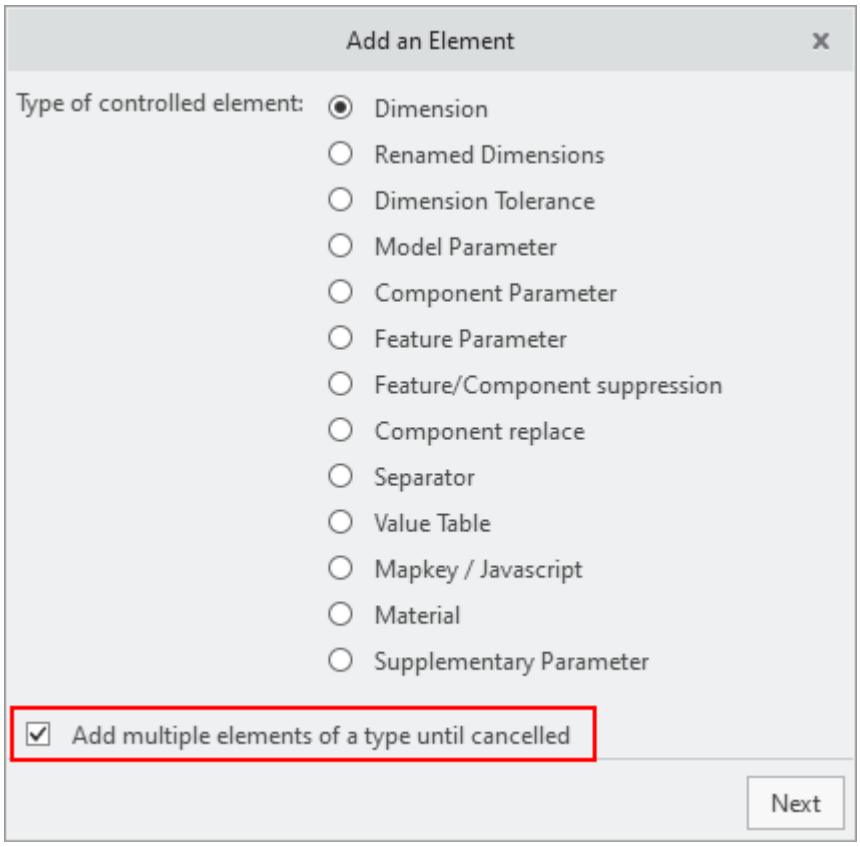

To move a group or a form element to another group, right-click to open the context menu and select *Move to new group* or *Move to group* (with selection of an existing group).

You can also select multiple elements and move them using the *Move* buttons or drag-and-drop. Right-click on an item to move the items within the created groups. Select an item to view its details and make changes.

## Adding form elements

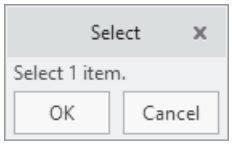
Use the button *Add new item to list* ✚ in the list control to add new form elements. Click *Next* to open the selection dialog.


© 2026 INNEO Solutions GmbH




*This control button allows you to select multiple elements, here using the example of the type "Dimension"*

You can select multiple elements by ticking the checkbox *Add multiple elements of a type until cancelled*. The function is available for selection until you click *Cancel*.




*Creo button for item selection*

After selecting one or more elements in the model, fill in the corresponding *Detail View*[141] dialog box.

# Types of form elements

| Form element | Description | Select element |
|---|---|---|
| Dimension | The input type[143] can be an input field, a radio button, a dropdown box, or an extended dropdown box. | X |


© 2026 INNEO Solutions GmbH

| Form element | Description | Select element |
|---|---|---|
| Renamed dimensions | All dimensions whose names have been edited are automatically transferred as dimension form elements from the selected model or feature to the current form definition. | X |
| Dimension tolerance | The input type[143] can be an input field, a radio button, a dropdown box, or an extended dropdown box. | X |
| Model parameter | Different input types[143] are possible. | X |
| Component parameter | Different input types[143] are possible. | X |
| Feature parameter | Different input types[143] are possible. | X |
| Suppress feature/component suppression[146] | Separate configuration options. | X |
| Replace component[146] | Separate configuration options. | X |
| Separator[147] | A separator used to structure a form. | |
| Value table[149] | Sets values for multiple existing form elements. | |
| Mapkey / Javascript[148] | Mapkey and JavaScript elements are displayed as buttons. | |
| Material[148] | Separate configuration options. | X |
| Supplementary Parameter | Do not exist in the model and cannot be saved. Used for JavaScript functions only. Accept any of the four available data types and can be configured with different input options depending on their data type, see input types[145]. | |


© 2026 INNEO Solutions GmbH

**Please note:** If you have added a read-only dimension, the *Read-only* property is automatically checked and the *Save* property is unchecked in the form element created with that dimension.

## Configuration of the form element list

The `gtf_editor_multiselect` configuration option allows you to set the checkbox *Add multiple elements of a type until cancelled* to be checked when starting *Forms* (default: 0 = off).

## 8.3.1.4 Describing a form

The *Detail View* is used to configure individual form elements. Click an element in the element list to display its *Detail view*.

## General settings

All form elements have the following entries.




*Detail view with general options of an element of the type "dimension"*

**Name / ID**
The displayed name of a form element (second column in the form).

**Title**
For groups only: The title is used as the tab caption for the group and can be maintained language-dependent. If there is no title, the name of the group appears as the tab caption.

**Description**
The localized description of a form element (fourth column in the form section). The displayed description depends on the configuration options `gtf_lang` and `gtf_def_lang`.


© 2026 INNEO Solutions GmbH

Default texts can be added using the button (Description of the standard text selection dialog[678]).

Use the languages dialog (flag icon in the command bar) to enter descriptions in multiple languages.

**Please note**: Description texts are limited to 80 characters.

**Color**
The color of the row of the form element. Enter a color directly from the RGB color space (comma separated) or click on the color circle to select a color (Description of the color dialog[796]). The brightness of the background color determines whether the foreground color of the element is white or black.

## Display settings

The following specifications depend on the form element.

**Visible**
Defines whether an element is displayed in the form area (e. g. for elements influenced by value tables).

**Input type**
- Input panel: input fields accept different inputs depending on the form element.
- Radio button, dropdown box, extended dropdown box: the section *Input values* opens.

**Default value**
Specifies the preset default value of a form element.

**Please note**: The default value is displayed only if it is a valid input. If not, the current value from the model is displayed.

**Read only**
Defines whether the field is read-only.

**Save**
Defines whether the element value can be stored in the model.

**Tip**: Use the *Read-only* option for input fields in combination with value tables[149] or JavaScript[154]. In this way, changes are visible without the individual values being edited by the user.

## Input values

Enter values for selection fields[143].


© 2026 INNEO Solutions GmbH

## Functions

You can assign an *OnChange* function to the element, see JavaScript in Forms.[154]

The *OnChange* function must be defined in the respective configuration via the *JavaScript editor*[784]. OnChange is executed when the value of a form element is changed automatically (e. g. by value tables) or manually by exiting a field or by pressing Enter.

**Please note**: Avoid creating infinite loops with your functions.

### 8.3.1.5 Creating form elements

This section contains information about defining different input types. To set several multiple properties at once, first define all the form elements you need, and then maintain them in value tables[149].

## Input field

If you select *Input field* as the input type, you can enter any string into the form. You can use a JavaScript function[783] to limit the values that can be entered, for example, to a range of values.

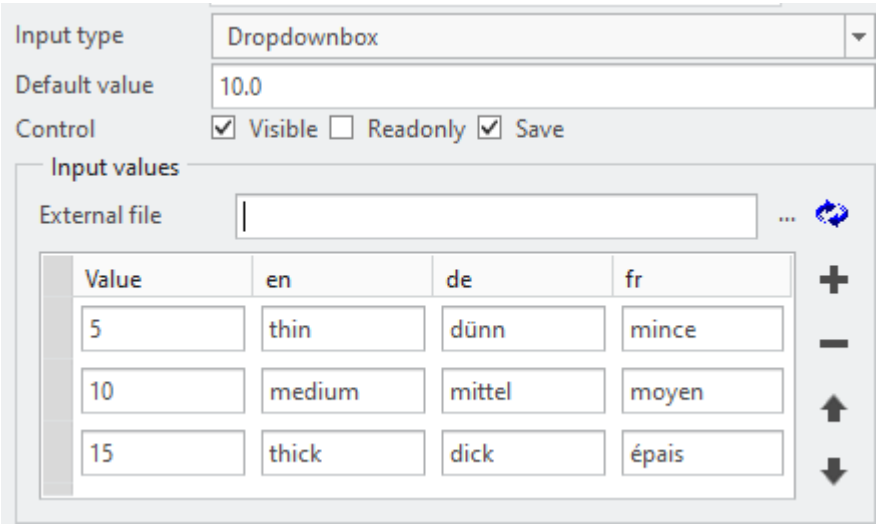
## Creating (extended) selection fields and radio buttons

In *Forms*, different types of input can be used. Selection fields (dropdown), extended selection fields (writable) and option fields are controlled by a table consisting of values and labels.

These value tables provide combinations of existing form elements (dimensions, parameters, substitutions and suppressions).

**Tip:** If you do not want a form element to be set in a particular variant, enter * in the cell for that element. In this case, the form element retains its previous value.


© 2026 INNEO Solutions GmbH




*Defined selection fields*

### Creating a table

Each table row describes one variant.

– Use the plus button ✚ to add a row. Select a row and click the minus button ━ to remove a row.

– Use the arrow buttons ⬆⬇ to move rows up and down.

– Enter values and the associated labels in the table. The label is displayed in the form area instead of the value in the field and is language-dependent.

### Using values from external file

You can also store the values for the selection in an external CSV file.

– The values from the CSV file are saved in the form.

– When the form is called, it checks to see if the linked CSV file exists and applies the changes.

– Select the required file under *External file* with the button .... By default, the file explorer opens in the folder *%GT_RESOURCE_FOLDER%forms*. You can define a different folder with the option `gtf_external_data_folder`.

– Click *Update* to re-read the displayed values from the CSV file.

### Linking an external file

There are three ways to specify an external CSV file:

1. Filename with extension, no path: the file must be located in the folder defined by the `gtf_external_data_folder` configuration option. The default is *%GT_RESOURCE_FOLDER%forms*.
2. A relative specification to the folder defined in the `gtf_external_data_folder` configuration option. Use this if you want the CSV file to be in a different folder.


© 2026 INNEO Solutions GmbH

3. An absolute path specification: Avoid using this if possible, as changes cannot be committed.

**Creating an external file**
The CSV file must contain the selection values in the first column and the labels in the other columns. The header cell of the first column is ignored. The header cells of the other columns must contain the two-character abbreviations of the corresponding languages.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | en | de | fr |
| 2 | 5 | thin | dünn | mince |
| 3 | 10 | medium | mittel | moyen |
| 4 | 15 | thick | dick | épais |
| 5 | | | | |

# Editing parameters

Four parameter types are available for editing model, component and feature parameters. The parameter types are editable with these field types:

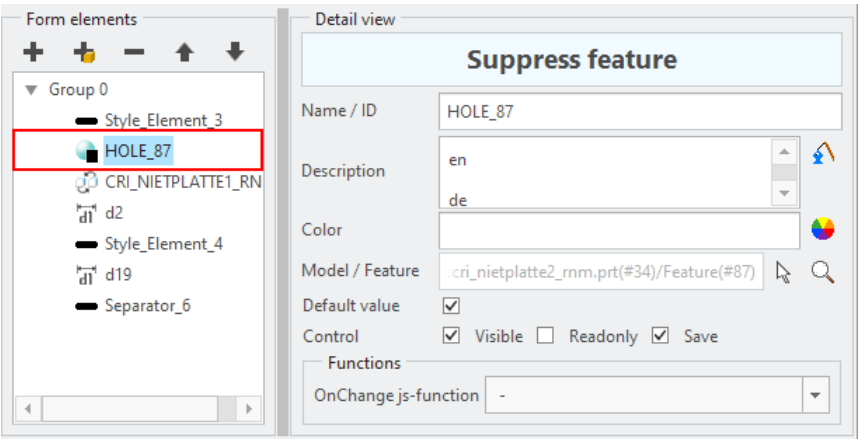
| Icon | Name | Editable by |
|---|---|---|
| [B] | **Boolean parameter** (Yes/No parameter) | – Radiobox |
| [I] | **Integer parameter**<br>This parameter type accepts only integer input. All inputs following a decimal point will be ignored. | – Radiobox<br>– Inputpanel<br>– Dropdownbox<br>– Extended dropdownbox |
| [D] | **Real parameters**<br>Real parameters accept integer input as well as point-separated values. | – Radiobox<br>– Inputpanel<br>– Dropdownbox<br>– Extended dropdownbox |
| [S] | **String parameters**<br>String parameters accept any character string. | – Radiobox<br>– Inputpanel<br>– Dropdownbox<br>– Extended dropdownbox |

The asterisk [*] indicates help parameters. Help parameters do not exist in the model and cannot be saved. Their only purpose is to be used in JavaScript functions[154]. Help parameters can have any of the four available data types.


© 2026 INNEO Solutions GmbH

# Feature / component suppression

Form elements for suppress actions define the suppress state of a model or feature.




Use the Selection button to select the model / feature to be suppressed from the model tree or directly from the model. Under *Scope*, specify whether to replace the selected model instance or all assembled models with the same name.

Suppress actions are displayed as a checkbox[149] in the form. Select the default state of the checkbox under *View*.

**Please note**: Suppressing all parts only affects parts in the same assembly.

# Component replace

Replacement form elements are used to replace models or features with other models or features. Component replacements can be managed using any type of input field, but the use of drop down lists is recommended.

Use the Creo Selection button to define the model or feature to be replaced using the model tree or a model. Under *Scope*, specify whether to replace the selected model instance or all component models with the same name.

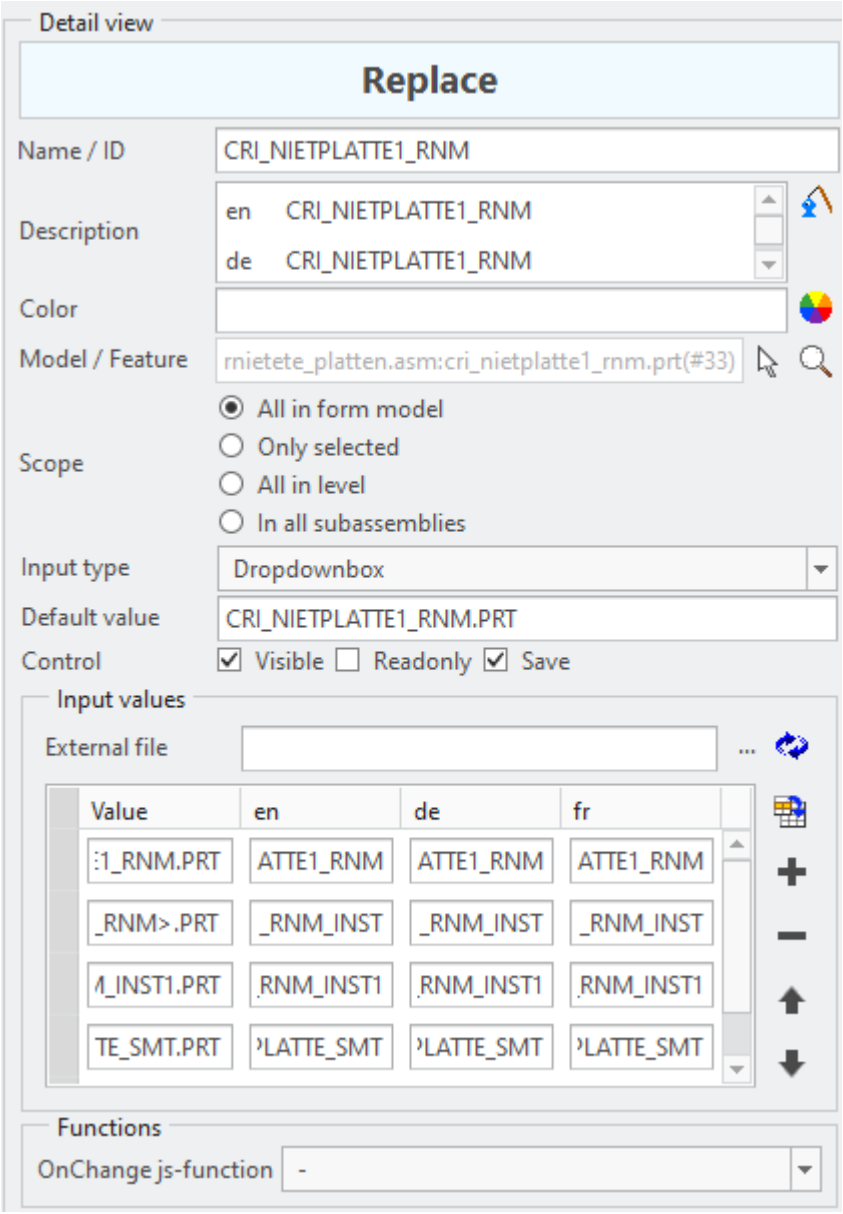
To name instances, use the following format: `Instance<Generic>.prt`.

If you are using a value list, you can import the instances from the family table of the selected model. To do this, click *Import instances* .

**Warning**: Take care to use the correct spelling for instances.

You can use models or features from the current work directory, from the directory of the assembly or from family tables.


© 2026 INNEO Solutions GmbH




*Configured replacement element using a dropdownbox*

**Please note**:Forms are processed from the top-down. Everything before a replace action is applied to the old model. Everything that comes after a replacement action is applied to the new model.
Replacement actions are only supported for interchange assemblies and instances from family tables.
Replacing all models with same name only works within the same assembly.

## Separator

Separators are used to organize a form and group its elements. To configure a separator, enter a name and localized descriptions for it.

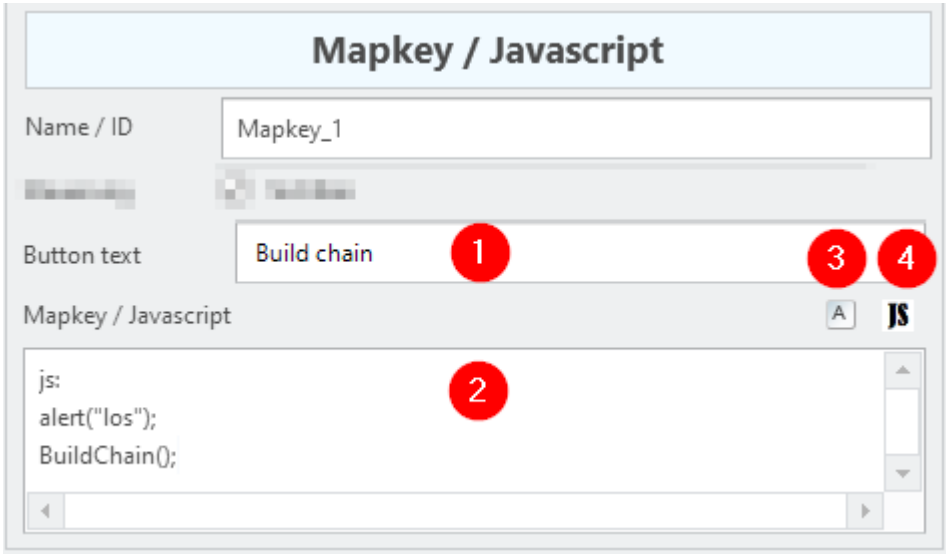
Use the *Visible* option to use a separator only in the Editor.


© 2026 INNEO Solutions GmbH

**Tip**: The name of a separator can be edited using JavaScript[784].

## Mapkey / JavaScript

Elements that execute a mapkey command or a Javascript function, are displayed as buttons on the form and require special configuration.

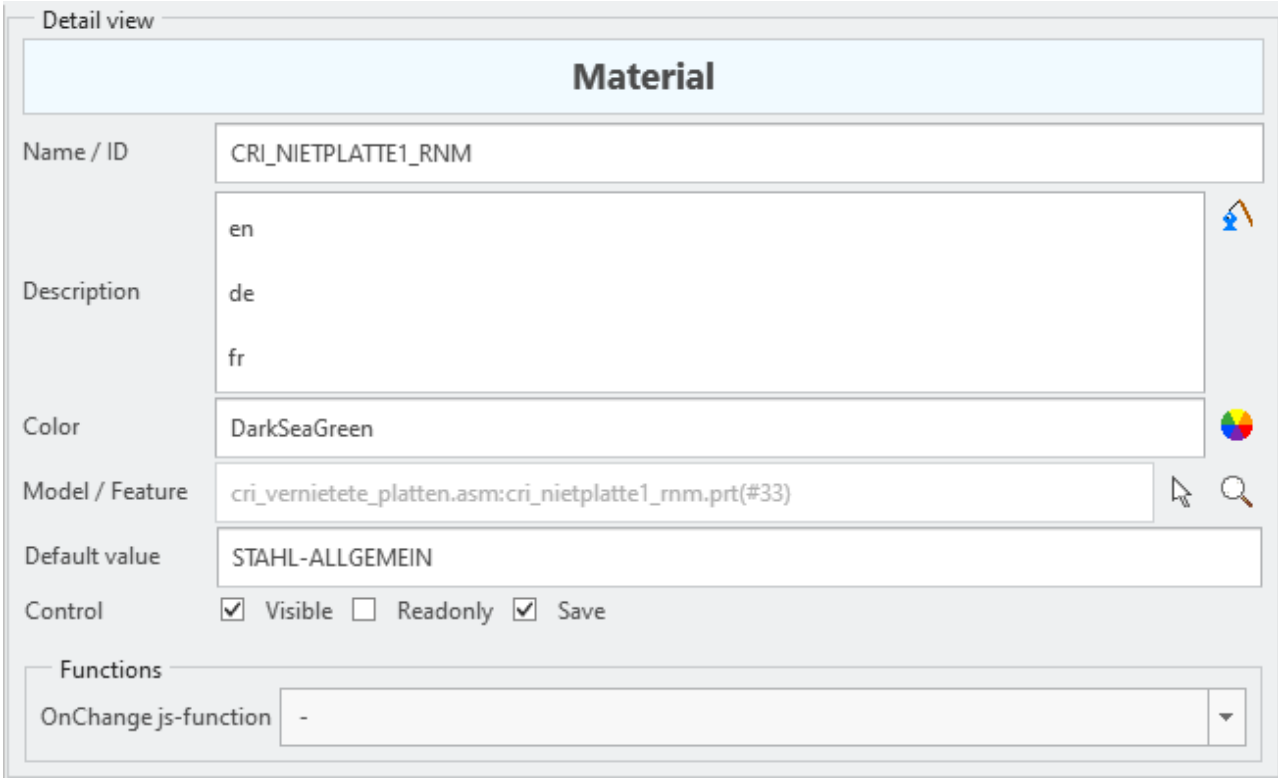



1. Define the label for the button under *Button text*.
2. In the *Mapkey/Javascript* section, enter the code to execute
   – a Mapkey command or
   – a Javascript function[783] (Javascript functions must begin with "js:")
3. Use the *Shorten Mapkey* A button to display long Mapkeys in a more compact format.
4. The button JS launches the *JavaScript Editor*[784] to help you write JavaScript code.

## Material

Material input fields are used to assign materials to individual models. Enter a name and a localized description to configure a material input field. Enter a default value. The default value is descriptive. A matching material must be entered by the user.


© 2026 INNEO Solutions GmbH




## Creating a binary checkbox

To use this input type, you must have a Boolean parameter in your model. Create this Boolean parameter as a form element[137] to create the *Binary Checkbox* input type. The input type is then uneditable.

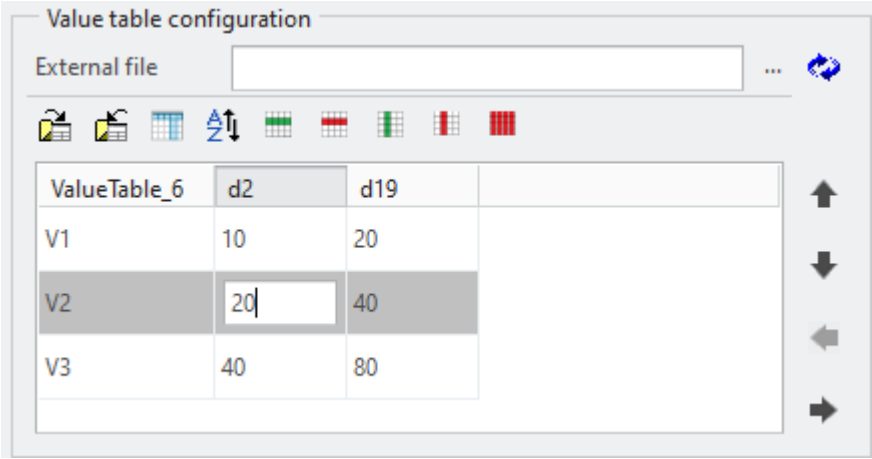
You can specify whether the value is enabled or disabled by default.

## Creating value tables

This section provides instructions for defining value tables that allow you to set multiple model properties simultaneously. Value tables control variants for already created form elements (dimensions, parameters, replacements and suppressions). Each table row is a single variant. Use the arrow buttons on the right to move rows or columns.

**Tip**: If you do not want a form element to be addressed in a particular variant, enter * in the cell for that element.


© 2026 INNEO Solutions GmbH




*Value table*

**Creating an external file as value table**

You can also store the values for the selection in an external CSV file. The first two rows of the CSV file link the values given below to the required form elements.

There are two options for creating tables:

– Rows 1 and 2 are empty: the columns are assigned according to their order. To force this behavior, the 2nd header row must be empty.

– Specifying the table header in lines 1 and 2 (*see figure*). Based on this information, the header is created in the value table and the table is filled in. The columns are assigned according to their ID in the 2nd header row.
To create the header rows for a CSV file, first create the required columns in the value table. Then, export the table as a CSV file. The exported file contains the required table header.

| | A | B | C |
|---|---|---|---|
| 1 | variant_table\|Dimensions | d2 | d19 |
| 2 | variant_table\|Dimensions | dim\|33\|2 | dim\|33\|19 |
| 3 | V1 | 10 | 20 |
| 4 | V2 | 20 | 40 |
| 5 | V3 | 40 | 80 |

Enter the desired values into the CSV file. Then select the CSV file under *External file*. The values from the CSV file are saved in the form. When the form is called up, it checks whether the linked CSV file exists, and changes are applied.

**Linking external files**

There are three ways of specifying a CSV file. They are the same as for Selection and option fields.[144]

You can click *Update* to re-read the displayed values from the CSV file.


© 2026 INNEO Solutions GmbH

# Editing value tables

The following control buttons are available for editing value tables.

| Icon | Name | Description |
|---|---|---|
|  | Import value table from file | Imports a value table from the following file formats:<br>– xls (Excel 97-2003)<br>– xlsx (Excel 2003-2016)<br>– csv (Comma-separated values)<br>– txt (text files with UTF-16LE/Unicode formatting) |
|  | Export value table to file | Exports a value table to the formats:<br>– xls (Excel 97-2003)<br>– xslx (Excel 2003-2016)<br>– csv (Comma-separated values) |
|  | Edit value table in spreadsheet software | The value table is opened in a spreadsheet program (depending on the client computer) and can be edited. Then it is reimported. |
|  | Sort selected column | Sorts the rows by the active column. Three sorting orders are available:<br>– 0-Z<br>– Z-0<br>– initial |
|  | Add/Remove row | Adds a new line under the current cursor position. To remove, select a row and click *Remove row*. |
|  | Add/Remove column | Adds a new column to the right of the current cursor position. To remove, select a column and click *Remove column*. |


© 2026 INNEO Solutions GmbH

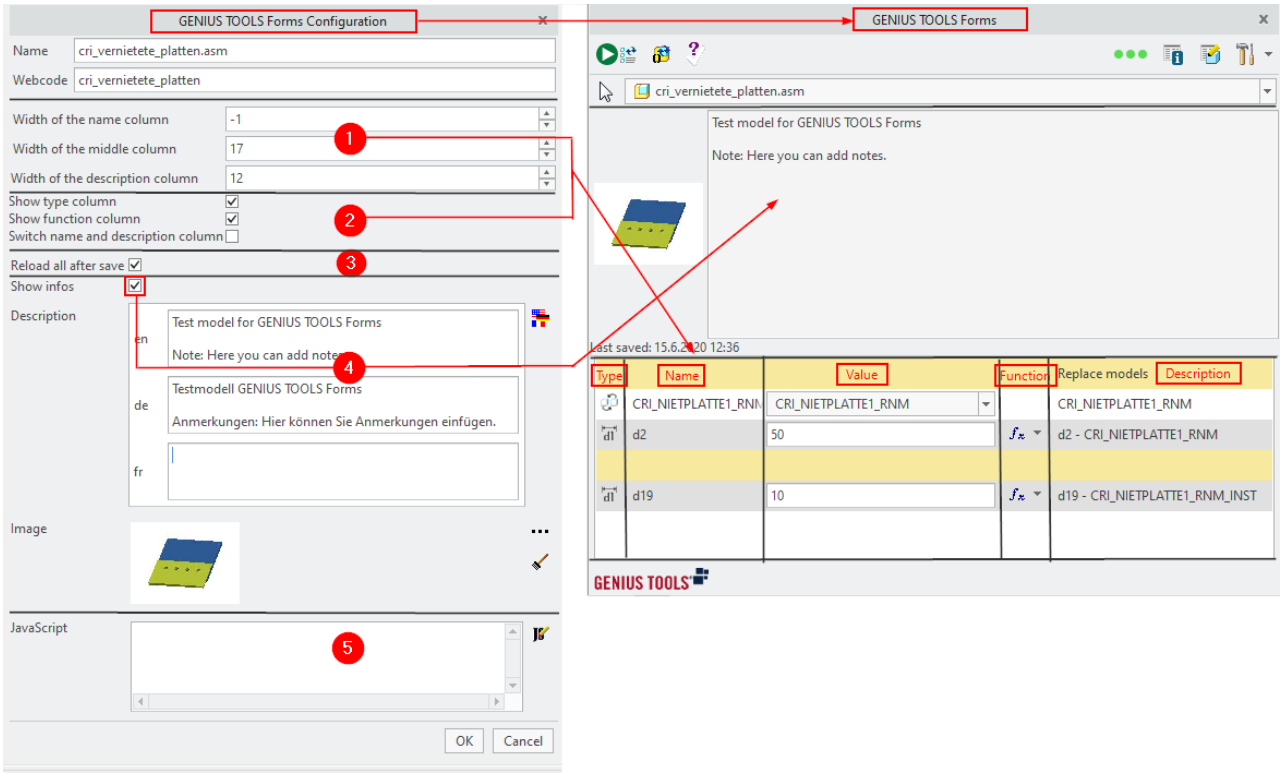
| Icon | Name | Description |
|---|---|---|
| | Remove all rows and columns | Deletes all rows and columns from the table. |

**Please note**: All changes made to a value table during editing in a spreadsheet program (delete, move or change rows/columns) are applied by *GENIUS TOOLS Forms*.

Find an example on how to **create a value table** in the section Use cases.[156]

## 8.3.1.6 Editing form display

The button *Opens Forms configuration* opens the Forms Configuration dialog which allows you to specify the user interface of *Forms*[125] as follows:




*Structure of Forms Configuration and impact on Forms user interface*

1. defining column widths of a form[153]
2. displaying and arranging columns[153]
3. renewing Creo values[153]
4. providing description and sample image[154]
5. entering JavaScript code[154] to be used in the form


© 2026 INNEO Solutions GmbH

## Column widths

**Width of the name / value / description columns:** Specifies the width of the respective column in characters.

– Value "0": hides the column

– Value "-1": applies the default values of the configuration options to the column

**Please note**: Do not hide the value column if you want users to fill out the form.

## Displaying and arranging columns

In this segment you can:

– **Show Type column:** removes or displays type column (first column)
To apply this setting, save it in the *Forms Editor*.

– **Show Function column:** removes or displays function column (fourth column)
To apply this setting, save it in the *Forms Editor*.

– **Switch Name and Description column:** If you switch the Name and Description columns, the description will appear before the value column, which can be used if you prefer to write your own descriptions.
To apply this setting, save it in the *Forms Editor* and restart *Forms*.

You can hide the Name, Value and Description columns in the previous segment by entering "0".

## Renewing Creo-values

**Display Creo values after saving:** Re-read all values of the model into the form after saving.

On
- displays the current values of the model (calculated values e.g. from relations / dependencies) in the form.
- in case of an error message - i.e. if the input value could not be transferred to the model, the previous value is displayed again in the form

Off
– all entered values are kept in the form, including those values that cause errors (i.e. that cannot be applied to the model)
– does not re-read the dependent values from the model
- useful for not losing incorrect entries

**Please note:** Before disabling this feature, test to determine how dependencies are affected.


© 2026 INNEO Solutions GmbH

## Providing description and sample image

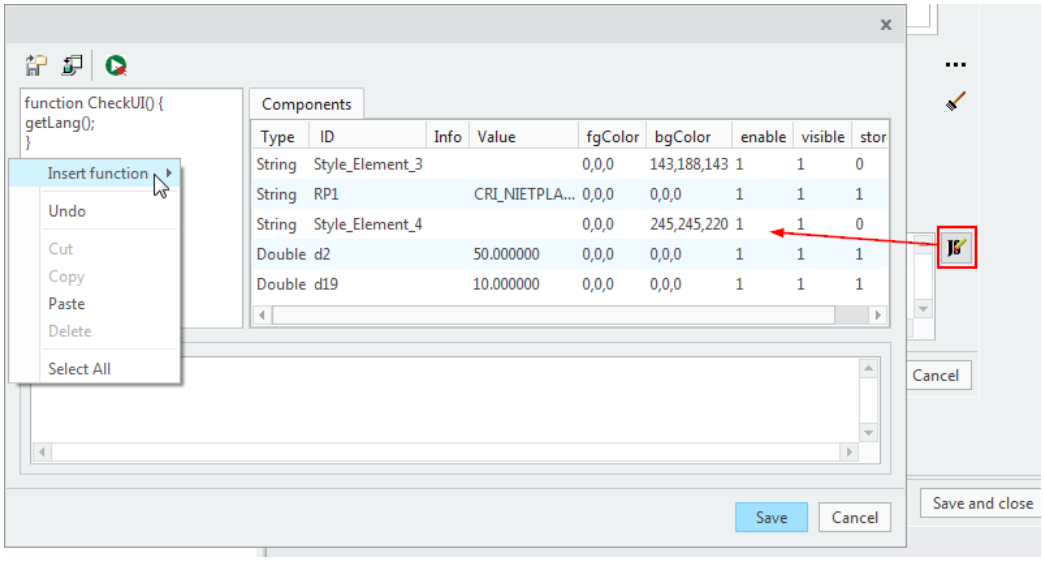
**Display information**: Automatically displays the information area when a form is opened.

**Description**: Enter a descriptive text for the form in the fields. Use the button *Open the multilanguage description window* to manage the languages.

**Preview image**: Enter any image here. Images are saved directly in the forms definition. The size of the saved image depends on the configuration options `gt_image_width` and `gt_image_height`. An image can be deleted by clicking the broom icon.

## 8.3.1.7 JavaScript in Forms

Executable JavaScript code is stored in the JavaScript field. Enter the code directly into the input field or click on the JavaScript icon to open the *JavaScript Editor*[784]. Right-click to open the context menu for quick function entry (Insert function). For more information, consult the chapter *JavaScript Editor*[784] and the application example for using JavaScript with a GENIUS TOOLS module[535].




*JavaScript Editor in Forms*

## Time of execution

JavaScript can be executed at different times. You can add as many functions as required to your JavaScript code.

| Time of execution | Function |
|---|---|
| After loading a form | PostLoad |
| After loading of a value table (CSV/XML) | PostLoadFromFile |


© 2026 INNEO Solutions GmbH

| Time of execution | Function |
|---|---|
| Before saving values in a form | PreSave |
| After saving values in a form | PostSave |
| After clicking the button *Check values* in the *Forms* dialog | CheckUI is a function that must be set directly in the JavaScript Editor of the Form[136]. The function cannot be set in a single form element. |
| After editing a *form* value or after clicking Enter | OnChange, is a function activated in the Functions[143] section of the *Detail view*. |

The names of *PostLoad, PostLoadFromFile, PreSave, PostSave* and *CheckUI* are fixed. Functions of the type *OnChange* can have any name.

Loading values from a CSV or XML file does not trigger any JavaScript OnChange functions.

**Please note:** Use the JavaScript function *creoMapkeyAddToStack* only as a PostSave function, because it executes mapkeys and mapkeys usually close windows.

## 8.3.1.8 Automatic load of external data

The automatic reloading of external forms can be used, for example, to transfer legacy data.

If the selected or current model does not contain a form, but a webcode, a check is made in a defined directory to see if a form definition (XML) with the parameter name WEBCODE is available. If so, it is loaded into the model and the form is opened. Other parameter names can be specified in the `gtf_autoload_parameter` configuration option to reload externally stored forms. You may need to do this, for example, when importing models created with MUI - a component of Startup TOOLS up to version 2018.

If you want to add forms to models created with MUI, you must first load, update, and validate the forms in the template model. Then they are saved as XML files in a specified folder.

1. Specify a folder with form configurations (XML) under `gtf_autoload_folder`.
2. A parameter must now be found in the model for automatic loading. Specify the parameter name in `gtf_autoload_parameter`.
3. If the parameter name exists, but there is no form in the model yet, an XML file with the same name will be searched and loaded. The `gtf_autoload_overwrite` configuration option can be used to specify whether this operation should also be performed for models that already contain a form. In this case, the existing form will be overwritten.


© 2026 INNEO Solutions GmbH

## 8.3.2 Use cases

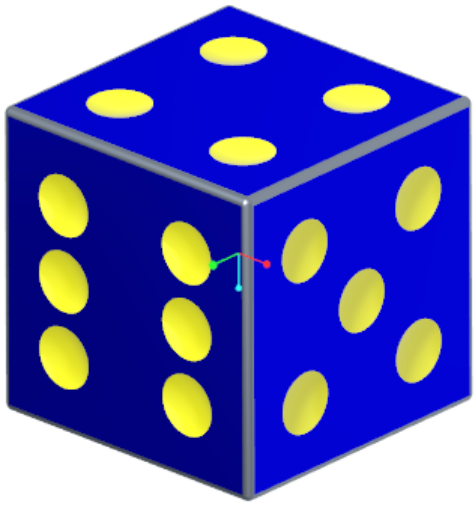
This section provides an example of how to create and use a form.

### Requirements

The example model is a cube. The size of the cube, the radius of the individual numbers of pips and the depth of the numbers of pips are supposed to be controllable. The positions of the numbers of pips in relation to each other and on the cube are fixed. They are not changed by the form.

### Creating a form

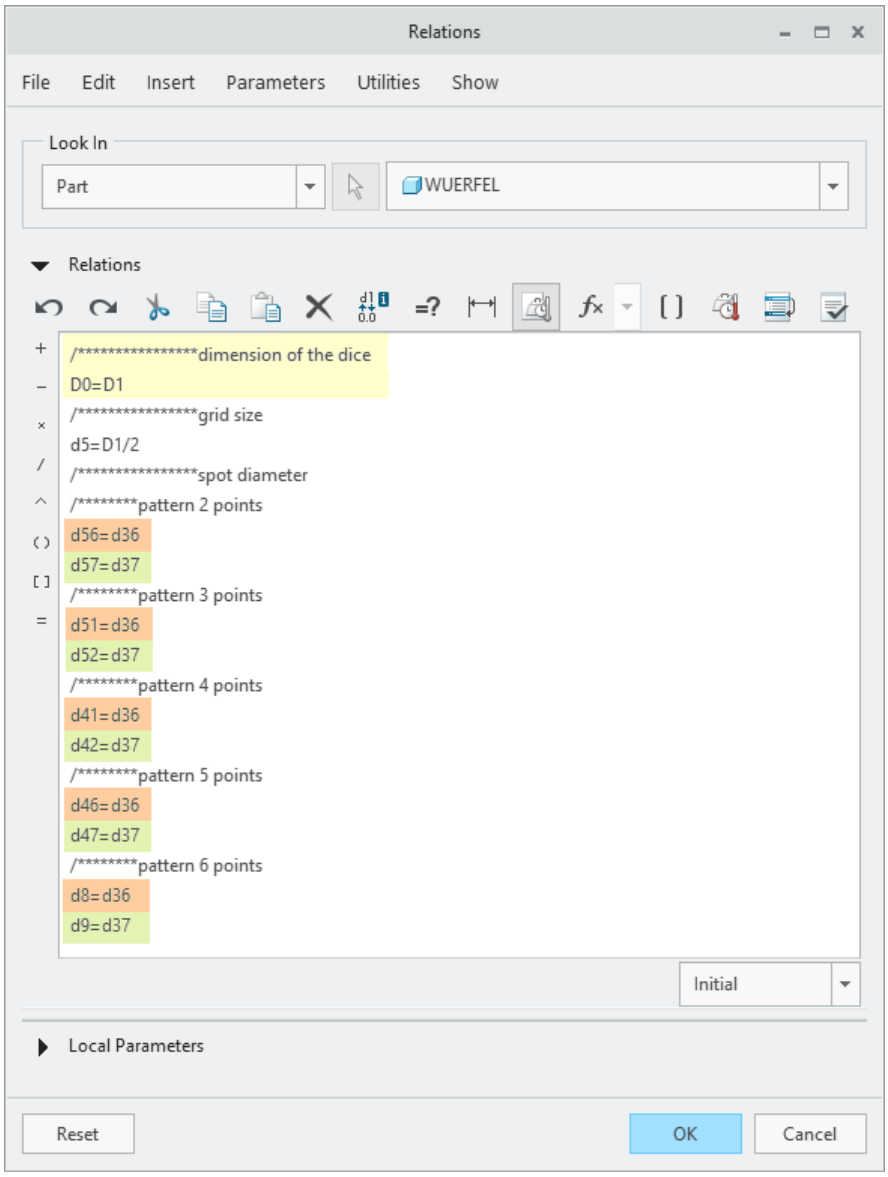
1. Open the model in which you want to create the form.



*Initial model of a dice*

2. Open the Creo window *Relations*.
3. Check the stored relations for compatibility and delete existing redundancies, if necessary.
4. Store comments in *Relations* to record the purpose of the relations. You can use this information when you create the form.

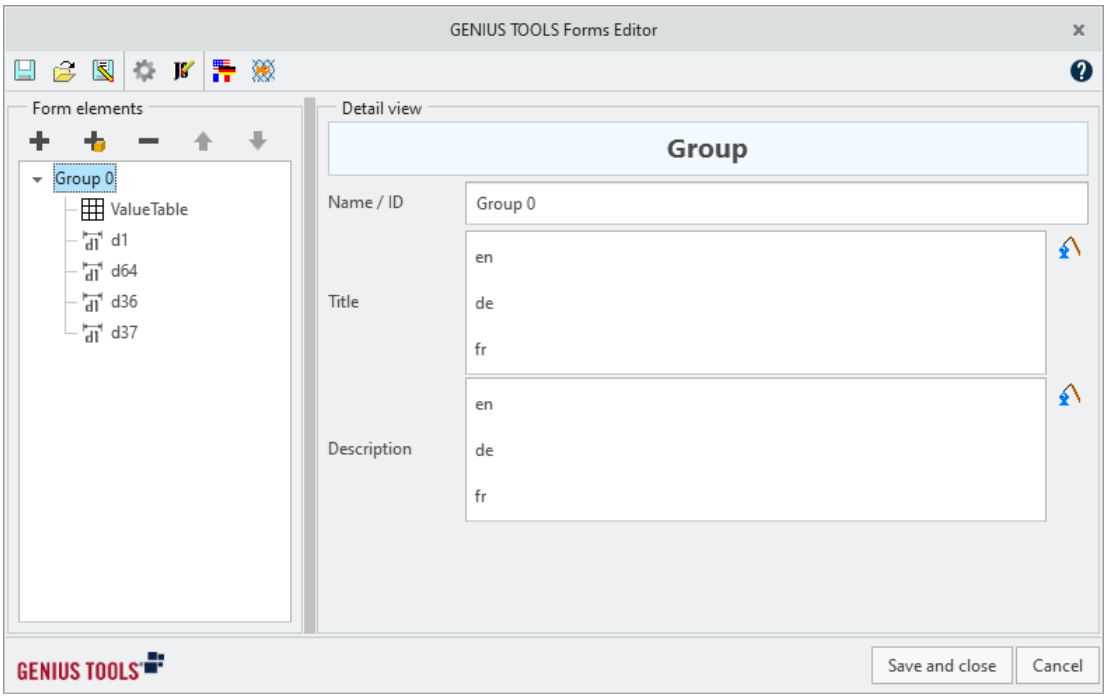

© 2026 INNEO Solutions GmbH




*Clearly defined relation overview for sample model*

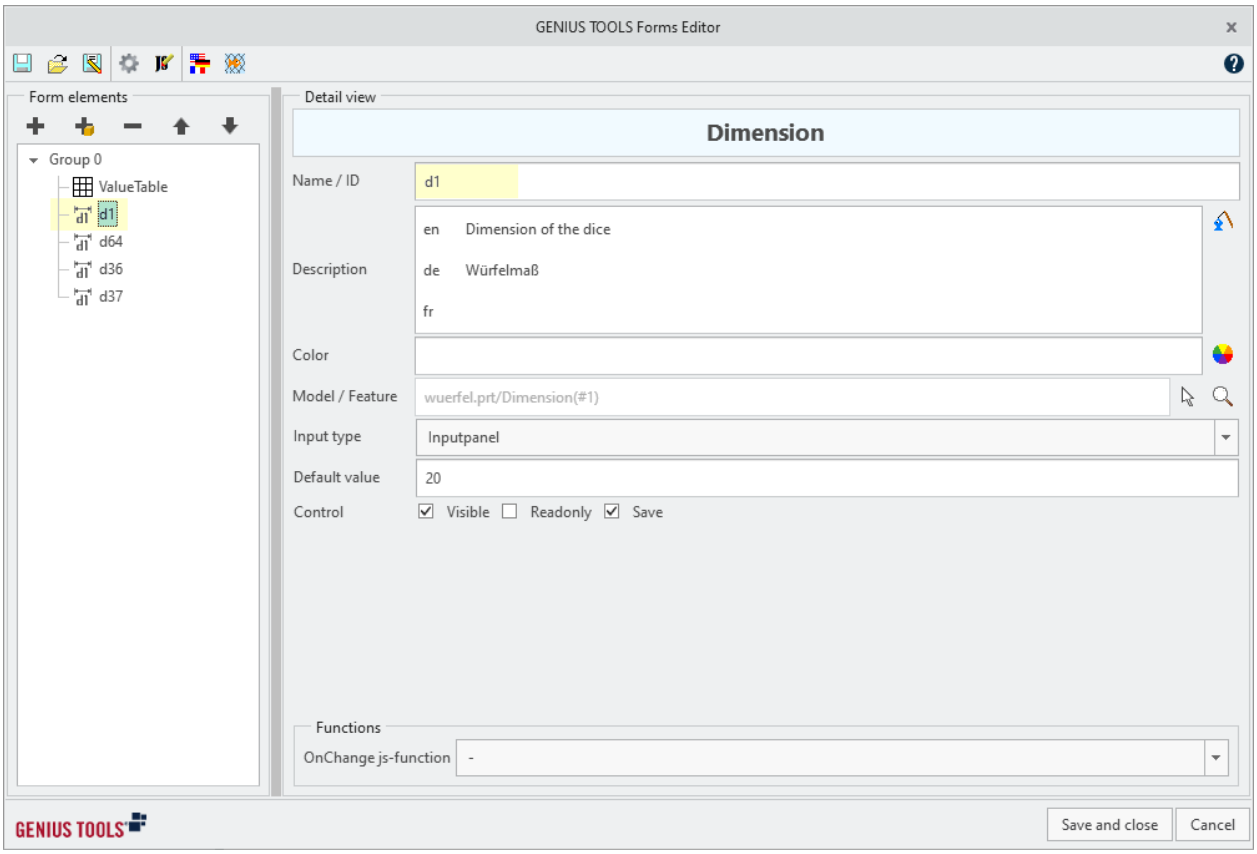
5. Click *OK* to close the Creo window *Relations*.
6. Go to the *GENIUS TOOLS* tab and open the *Forms* module.
7. Open the *Forms Editor* to create the desired form.
8. *Add a new group* to which the form elements will be attached.
9. Define properties of the group:


© 2026 INNEO Solutions GmbH



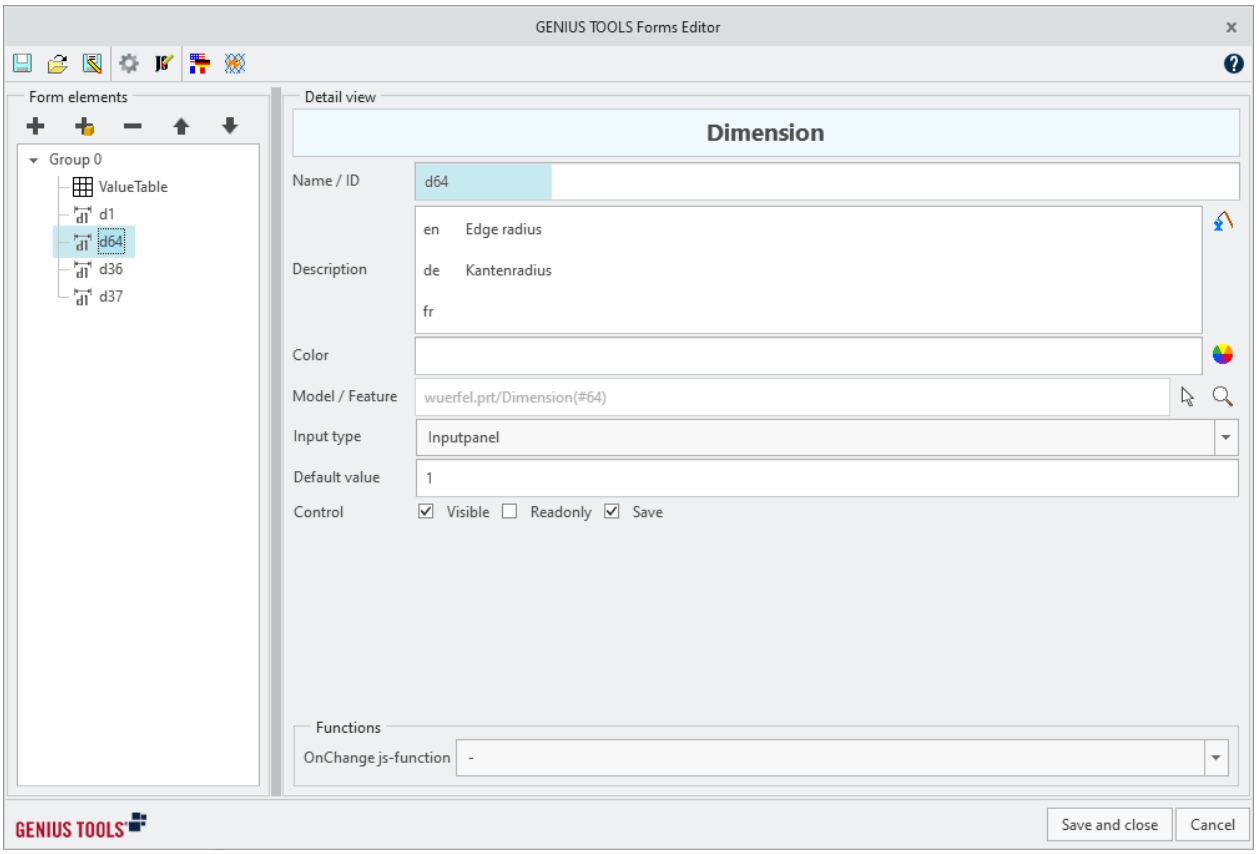

10. *Add new items* ✚ to this group.
11. Select the *type of controlled element*: *Dimension*.
12. Select one side of the cube: *d1*. Since the size of the cube is defined so that all side edges have the same length, it does not matter which side you select at this point. All edges of the cube in the form to be created are controlled by the relation defined above.
13. Fill in the detail view:
    a. Name / ID *d1* is filled in automatically. This name clearly indicates the relation.
    b. The *Descriptions* are used to store the descriptions that will be displayed in the form for this form element in both English and German.
    c. The input type *Inputpanel* allows to enter any values in the form.
    d. An unchecked *Readonly* allows to assign different values to the input field in the form.
    e. When the *Visible* and *Save* checkboxes are selected, the form element is visible and editable.


© 2026 INNEO Solutions GmbH

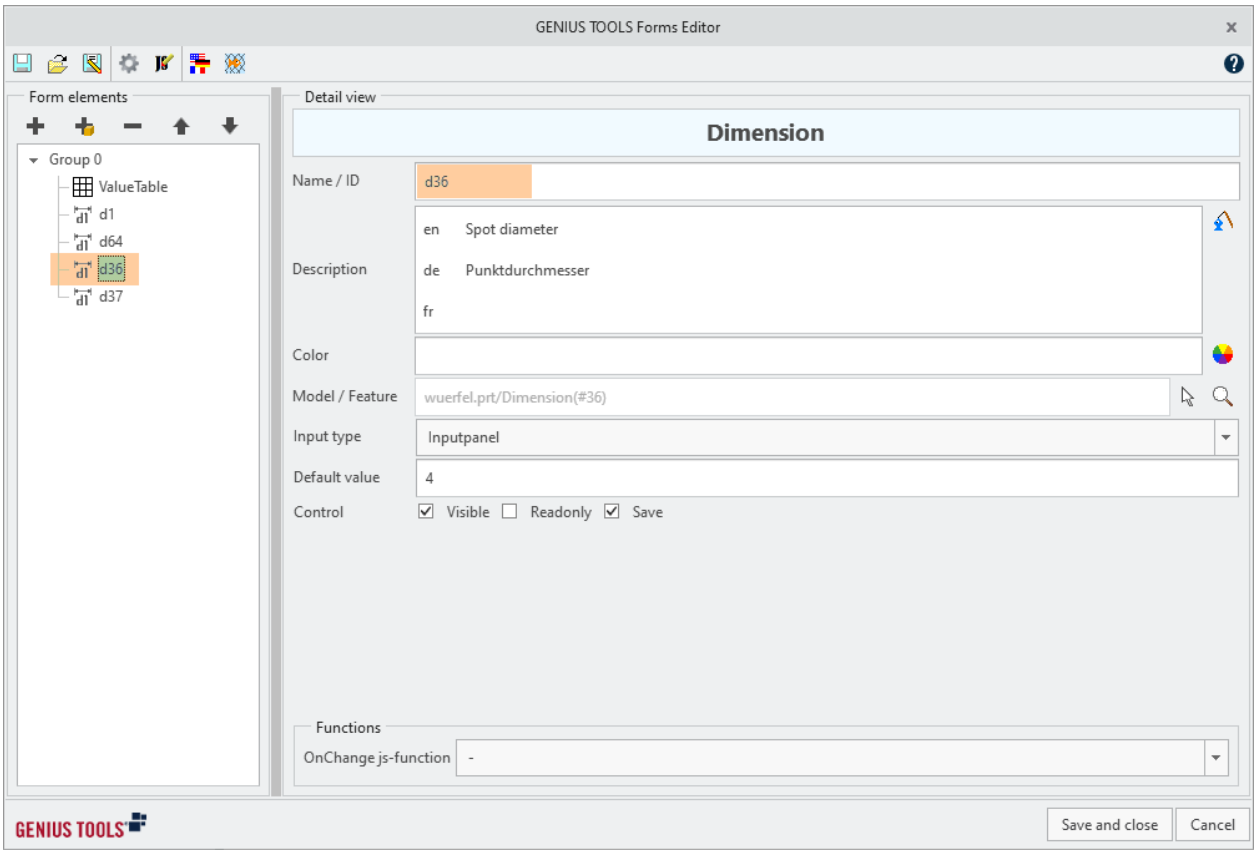



14. The form element for the dimension of the dice is now created. Repeat steps 10 and 11 for the edge radius, point diameter, and point depth. *Add new items* ➕ to this group.
15. Select the *type of controlled element*: *Dimension*.
16. Select one side of the dice: *d64*. The round that defines the edge radius was inserted into the model last. Thus, the edge radius is the same for all cube edges and does not need to be defined via relations.


© 2026 INNEO Solutions GmbH

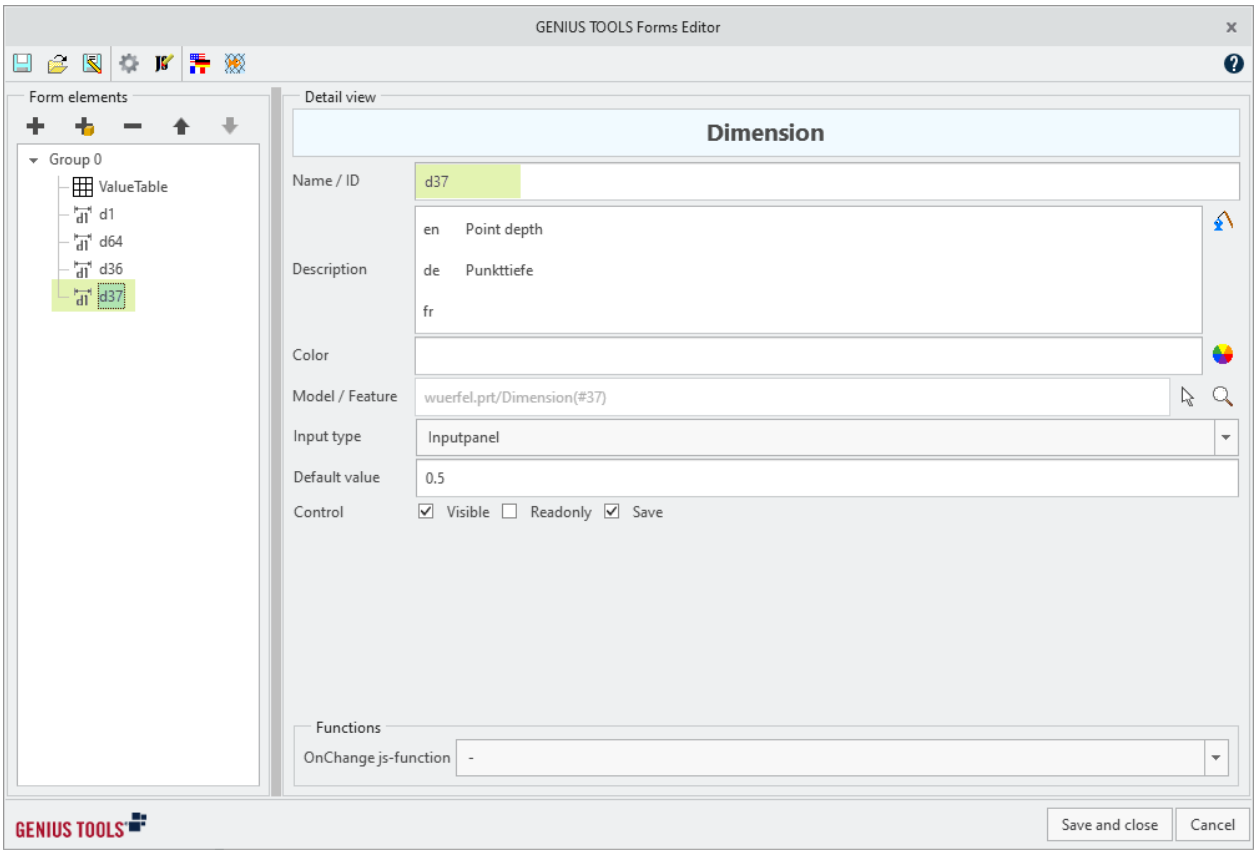



17. *Add new items* ✚ to this group.
18. Select the *type of controlled element*: *Dimension*.
19. Select the point diameter of a number of pips of the cube: *d36*. The point diameter is defined the same for all numbers of pips.
20. Complete the detail view in the same way as the *d1* view.




21. *Add new items* ✚ to this group.
22. Select the *type of controlled element*: *Dimension*.
23. Select the point depth of a number of pips on the dice: *d37*. The depth of the point is defined in the same way as the diameter of the point for all numbers of pips.
24. Complete the detail view in the same way as the *d1* view.

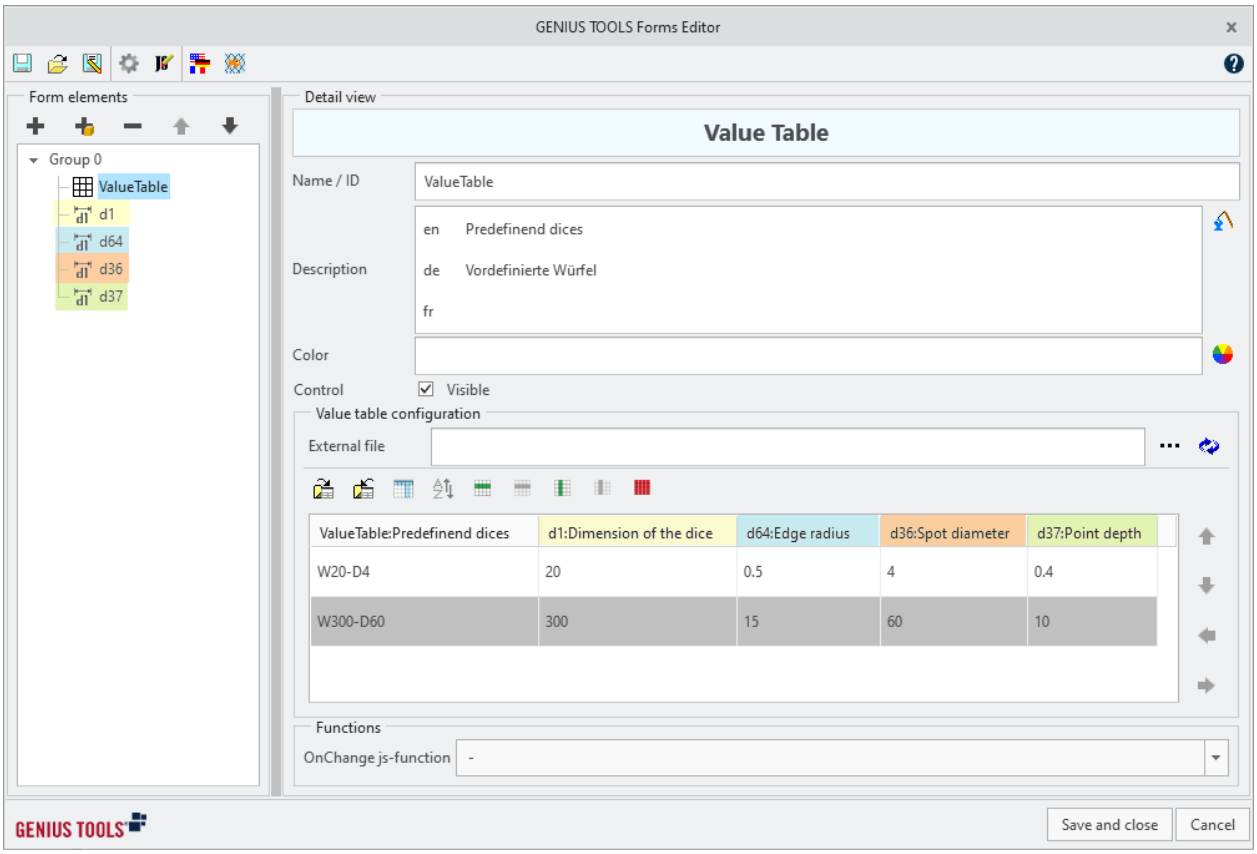

*© 2026 INNEO Solutions GmbH*



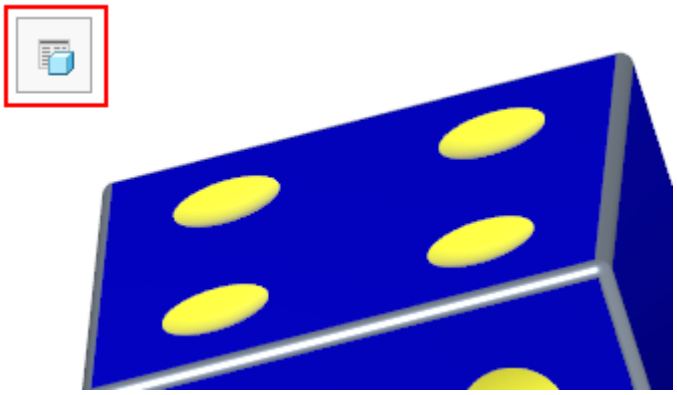

25. Create a value table. *Add new items* ✚ to this group.
26. Select the *type of controlled element*: *Value Table*.
27. Use  to add a column to the value table for each of the form elements you created.

**Please note:** Value tables can only be created for existing form elements.

28. Add two rows to the value table using  .
29. Fill in the rows with predefined cube sizes.
    In the first row of the table, the predefined cube W20-D4 describes the initial model of the cube. With this predefinition, the initial state of the cube can be restored at any time.
    The second row of the table describes a variance of all form elements.

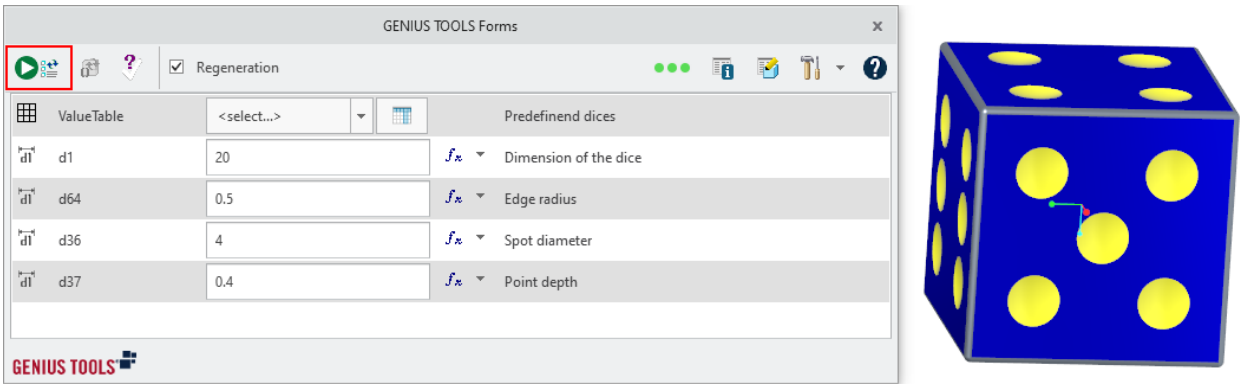



30. Click *Save and close*.
31. When you open *Forms*, you will see the created form directly. When you close *Forms*, you can also access the form directly in the open model.

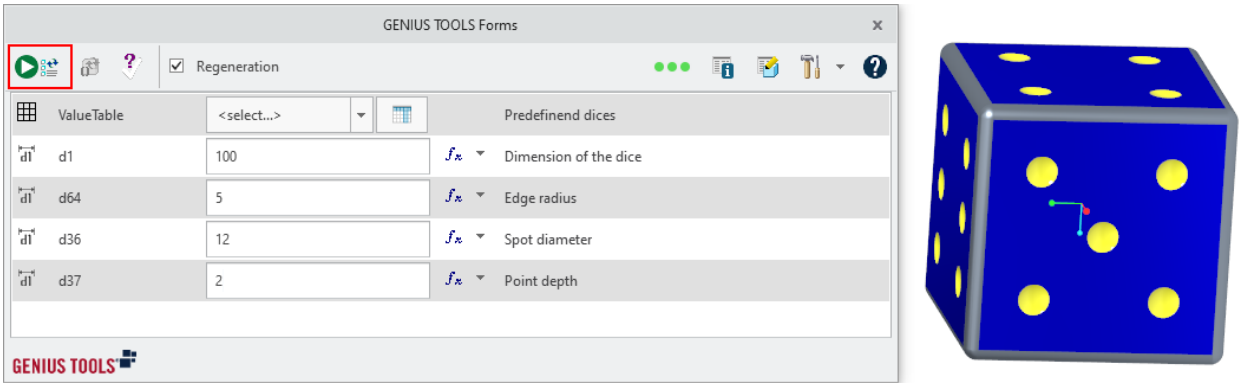


32. The following result is obtained by setting the predefined cube W300-D60 and then applying the value table by clicking the button .


© 2026 INNEO Solutions GmbH



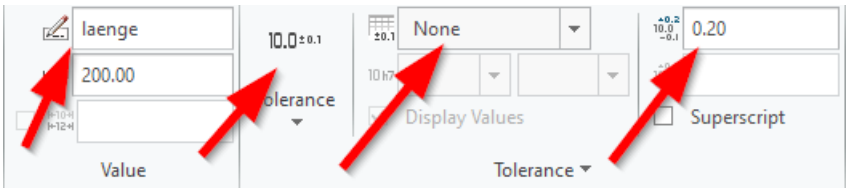

33. The form elements can be set individually and the settings can be applied by clicking the button . For example, it looks like this



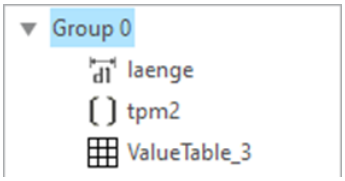

## Setting tolerance from a value table automatically

Depending on the value of a dimension, the tolerance is to be set automatically. In a form with three elements (dimension, tolerance, value table), a JavaScript function[784] is stored that is called when the dimension is changed. The JavaScript function sets the value of the value table from which the tolerance is set.

1. Create a new component with a cuboid.
2. Edit the dimensions of the cuboid.
   name, tolerance table, display of the tolerance (Symmetric), tolerance value

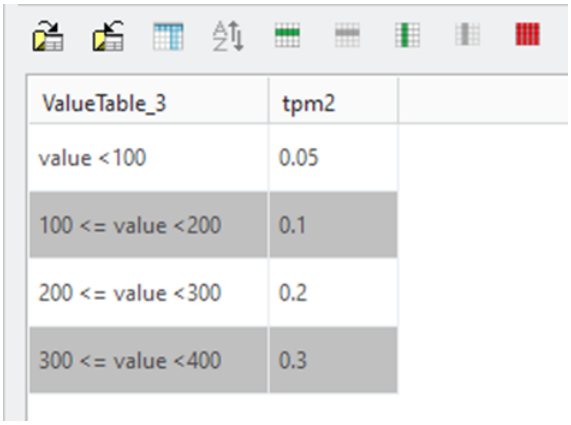



3. Create a form with three elements:



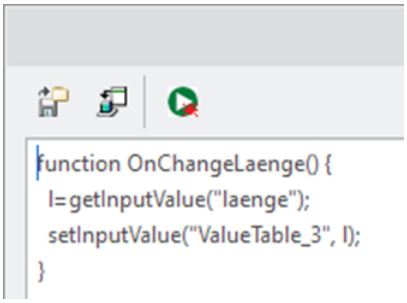

4. Fill in the tolerance value table:


© 2026 INNEO Solutions GmbH



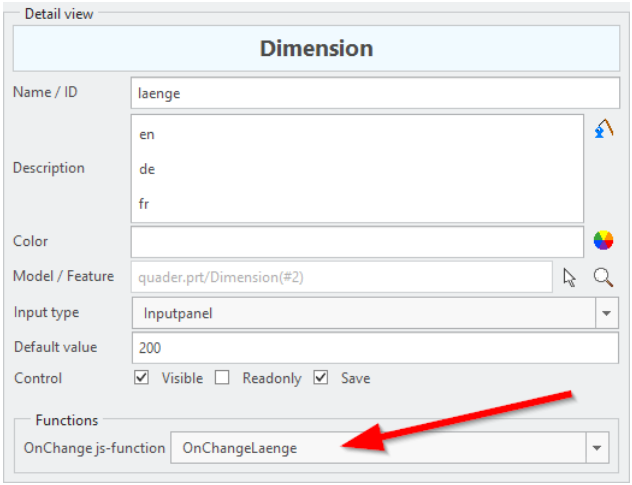
| ValueTable_3 | tpm2 |
|---|---|
| value <100 | 0.05 |
| 100 <= value <200 | 0.1 |
| 200 <= value <300 | 0.2 |
| 300 <= value <400 | 0.3 |

5. Create the following JavaScript function:

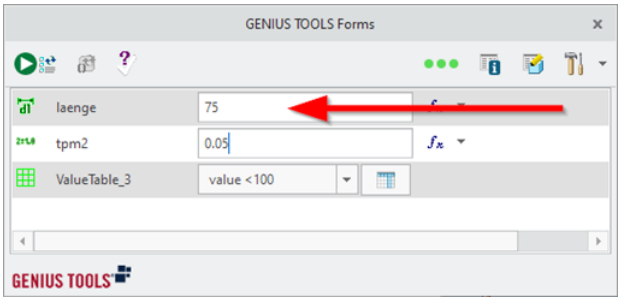


```
function OnChangeLaenge() {
  l=getInputValue("laenge");
  setInputValue("ValueTable_3", l);
}
```

6. Specify the time at which the JavaScript function is to be executed. The function should be applied when there is a change in length.



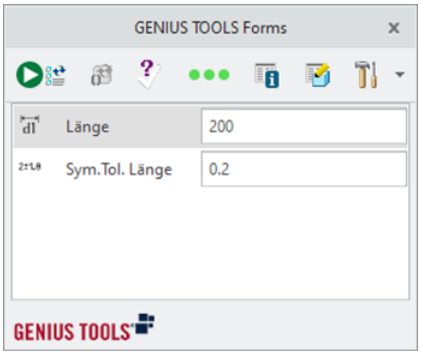

7. Test that the JavaScript code works as expected. Change the length in the form. The value table and tolerance are automatically adjusted.
The value table and tolerance change automatically.





© 2026 INNEO Solutions GmbH

8. Optionally, you can optimize the display of the *Forms* dialog, e.g. by hiding the value table and configuring the column display.





© 2026 INNEO Solutions GmbH

# 9 Function Manager

Available in the ribbon menu GT MBD

The module *Function Manager* allows you to create and manage **functions** and associated **functional components** that you need for **functional construction and specification in 3D**. Templates for functions and functional components can be read as XML files. See *Glossary* for GT MBD[30] for explanations of the terms in bold.

*Function Manager* is available in part mode and in assembly mode with the following features:

1. Overview of existing functions and their functional components (combined views, appearances, etc.)
2. Real-time editing of functions and functional objects: adding, renaming, extending, deleting
3. Generating several combined views at the same time and displaying them in the model tree
   - Generating combined views alphabetically without scrolling in the Creo Parametric main window
4. Switching between combined views without switching views
5. Editing, grouping and deleting combined views
6. Automatic naming of functions
7. Coloring of surfaces, features and parts
   - A predefined color scheme is provided and accessible via *View* > *Appearances* > *Library* > *startuptools*. This color scheme is ready to use and can also be customized.

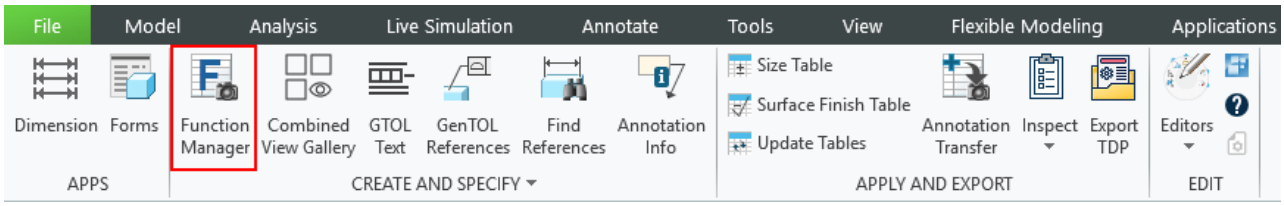
## 9.1 Usage

In this section you will find information about the use of *Function Manager*. The general structure of the program is explained.

### Starting the program: in part mode and in assembly mode

To start *Function Manager*  , click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY*. Click on the button to open the user interface. By default, when the program is opened, an overview of the functions already available in a model is displayed.

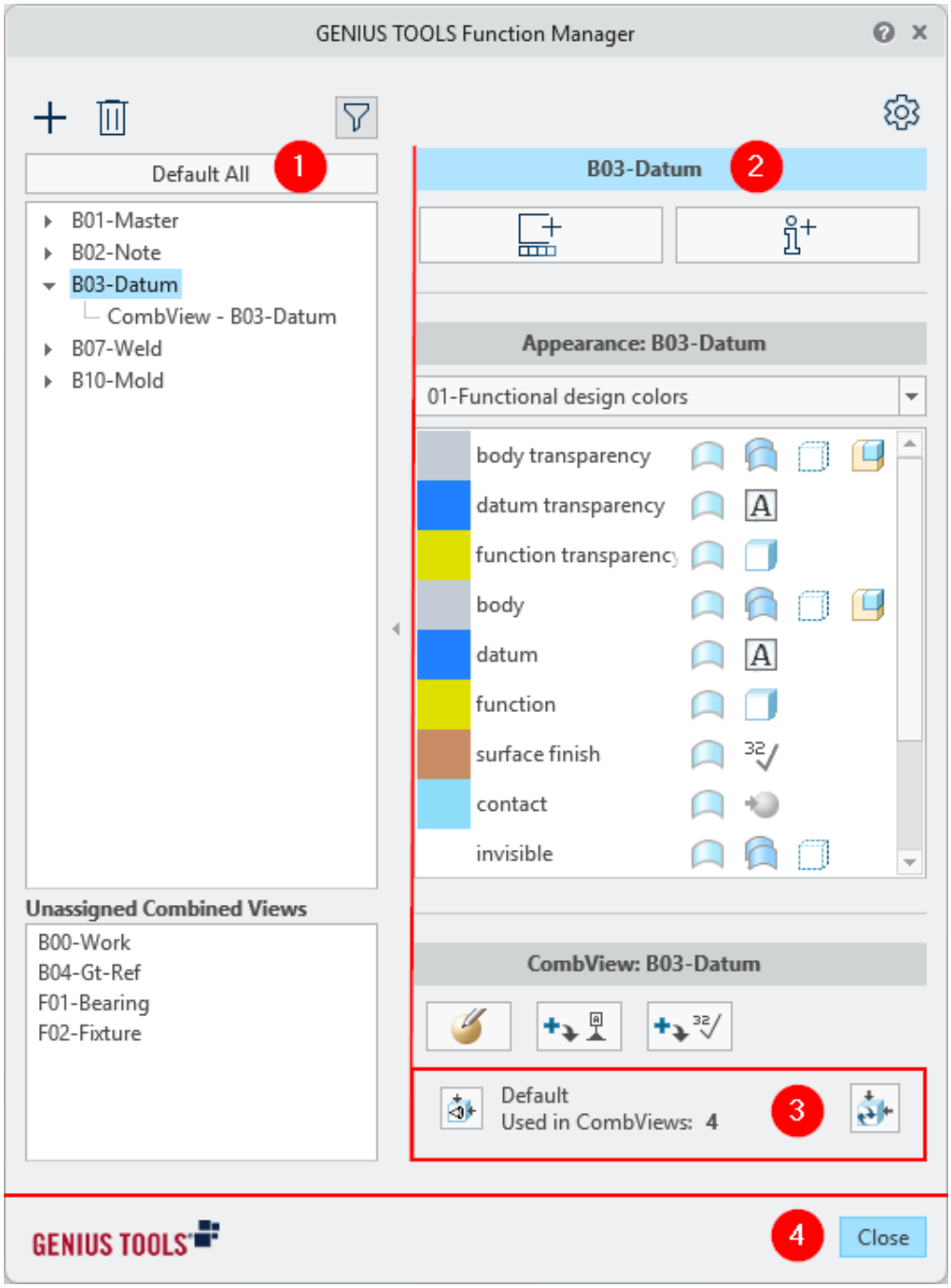

© 2026 INNEO Solutions GmbH




*Starting via the GT MBD in the ribbon menu*

## 9.1.1 User interface

The user interface of *Function Manager* consists of the following elements:



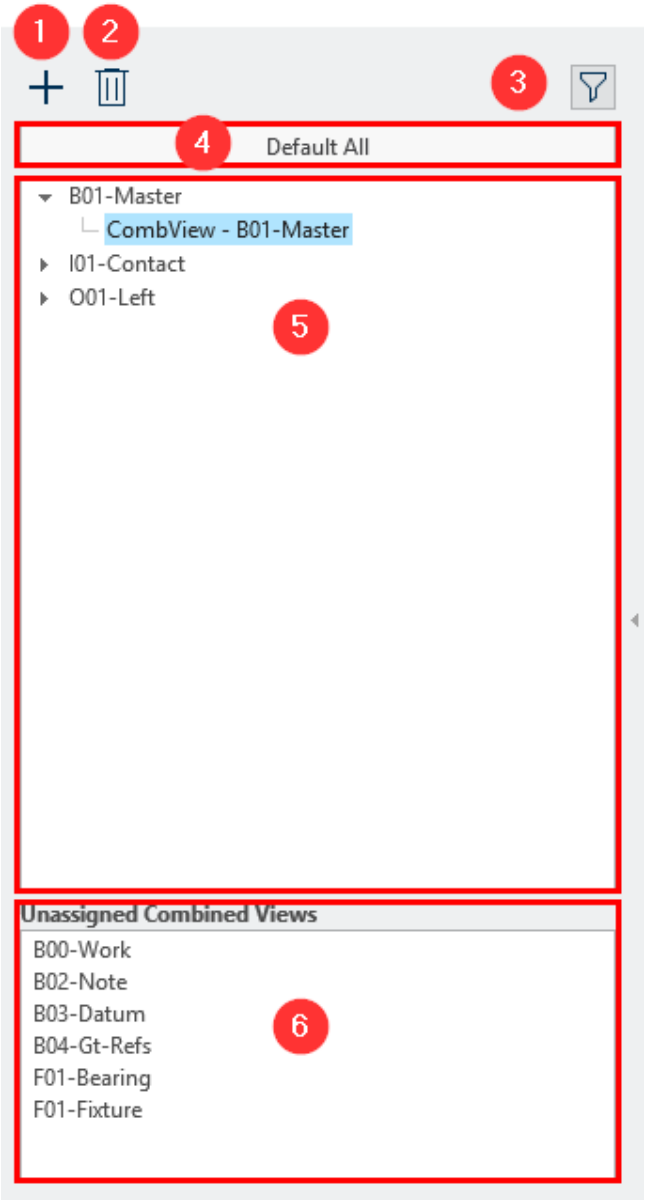


© 2026 INNEO Solutions GmbH

1. Function overview[169]
2. Detailed view of the selected function with editing options:
   Changing the naming scheme[179], Renaming and storing additional information[170], Coloring surfaces, features and parts[175], Editing Combined Views[178]
3. Direct connection to the Orientation Manager[460] to redefine orientation as needed.
4. Close

## 9.1.2 Managing functions

In the following area of the user interface, you see a function overview of existing functions, functional components and commands for adding, deleting and filtering.




1. Adding one or more functions[171]


© 2026 INNEO Solutions GmbH

2. Deleting a function
   A new dialog box opens in which you can select whether the functional components are also deleted in the model. If a functional object is not deleted, it is listed under *Unassigned Combined Views*. It can then be reassigned to a function without the need for re-creation.

**Please note:** If a functional component is not deleted, it is listed under *Unassigned Comb. Views*. This means it can be reassigned to a function without being recreated.

3. Filtering functions
   The filter is enabled by default when the user interface is opened. The filter is displayed in one of the following states:

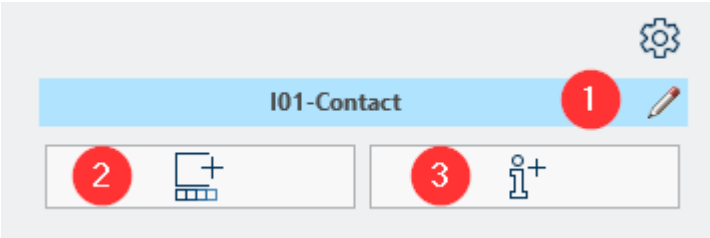
| | | |
|---|---|---|
| gray | Filter enabled | Appearances and construction groups are filtered. They are not displayed. |
| white | Filter disabled | Appearances and construction groups are not filtered. They are displayed. |

4. Displaying the model and appearance of the combined view *Default All*
5. If an existing function is selected, the first combined state is selected. If no combined view has yet been defined in the function, the combined view *Default All* is displayed.
6. Unassigned combined views
   Manually created combined views that do not yet exist in any function. Combined views can be assigned in various ways:
   a. Combined views can be added by right-clicking. If the function does not have a color effect, it can be created.
   b. If a function with the same name is created, the corresponding combined view is automatically assigned to that function.

**Please note:** If you create a combined view in the Creo View Manager while *Function Manager* is open, you must restart *Function Manager* to see the newly created combined view.

## 9.1.3 Editing metadata of a function

In this section, you will find a brief overview of the currently selected function:





© 2026 INNEO Solutions GmbH

1. **Name and number of the selected function**
   The button *Rename function* appears if the value `nameedit` is set to `true` in the template[182].
2. **Adding combined views**
   The XML file saved in the configuration option `gtfm_add_cs_to_existing_function` is read. Default: `%gt_resource_folder%function_manager\AddCS.xml`.
   Functions without combined views are skipped.
   If additional information is stored in this XML file, a link to that information is created for each piece of information.

**Tip:** For better clarity in this use case, it is recommended to define one function per XML file. If multiple functions are defined, subsequent functions with combined views will not be read and cannot be selected and added under *Functional Templates*.

3. **Adding additional information (PDF, web links, etc.)**
   The XML file saved in the configuration option `gtfm_add_info_to_existing_function` is read. Default: `%gt_resource_folder%function_manager\AddInfo.xml`
   All links saved in this function will be generated. Functions without links will be skipped during import.
   Web links must begin with `https://` or `http://`, e.g. `https://www.inneo.co.uk/en`.

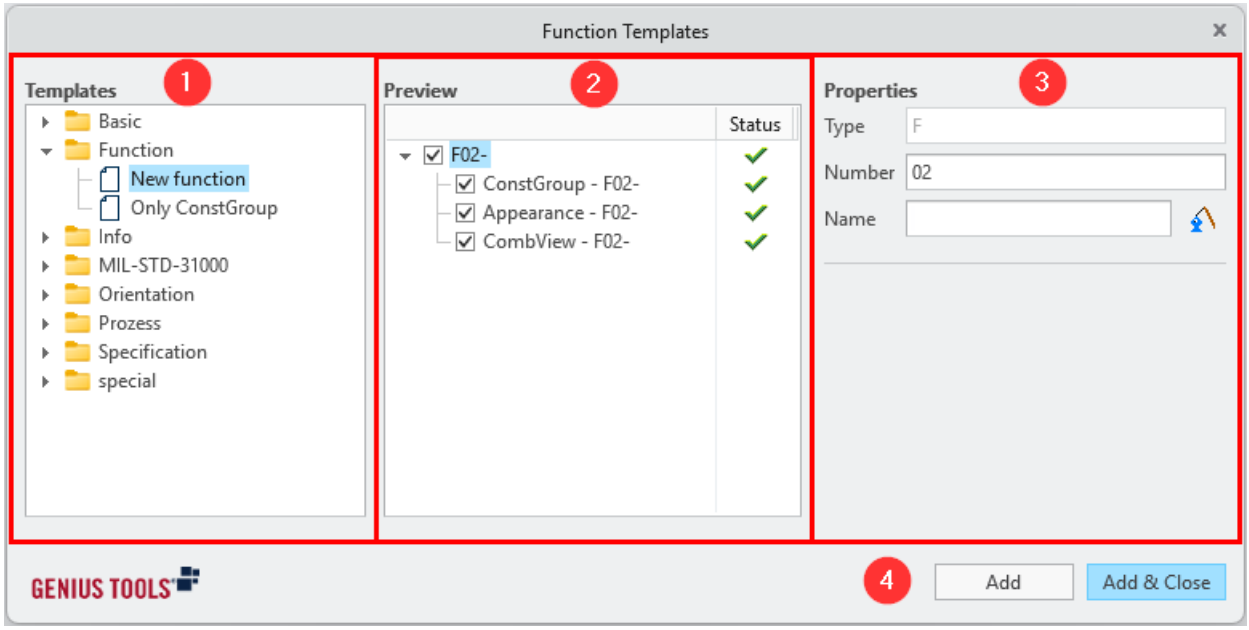
**Tip:** For better clarity in this use case, it is recommended to define one function per XML file. If multiple functions are defined, subsequent functions with additional information will not be read and cannot be selected and added under *Functional Templates*.

**Please note:** A function or functional component can be renamed if the corresponding `edit` attributes are set to `true` in the template. If such an attribute is set to `false`, the *Rename* button does not appear.

## 9.1.4 Adding functions

Clicking the button *Add* + opens the following user interface for specifying the functions to be added.


© 2026 INNEO Solutions GmbH




1. Templates for functions
   Functions are stored in the path specified in the configuration option `gtfm_template_folder` (Default: *%gt_resource_folder% function_manager\function_templates*).
   Templates are maintained as XML files.[180]
2. Detail view of the selected template with functional components and status indication
   a. Possible functional components:
      i. Construction group (assembly only)
      ii. Appearance state
      iii. Combined view[173]
      iv. Info[173]
      v. Style (assembly only)
   b. Possible status indications:

| | | |
|---|---|---|
| ✓ | green | Function / functional component can be created and added with the specified properties. |
| ✓ | yellow | The functional component was already created and can be added to the selected function. |
| ✗ | red | There is a naming conflict. Function / functional component already exists and cannot be added. |

3. Automatically filled section with information about the unique name of the function / functional component to be added.
   To change the predefined naming scheme, see Changing the naming scheme[179].
   Type, number and name are displayed for each selected function / functional


© 2026 INNEO Solutions GmbH

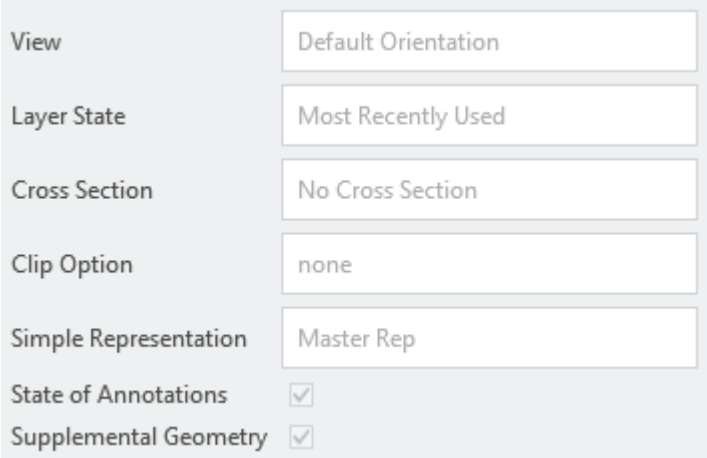
component. Further properties are displayed depending on the function / functional component (see point 2).
The *Select default text* button allows you to select the names of the combined views from a database. These can be multilingual and stored in different categories.
The configuration option `gtfm_default_text_database` stores the folder path to the default text database (default: *%GT_RESOURCE_FOLDER %/function_manager/gt_function_db.sqlite*).
The configuration option `gtfm_default_text_for_name` defines the language used when filling in function names with default texts in the Function Manager (default: key).
The configuration option `gtfm_default_text_table` defines the database table from which the default texts are used (default: selection).

4. Adding a function / Closing the user interface
   Click *Add* and the function is automatically created in the model tree. The user interface remains open for you to add more functions.
   Clicking *Add & Close* automatically creates the function in the model tree and closes the user interface.

**Tip:** You can enter UTF-16 encoded characters in all textboxes in *Function Manager*.

## 9.1.4.1 Combined view

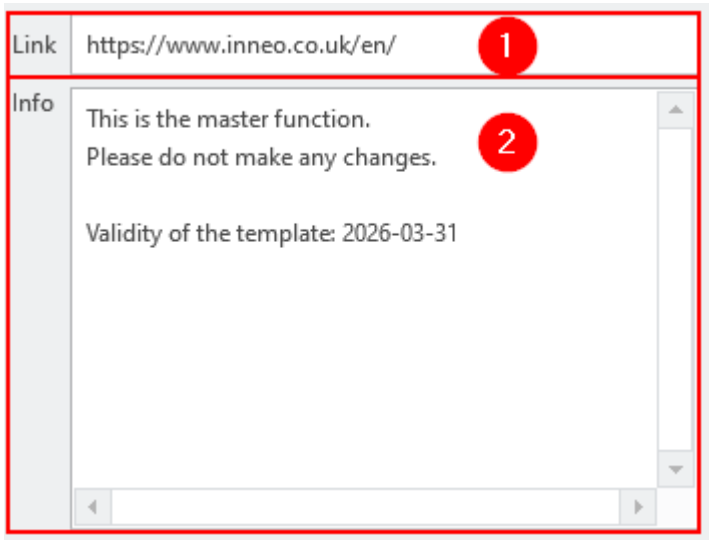
When you add a combined view as a functional component, additional information loaded from Creo *View Manager* is visible in the section *Properties*. This is for information purposes only. After creating the combined view, you can change the information displayed here in Creo *View Manager*. Alternatively, you can specify the information using *Function Manager* templates[180].




## 9.1.4.2 Links to additional information

Once you have created a function, you can subsequently enter additional information or further edit the template.


© 2026 INNEO Solutions GmbH




You can enter the following information:

1. **Link**
   Link to a website (in the format `https://www.[websitename].uk/en`) or a complete file path with a file, e.g. PDF.
2. **Info**
   Additional text can be entered, which you can only see and edit here.

## 9.1.4.3 Predefined functions

The Function Manager module contains predefined functions that can be directly used or customized. The templates are located in the resource directory in the corresponding subdirectories, e.g. *%gt_resource_folder%\function_manager\function_templates\Basic*.

The following table lists the predefined functions supplied.

| Basic functions | PRT | ASM | Description |
|---|---|---|---|
| B00-Work | ✓ | ✓ | Work region |
| B01-Master | ✓ | ✓ | Model overview |
| B02-Note | ✓ | ✓ | "Title block", notes on text information |
| B03-Datum | ✓ | ✓ | All datums |
| B04-Gt-Ref | ✓ | ✓ | All GPS specifications |
| B05-Surf-Finish | ✓ | ✓ | All surface specifications |
| B06-Design-Intent | ✓ | ✓ | Design models (skeletons) |
| B07-Weld | ✓ | ✓ | Welding specification |


© 2026 INNEO Solutions GmbH

| Basic functions | PRT | ASM | Description |
|---|---|---|---|
| B08-Ecad | ✓ | ✓ | ECAD components, wiring |
| B09-Piping | ✓ | ✓ | Piping components, piping |
| B10-Mold | ✓ | ✓ | Specific molding data |
| B11-Cast | ✓ | ✓ | Specific casting data |
| B12-MFG | ✓ | ✓ | Specific manufacturing data |
| B13-Simulation | ✓ | ✓ | Specific analysis data |
| B14-Bom | | ✓ | Bill of materials data |
| B15-Explosion | | ✓ | Exploded views |
| B16-Kinematic | | ✓ | Movable component relationships |
| B17-Assembly | | ✓ | Assembly / installation information |
| B18-Interchange | ✓ | ✓ | Interchange references |
| B19-Sheetmetal | ✓ | | Specific sheet metal information / folding |

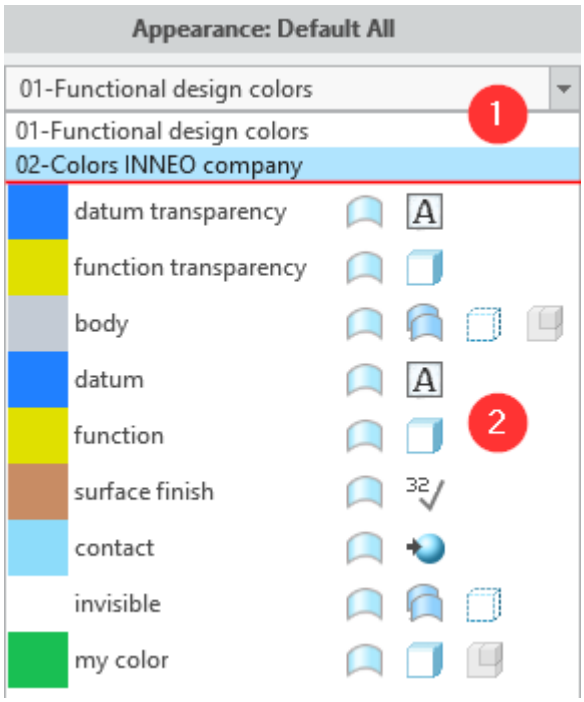
| Orientations functions | PRT | ASM | Description |
|---|---|---|---|
| O01-Front | ✓ | ✓ | Front view |
| O02-Left | ✓ | ✓ | Left view |
| O03-Top | ✓ | ✓ | Top view |

| Process functions | PRT | ASM | Description |
|---|---|---|---|
| P00-Calypso | ✓ | ✓ | |

## 9.1.5 Coloring functions

The section *Appearance* is displayed when an appearance state exists in a function. You can store several XML templates and thus several color schemes for selection.


© 2026 INNEO Solutions GmbH




### 1. Selection of the color template

In this selection field, select the color scheme that you want to apply to the function.

*GENIUS TOOLS MBD for ISO GPS* provides a predefined color scheme. The configuration option `gtfm_color_definition` specifies the relative folder path to the used color schemes. Default: `%gt_resource_folder%function_manager\color_templates\gtfm_color_definition.xml`. You can work with this scheme and customize it.

### 2. Available colors and coloring commands

The available colors and coloring commands are displayed under the color scheme selection.

These predefined colors cover the most common application areas:

| Color | Color name | Explanation |
|---|---|---|
| | body | Semi-transparent surface of a basic body |
| | datum | Datum surface of the model frame |
| | function | Functional surfaces |
| | surface finish | Surfaces with roughness data |
| | contact | Contact surfaces |

The description of the colors in the dialog box can be hidden with the configuration option `gtfm_appTable_description_visible`.


© 2026 INNEO Solutions GmbH

All coloring commands set appearances in real time. Depending on the XML file, up to nine coloring commands are available for each color:

| Icon | Part mode | Assembly mode |
|---|---|---|
|  | Selected surfaces are colored.<br>You can color multiple surfaces at once. Cancel the command with the middle mouse button. | Selected component surfaces are colored.<br>You can color multiple component surfaces at once. Cancel the command with the middle mouse button. |
|  | All surfaces are colored. | All component surfaces are colored. |
|  | All surfaces that do not yet have a color are colored. | All component surfaces that do not yet have a color are colored. |
|  | All surfaces that have a datum assigned to them are colored. | All component surfaces that have a datum assigned to them are colored. |
|  | All feature surfaces of the selected feature are colored. When a pattern or a group of features is selected, the features in the tree below are colored.<br>You can color multiple surfaces at once. Cancel the command with the middle mouse button. | All feature surfaces of the selected feature are colored. When a pattern or a group of features is selected, the features in the tree below are colored. It does not work for features created in the assembly.<br>You can color multiple surfaces one after another. Cancel the command with the middle mouse button. |
|  | *Not available* | All component surfaces of the selected component are colored. You can color multiple component surfaces one after another. Cancel the command with the middle mouse button. |
|  | All surfaces identified as contact surfaces by *Find Contact Surfaces*[709] are colored by opening and executing *Select Contact Surfaces*[734]. | *Not available* |

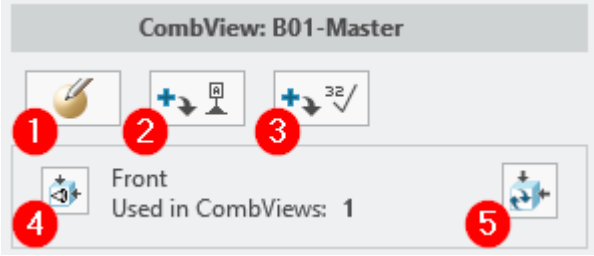

© 2026 INNEO Solutions GmbH

| Icon | Part mode | Assembly mode |
|---|---|---|
| 32 | All surfaces that have a surface symbol assigned to them are colored. | All component surfaces that have a surface symbol assigned to them are colored. |

**Tip:** Use the utility *Select Surfaces by Color*[737] to select all surfaces of a color. You can then use the *Function Manager* to color these selected surfaces at once.

## 9.1.6 Editing combined views

In this section of the detail view, you can redefine combined views via a direct link to the Creo *View Manager*.




1. **Redefining the definition**

   The Creo dialog *Edit definition* opens.

2. **Transfer all datum tags**

   All datum tags, including invisible ones, are transferred to the active combined view and made visible.

3. **Transfer all surface finishes**

   All surface finishes are transferred and visualized.

The buttons *Transfer all datum tags* and *Transfer all surface finishes* operate according to the same principle as *Annotation Transfer*[649], i.e. groups of annotations are fully transferred.

**Tip:** The function *Annotation Transfer*[649] can be used to manually transfer datums tags, surface finishes and other annotations.

4. **Orientation Manager**

   Open the Orientation Manager.

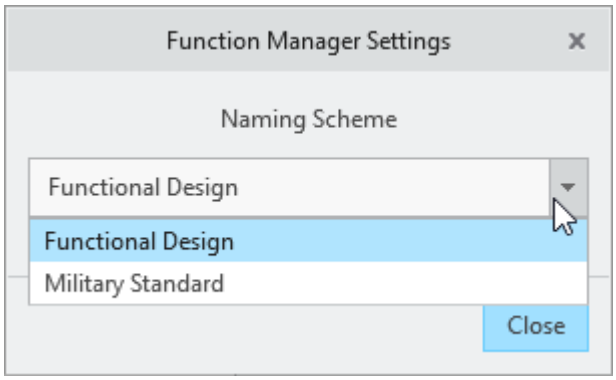
5. **Update the assigned orientation of the active CombView**

   Update the orientation of the combined view.


© 2026 INNEO Solutions GmbH

## 9.1.7 Changing the naming scheme

The function *Change naming scheme* allows you to choose between two variants for assigning names to newly created functions:




The naming schemes are structured as follows:

| Naming scheme | Structure |
|---|---|
| Functional Design | `TypeNumber-Name`<br>– `Type`: generated capital letter (not editable)<br>– `Number`: generated two-digit number (not editable)<br>– `Name`: enter any function name; the function can also be created without a function name |
| Military Standard | `TypeLetter_Name`<br>– `Type`: generated capital letter (not editable)<br>– `Letter`: generated lowercase letter (not editable)<br>– `Name`: enter any function name; the function can also be created without a function name |

Use the configuration option `gtfm_naming_scheme` to choose the preset naming scheme. Default: 1 (=Functional Design)

**Please note:** You must first delete all functions before you can change the naming scheme. The naming scheme cannot be changed later.

**Please note:** If a functional component is not deleted, it is listed under *Unassigned Comb. Views*. This means it can be reassigned to a function without being recreated.


© 2026 INNEO Solutions GmbH

# 9.2 Configuration

You can work with **function templates**[180] in which you specify properties for functions to be created.

The configuration option *gtfm_template_folder* defines the path to the directory that contains the templates. Default setting*: %gt_resource_folder% function_manager\function_templates*.

The template file that opens when the *Add* dialog is opened is defined in the configuration option *gtfm_init_template_node_selection*. Default: *Functions\B01-Master.xml*.

The function templates directory contains predefined functions[174] which you can customize.

You can also work with **color templates**[183]. You can use the supplied templates for this or create your own.

The configuration option *gtfm_color_folder* defines the path for the color tables. Default setting: *%gt_resource_folder%function_manager\color_templates*

## 9.2.1 Editing function templates

The templates are files in XML format that are edited with an editor, e. g. Notepad. The file name under which the template is saved is the name under which the template is displayed in the *Function Manager* user interface.

**Please note:** Use tags to apply the necessary formatting to the XML files. Formatting errors mean that an XML template cannot be read and therefore cannot be used.

Help for customizing XML templates:

| Inserting tags | Purpose | Formatting | Formatting example |
|---|---|---|---|
| – with start and end formatting | Adding an element with detailed description | `<description>`<br>`[content]`<br>`</description>` | `<de>function name</de>` |
| – as one-liner | Adding an element so that it can be added to a function via checkbox | `<element to be added />` | `<constgroup />` |

The illustration shows an example of a function template. This example is used to explain the different sections of a template.


© 2026 INNEO Solutions GmbH




1. `<gtfm>...</gtfm>`
   Frame for the template: This XML file shows a template in which one function has been defined. You can define several functions in one XML file.
   - `version`=Specification of a number for versioning the template
   - `name`=Name of the template
2. Information about the function[181]
3. Functional components[182]

# Information about the function

Each function requires the following tags:

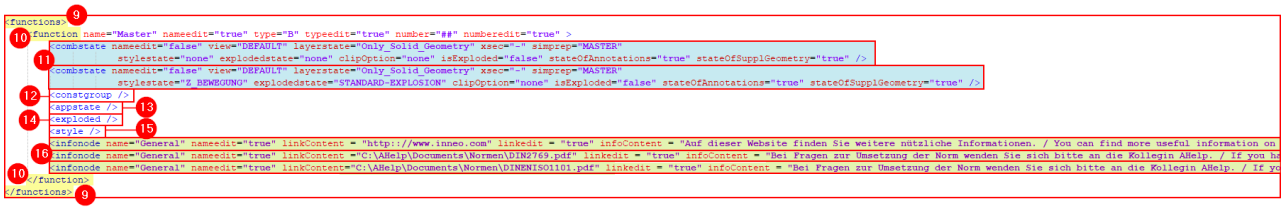



4. `<info>... </info>`
   Tags to frame the information section
5. `<title>... </title>`
   Title in English `<en>` and German `<de>`
6. `<description>... </description>`
   Description text in English `<en>` and German `<de>` (optional)
7. *Currently not used*
8. *Currently not used*


© 2026 INNEO Solutions GmbH

# Functional components

The following functional components can be added to a function:




**Please note:** A function or functional component can be renamed if the corresponding `edit` attributes are set to `true` in the template. If such an attribute is set to `false`, the *Rename* button does not appear.

9. `<functions>...</functions>`
   Frame for the part that contains one or more functions.

10. `<function name...</function>`
    Frame for the function. Enter the properties of the function in quotes:
    - `name`=function name
    - `nameedit`=function name can be edited in the *Function Manager* user interface yes/no (`true/false`)
    - `type`=function type
    - `typeedit`=function type can be edited in the Function Manager user interface yes/no (`true/false`)
    - `number`=type of numbering, one hashtag is entered per digit, i.e. `##` starts with `01`
    - `numberedit`=numbering can be edited in the Function Manager user interface yes/no (`true/false`)

11. `<combstate/>`
    Defining a combined view with properties. You can create several combined views. Enter the properties of the combined view in the quotation marks:
    - `nameedit`=name can be edited in the Function Manager user interface yes/no (`true/false`)
    - `view`=view
    - `layerstate`=layer state
    - `xsec`=cross section
    - `simp`=simple representation
    - `stylestate`=style
    - `explodestate`=exploded view


*© 2026 INNEO Solutions GmbH*

- `clipOption=`clip option
- `isExploded=`exploded view applied yes/no (`true/false`)
- `stateOfAnnotations=`annotations yes/no (`true/false`)
- `stateOfSupplGeometry=`supplementary geometry yes/no (`true/false`)

**Tip:** For better clarity in this use case, it is recommended to define one function per XML file. If multiple functions are defined, subsequent functions with combined views will not be read and cannot be selected and added under *Functional Templates*.

12. `<constgroup />`
    Creating a construction group. If a function template should not contain a construction group, delete this line.
13. `<appstate template="Name"/>`
    Adding an appearance state. If a function template should not contain an appearance state, delete this line.
14. `<exploded />`
    Adding an exploded view. If a function template should not contain an exploded view, delete this line.
15. `<style />`
    Adding a style. If a function template should not contain a style, delete this line.
16. `<infonode/>`
    Adding an info node. You can create several info nodes.
    - `name=`name of the info node
    - `nameedit=`name of the info node can be edited in the *Function Manager* user interface yes/no (`true/false`)
    - `linkContent=l`ink to a website or to a file with file path
    - `infoContent=`additional information text to expand / explain the link (optional)
    - `infoedit=`info node can be edited in the *Function Manager* user interface yes/no (`true/false`)

**Tip:** For better clarity in this use case, it is recommended to define one function per XML file. If multiple functions are defined, subsequent functions with additional information will not be read and cannot be selected and added under *Functional Templates*.

## 9.2.2 Editing color templates

The templates are files in XML format that are edited with an editor, e. g. Notepad. The file name under which the template is saved is the name under which the template is displayed


© 2026 INNEO Solutions GmbH

in the *Function Manager* user interface.

**Please note:** Use tags to apply the necessary formatting to the XML files. Formatting errors mean that an XML template cannot be read and therefore cannot be used.

Help for customizing XML templates:

| Inserting tags | Purpose | Formatting | Formatting example |
|---|---|---|---|
| – with start and end formatting | Adding an element with detailed description | `<description>` `[content]` `</description>` | `<de>function name</de>` |
| – as one-liner | Adding an element so that it can be added to a function via checkbox | `<element to be added />` | `<constgroup />` |

The illustration shows an example of a template for a color table. This example is used to explain the different sections of a template.




1. `<gtfmColor>...</gtfmColor>`
   Frame for the template: This XML file shows a defined color template. You can define any number of colors in an XML file.
   - `version`=Specification of a number for versioning the template

**Tip:** It is at the user's discretion how many colors need to be entered and maintained in an XML template. Clarity should be preserved.


*© 2026 INNEO Solutions GmbH*

2. `<info>... </info>`
   Tags to frame the information section
   - `<title>... </title>`
     Title in English `<en>` and German `<de>`
3. Color definitions[185]

# Color definitions

Color definitions can be created with the following properties:




4. `<color_definitions>...</color_definitions>`
   Frame for the color definitions: All colors and associated commands are defined within this frame.
5. `<color>...</color>`
   Definition of the description of the color. Enter the properties of the color definition in quotes:
   - `color name=`*currently not used*
   - `id=`*currently not used*
   - `tooltip=`*currently not used*
   - `description=`display name of the color in the color table of Function Manager
   - `keywords=`*currently not used*
6. `<appearance_data>...</appearance_data>`
   Definition of a color value of the class *Generic* with all color values from the Creo *Model Appearance Editor* (value between 0.00 and 1.00).
   - `color_rgb chosen_color=`color as RGB value


© 2026 INNEO Solutions GmbH

- `intensity value=`intensity
- `ambient value=`ambient
- `color_highlight chosen_color=`highlight color as RGB value
- `shine value=`shine
- `highlight value=`highlight
- `reflection value=`reflection
- `transparency value=`transparency

7. `<actions>...</actions>`
   Definition of the coloring commands that are available for this color, see Coloring functions[175].
   Possible action types in the user interface of *Function Manager*:

- `surface=` 
- `allSurfaces=` 
- `colorlessSurfaces=` 
- `datumSurfaces=` 
- `feature=` 
- `component=` 
- `contactSurfaces=` 
- `surFinSurfaces=` 

8. Entries 5-7 must be made for each color to appear in the color table.


© 2026 INNEO Solutions GmbH

# 10 Inspect

Available in the ribbon menu GENIUS TOOLS

Use *Inspect* to add inspection symbols and inspection symbol tables to drawings in Creo Parametric.

*Inspect* is available in drawing mode with the following features.

1. Free placement of inspection symbols
2. Placement of inspection symbols linked to
   - dimensions
   - shape and position tolerances
   - surface quality symbols
   - notes
   - symbols
3. Free placement of inspection symbol tables
   - numbering tables (overview of the types of inspection symbols and the numbers used)
   - report tables (overview of all properties of inspection symbols)
4. Numbering of inspection symbols
   - by height,
   - by symbol type
   - similar to DIN 6770 (numbers are not assigned anew)
5. Data export to Excel

**Warning:** To use *Inspect*, the configuration `PRO_SYMBOL_DIR` must be editable. However, if `PRO_SYMBOL_DIR` is included in the Creo file *config.sup*, `PRO_SYMBOL_DIR` cannot be edited. In this case, *Inspect* will not work.

The component *Inspect Revision*[231] creates a snapshot of all inspection symbols on a drawing at one point in time, as well as a history of all snapshots.

You can also use symbols in *Inspect* to mark changes on a drawing[226].


© 2026 INNEO Solutions GmbH

# 10.1 Fundamentals

## Glossary

**Inspection symbols**
Numbered symbols marking characteristics that have to be considered in quality control because they are critical to the quality of function of the finished part or product.

**Inspection symbols table**
Overview of all inspection symbols used in a drawing.

**Inspect configuration**
XML file that contains definitions for symbols, tables and display settings. (See also Inspect configuration 207.)

## Structure of an inspection symbol

The figure below shows an inspection symbol generated with *Inspect* with its individual components.

Each inspection symbol (numbered symbol) marks a characteristic in a technical drawing.

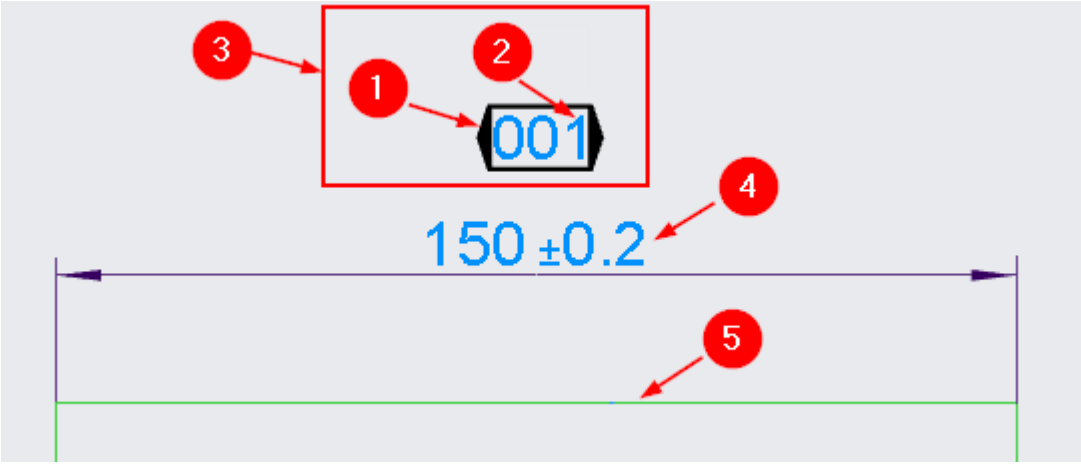
1) symbol

2) number of the characteristic

3) inspection symbol, consisting of the symbol (1) and the number of the characteristic (2)

4) value of the characteristic

5) characteristic: length of a geometry element




## Font used for out-of-the-box symbols

The symbols delivered with *Inspect* use the font *ISONORM LT Regular* based on ISO 3098-5.


© 2026 INNEO Solutions GmbH

If you experience issues with displaying the out-of-the-box symbols, please check whether this font is available in your system.

You can create your own symbol files to use as inspection symbols. For more information on how to do this, please refer to Creating a Creo symbol for Inspect 220.

## Integrating additional texts according to DIN EN ISO 14405-1

You can use the Creo menu *Drawing Properties* > *Detailing* to set that symbols can also contain texts in accordance with DIN EN ISO 14405-1. The following settings are required for this:

| Option | Value |
|---|---|
| default_annotation_font | isonormlt-regular.ttf |
| symbol_font | ISO |

**Tip:** The detail option symbol_font = legacy is not supported.

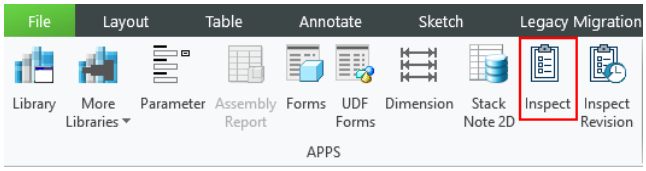
# 10.2 Usage

This section contains information on using *Inspect*. It describes the general structure of the program.

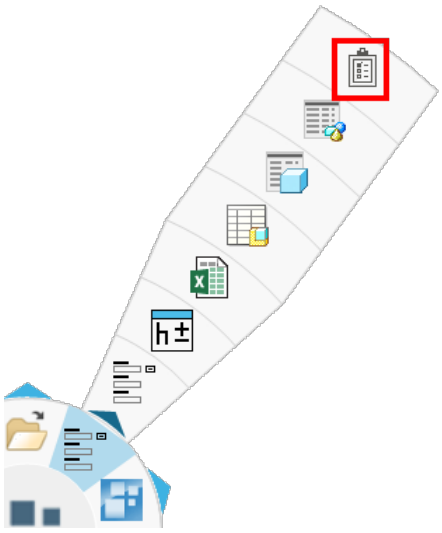
## Starting the program: in drawing mode

To start *Inspect* , click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section by using *Quick Access* (key [<]).


© 2026 INNEO Solutions GmbH



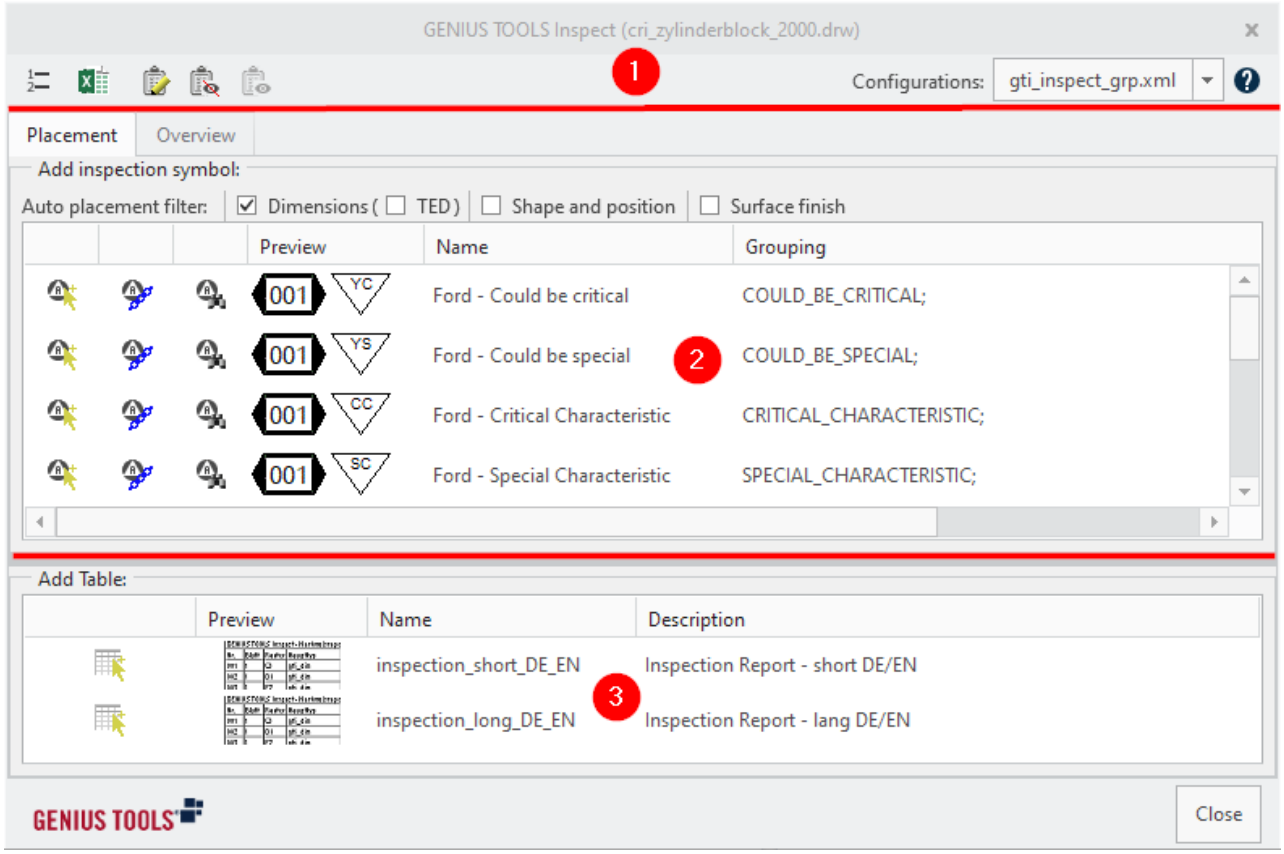

*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

## 10.2.1 User interface

The user interface of *Inspect* consists the following elements:




1. Command bar[191]
2. Symbol overview: divided in two tabs, Placement and overview[192]
3. Table overview[192]


© 2026 INNEO Solutions GmbH

## 10.2.2 Command bar

The command bar displays general control elements. The following buttons are included:

| Icon | Name | Description |
|---|---|---|
|  | Number symbols | Numbers all placed symbols. The numbering is determined by configuration options. |
|  | Export overview to Excel | Opens the export dialog for the symbol table[202]. |
|  | Hide Inspect Elements | Hide all Inspect elements. **Warning:** Linked symbols are unlinked from their targets. This does not make them free symbols, see also Hide and unhide[201]. |
|  | Unhide Inspect Elements | Unhide all Inspect elements. **Warning:** Linked symbols are re-linked to their targets. If the target for a symbol cannot be found in the drawing, the symbol is deleted. |
|  | Open Inspect Editor | Opens the *Editor*[206]. |
|   | Configuration selection | Switches between multiple Inspect configurations that can contain different symbol and table definitions as well as view settings. |
|  | Open Help | Opens the help. |

**Please note:** Always place[192] all symbols before numbering them.

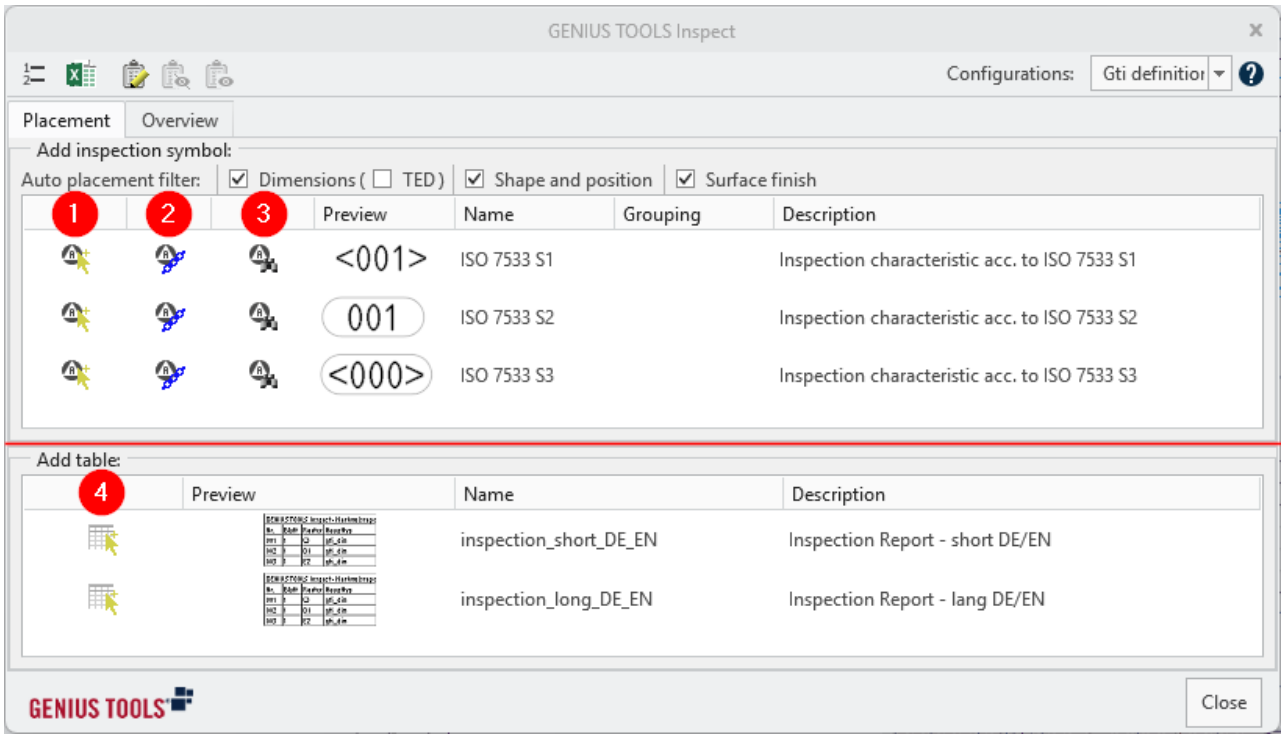
The numbering of symbols is influenced by the configuration options[825] `gti_number_sort_at_height`, `gti_number_sort_at_type`, `gti_start_number` and `gti_din_compliant`, see Numbering[196].

**Warning:** Do not use the same symbol in different Inspect configurations if you are using numbering similar to DIN 6770. Otherwise, you may experience numbering conflicts.


© 2026 INNEO Solutions GmbH

## 10.2.3 Placing

The tab *Placement* is used to place and link inspection symbols and tables on drawings. The columns displayed, the order of the columns, and the order of the inspection symbols are defined in the *Editor*[205].




Available inspection symbols are displayed under *Add inspection symbol*. Under *Auto Placement Filter*, select the inspection symbols to be added.

**Please note:** *Auto placement filter*




can only be used for **linked insertion with previous selection** or **automatic placement**.

Each inspection symbol can be added to a drawing as an independent element or linked to an object. Click on the respective icon to add inspection symbols to a drawing.

### Inspection symbols

Use *Auto Placement Filter* to specify which inspection characteristics to add.

**Free insertion (1)** : Press the button for free placement of an inspection symbol and then select its position on the drawing with the left mouse button. You can place more than one characteristic. Press the middle mouse button to cancel the placement.

**Linked insertion (2)** : Inspection symbols can be linked to dimensions, shape and position tolerance symbols, surface quality symbols, notes and drawing symbols displayed


© 2026 INNEO Solutions GmbH

on drawings. These inspection symbols will be attached to the drawing item with equal alignment.

– **Simple linked placement:**
  Click the button , then select a drawing item using the left mouse button. Now place the inspection symbol on the drawing using the left mouse button. You can position several symbols, then stop the process by clicking the middle mouse button.
  Use the configuration option `gti_activate_simple_linked_auto_placement` to control whether the symbol is automatically placed (*Automatic insertion* placement). Default: 1 (=yes)
– **Linked placement by previous selection (single elements):**
  Select one or more elements in the drawing. Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. You can then insert linked inspection symbols by clicking the button . These inspection symbols will be attached to the drawing item with equal alignment.
– **Linked placement by previous selection (multiple elements):**
  Select multiple elements or an area in the drawing. Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. Then you can insert linked inspection symbols by pressing the button . Inspection symbols will be linked to all dimensions in the selection. These inspection symbols will be attached to the drawing item with equal alignment.

**Automatic insertion (3)** : Inspection symbols are automatically placed on all unlinked drawing items (dimensions, shape and position, surface). Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. Items which are already linked to the selected inspection symbol are not re-linked. *Inspect* checks whether the SYM file of the selected inspection symbol matches the existing inspection symbol. These inspection symbols will be attached to the drawing item with equal alignment.

No inspection characteristics are created for TEDs.

**Please note:** Automatic insertion applies inspection symbols only to displayed dimensions.

**Please note:** If some inspection symbols were not placed during linked or automatic insertion, check if a placing exception rule[194] has been defined and delete it.

## Tables

**Free insertion (4)** **:** Available tables for inspection symbols are displayed under *Add Table*. Click on the placement icon and place the table freely on the drawing. There are two types of tables:

– **Numbering table:** Overview of the types of inspection symbols and the numbers used.
– **Report table:** Overview of all properties of the inspection symbols.


© 2026 INNEO Solutions GmbH

Tables are created in the *Editor*[205].

**Please note:** If not all types of inspection symbols were added during insertion, check that the inspection symbols are checked under *Auto Placement Filter*.

In drawing tables, you can manually link inspection symbols to drawing table cells.

### 10.2.3.1 Placing with exceptions

You can define exceptions for **placement by previous selection**[193] and **automatic placement**[193] to exclude target elements with certain phrases from placement. The exceptions are defined in the configuration option `gti_placement_exception_rule`. The configuration option has no default setting, so by default, no target elements are excluded from placement. You can enter fixed formulations, Creo special characters or Regular Expressions[487].

**Please note:** First define the exception in the configuration option `gti_placement_exception_rule`. Then place the inspection symbols. Inspection symbols already placed are not removed by this configuration option, but must be deleted manually.

## Use cases

Possible phrases to exclude target elements from the positioning:

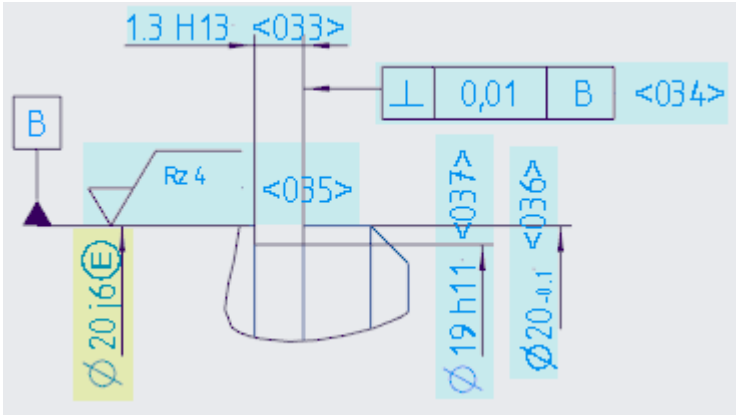
**Excluding special characters from placement**

– Setting with color coding: Ⓔ

– Explanation of the setting:
Ⓔ was entered as a Creo special character. The special characters must be copied and pasted from Creo Parametric and inserted into *Configuration Utility*[765]. The special characters are usually not resolved, but the filter works.
Entered special characters are displayed correctly in the configuration overview *GENIUS TOOLS* for Creo Settings[763].


© 2026 INNEO Solutions GmbH

– Result in the user interface:



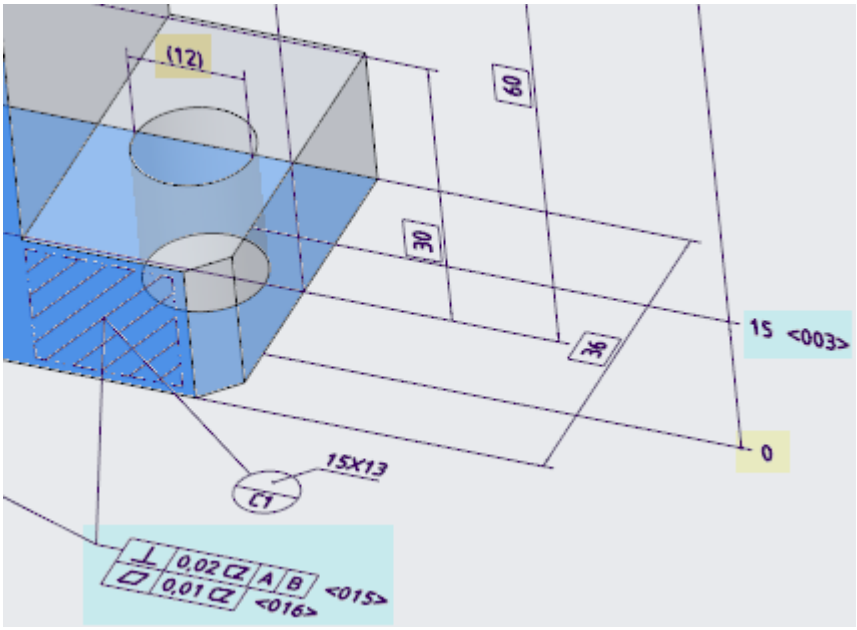

*Excerpt from gti-antriebswelle.drw*
*green=target element with not set inspection characteristic*
*blue=target element with set inspection characteristic*

**Excluding clamp dimensions or dimensions (incl. ordinate dimensions) with the value 0 from placement**

– Setting with color coding: ^\(.+\)$|^0$

– Explanation of the setting:
The exception was entered as Regular Expression[487] with 'or' operator. One of two exceptions is applied:
Either no inspection characteristics are placed on target elements with clamps, or no inspection characteristics are placed on target elements with a value of 0.

– Result in the user interface:




*Excerpt from lagerbock-mbd-gps-v4-col, B01-Master*
*orange=target element with inspection characteristic not set due to dimensioning in clamps*


© 2026 INNEO Solutions GmbH

*yellow=target element with inspection characteristic not set due to ordinate dimensioning 0*
*blue=target element with set inspection characteristic*

## 10.2.4 Numbering

The following section describes the numbering options. The numbering of the inspection symbols is influenced by the following configuration options:

| Configuration option | Input value | | |
|---|---|---|---|
| gti_number_sort_at_height | -1 (descending) | 0 (no sorting) | 1 (ascending) |
| gti_number_sort_at_type | -1 (descending) | 0 (no sorting) | 1 (ascending) |
| gti_start_number | Numerical value for numbering start (used only when gti_din_compliant=0 is set) | | |
| gti_din_compliant | 0 (off) | 1 (on) | |
| gti_numbering_all_sheets | 0 (off) | 1 (on) | |
| gti_number_range_per_sheet | 0 (off) | 1 (on) | |

### Numbering per sheet or drawing

With the configuration option gti_numbering_all_sheets you can control whether symbols are sorted and numbered per sheet (gti_numbering_all_sheets=0) or globally over the entire drawing (gti_numbering_all_sheets=1). Whether in the case of gti_numbering_all_sheets = 0 a separate number range or a consecutive numbering should be used is controlled with the configuration option gti_number_range_per_sheet.

**Please note:** Numbering with these settings is very time consuming.

### Number ranges

With the configuration option gti_number_range_per_sheet you can control whether each sheet (gti_number_range_per_sheet=1) has its own number range. Otherwise (gti_number_range_per_sheet=0) one number range is used for the whole drawing.


© 2026 INNEO Solutions GmbH

**Warning:** This configuration option will only be read if the options `gti_numbering_all_sheets` is set to `0` or `gti_din_compliant` is set to `1`.

## Numbering similar to DIN 6770

The configuration option `gti_din_compliant` determines whether numbering is similar to DIN 6770.

To be able to determine the next number, the highest number assigned is stored in the parameter GTI_LAST_SYM_<*configuration name*><*sheet index*>. If `gti_number_range_per_sheet` is set to `1`, the highest number per sheet is remembered.

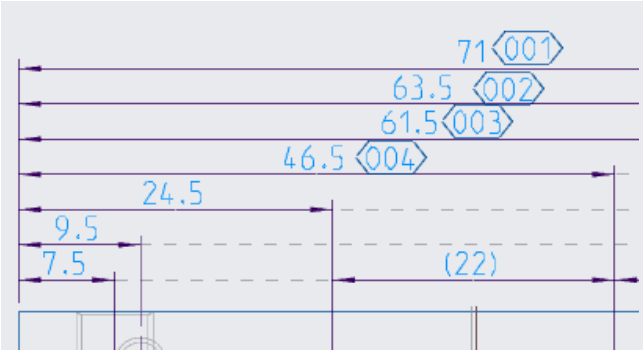
Only inspection features with the number 000 are re-numbered. 000 is the default value for newly placed inspection symbols.

The configurations `gti_numbering_all_sheets`, `gti_number_sort_at_height` and `gti_number_sort_at_type` and are not read out when using DIN 6770 (`gti_din_compliant=1`) numbering. It is automatically sorted by height *from top to bottom* (`gti_number_sort_at_height=-1`) and over each sheet (`gti_numbering_all_sheets=0`).

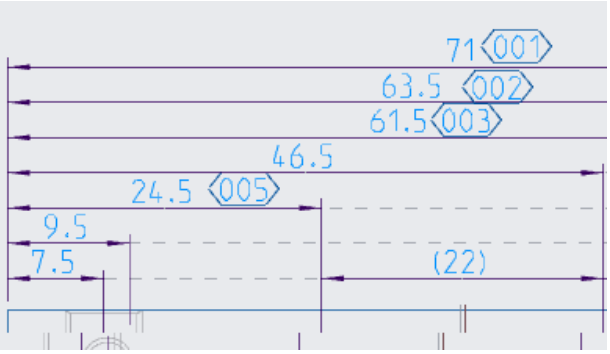
If you do not set the option, i. e. if `gti_din_compliant=0`, values from `gti_numbering_all_sheets`, `gti_number_sort_at_height` and `gti_number_sort_at_type` are read.

**Example**

In the left figure, four new inspection symbols have been placed and numbered consecutively. If the inspection symbol with the number 004 is deleted, the number 004 remains stored in the parameter `GTI_LAST_SYMBOL`. In the right figure, a new inspection symbol has been placed. When the inspection symbols are re-numbered, the new symbol is numbered 005, not 004.




*Placement of the inspection features and subsequent numbering with 001 - 004.*



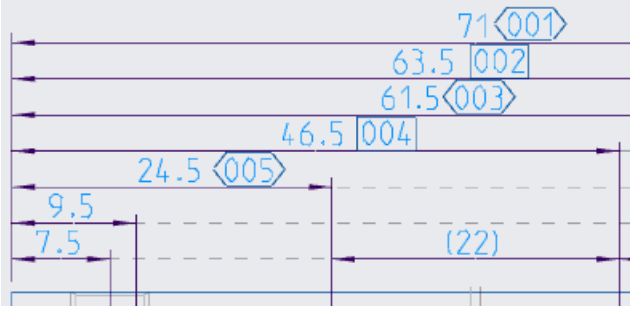

*Placement of a new inspection feature and subsequent numbering with 005.*

## Numbering order

Numbering order is determined by the configuration options `gti_number_sort_at_height` and `gti_number_sort_at_type`.

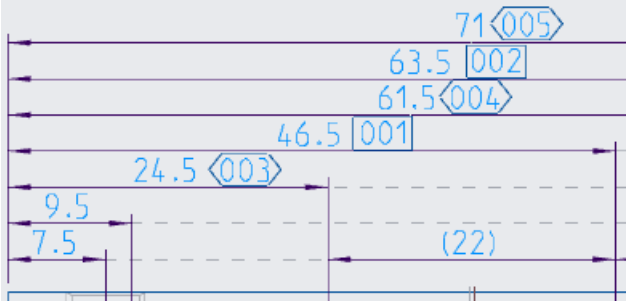

© 2026 INNEO Solutions GmbH

The option `gti_number_sort_at_height` specifies whether the inspection symbols should be numbered by height, from *top to bottom* (`gti_number_sort_at_height=-1`) or from *bottom to top* (`gti_number_sort_at_height=1`).



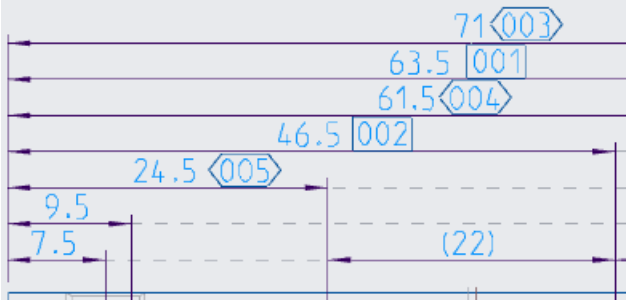

*Numbering by height.*

The option `gti_number_sort_at_type` specifies whether the inspections symbols are to be numbered by inspection symbol type, should be numbered alphabetically *descending* (`gti_number_sort_at_type=-1`) or *ascending* (`gti_number_sort_at_type=1`).




*Numbering by type.*

If both numbering options are active at the same time, numbering is first by inspection symbol type and then by height within a type.

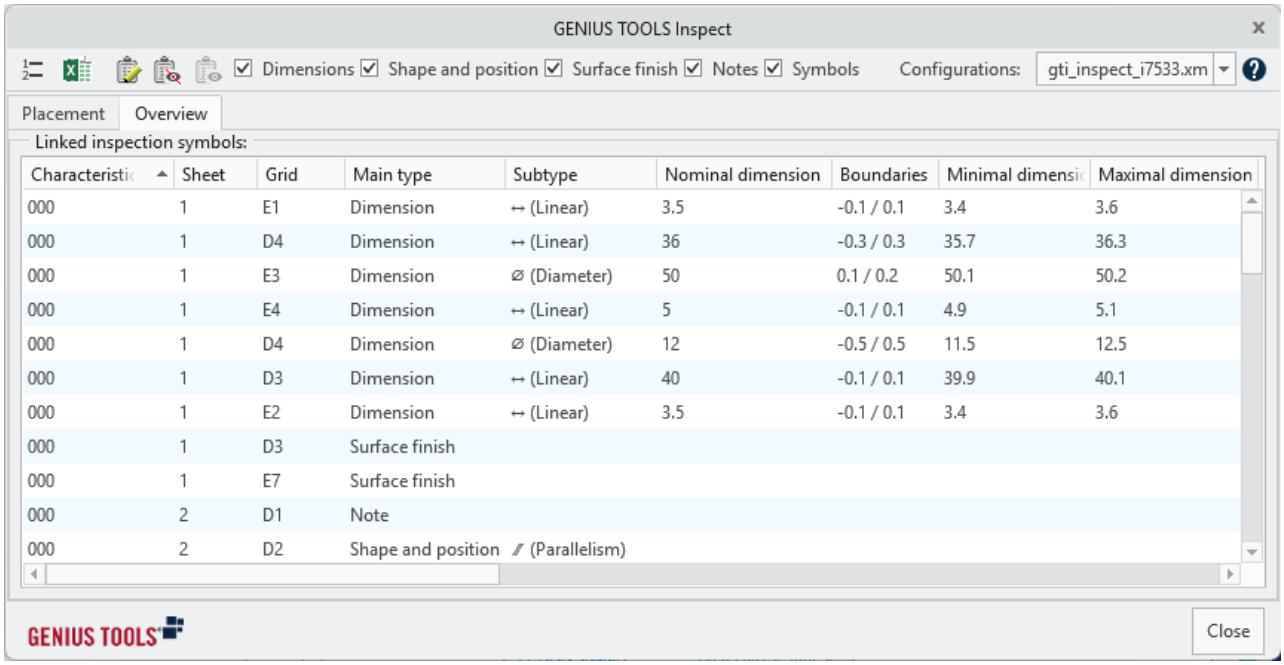



*Numbering by type and height.*

## 10.2.5 Overview

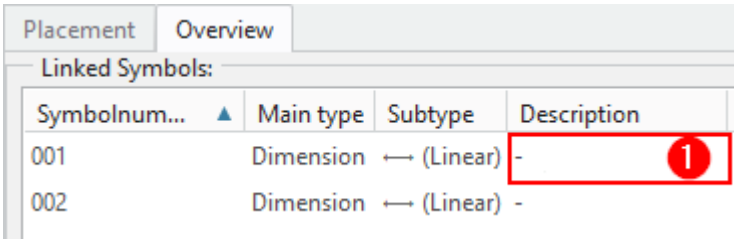
The *Overview* tab displays all previously placed inspection symbols and relevant data for these inspection symbols in a table. Additional information (e.g. tolerance information) is displayed for linked inspection symbols. The columns displayed, the column order, and the order of the inspection symbols are defined in the *Editor*[205].


© 2026 INNEO Solutions GmbH




### Linked inspection symbols

The out-of-the-box inspection symbols include a description parameter. Its content is displayed in the *Description* column and can be changed. A left click in the description column opens an input dialog for entering a description (1). This description is saved in the inspection symbol and will be available the next time *Inspect* is started.




The description field is variable text[221], which is saved in a Creo symbol.

**Tip**: When you are in an input field, you can use *Tab* to move to the next input line.

You can add information (e.g. from a revision parameter) by creating a new column. In *Inspect Editor* go to the tab *Inspection symbol > Parameters*[208].

### Linking free inspection symbols

Right-click on an independently-placed inspection symbol. Then select the context menu item *Link element* and select a drawing item to link to.

**Please note:** Use the configuration option `gti_lang` to define - not depending on the system language - the language used for the contents of the overview table.


© 2026 INNEO Solutions GmbH

## Filtering inspection symbols

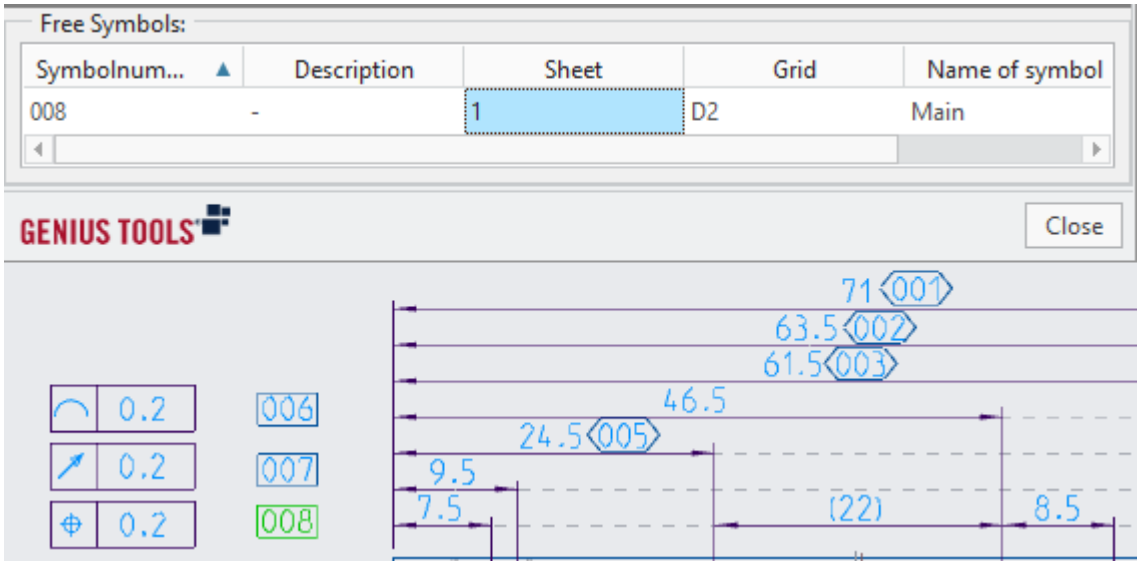
Use the filters in the top bar to show and hide inspection symbols linked to dimensions, shape and position tolerances, surface finishes, notes or symbols. If you hide inspections symbols that are linked to a defined type of drawing element (e. g. dimensions or tolerances), columns that are only relevant for this type will also be hidden.

## Sorting inspection symbols

By clicking on a column header, the inspection symbols are sorted according to this column. By default, the symbols are sorted by inspection symbol number (ascending).

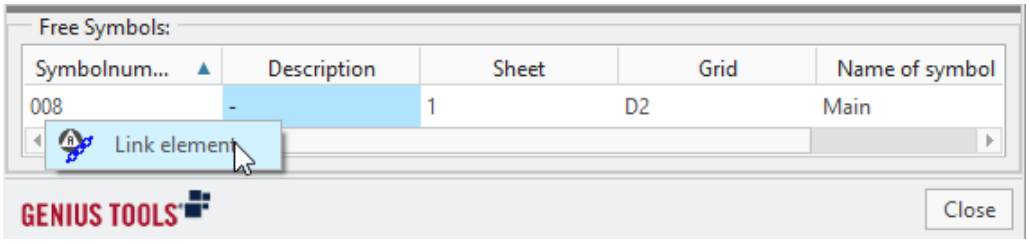
## Highlighting inspection symbols

A click on any cell in an inspection symbol's row highlights the inspection symbol in the drawing.




## Linking free symbols

You can link inspection symbols to drawing elements later. To do this, right-click the inspection symbol to be linked in the *Free symbols* table. Now click *Link element* and select the corresponding drawing element with the left mouse button.





© 2026 INNEO Solutions GmbH

## 10.2.6 Hide and unhide

### Hiding Inspect elements




To hide all *Inspect* elements, click the button shown above (1).

All *Inspect* tables and *Inspect* inspection symbols that contain the variable text `LinkedTo` are hidden.

Symbols without the variable text `LinkedTo` are not hidden. This can be the case for inspection symbols created with version of *Inspect* prior to 6.0. The variable text has to be added to these symbols if you want to use the hide functionality.

When hiding *Inspect* elements, linked inspection symbols are unlinked from their target elements. This does not turn them into free symbols – *Inspect* saves the attributes of the target element to the variable text `LinkedTo` to be able to re-link the inspection symbols when they are unhidden.

### Unhiding Inspect elements




To unhide all *Inspect* elements, click the button shown above (2).

All inspection symbols which were linked before hiding are re-linked to their target elements.

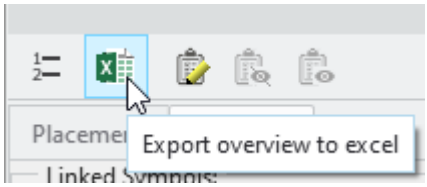
**Warning**: Inspection symbols whose target elements have been deleted will be deleted when unhiding!


© 2026 INNEO Solutions GmbH

## 10.2.7 Export to Excel

### Exporting inspection symbols to Excel

To export your inspection symbols to Excel, click on the command *Export overview to Excel* . The dialog Export table to Excel[690] opens, in which you select the Excel template, export file and configuration.




Starting the Excel export can take some time because the Excel template has to be checked. Once the export is complete, the Excel file will open automatically.

**Warning:** Close all Excel windows before starting the export.

### Export dialog

By default, the out-of-the-box template is selected. The file name is generated from the name of the drawing. You can also design an individual Excel template[693].

The selected configuration determines which inspection symbols are exported. Inspection symbols will be exported if the selected configuration refers to their symbol file, i.e. if the configuration contains their inspection symbol type.




### Configuring path to template

The default selection of a template can be changed by using these options:

**gti_excel_template**
Defines the Excel template. Default: *gti_inspection_template_de_en.xlsx*

**gti_folder**
Defines the directory, which contains the Excel template. Default: <*gt_resource_folder*>

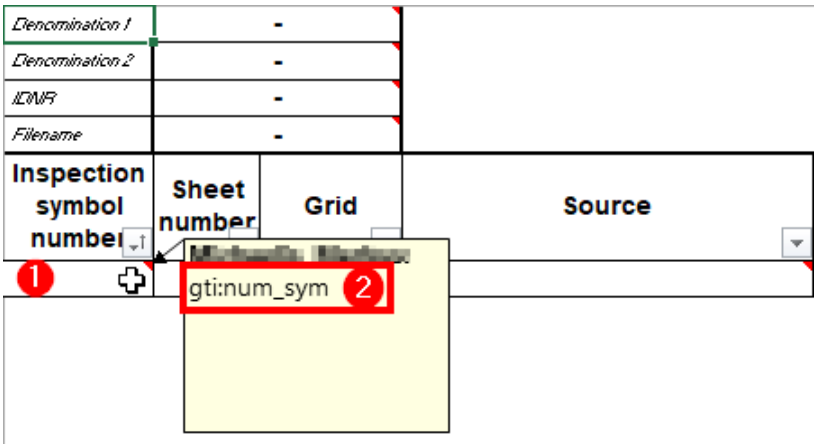
### Customizing the Excel template

If you want to create your own template, you can use the supplied template *gti_inspection_template_de_en.xlsx* from the directory *%gt_resource_folder%\inspect* as a


© 2026 INNEO Solutions GmbH

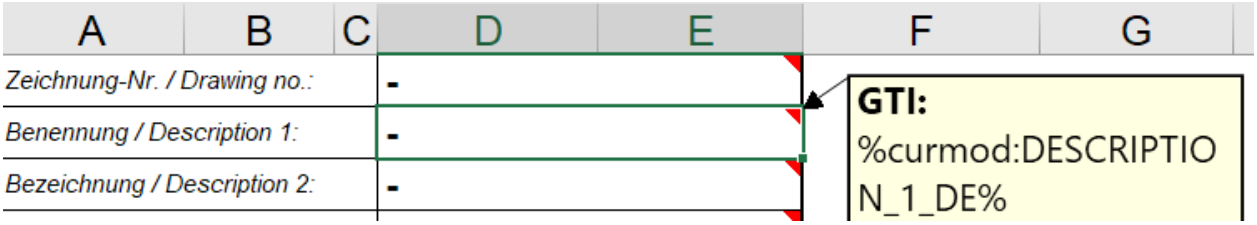
basis and add a comment (2) to the first value cell (1) of the desired column. The chapter Create template[693] describes how to set up an export template step by step.

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.



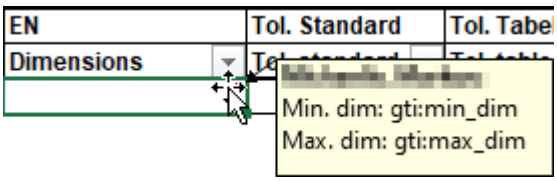

*Comment used for column assignment*

The text of the comment determines which values are to be copied into the column. Head parameters allow variables without module abbreviations, e.g. *%DRAWING_NO%, % curmod:DESCRIPTION_1_DE%*.

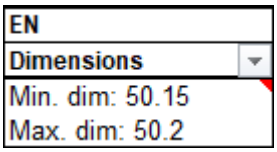



*Head parameter for Inspect*

During table export, all tags in a comment are searched and replaced with the appropriate value. The remaining text in the comment is preserved.




*Comment on the filling of the column with tags and additional text to be displayed.*




*Display in the exported table*


© 2026 INNEO Solutions GmbH

For report parameters, the text in the comment must consist of the component abbreviation *gti:* and a keyword:

| Comment text | Column name |
|---|---|
| gti:<columnName> | Output of additional user-defined parameters |
| gti:gtol_bottom_text | Bottom text |
| gti:val_tol | Boundaries |
| gti:tpe_sym | Creo symbol |
| gti:gtol_datum_references | Datum references |
| gti:descr | Description |
| gti:dim_value_text | Dimension text |
| gti:grd | Grid |
| gti:ipc_dim | Inspection dimension |
| gti:gtol_left_text | Left text |
| gti:low_tol | Lower boundary |
| gti:tpe_main | Main type |
| gti:max_dim | Maximal dimension |
| gti:min_dim | Minimal dimension |
| gti:nme_sym | Name of symbol |
| gti:bse_dim | Nominal dimension |
| gti:gti_note | Note |
| gti:gti_param | Parameter |
| gti:gtol_right_text | Right Text |
| gti:num_sheet | Sheet |
| gti:src | Source |
| gti:tpe_sub | Subtype |
| gti:num_sym | Symbol number |
| gti:tpd_dim | Theoretically precise dimension |


© 2026 INNEO Solutions GmbH

| Comment text | Column name |
|---|---|
| gti:cls_tol | Tolerance class |
| gti:tpe_tol | Tolerance standard |
| gti:mod | Tolerance table |
| gti:gtol_value | Tolerance value |
| gti:gtol_top_text | Top text |
| gti:upp_tol | Upper boundary |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gti*. Thus you can export data from *Inspect* and *Inspect Revision* together, see Export data from several GENIUS-TOOLS-components[701] for an example.

# 10.3 Configuration

This section contains information about the configuration of *Inspect 3D*, which is performed in *Inspect > Inspect Editor*[205].

## 10.3.1 Inspect Editor

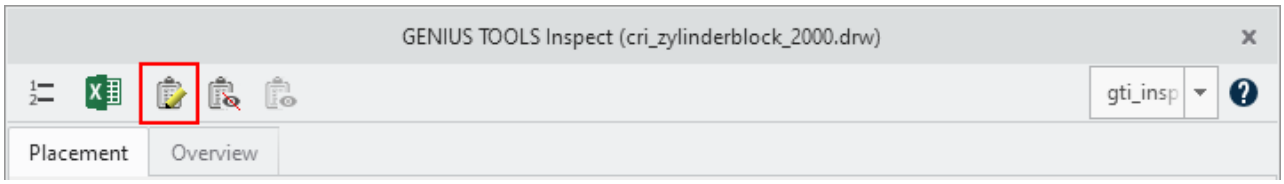
The *Inspect Editor* is used to create and manage settings for *Inspect* and *Inspect 3D*. Settings that are valid only for Inspect or only for Inspect 3D are also only applied in the corresponding module. If a setting for 3D is made in 2D or vice versa, it **cannot** be read, i.e., the setting has no effect.

Multiple groups of settings consisting of inspection symbols, tables, and view settings can be managed. Each configuration is saved as an XML file in the `gti_folder`.

Users can switch between different configurations in the program interface.

## Starting the program

Start the editor  from the command bar of *Inspect* or *Inspect 3D*.





© 2026 INNEO Solutions GmbH

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

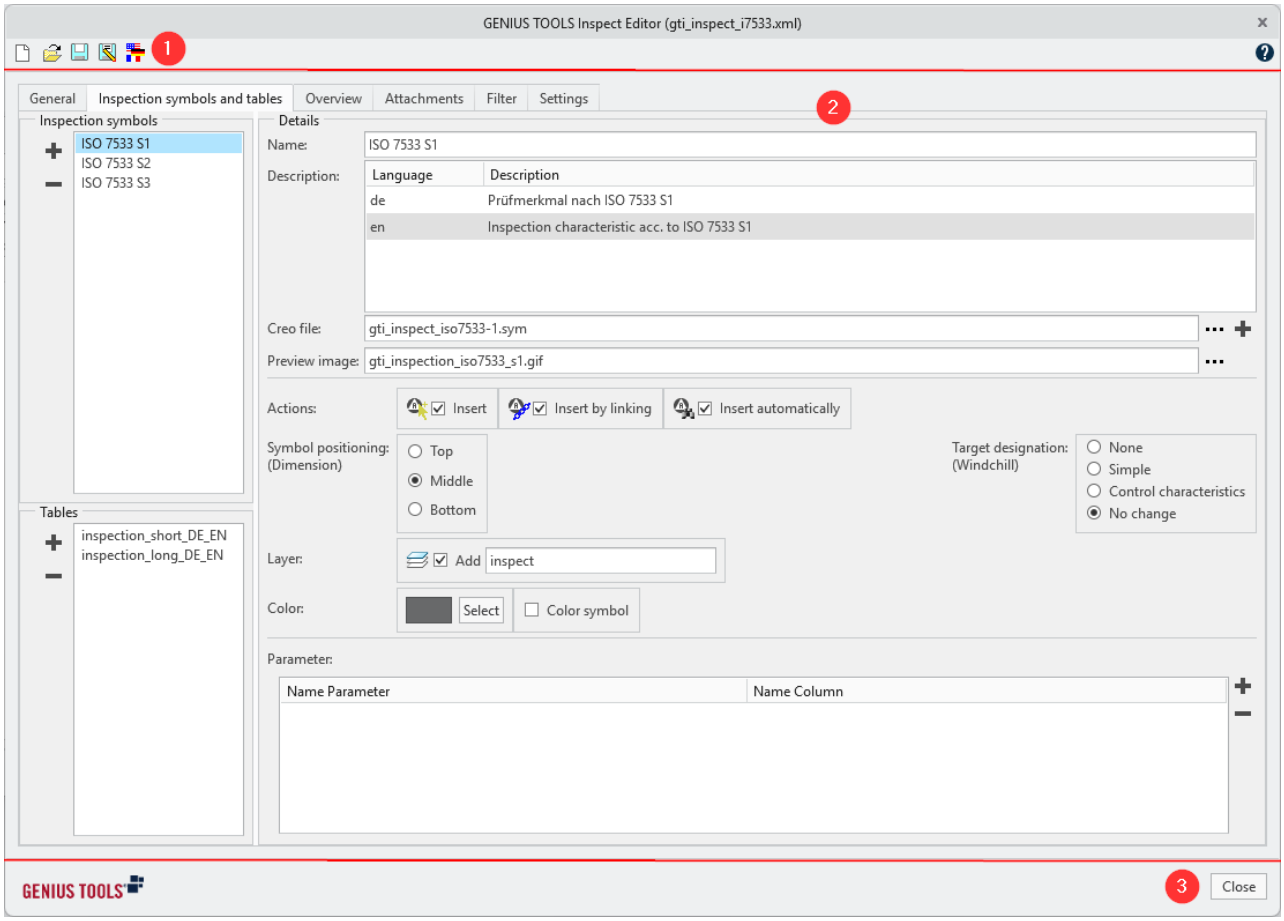
SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## 10.3.1.1 User interface

The user interface of *Inspect Editor* consists of the following elements:





© 2026 INNEO Solutions GmbH

1. Command bar
2. Detail view of the tab *Inspection symbols and tables*
   The editor is divided into these tabs: Inspection symbols and tables[208], Overview[215], Attachments[215], Filter[216] and GT MBD.
3. Close

### 10.3.1.2 Inspect configuration

A configuration in *Inspect* is an XML file that contains definitions for symbols, tables and display settings. Create new configurations in the Editor by clicking the  button.

**Inspect directory:** Configurations are by default saved in the inspect directory in the resource folder *(%gt_resource_folder%inspect)*.

**Find inspect directory:** The path to the directory can be edited by the option `gti_folder`.

**Save configuration:** The name of a configuration file must not exceed 18 characters.

**Warning:** Configuration files that have been manually created with names longer than 18 characters cannot be opened by the editor. They must be edited manually.

## Command bar

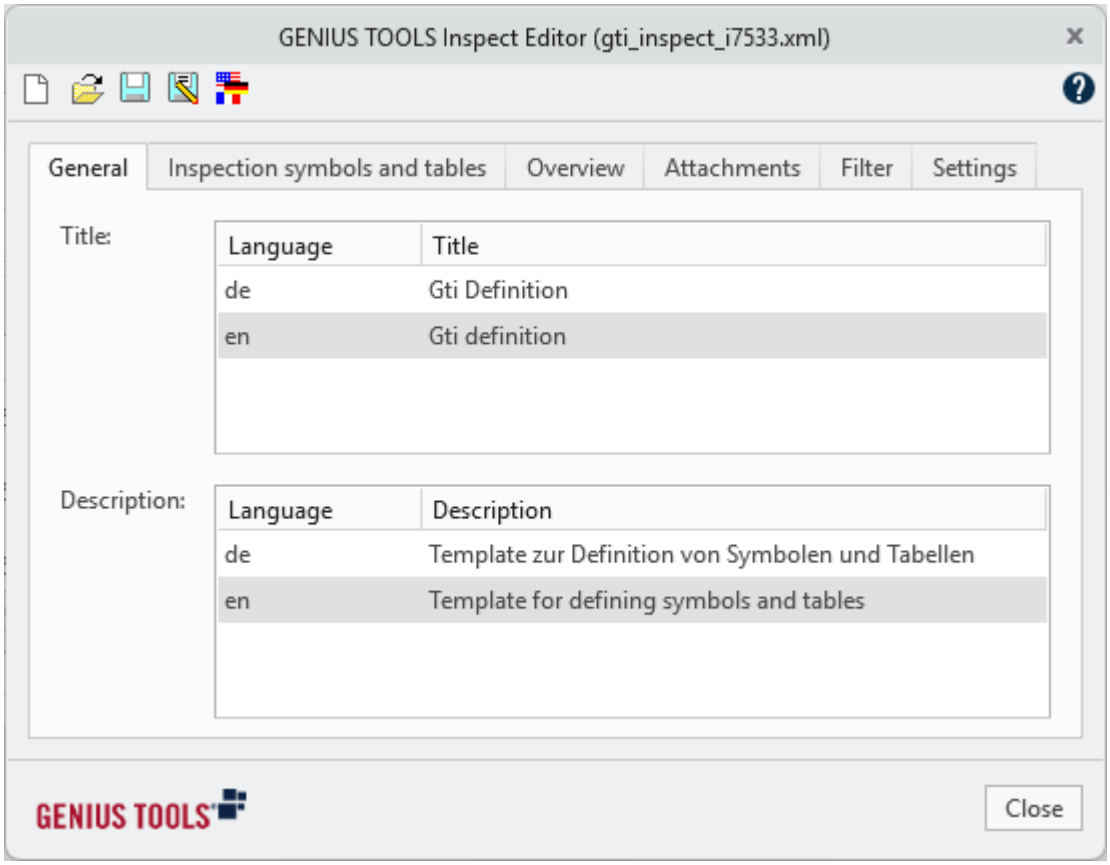
The command bar contains the following buttons:

| Symbol | Name | Description |
|---|---|---|
|  | New configuration | Creates a new Inspect configuration[207] (XML). Each configuration can contain different symbol and table definitions as well as view settings. |
|  | Open configuration | Opens an existing configuration from an XML file. |
|  | Save | Saves the current configuration to an XML file. |
|  | Save as | Saves the current configuration under a new name. |
|  | Language dialog | Adds or deletes a language from a configuration. |
|  | Open help | Opens the help. |

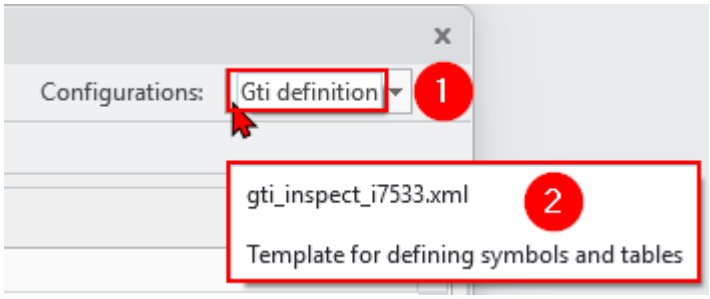

© 2026 INNEO Solutions GmbH

## 10.3.1.3 General

In the *General* tab, you can edit the configuration data in multiple languages.




1. **Title**: The name of the title is displayed in the selection menu.
2. **Description**: Tooltip that is displayed for the title.

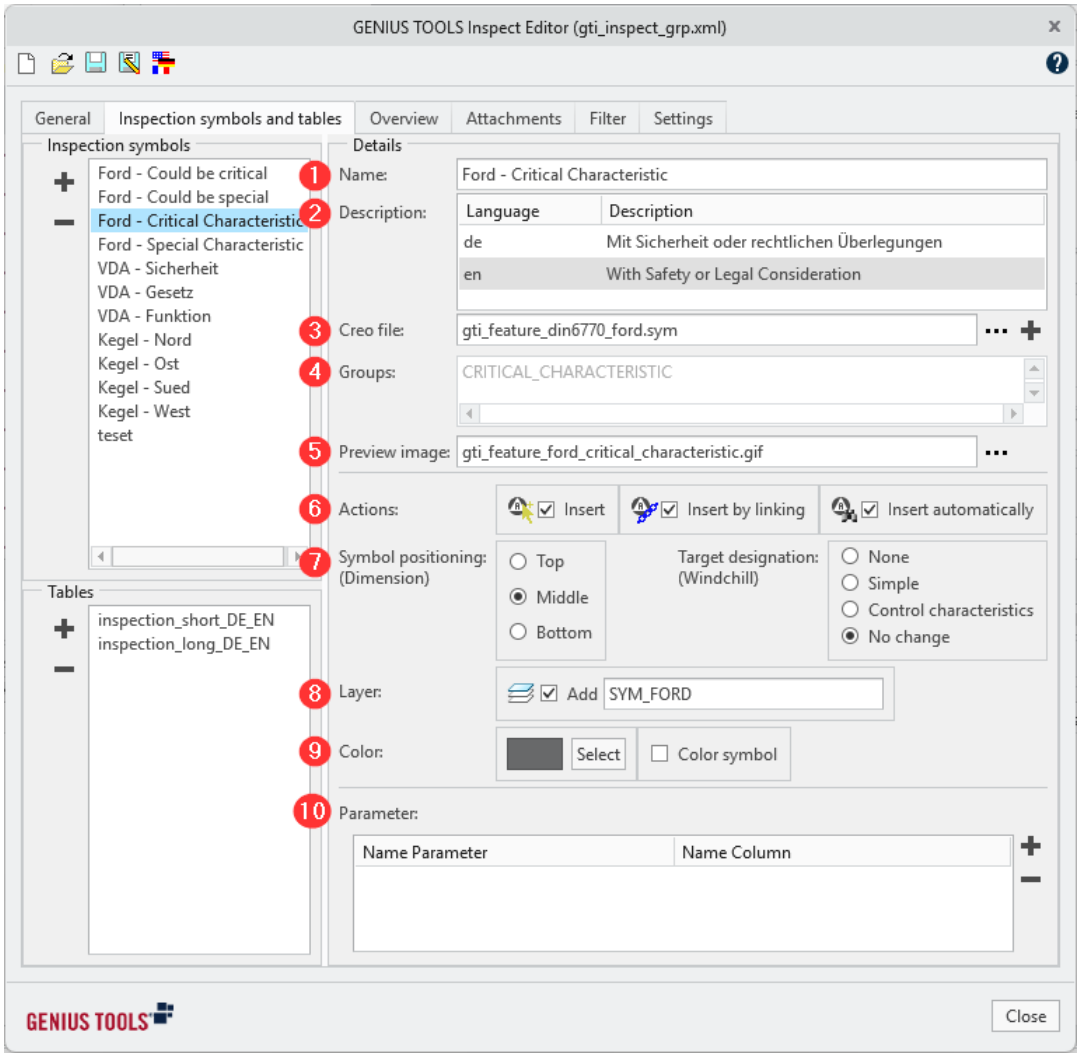



## 10.3.1.4 Editing inspection symbols and tables

In the tab *Inspection symbols and Tables*, the different sets of settings for inspection symbols and tables are configured. Use the (+/-) buttons to add or remove elements.


© 2026 INNEO Solutions GmbH

The order of the inspection symbols and tables determines their order in the user interface of *Inspect*. An elements position can be changed by drag-and-drop, dropping the element below the target element.




Click on an inspection symbol or table in the element selection to view the details.

## Inspection symbols

1. **Name**
   Specifies the displayed name of the inspection symbol.

2. **Description**
   A localized description of the inspection symbol. Use the localization button in the command bar to manage the languages.

3. **Creo-File**
   Specifies the path to a SYM file. Default is *%gt_resource_folder%\inspect*.


*© 2026 INNEO Solutions GmbH*

**Warning:** Do not use the same symbol in different Inspect configurations if you are using numbering similar to DIN 6770. Otherwise, you may experience numbering conflicts.

4. **Grouping**
   If you are using a symbol that contains groupings, create a separate inspection symbol for each symbol variant that you want to place using *Inspect*. If you select a symbol file that contains groupings, the first grouping found is selected for the inspection symbol. To select the required symbol variant for the inspection symbol, proceed as follows.
   - Open a drawing that contains the required symbol variant, or place the required symbol variant on a drawing.
   - Open Inspect Editor and create the new inspection symbol. Select the symbol file under Creo-File using the Browse button (...).
   - Click on the plus symbol next to the Creo-File input field. You are asked to select an element.
   - Select the required symbol in the drawing. The grouping settings of this symbol are written to the inspection symbol definition.
5. **Preview image**: Specifies the path to a preview file of an inspection symbol.
6. **Actions**: Defines the actions that can be used on an inspection symbol: free insertion, linked insertion and linked, automatic insertion.
7. **Symbol positioning (Dimension)**
   Defines the positioning of the symbol in relation to the dimension: Top, Middle, Bottom. Defines whether and in what form the data is displayed in Windchill: None, Simple, Control characteristics, No change
8. **Layer**: Defines the layer on which the symbols are placed. Enter the layer under *Name*.

**Please note**: When you hide the layer, only inspection symbols that have been freely placed will be hidden with it. Linked inspection symbols can only be hidden using the *Inspect* hide and unhide functionality, see Hide and unhide 201.

9. **Color**
   Specify the color to be used for the selected inspection characteristic. Use the checkbox to toggle the coloring.
10. **Parameter**: You can add additional information from drawing or model parameters to an inspection symbol. To do this, a symbol must have variable text, in which the necessary drawing or model parameter is defined. See Creating variable text 221 and Adding parameter values to a symbol 224 for more information.

Under *Name Parameter,* enter the name of the variable text. Under *Name Column*, enter a name for the table column that displays the values from the variable text. This column is added to the symbol, which you can check in the *Overview* 198 tab in *Inspect*.


© 2026 INNEO Solutions GmbH

**Warning:** The following terms are used by *Inspect* and cannot be used for naming parameters and their columns, both upper and lower case.

| | | | |
|---|---|---|---|
| bse_dim | gti_param | max_dim | tpe_sub |
| cls_tol | linkedto | min_dim | tpe_tol |
| creo_id | nme_sym | mod | val_tol |
| descr | num | revision | src |
| grd | num_sheet | separator | tpe_sym |
| gti_note | num_sym | tpe_main | tpe_tol |
| ipc_dim | | | |

To output the additional parameters in the Excel export as well, you have to assign a column in the Excel template to each parameter. In the Excel template, specify the column name (**Name Column**) in the format *gti:<columnName>*, see also Export[202].

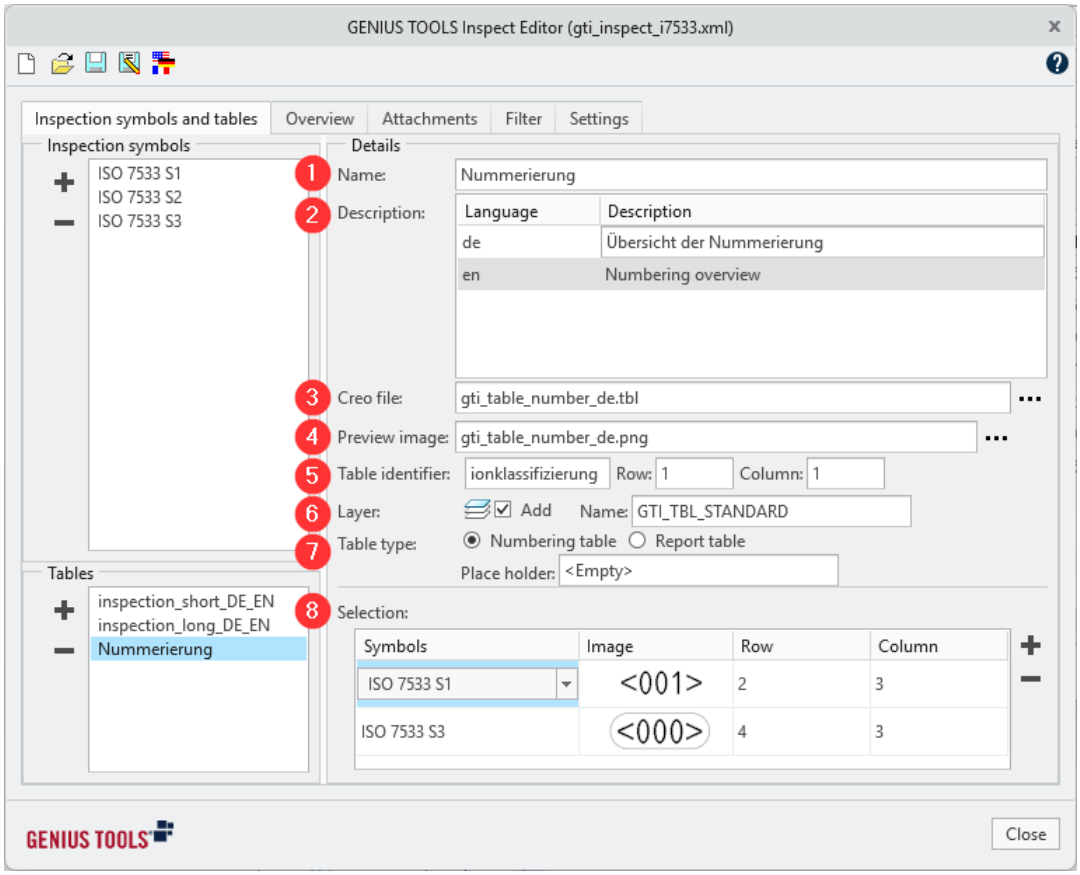
## Tables

*Inspect* comes with two out-of-the-box tables, one numbering table and one report table. The configuration for the settings *Table identifier*, *Table type* and *Selection* differs according to the table type. The Creo table used also has some different characteristics.

- **Numbering table:** lists inspection symbols with the symbol image, a description and a list of numbers.
- **Report table:** displays detailed information on individual inspection symbols.

For both table types, you can customize the standard template or create your own table templates. Please refer to Configuration for numbering tables[213] and Configuration for report tables[214].


© 2026 INNEO Solutions GmbH

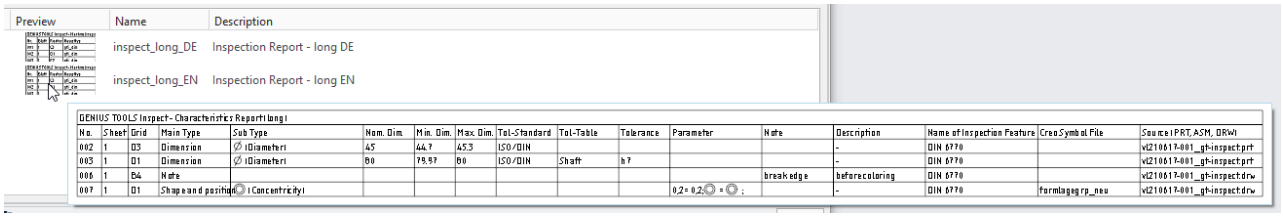



1. **Name**
   Specifies the displayed name of the table.

2. **Description**
   A localized description of the table. Use the localization button in the command bar to manage the languages.

3. **Creo-File**
   Specifies the path to a TBL file. The default path is *%gt_resource_folder%\inspect*. Numbering tables [213] and report tables [214] differ.

4. **Preview image**
   Specifies the path to a preview file of a table image. The tooltip displays the image in its original size, so a large image may improve the legibility of the preview.




5. **Table identifier**
   Used to identify a table. There are differences between numbering tables [213] and report tables [214].


© 2026 INNEO Solutions GmbH

6. **Layer**
   Defines the layer on which the placed table will be placed when *Add* is set. Enter the layer name under *Name*.
7. **Table type**
   The table type is either *Numbering table* or *Report table*. In addition, a place holder for empty cells can be entered for a *Numbering table*, e.g. <Empty>.
8. **Selection**
   This section is displayed only for numbering tables[213].

## Configuring numbering tables

**Creo-File**: Specifies the path to a TBL file. The default path is `gt_resource_folder`. The TBL file for a numbering table specifies symbols and descriptions line by line. For each symbol, a cell is provided for the list of numbers used. The table cell for the numbering is specified in the section *Selection* under *Row* and *Column*.

| Function related classification | | Inspection No. |
|---|---|---|
| Symbol | Critical Feature | 1 |
|  | Hazardous to human beings | |
| Symbol | Main feature | <Empty> |
|  | Restricted functionality, reduced output/performance | |
| Symbol | Standard feature | all other quality parameter |
| | no direct effect on funktionality | |

*Example of a Creo table for a numbering table*

**Table identifier / Row / Column**: The table identifier is used to identify a table and must be unique. The table identifier is *Function related classification* for the example numbering table. If you want to define your own tables, use a unique table identifier.

Enter a row and column to specify where the table identifier is located in the table. By default, the table identifier is located in the first cell of the first row.

**Selection**: The position of the displayed inspection symbols is defined here. Click on the displayed name of an inspection symbol to display a drop-down list of symbols not yet included. Use the (+/-) buttons to add or remove rows.

Under *Row* and *Column*, specify the table cell in which to display the list of numbers used. For summarized cells, specify the top row.

**Please note**: Numbering tables cannot contain multiple inspection symbols that use the same Creo symbol file. Only the last row defined for each symbol file is saved in the table definition.


© 2026 INNEO Solutions GmbH

# Configuring report tables

**Creo-File**: Specifies the path to a TBL file. The default path is `gt_resource_folder`.

For report tables, the second row of the table defines which data to write to each column. This is done by specifying the abbreviation for the desired column. The available column codes are listed in the section on adapting the Excel export template for *Inspect* under Export 202.

The first cell to be filled with data contains the text *Wertezeile* (value row).

If you insert a report table with *Inspect*, the table is filled automatically and the first two rows are hidden, making the table identifier and column types invisible.

**Please note:** There is no way to display the top two rows of the table once you have inserted it using *Inspect*. If you want to make changes to the TBL template, you have to edit it in Creo without using *Inspect*.

| gt_tbl_gti | | |
|---|---|---|
| num_sym | tpe_main | tpe_sub |
| GENIUS TOOLS Inspect Tabelle | | |
| Number | Main type | Subtype |
| Wertezeile | | |

*Example of a Creo table for a report table*

**Table identifier / Row / Column**: The table identifier is used to identify a table and must be unique. The table identifier is *gti_tbl_report_long* for the example report table. If you want to define your own tables, use a unique table identifier.

Enter a row and column to specify where the table identifier is located in the table. By default, the table identifier is located in the first cell of the first row. If you insert a report table with *Inspect*, the table is filled automatically and the first two rows are hidden, making the table identifier and column types invisible.

**Please note:** Using the configuration option `gti_lang` you define - not depending on the system language - the language used for the contents of the report table.

# Customizing tables

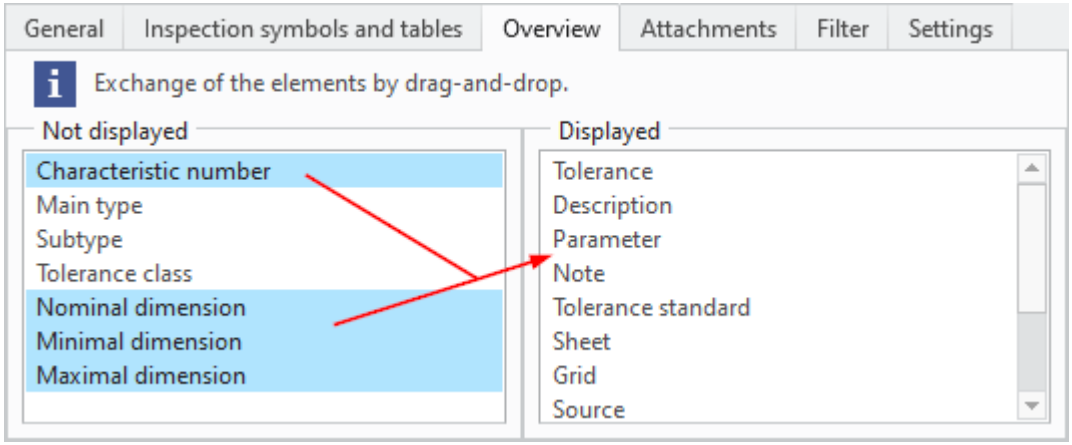
For instructions on creating a customized report table see the chapter Creating a custom report table 229.


© 2026 INNEO Solutions GmbH

## 10.3.1.5 Editing column display

The *Overview* tab is used to manage the display of the *Linked Symbols* table.

Drag and drop elements to be displayed in the table into the desired position. Remove items that you do not want to display in the same way.

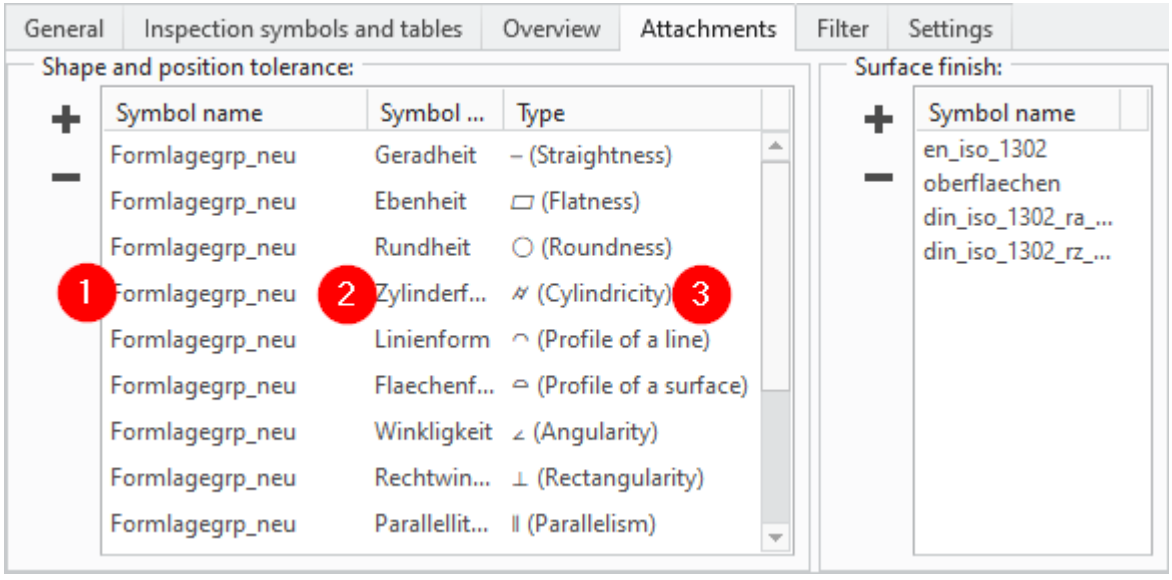



*Use Drag and Drop to compile the overview table*

## 10.3.1.6 Assigning shape and position tolerances / surface finish symbols

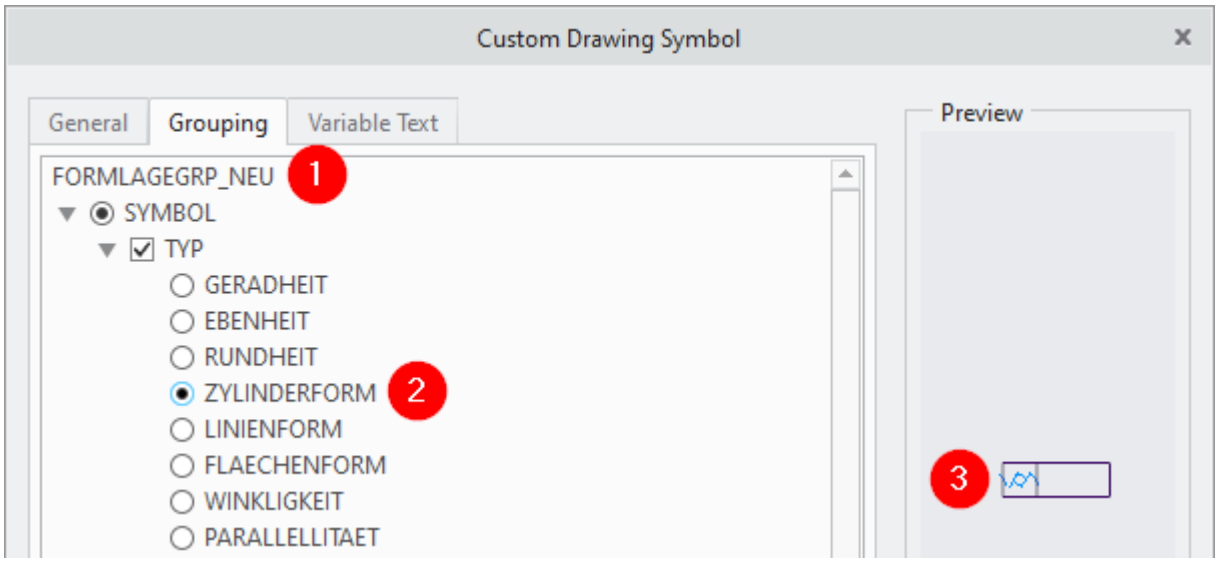
The *Attachments* tab manages the assignment of various shape and position tolerances or surface quality symbols to user-defined symbols.

This makes it possible to filter for inspection symbols linked to defined types of elements.




*Entries in the shape and position tolerance table or surface quality table correspond to properties of symbols in the symbol library.*


© 2026 INNEO Solutions GmbH




*Creo dialog for groupings in drawing symbols*

Enter the file names of the custom drawing symbols (as defined in the Creo dialog box) in the corresponding tables to assign them to the correct main types. In the case of shape and position tolerances, the main type (1) and subtype (2 and 3) can be determined.

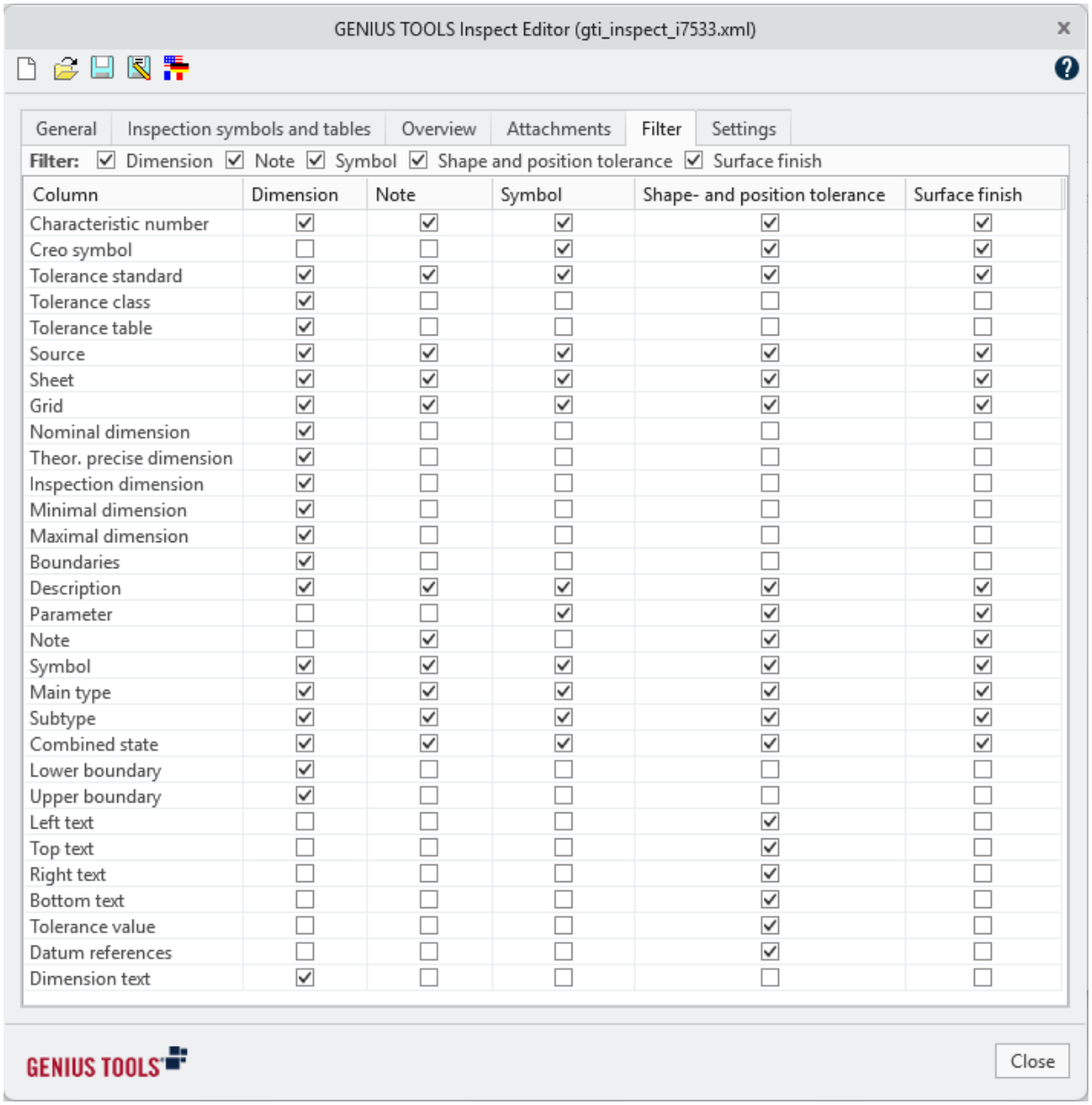
## 10.3.1.7 Filtering displayed inspection symbols

The *Filter* tab defines the view of the usable filters for the *Linked Symbols* overview. The configuration is stored per definition.

First, specify which filters (dimension, note, symbol, shape and position, surface) are to be activated by default.

In the second step, configure the columns to be displayed. Then save the configuration. The overview table *Linked Symbols* will always use the filters configured here when applying the definition.


© 2026 INNEO Solutions GmbH

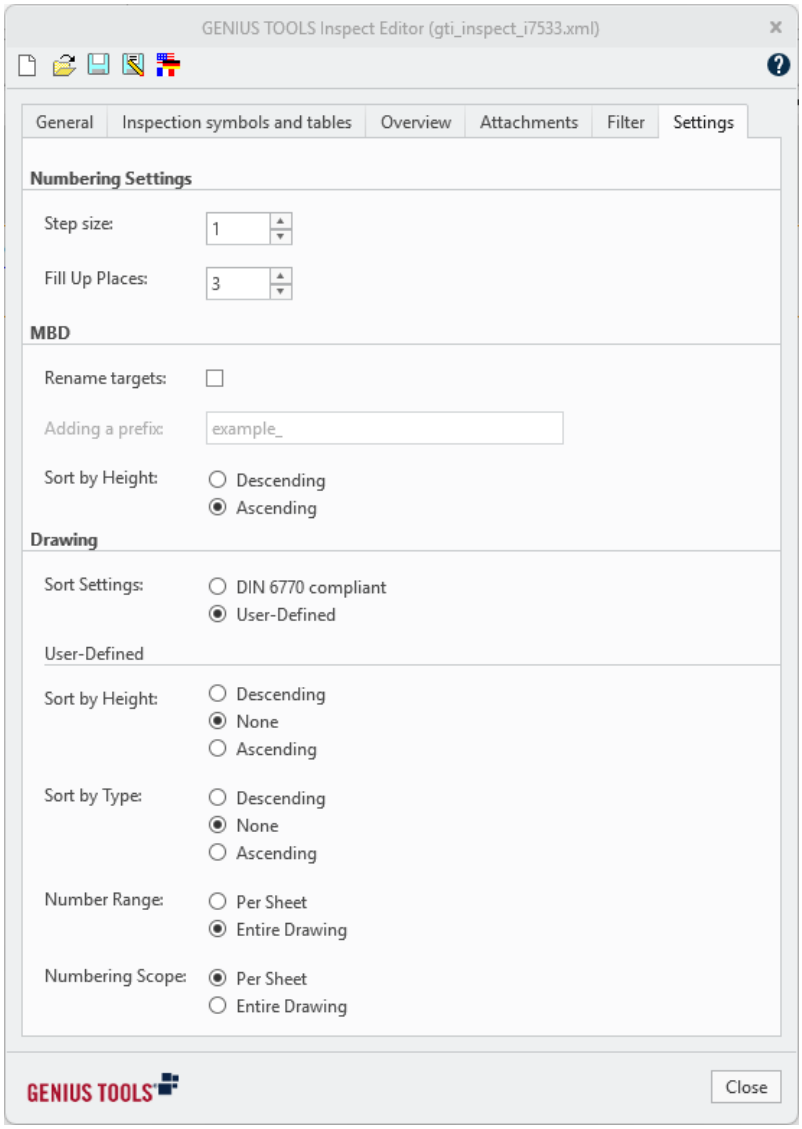



*In this example, the Tolerance class column will only be displayed, if the Dimension filter is active, i.e. if inspection symbols linked to dimensions are displayed.*

## 10.3.1.8 Settings for 3D models

**Please note:** In the tab *GT MBD*, settings can only be made for 3D models that are executed via *Inspect 3D*[243]. In drawing mode, the settings made here are not read and have no effect.


© 2026 INNEO Solutions GmbH




### Numbering Settings

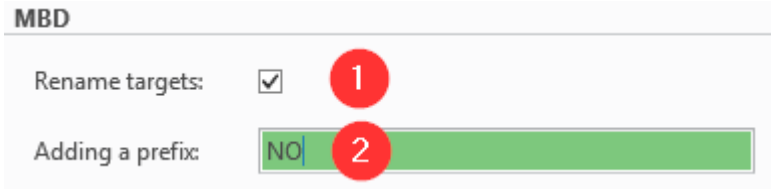
Step size: Set the step size in which symbols are to be numbered for each configuration.

Fill Up Places: Fill with leading zeros up to a certain number of digits.

### MBD

Inspection characteristics can be displayed in the annotation tree with an additional description. The inspection characteristics are displayed in the model as they were specified in the *Editor*. A prefix can be displayed in the annotation tree, e.g. NO.

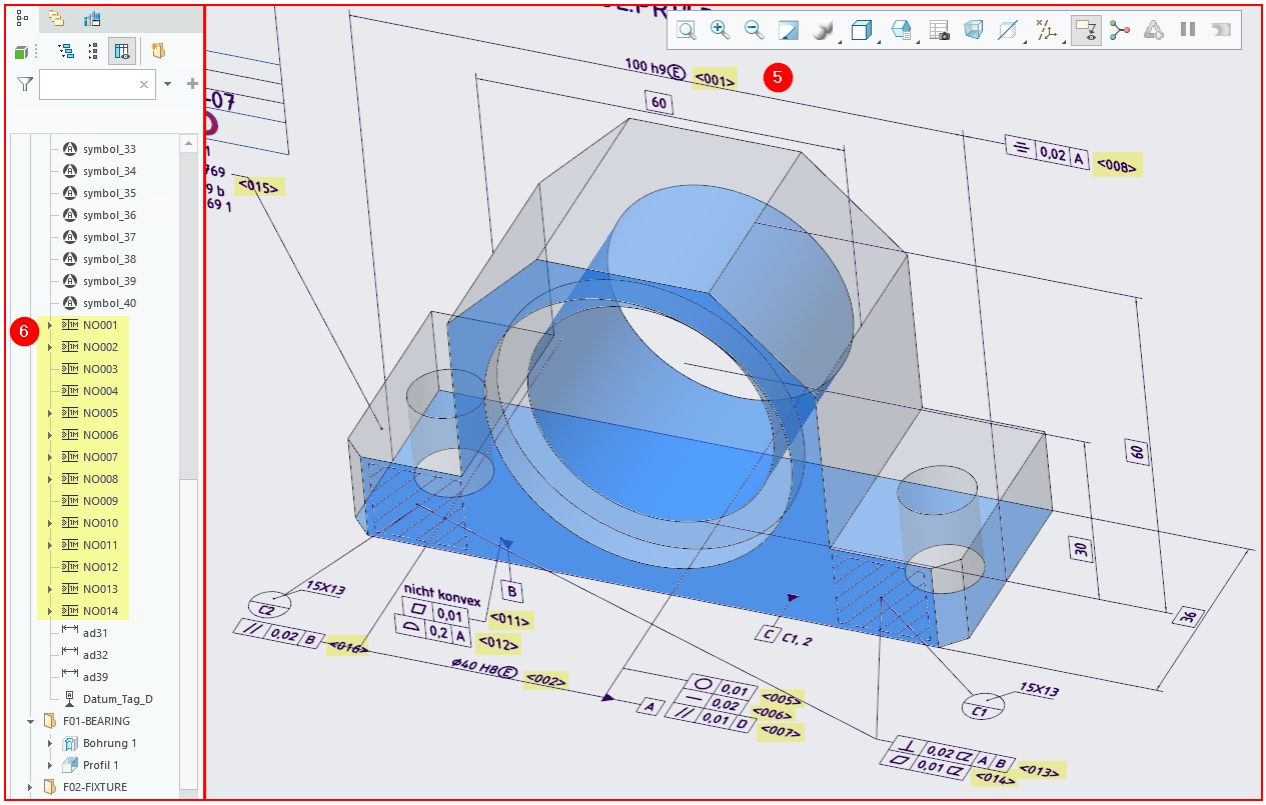
To obtain this display, set the following settings:




1. Select the checkbox for *Rename targets?*

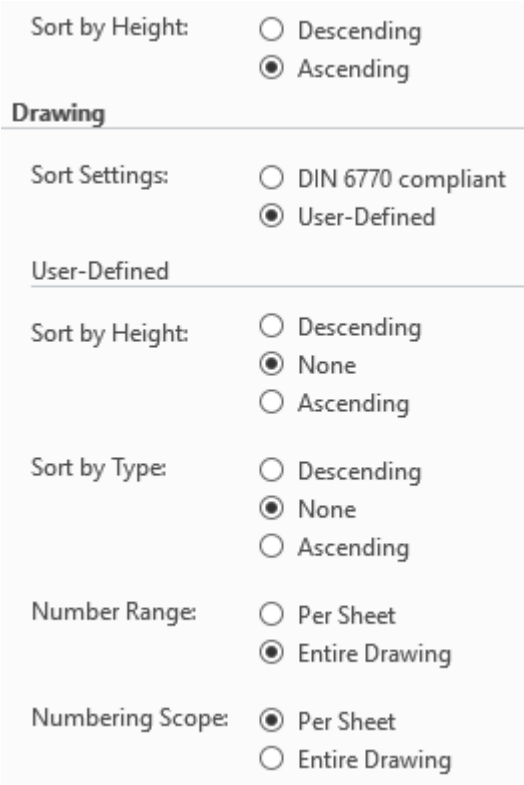

© 2026 INNEO Solutions GmbH

2. Enter the prefix you want to use, here: NO for numbering.
3. Save settings in *Inspect Editor* and close the Editor.
4. Click on *Update inspection features*.
5. You receive the following display:
   - In the model, the inspection characteristics are displayed as defined, e.g. <001> (5).
   - In the annotation tree, the same inspection characteristics are displayed with the defined prefix, e.g. NO001 (6).




**Drawing**
The numbering can be defined either in accordance with DIN 6770 or user-defined.


© 2026 INNEO Solutions GmbH




## 10.3.2 Creating a Creo symbol for Inspect

This section explains how to create a new symbol for use as an inspection symbol.

**Please note:** The symbols delivered with *Inspect* use the font *ISONORM LT Regular* based on ISO 3098-5. If you experience issues with displaying the out-of-the-box symbols, please check whether this font is available in your system.

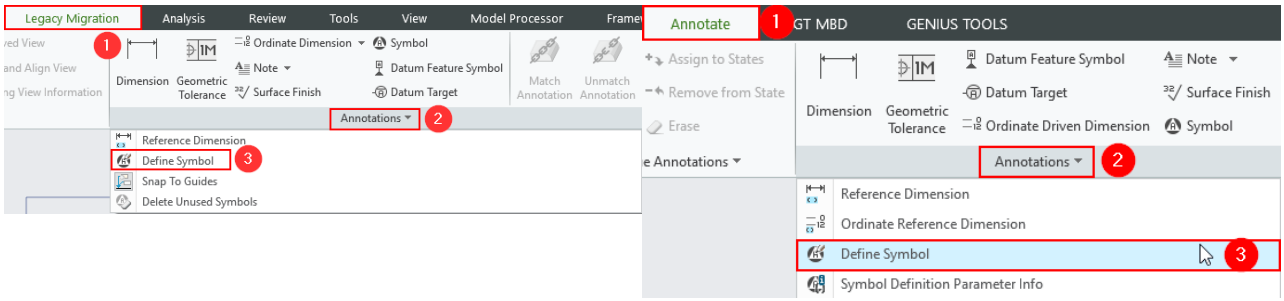
## 1. Sketching the symbol

Open the command *Define Symbol*.

For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:

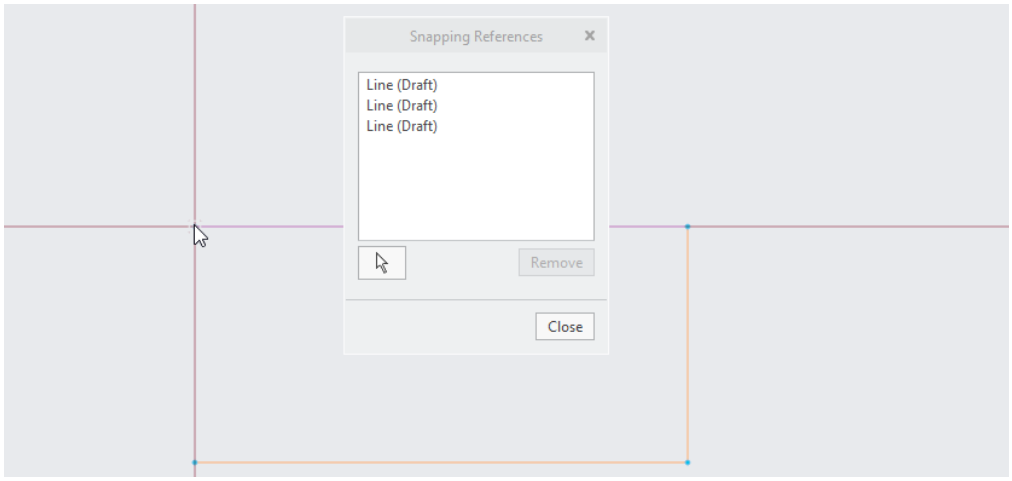
For *Inspect 3D* in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:


© 2026 INNEO Solutions GmbH




In the Menu Manager in the section SYM DEFINITION click *Define* and enter a name for the symbol.

To define your own symbol, create a Creo Parametric symbol in the sketcher.



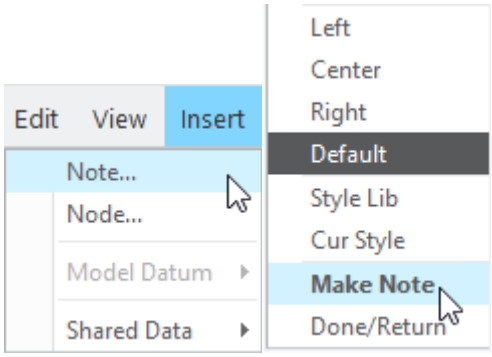

*Sketching a symbol by lines.*

# 2. Creating variable text

Variable text is data that is saved on a symbol as a note.

## 2.1. Create a note

Click *Insert* > *Note* and in the Menu Manager that opens click *Make Note* to create and place a new note.




Next you will be asked for the note name.

To ensure compatibility with *Inspect*, you have to create three notes. Enter the following names:


© 2026 INNEO Solutions GmbH

– \num\ to save the number of the inspection symbol

– \descr\ to save the description

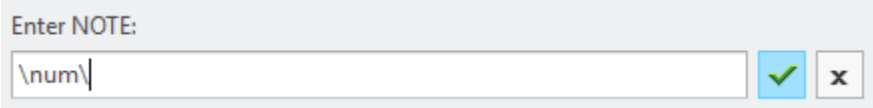
– \LinkedTo\ to save the linked drawing element for the symbol

**Warning:** The following names must not be given to a note – both in upper and lower case –, as they are in use by the program.

| | | | |
|---|---|---|---|
| bse_dim | gti_param | min_dim | tpe_sub |
| cls_tol | nme_sym | mod | tpe_tol |
| creo_id | num_sheet | revision | val_tol |
| grd | num_sym | separator | src |
| gti_note | max_dim | tpe_main | tpe_sym |

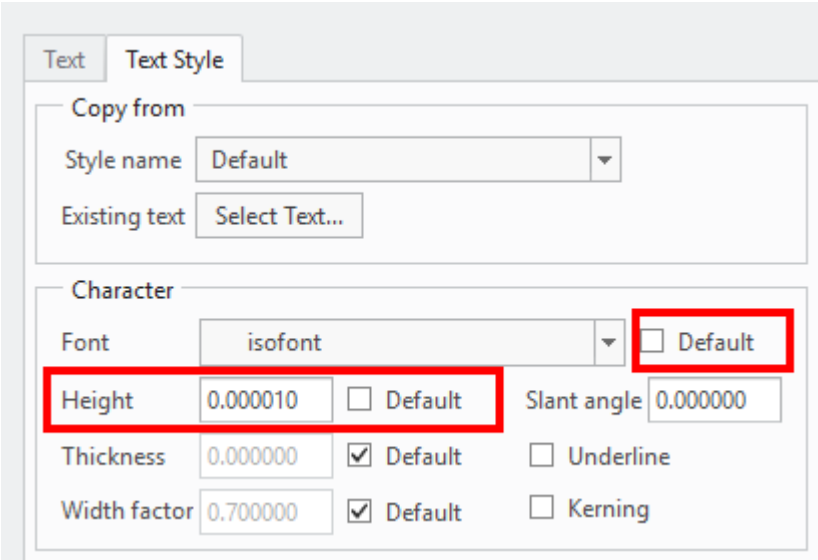
Enter the note name, confirm by clicking on the green arrow, then exit the input dialog by clicking on the X button.

The next notes can be created in the same way.




### 2.2. Change the size of the notes

Since the notes *descr*\ and *LinkedTo*\ should not appear when the drawing is printed, their size has to be reduced. You can change it via the context menu of the note by right clicking on it. Select *Properties* > *Text Style* to change the font and size.




*Text style of a note*

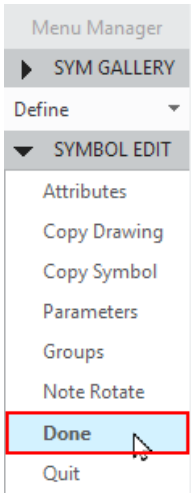

© 2026 INNEO Solutions GmbH

– Remove the checkmark from the properties for *Height* and set a very small value, e.g. 0.00001.

– Remove the checkmark from *Font*.

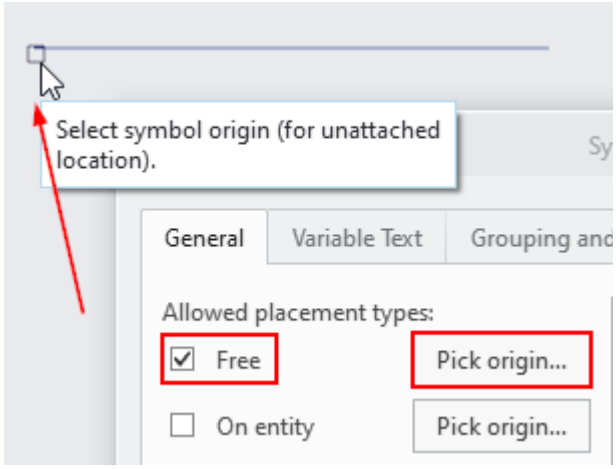
## 3. Define attributes for the symbol

To finish creating the symbol, you need to define the placement, symbol origin and predefined values of the variable texts. The dialog for these settings opens when you click *Done* in the *Menu manager*.




### 3.1. General attributes

Check *Free* and select the lower left corner as origin.




*Choosing the placement and origin*

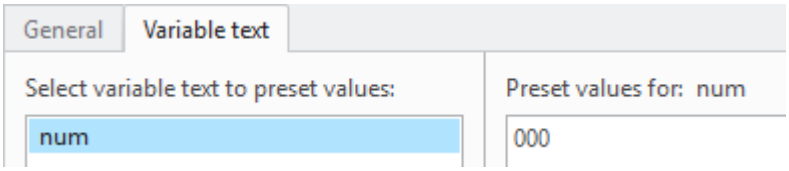
### 3.2. Variable text

In the second tab *Variable text* enter in the section *Preset values for* the following characters:

| Name | Value |
|---|---|
| num | 000 |


© 2026 INNEO Solutions GmbH

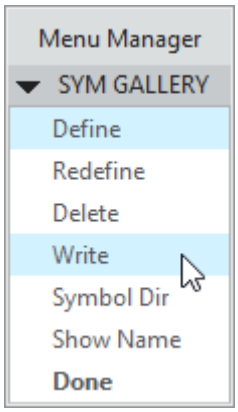
| Name | Value |
|---|---|
| descr | - |
| LinkedTo | -1,-,-1,- |




*Predefined value for /num/.*

## 5. Saving the symbol

As the last step, the symbol has to be saved. You can do this by clicking *Write* in the *Menu manager*. Enter the path, or leave it empty to use the current symbol directory for saving.




## 10.3.3 Adding parameter values to a symbol

You can assign values of drawing or model parameters to an inspection symbol. Parameter values can be either copied or linked to the symbol.

- Copy: The parameter value is copied to the inspection symbol when it is placed on the drawing and is static thereafter. (See Copying parameter values[225].)
- Link: The parameter value is updated on the inspection symbol when the drawing or the model is opened. (See Linking parameter values[226].)

For general information on parameters consult the chapter Variables.[916]

Parameters are written into the variable text in the dialog *Symbol Definition Attribute*.

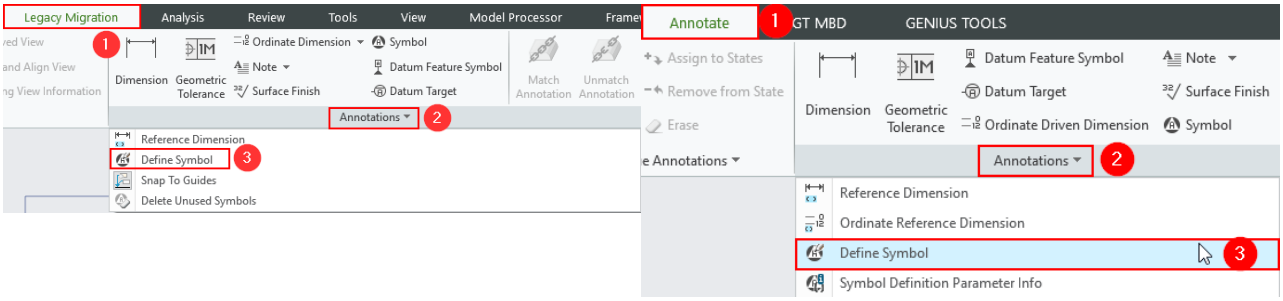
## Defining a symbol

1. Open the command *Define Symbol*.

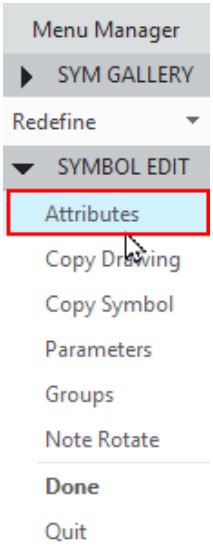

© 2026 INNEO Solutions GmbH

For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:

For *Inspect 3D* in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:




2. In the Menu Manager in the section SYM GALLERY click *Redefine*.
3. Click on the symbol. The Creo symbol editor is opened.
4. In the Menu Manager click on *Attributes*. The dialog *Symbol Definition Attributes* is opened.



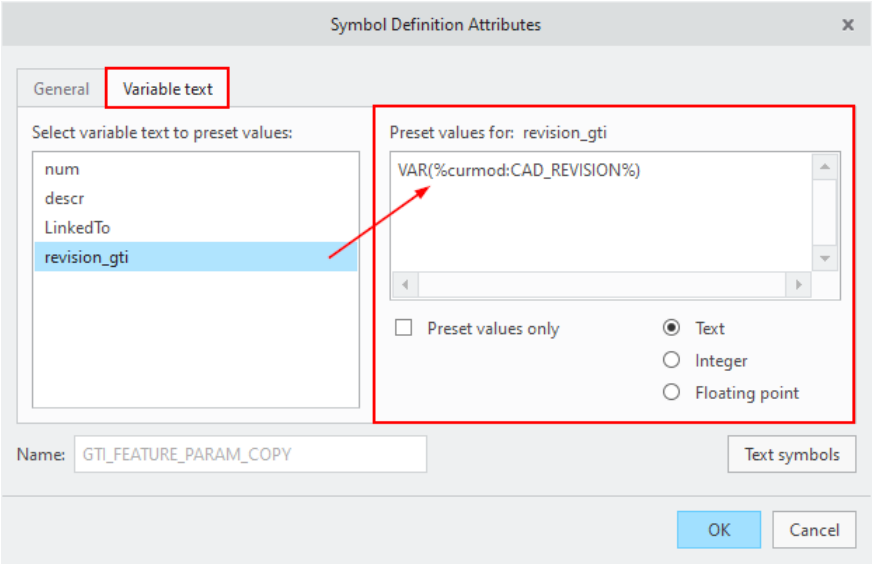

## Copying parameter values

Enter the parameter name in the variable text on the symbol in the Symbol Definition Attributes[224] dialog. In the tab *Variable text* > field *Preset values for* type:

– for a revision parameter from the model: *VAR(%curmod:Parameter%)*

– for a revision parameter from the drawing: *VAR(%Parameter%)*


© 2026 INNEO Solutions GmbH




*Preset value: Copy value from parameter CAD_REVISION*

For a step-by-step description of how to copy parameter values into symbols, see the example Creating Change Symbols[226].

## Linking parameter values (Referencing)

The parameter value is updated at the inspection symbol when the drawing is opened. In the Symbol Definition Attributes[224] dialog > *Variable text* tab > *Preset values for* field, type the parameter name in the following format:

– for a parameter from a model: *&Parameter*

– for a parameter from a drawing: *&Parameter:D*

**Please note:** The revision parameter CAD_REVISION should not be used for change symbols because the revision information on the drawing would change with each new revision.

## Redefine parameters

Existing parameters can be redefined in variable text in the drawing mode. Enter the changes for the variable text in the Symbol Definition Attributes[224] dialog.

## 10.3.4 Creating change symbols and tables

Use change symbols to mark changes to elements. Revision information can be used for this.

To display changes in a drawing, the revision must be copied to the change icon at the time of placement.


© 2026 INNEO Solutions GmbH

With the variable *revision_gti*, *Inspect* provides an interface that copies values of drawing or model parameters to an inspection symbol. It can also be used in an Inspect table.

For a revision value to be copied into a change symbol, the following preset values must be assigned in the variable text *revision_gti:*

– for a revision parameter from the model: `VAR(%curmod:Parameter%)`

e.g.: `VAR(%curmod:CAD_REVISION%)`

– for a revision parameter from the drawing: `VAR(%Parameter%)`

e.g.: `VAR(%CAD_REVISION%)`

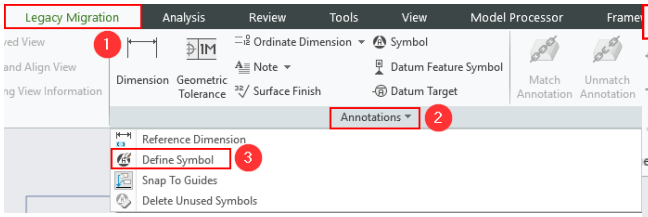
The step-by-step procedure is described below.

# Creating change symbols

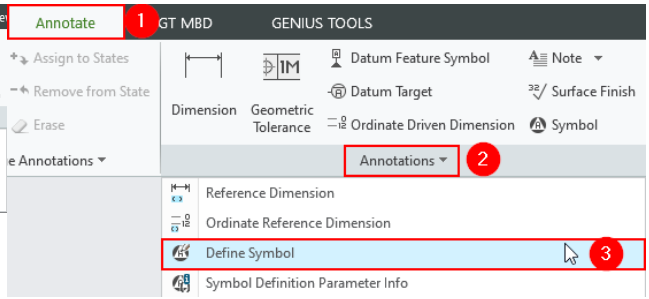
**Procedure for copying the values of a revision parameter to a symbol**

1. Open a drawing which includes the symbol in question or create a new symbol[220] first.
2. Open the command *Define Symbol*.

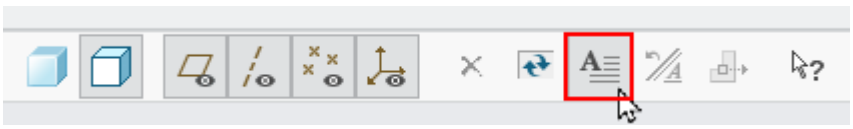
For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:




For Inspect 3D in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:



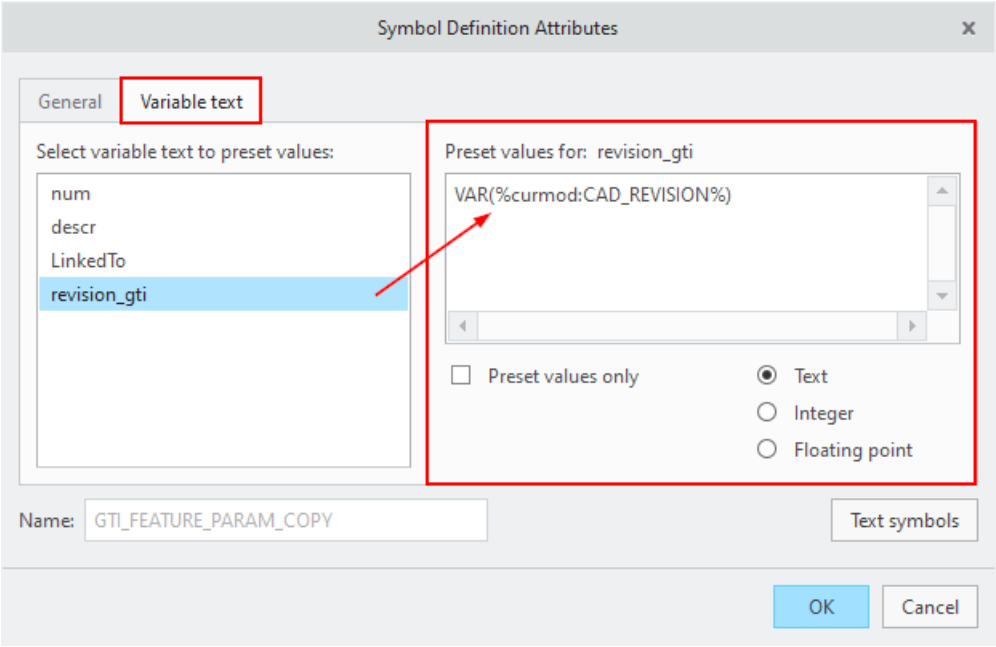

3. In the Menu Manager / SYM DEFINITION, go to *Redefine* and *Pick Inst.*
4. Click on the symbol in the drawing.
5. The Creo symbol editor opens. Go to the *Create Note* icon in the menu bar:



6. In the Menu Manager, go to *Notes* > *Create Note*. Select the style of the note and click *Make Note*.
7. The *Select item* dialog opens. Select a point on the sketch where the note should be located.


© 2026 INNEO Solutions GmbH

8. Enter this name for the note: *revision_gti*\ and click on the green arrow and then on the cross beside it.
9. Click *Done* in the Menu Manager.
10. In the Menu Manager, go to *SYMBOL EDIT* > *Attributes*.
11. This opens the *Symbol Definition Attributes* dialog. Specify the revision parameter for *revision_gti* in the tab *Variable Text*:
   – for a revision parameter from the model: *VAR(%curmod:Parameter%)*
   – for a revision parameter from the drawing: *VAR(%Parameter%)*




*Copy parameter values for the variable text revision_gti*

## Creating change tables

Changes can be displayed on a drawing by inserting a table with a column that shows the copied value of the revision parameter from the icon. You can either use the file *gti_table_param_en.tbl* or create the required column yourself in a table as follows.

1. The Creo symbol must have a variable text – here: *revision_gti* – and the revision parameter must be copied into it. (See step 11 above.)
2. This variable text must be defined for the symbol in *Inspect Editor*. To do this, go to the *Inspect Characteristics and Tables > Parameters*[208] tab and type the name of the variable text under *Name Parameter* – here: *revision_gti* – and the column name under *Name Column* – here: *Revision*.

   This column is now added to the symbol, as you can check in *Inspect* on the *Overview* tab.
3. Now create your own report table as in the example in the Creating a custom report table[229] chapter and perform step 9 and 10 (naming new column) as follows.


© 2026 INNEO Solutions GmbH

– write in the second line the same name as you have defined in step 2 under *Name column* – here: *Revision* (this line will become invisible when the table is placed using *Inspect*)

– write in the fourth line *Revision* (this is the displayed line in the table)

4. Save the table (TBL file) in the Caddepot directory of the synchronization server in the *Inspect* folder.

## 10.3.5 Creating a custom report table

This example shows how to create your own report table from a template from the *Startup TOOLS*. (See also differences between numbering and report tables208 ).

## Step-by-step guide

The template is created in the resource folder of the caddepot.

**Pause synchronization**

1. In *GENIUS TOOLS Starter App*, go to the user menu and select *Pause synchronization*.

**Open template**

2. Open a drawing.
3. In the Creo ribbon, go to the tab *Table* > *Table from File*.
4. Open a report table from the *inspect* directory in the folder *gt_resource_folder* (path: %gt_resource_folder%\inspect) the TBL file *gti_table_inspect_l_en.tb*.
5. Place the table on the drawing.

**Specify new table identifier**

6. Go to the cell defined for the table identifier212 (default: first row, first column).
   a. For the default report table, the table identifier is defined as follows: *gti_tbl_inspection_long_en*
   In this case, the table is filled starting from row 5.
   b. If you want to fill the table from a different row, enter: *gti_tbl_inspection_long_en|row number*
   The line number must be a number ≥ 3 because the first two lines are reserved and hidden.

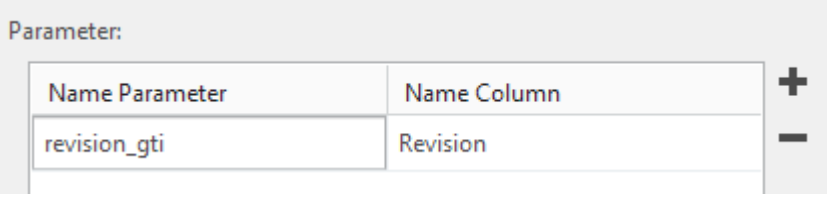
**Adjust formatting**

7. In the first empty value row, set the formatting you want to use, e.g., center-aligned instead of left-aligned.


© 2026 INNEO Solutions GmbH

**Insert and name columns**

8. In the *Table* tab, go to *Add Column* and click with the mouse on the line of between two columns. The new column is created to the right of it.
9. Write the desired variable in the second line of the new column. This line becomes invisible when the table is placed by using *Inspect*. You can either
   – use variables that are predefined by *Inspect* (see table[210]) or
   – use your own variables which have been defined in the variable text[221] of the symbol. In this case enter here the entry in *Name column* as spefied in the *Inspect Editor* under Inspection Characteristics and Tables > Details > Parameters > Name Column[208].




*Variable (here: Revision) in "Name Column" in the settings for parameter in Inspect Editor*

10. Write in the fourth line the name that the column shall display in the table.
11. Add further columns.

**Deleting Columns**

12. Delete existing columns by selecting them (left-click and drag the mouse over them) and press the *Delete* key.

**Note:** Make sure that your table does not contain empty columns. *Inspect* is unable to read tables with empty columns.

**Save changed table**

13. Mark the entire finished customized table by left-clicking and dragging the mouse over it.
14. In the *Table* tab, click on *Save as Table* and save the table under a new name. It is recommended to use the table identification. Creo creates a TBL file.

**Storage location**

15. To make this TBL file permanently available, you must store the file in the Caddepot of the synchronization server in the inspect folder. (The inspect folder in the Cadpool will be overwritten by Caddepot after the synchronization pause.)
16. Start the synchronization again in the *GENIUS TOOLS Starter App* in the user menu with *Synchronize Now*.


© 2026 INNEO Solutions GmbH

# 10.4 Inspect Revision

*Inspect Revision* allows you to create a snapshot of all inspection symbols on a drawing at a given time. Using a drawing revision parameter you can define the revision level of a drawing and generate a history of all revisions. Revision histories can be exported to Excel.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## 10.4.1 Fundamentals

### Glossary

**Revision**
Reviewed state of a technical document.

**Revision status**
Change of a revision identified by a number, letters and / or date.

**Drawing revision parameter**
Parameter indicating the revision status of a drawing.

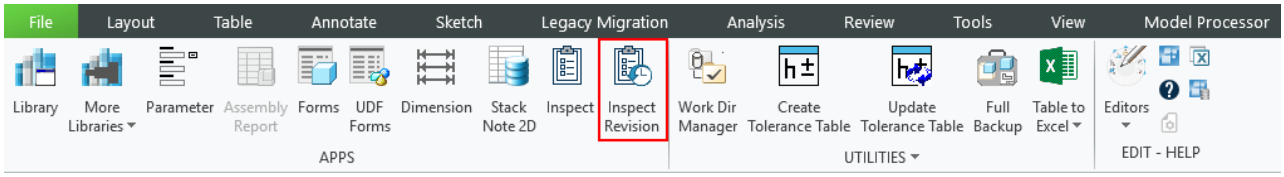
**Snapshot**
A snapshot shows all changes to inspection-relevant properties. This allows you to see whether, for example, the values of dimensions with inspection characteristics have been changed.

**Snapshot history**
Overview of all taken snapshots. Snapshot history can be exported.

## 10.4.2 Starting the program

### Starting the program: in drawing mode

To start *Inspect Revision*, click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section.

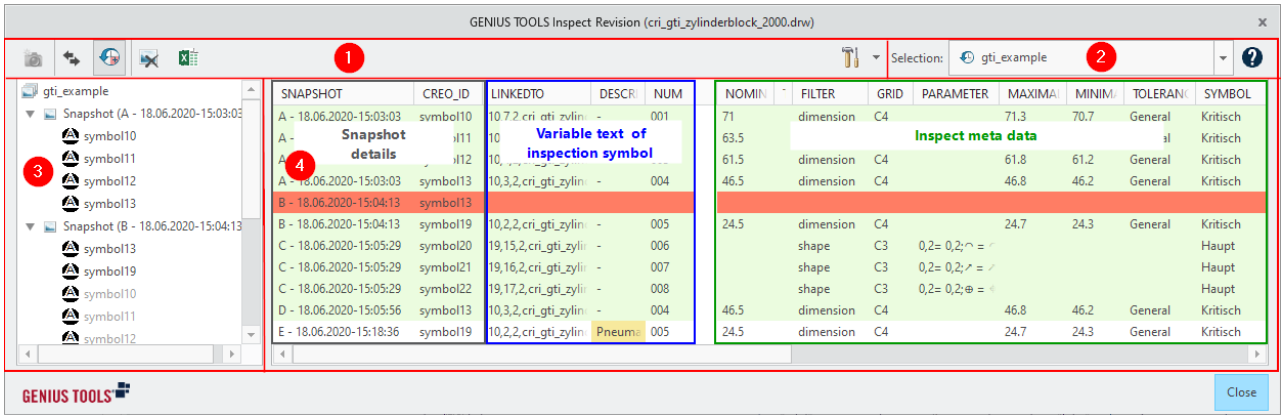



*Starting via the GENIUS TOOLS tab in the ribbon menu*


© 2026 INNEO Solutions GmbH

## 10.4.3 User interface

The user interface of *Inspect Revision* / *Inspect 3D Revision* contains the following elements:




1. Command bar[232]
2. Select configuration file[233]
3. Symbol / snapshot tree[234]
4. Overview of snapshots with
   - Snapshot details:

     Drawing revision parameter[238],
     Time of creation of snapshot and
     Creo-ID of modified symbol. (Creo-ID is the identity number assigned by Creo.)
   - Variable text of the inspection symbol.

     Variable text[221] is data stored in the SYM file of the symbol.
   - Inspect metadata:

     All data extracted by *Inspect* / *Inspect 3D*.

## 10.4.4 Command bar

The command bar contains these control elements:

| Icon | Name | Description |
|---|---|---|
|  | Create snapshot[235] | Displays a snapshot of all inspection symbols on a drawing |
|  | Switch tree representation[234] | Switches display between symbol tree and snapshot tree |


© 2026 INNEO Solutions GmbH

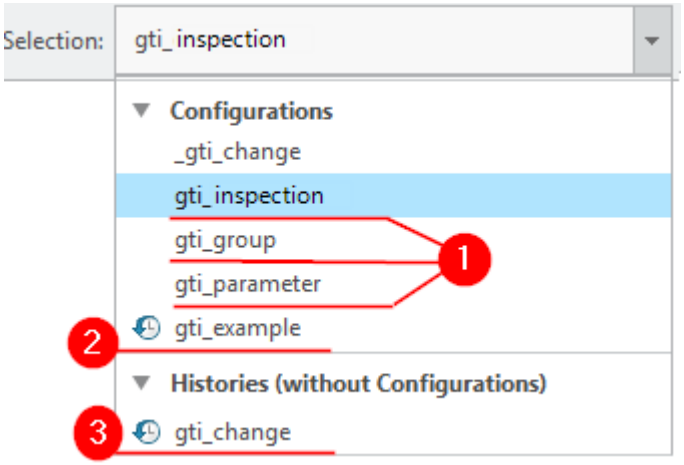
| Icon | Name | Description |
|---|---|---|
|  | Show complete state | Shows all existing inspection characteristics at the time of the respective snapshot |
|  | Delete latest snapshot | Deletes the latest snapshot |
|  | Export history to excel240 | Exports snapshot history as XLSX file |
|  | Tools | Contains supportive functions:<br>- Save values to XML |
|  | Configuration / history233 | Select configuration file and snapshot history |
|  | Help | Opens help page for *Inspect Revision / Inspect 3D* |

## 10.4.5 Selecting a configuration file

The command bar contains a field for selecting a configuration. A configuration is an XML-file that contains symbols, table definitions and display settings and that is created in *Inspect Editor*205. The clock symbol next to the name  indicates whether the configuration file contains snapshot data, i. e. a history.

Snapshot data is data stored in a Creo file (DRW, PRT, ASM). They can be exported from the XML file but also as stand-alone data without data of the XML file. Thus you may send a model with an inspection history without revealing the configuration settings.


© 2026 INNEO Solutions GmbH




*Selection of configurations in Inspect Revision*

**Configuration without history (1)**
No data in Inspect Revision dialog
**Configuration with history (2)**
Snapshot history displayed in Inspect Revision dialog
**History without configuration (3)**
Snapshot cannot be created

## Save history as XML file

A history of snapshots can be exported without the corresponding configuration settings by saving a separate XML file in *Tools* > *Save history as XML file*.

## Configuring the selection field

The configuration file is by default set to *gti_inspection*. The displayed default file can be changed in the configuration option `gti_start_file`.

## 10.4.6 Selecting the tree view

You can expand the snapshot / symbol tree by clicking on the arrow symbol and switch between the two tree displays by clicking on the symbol :

1. **Snapshot view:** Lists all snapshots and the corresponding inspection symbols
2. **Symbol view:** Lists all inspection symbols and the corresponding snapshots

The selection in the snapshot / symbol tree display area determines how snapshot data is displayed. These are the possibilities:

– Select the configuration file: displays complete history

– Select a snapshot (e.g. Revision A): displays a single snapshot


© 2026 INNEO Solutions GmbH

- in snapshot tree: with all modified inspection symbols
- in symbol tree: of selected inspection symbol

– Select an inspection symbol (e.g. symbol 24): displays a single inspection symbol

- in snapshot tree: of selected snapshot
- in symbol tree: with all snapshots

**Please note:** When using *Inspect 3D Revision*, all inspection symbols that are available in the currently selected combined view are displayed. Inspection symbols available in other combined views are grayed out in the snapshot tree.

## 10.4.7 Creating snapshots

The snapshot function  maps all inspection symbols on a drawing / in a model and creates a time stamp. In addition, *Inspect Revision* / *Inspect 3D Revision* can extract values from a Revision parameter[238].

**Tip:** A snapshot corresponds to the revision status, if the revision parameter has been changed for the snapshot.

When a snapshot is taken, all inspection symbols stored in the configuration file are compared to the previous snapshot. A snapshot can be created when at least one inspection symbol has been altered.

The following modifications are included in a snapshot:


© 2026 INNEO Solutions GmbH

| Modification | Color/ example |
|---|---|
| 1. A new inspection symbol was placed (symbol does not yet exist in any snapshot) | green row<br>A - 18.06.2020-15:( symbol12 10,4,2,cri_gti_zylinderblock_ |
| 2. A new value was added to an inspection symbol | green cell<br>General |
| 3. A value was edited in the model | yellow cell in Inspect metadata columns (3)<br>cri_gti_zylinderblo Dimension ISO/DIN |
| 4. A value on an inspection symbol was altered (e. g. in variable text) | yellow cell in Variable text columns (2)<br>10,3,2,cri_gti_zylin Pneumatic length measuring 004 |
| 5. An inspection symbol was deleted (symbol is in snapshot but not in the drawing / in the model) | red row<br>B - 18.06.2020-15:04:13 symbol13 |
| 6. An unchanged symbol from an older snapshot  | greyed out text<br>A - 09.02.2021-07:( symbol12 |

# Recorded inspection symbols

A snapshot unhides all hidden inspection symbols. This means that symbols without a target will be deleted in the process, see also Hide and unhide 201.

A snapshot does not capture all inspection symbols and will blend in all hidden symbols as follows:

| State of inspection symbol | Captured by snapshot? |
|---|---|
| Unnumbered inspection symbols (000 symbols) | no |
| Inspection symbols whose target was deleted | no<br>**Warning:** Inspection symbols, whose target was deleted, will be deleted. |


© 2026 INNEO Solutions GmbH

| State of inspection symbol | Captured by snapshot? |
|---|---|
| Hidden symbols | |
| – with a linked target | yes, are shown again in the drawing / mode |
| – without a target | no<br>**Warning:** Inspection symbols, whose target was deleted, will be deleted. |

## Example: Creating a snapshot of a revision level

**Steps:**

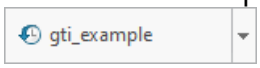
1. Place and number the needed inspection symbols.
2. Assign a new value to the revision parameter in *Tools* > *Parameter*. In the example: CAD_REVISION = D
3. In the command bar click the button *Create snapshot*  .

**Result:**

The snapshot will be displayed in the left section of the dialog (symbol- / revision tree). In the example: Snapshot (D - 15.06.2020 - 13:37:41)

## 10.4.8 History of snapshots and symbols

After the first snapshot is taken, a clock icon appears next to the configuration file.

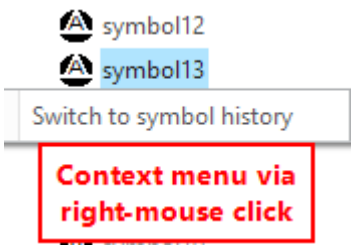



Configurations with a clock have a snapshot history, which is the sum of all snapshots. This history can be exported.[240]

**Tip:** You can get a history of revisions by mapping them with a snapshot after completing a new revision.

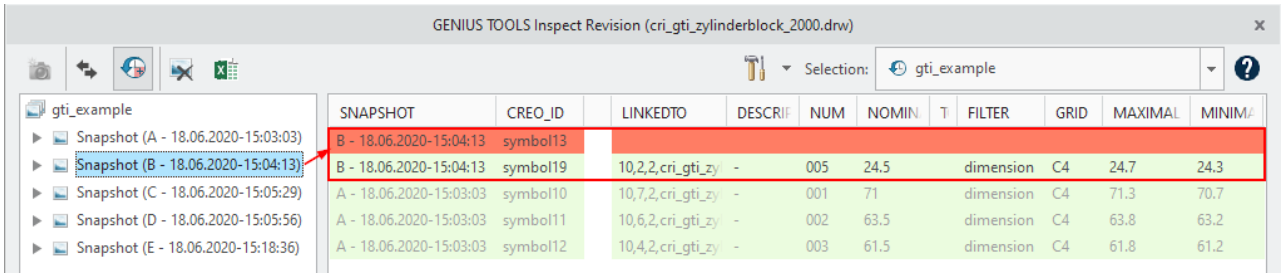
You can view a history by clicking on a snapshot or an icon.

You can use the  button to switch to the symbol tree or click directly on a symbol to switch to its history using the context menu.

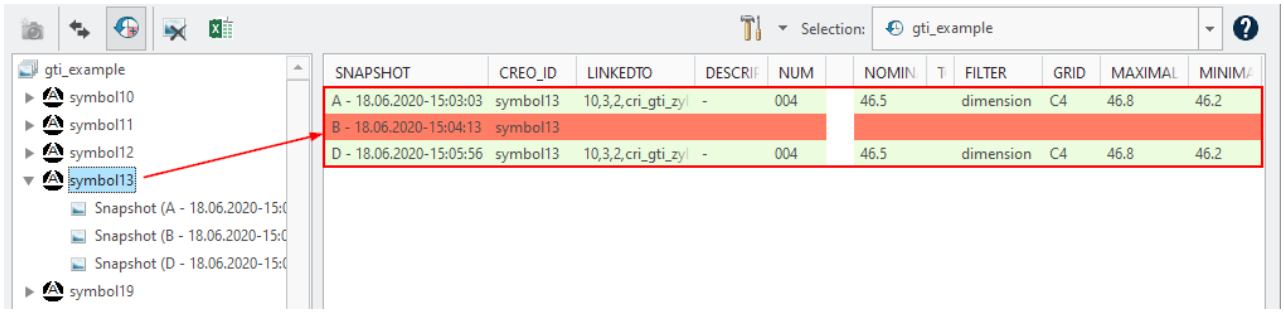




© 2026 INNEO Solutions GmbH




*History of snapshot of revision B*




*History of symbol 13*

## Revert symbol

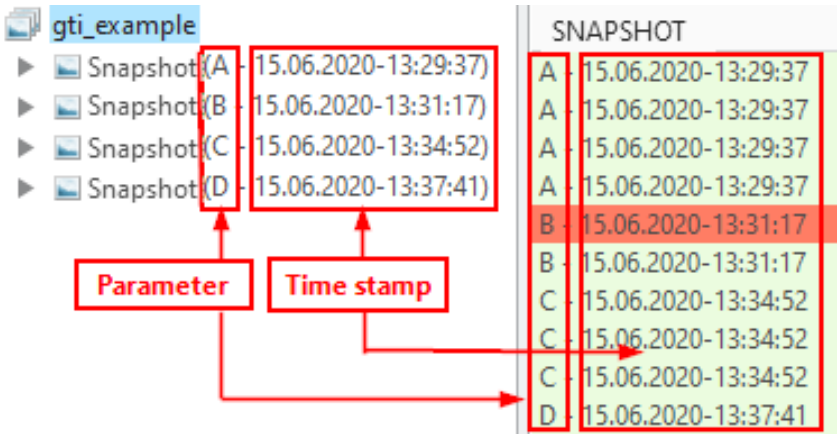
If you have removed a symbol from the drawing, you can restore it. The removed symbols are marked in orange in the table. Right-click on the corresponding row and then click *Revert Symbol* to restore the symbol.




## 10.4.9 Revision parameter

The revision parameter informs about the revision level of a drawing / model. A new parameter value should be assigned after any test-relevant change to the inspection symbols or their values. (In the example below: A - D.) If you do not work with a revision parameter, you can still distinguish snapshots by their time stamp.

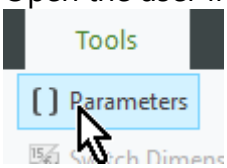

© 2026 INNEO Solutions GmbH



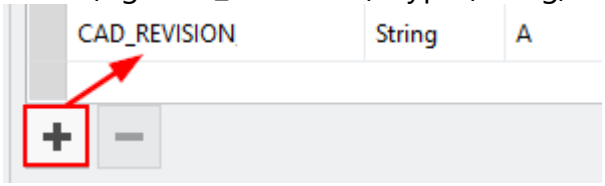

*Details of snapshots in the snapshot tree display*

There are different ways to use a revision parameter in *Inspect* / *Inspect 3D*:

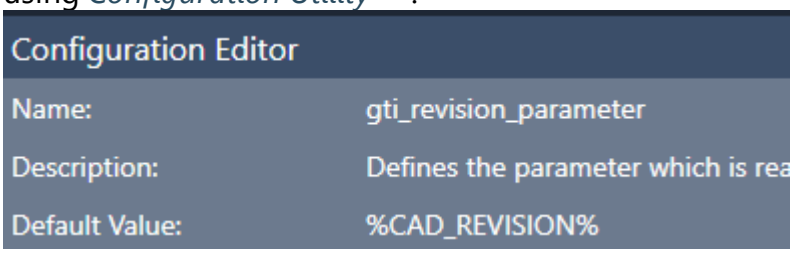
1. **Using a revision parameter from Startup TOOLS**
   If you work with *Startup TOOLS*, you use a preconfigured start part. The start part contains the revision parameter CAD_REVISION. The configuration option `gti_revision_parameter` is set to this by default. If you are creating inspection symbols on the drawing / in the model for the first time, CAD_REVISION is automatically set to A and then read.
2. **Using a revision parameter from Windchill**
   A revision parameter is provided by Windchill and must be specified in the configuration option `gti_revision_parameter`.
3. **Generating a revision parameter yourself**
   **a.** Open the user interface *Parameter* via the Creo ribbon menu *Tools*:




   **b.** Add a new parameter using the *Plus* Button (*Add new parameter*). Enter the parameter name (e.g. CAD_REVISION), Type (String) and Value (A):




   **c.** Enter the new parameter name in the configuration option `gti_revision_parameter` using *Configuration Utility*[765]:





© 2026 INNEO Solutions GmbH

## 10.4.10 Export snapshot history

The history of snapshots can be saved to an XLSX file using a template.




*Excel button in Inspect Revision*

The button opens the dialog Export table to Excel[690] which allows you to select the Excel template, the file to be exported and the snapshot history.

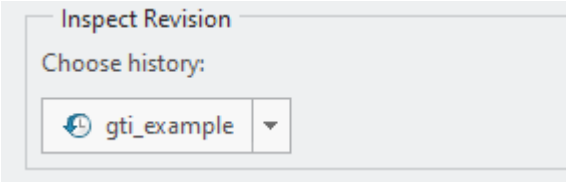
Starting the Excel export may take some time due to checking the Excel template.

**Warning:** Close all Excel windows before starting the export function.

## Export dialog

The supplied template `gti_revision_template_de_en.xlsx` (in the resource directory *gt_resource_folder\inspect*) is selected by default. The file name is generated from the name of the drawing / model.

The selected configuration determines the inspection symbols whose snapshot history is exported.




## Configuring export

Templates are controlled with these configuration options:

`gti_revision_excel_template` defines the name of the default Excel template for export. Default: gti_revision_template_de_en.xlsx

`gti_revision_folder` defines the folder in which the default Excel template is searched for. Default: %gt_resource_folder%/inspect

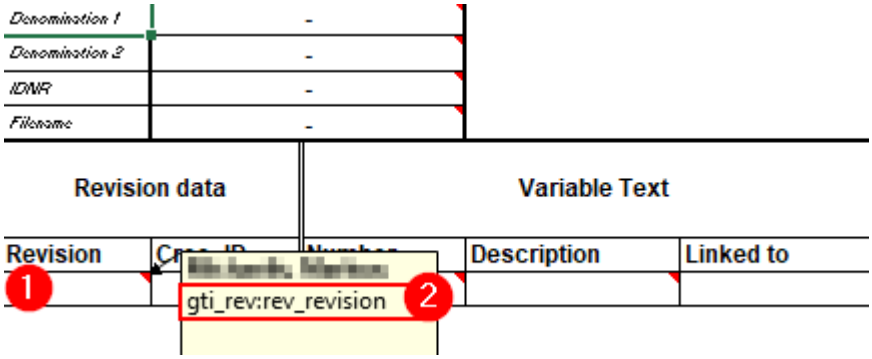
`gti_revision_excel_coloring` defines whether the coloring of the data view in the snapshot data area is also transferred to the XLSX file. Default: 1=yes

## Customizing an export template

You can customize a template by taking the template file *gti_revision_template_de_en.xlsx* from the directory *gt_resource_folder%/inspect* as a basis and adding a comment (2) to the


© 2026 INNEO Solutions GmbH

first line (1) of the column you wish to adapt. The chapter Create template[693] describes how to set up an export template step by step.




*Acronym gti_rev: in the comment that assigns a column*

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.

The code in the comment defines the values to be copied into a column. It consists of the component acronym *gti_rev:* and the fill command. Use the following codes:

| Command code | Column name |
|---|---|
| gti_rev:rev_revision | Revision |
| gti_rev:rev_id | Creo_ID |
| gti_rev:var_num | Number |
| gti_rev:var_descr | Description |
| gti_rev:var_LinkTo | Linked to |
| gti_rev:num_sym | Number of inspection symbol |
| gti_rev:num_sym | Symbol number |
| gti_rev:tpe_main | Main type |
| gti_rev:tpe_sub | Subtype |
| gti_rev:cls_tol | Tolerance class |
| gti_rev:bse_dim | Nominal dimension |
| gti_rev:min_dim | Minimal dimension |
| gti_rev:max_dim | Maximal dimension |
| gti_rev:val_tol | Tolerance |


© 2026 INNEO Solutions GmbH

| Command code | Column name |
|---|---|
| gti_rev:descr | Description |
| gti_rev:gti_param | Parameter |
| gti_rev:gti_note | Note |
| gti_rev:tpe_tol | Tolerance standard |
| gti_rev:num_sheet | Sheet |
| gti_rev:grd | Grid |
| gti_rev:src | Source |
| gti_rev:mod | Tolerance table |
| gti_rev:nme_sym | Name of symbol |
| gti_rev:tpe_sym | Creo symbol |
| gti_rev:var_<parametername> | Output of additional user-defined parameters |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gti_rev:*. Thus you can export data from *Inspect (3D)* and *Inspect (3D) Revision* together, see Export data from several GENIUS-TOOLS-components[701] for an example.


© 2026 INNEO Solutions GmbH

# 11 Inspect 3D

Available in the ribbon menu GT MBD

Use *Inspect 3D* to add inspection symbols to parts and assemblies in Creo Parametric.

*Inspect 3D* is available in part mode and assembly mode with the following features:

1. Placing of inspection symbols linked to
   - dimensions
   - shape and position tolerances
   - surface quality symbols
   - notes
   - symbols
   - linked elements
2. Numbering of inspection symbols
   - by axes
   - by symbol type
   - similar to DIN 6770 (numbers are not assigned anew)
3. Exporting data to Excel

**Warning:** To use *Inspect*, the configuration `PRO_SYMBOL_DIR` must be editable. However, if `PRO_SYMBOL_DIR` is included in the Creo file *config.sup*, `PRO_SYMBOL_DIR` cannot be edited. In this case, *Inspect* will not work.

The component *Inspect Revision 3D*[281] creates a snapshot of all inspection symbols in a part or assembly at one point in time, as well as a history of all snapshots.

You can also use symbols in *Inspect 3D* to mark changes on a part or on an assembly[226].

## 11.1 Fundamentals

### Glossary

**Inspection symbols**
Numbered symbols marking characteristics that have to be considered in quality control because they are critical to the quality of function of the finished part or product.


© 2026 INNEO Solutions GmbH

**Inspect configuration**
XML file that contains definitions for symbols and display settings. (See also Inspect configuration[207].)

## Structure of an inspection symbol

The figure below shows an inspection symbol generated with *Inspect 3D* with its individual components in the front view.

Each inspection symbol (numbered symbol) marks a characteristic in a part or assembly.

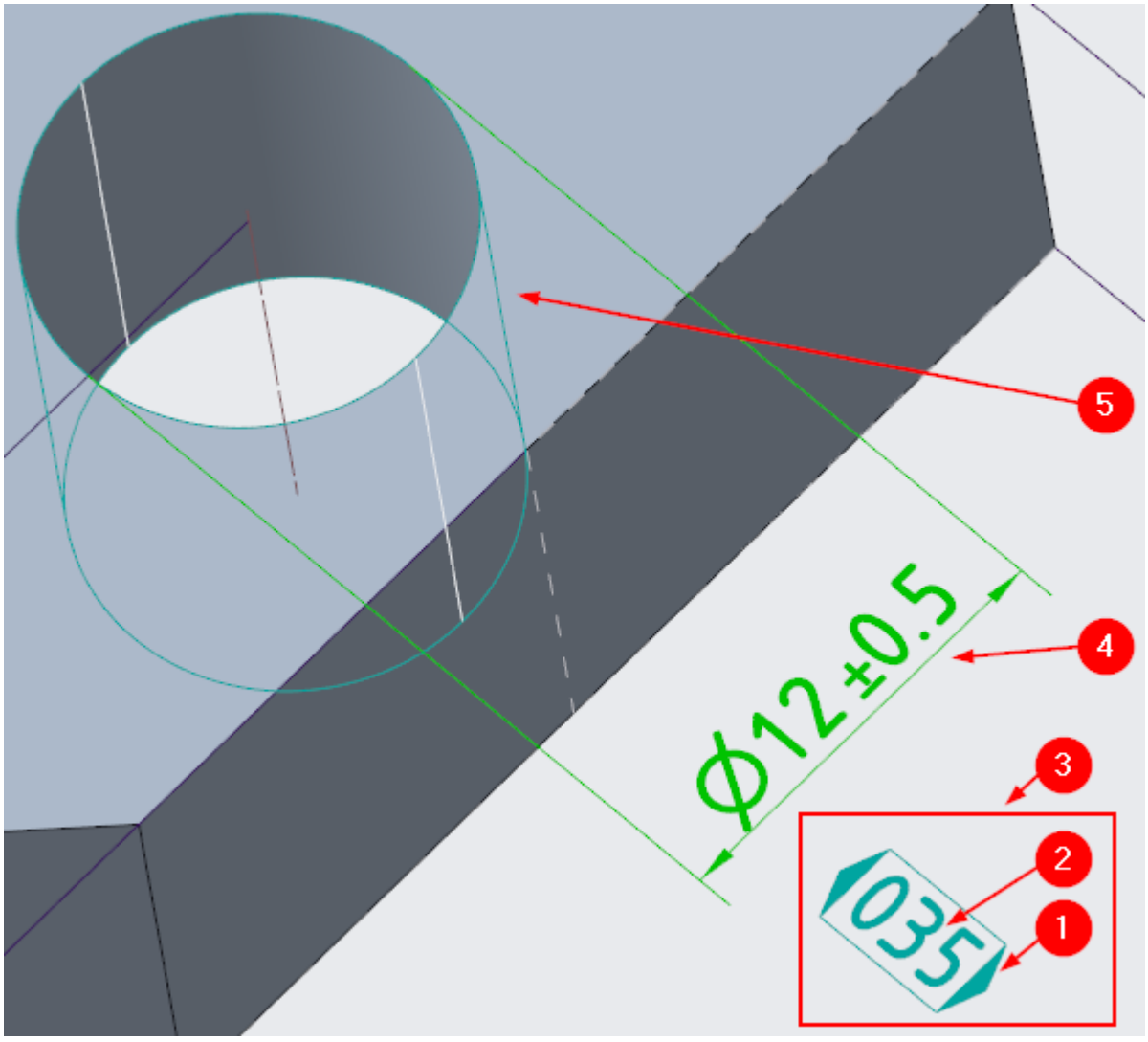
1) symbol

2) number of the characteristic

3) inspection symbol, consisting of the symbol (1) and the number of the characteristic (2)

4) value of the characteristic

5) characteristic: highlighted dimension of the geometry element





© 2026 INNEO Solutions GmbH

## Font used for out-of-the-box symbols

The symbols delivered with *Inspect 3D* use the font *ISONORM LT Regular* based on ISO 3098-5.

If you experience issues with displaying the out-of-the-box symbols, please check whether this font is available in your system.

You can create your own symbol files to use as inspection symbols. For more information on how to do this, please refer to Creating a Creo symbol for Inspect 220.

## Integrating additional texts according to DIN EN ISO 14405-1

You can use the Creo menu *Model Properties* > *Detailing* to set that symbols can also contain texts in accordance with DIN EN ISO 14405-1. The following settings are required for this:

| Option | Value |
|---|---|
| default_annotation_font | isonormlt-regular.ttf |
| symbol_font | ISO |

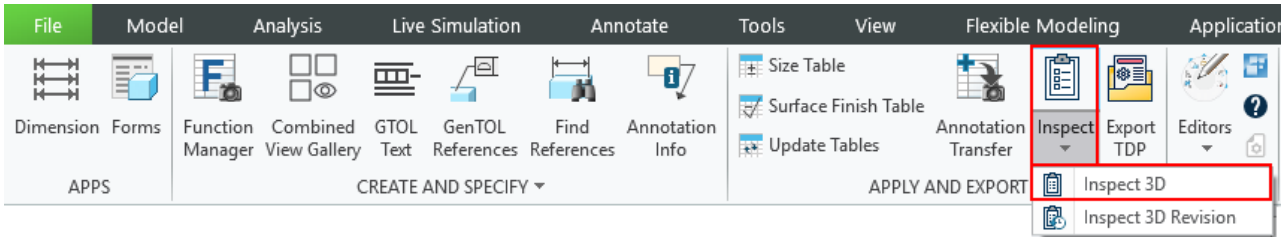
**Tip:** The detail option symbol_font = legacy is not supported.

# 11.2 Usage

This section contains information on using *Inspect 3D*. It describes the general structure of the program.

## Starting the program: in part / assembly mode

To start *Inspect 3D* , click *GT MBD* in the ribbon menu and go to *APPLY AND EXPORT* section.

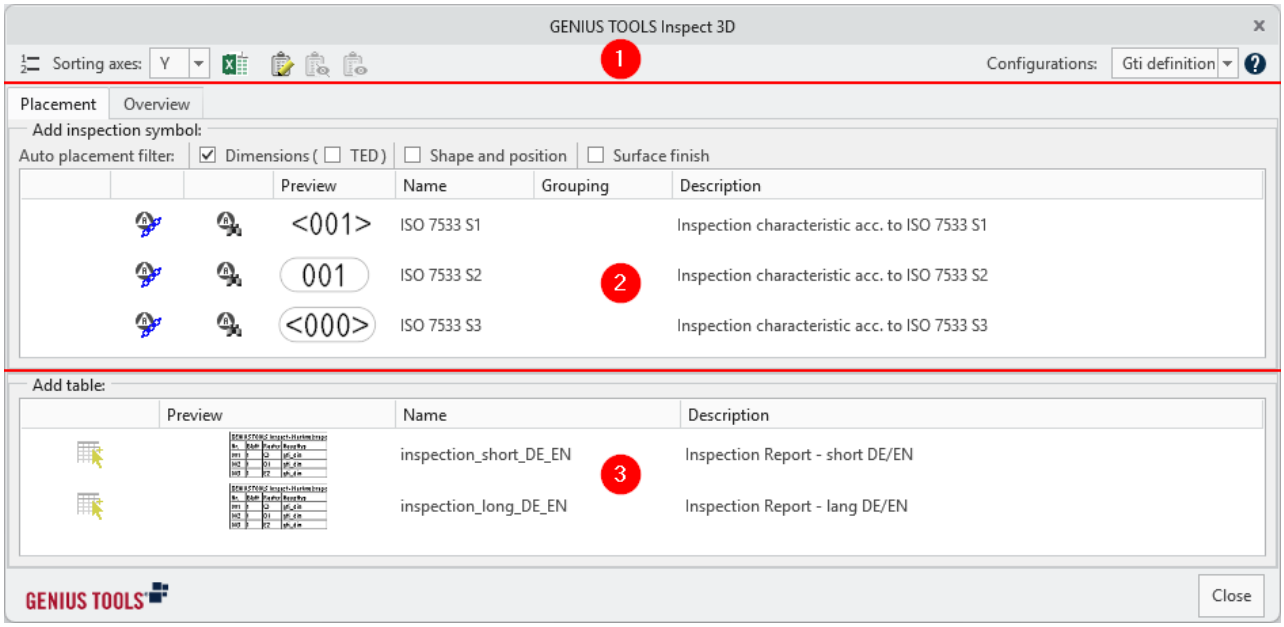



*Starting via the GT MBD tab in the ribbon menu*


© 2026 INNEO Solutions GmbH

## 11.2.1 User interface

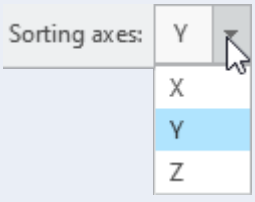
The user interface of consists of *Inspect 3D* the following elements:




1. Command bar[246]
2. Symbol overview: divided in two tabs, Placement and overview[247]
3. Table overview[247]

## 11.2.2 Command bar

The command bar displays general control elements. The following buttons are included:

| Icon | Name | Description |
|---|---|---|
| | Number symbols | Numbers all placed symbols. The numbering of symbols is determined by the configuration options[825] `gti_number_sort_at_height` and `gti_start_number`, see Numbering[250]. |
|   | Sorting Axes | Select the axis according to which the inspection symbols are created and sorted. The last selected sorting axis is saved and preselected when the module is restarted. |
| | Export overview to Excel | Opens the export dialog for the symbol table[254]. |


© 2026 INNEO Solutions GmbH

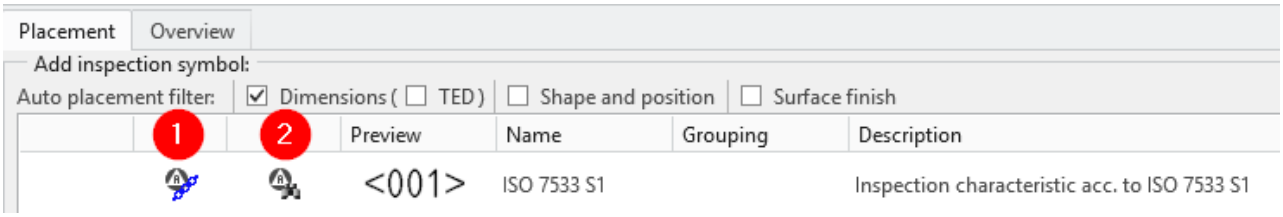
| Icon | Name | Description |
|---|---|---|
|  | Open Inspect Editor | Opens the *Editor*[206]. |
|  | Configuration selection | Switches between multiple Inspect configurations that can contain different symbol and table definitions as well as view settings. |
|  | Open Help | Opens the help. |

**Please note:** Always place[247] all symbols before numbering them.

**Warning:** Do not use the same symbol in different Inspect configurations if you are using numbering similar to DIN 6770. Otherwise, you may experience numbering conflicts.

## 11.2.3 Placing

The tab *Placement* is used to place and link inspection symbols. The displayed columns, the order of the columns, and the order of the inspection symbols are defined in the *Editor*[205].




Available inspection symbols are displayed under *Add inspection symbol*. Use *Auto Placement Filter* to specify which inspection characteristics to add.

**Please note:** *Auto placement filter*




can only be used for **linked insertion with previous selection**.

**Linked insertion (1)**  **:** Inspection symbols can be linked to displayed dimensions, shape and position tolerance symbols, surface quality symbols and notes. These inspection symbols will be attached to the target item with equal alignment.


© 2026 INNEO Solutions GmbH

– **Simple linked placement:**
Click the button  , then select an item using the left mouse button. Now place the inspection symbol using the left mouse button. You can place several symbols, then stop the process by clicking the middle mouse button.
Use the configuration option `gti_activate_simple_linked_auto_placement` to control whether the symbol is automatically placed (*Automatic insertion* placement). Default: 1 (=yes)
– **Linked placement by previous selection (single elements):**
Select one or more elements in the model. Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. You can then automatically insert inspection symbols by clicking the button  . These inspection symbols will be attached to the target item with equal alignment.
– **Linked placement by previous selection (multiple elements):**
Select multiple elements or an area in the model. Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. Then you can automatically insert inspection symbols by pressing the button  . Inspection symbols will be linked to all dimensions in the selection. These inspection symbols will be attached to the target item with equal alignment.

**Automatic insertion (2)**  : Inspection symbols are automatically placed on all unlinked items (dimensions, shape and position, surface) in all active combined views. Under *Auto Placement Filter*, mark the checkboxes for the inspection characteristics you want to add. Items which are already linked to the selected inspection symbol are not re-linked. *Inspect 3D* checks whether the SYM file of the selected inspection symbol matches the existing inspection symbol. These inspection symbols will be attached to the item with equal alignment.

No inspection characteristics are created for TEDs.

**Please note:** Automatic insertion applies inspection symbols only to displayed dimensions.

**Please note:** If some inspection symbols were not placed during linked or automatic insertion, check if a placing exception rule[194] has been defined and delete it.

## 11.2.3.1 Placing with exceptions

You can define exceptions for **placement by previous selection**[193] and **automatic placement**[193] to exclude target elements with certain phrases from placement. The exceptions are defined in the configuration option `gti_placement_exception_rule`. The configuration option has no default setting, so by default, no target elements are excluded


© 2026 INNEO Solutions GmbH

from placement. You can enter fixed formulations, Creo special characters or Regular Expressions[487].

**Please note:** First define the exception in the configuration option `gti_placement_exception_rule`. Then place the inspection symbols. Inspection symbols already placed are not removed by this configuration option, but must be deleted manually.

## Use cases

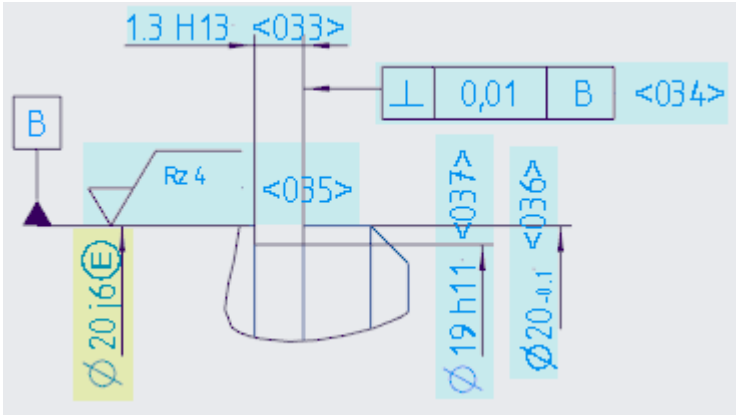
Possible phrases to exclude target elements from the positioning:

**Excluding special characters from placement**

– Setting with color coding: Ⓔ

– Explanation of the setting:
Ⓔ was entered as a Creo special character. The special characters must be copied and pasted from Creo Parametric and inserted into *Configuration Utility*[765]. The special characters are usually not resolved, but the filter works.
Entered special characters are displayed correctly in the configuration overview *GENIUS TOOLS for Creo Settings*[763].

– Result in the user interface:




*Excerpt from gti-antriebswelle.drw*
*green=target element with not set inspection characteristic*
*blue=target element with set inspection characteristic*

**Excluding clamp dimensions or dimensions (incl. ordinate dimensions) with the value 0 from placement**

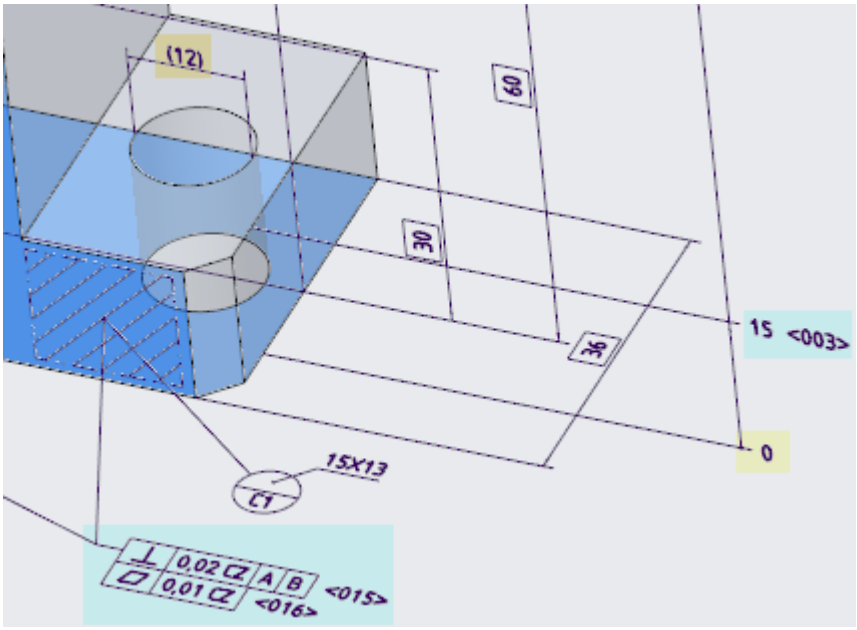
– Setting with color coding: `^\(.+\)$|^0$`

– Explanation of the setting:
The exception was entered as Regular Expression[487] with 'or' operator. One of two exceptions is applied:


© 2026 INNEO Solutions GmbH

Either no inspection characteristics are placed on target elements with clamps, or no inspection characteristics are placed on target elements with a value of 0.

– Result in the user interface:




*Excerpt from lagerbock-mbd-gps-v4-col, B01-Master*
*orange=target element with inspection characteristic not set due to dimensioning in clamps*
*yellow=target element with inspection characteristic not set due to ordinate dimensioning 0*
*blue=target element with set inspection characteristic*

## 11.2.4 Numbering

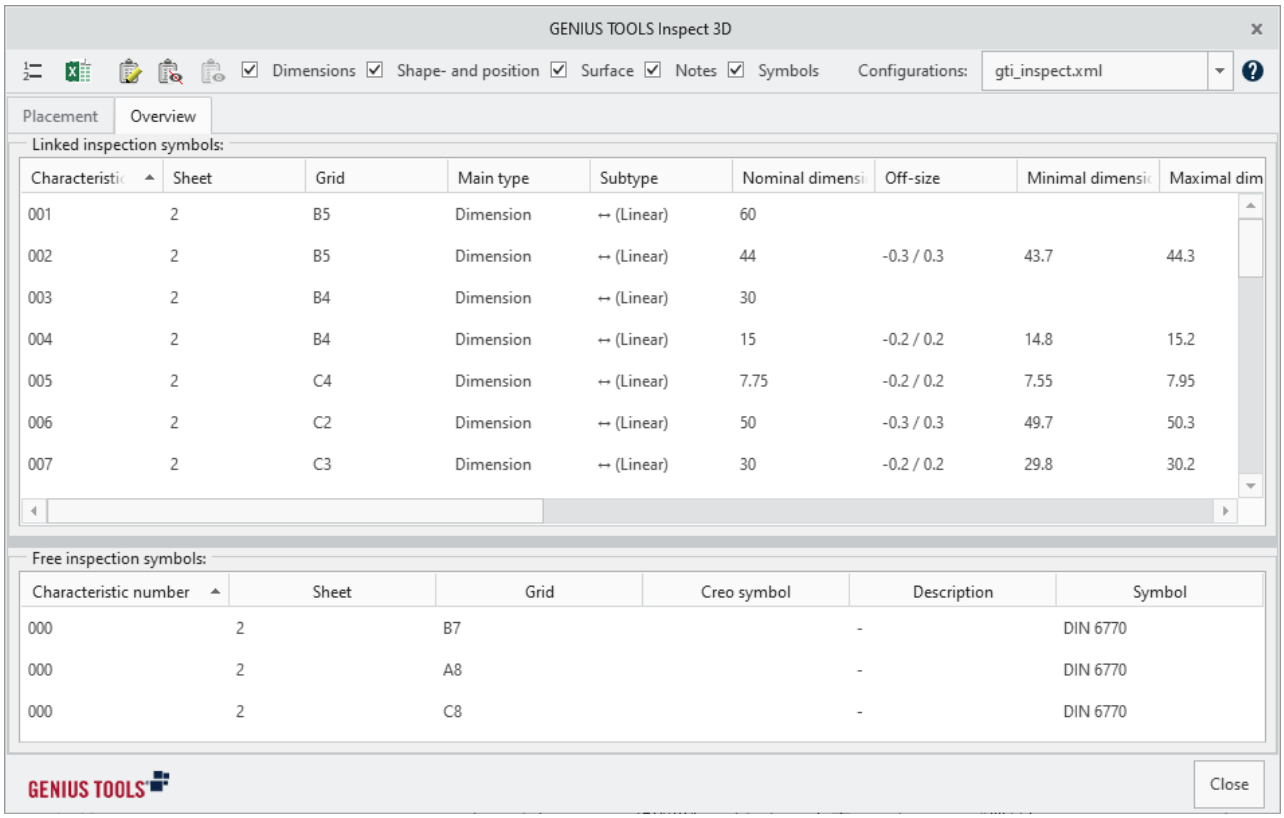
The following section describes the numbering options. The numbering of the inspection symbols is influenced by the following configuration options:

| Configuration option | Input value | | |
|---|---|---|---|
| `gti_number_sort_at_height` | -1 (descending numbering sequence) | 0 (no sorting) | 1 (ascending numbering sequence) |
| `gti_start_number` | Numerical value for numbering start (used only when `gti_din_compliant=0` is set) | | |

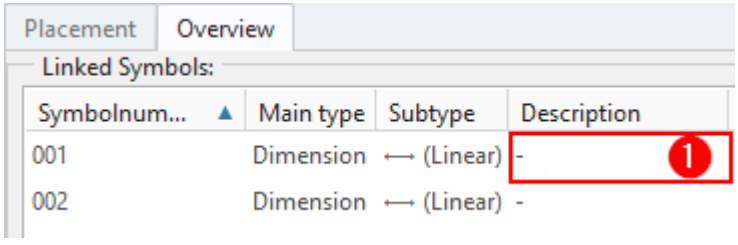

© 2026 INNEO Solutions GmbH

## 11.2.5 Overview

The *Overview* tab displays all previously placed inspection symbols and relevant data for these inspection symbols in a table. Additional information (e.g. tolerance information) is displayed for linked inspection symbols. The columns displayed, the column order, and the order of the inspection symbols are defined in the *Editor*[205] .




**Linked inspection symbols**
Use the filters in the top bar to show and hide inspection characteristics linked to dimensions, shape and position tolerances, surface qualities, notes or inspection symbols. The out-of-the-box inspection symbols include a description parameter. Its content is displayed in the *Description* column and can be changed. A left click in the description column opens an input dialog for entering a description. This description is saved in the inspection symbol and will be available the next time *Inspect* is started.




The description field is variable text[221] , which is saved in a Creo symbol.

**Tip**: When you are in an input field, you can use *Tab* to move to the next input line.


© 2026 INNEO Solutions GmbH

You can add information (e.g. from a revision parameter) by creating a new column. In *Inspect Editor* go to the tab Inspection symbol > Parameters[208] .

**Please note:** Use the configuration option `gti_lang` to define - not depending on the system language - the language used for the contents of the overview table.

## Filtering inspection symbols

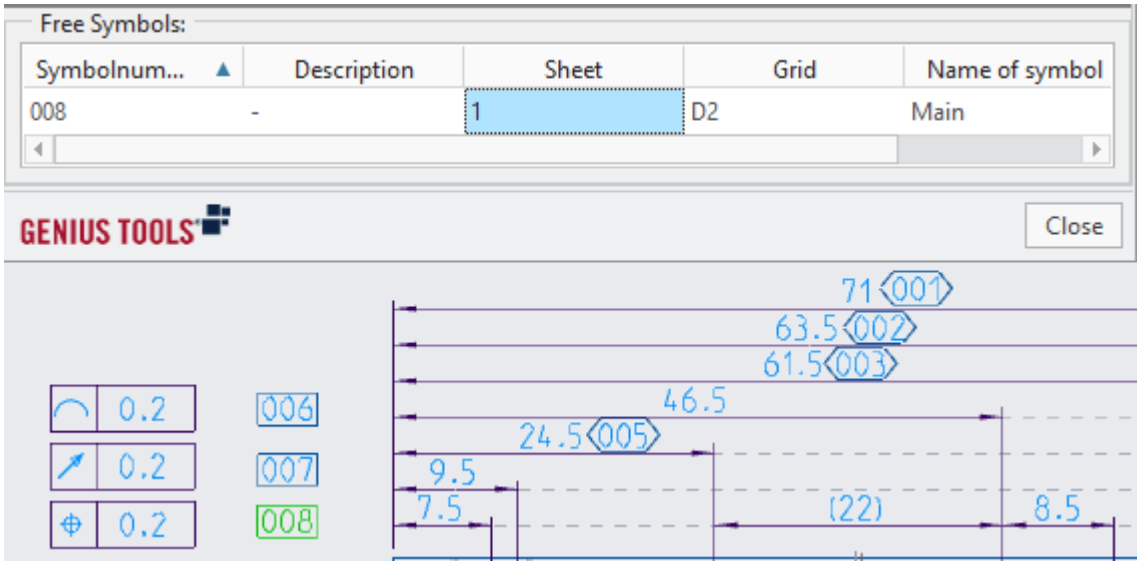
Use the filters in the top bar to show and hide inspection symbols linked to dimensions, shape and position tolerances, surface qualities, notes or symbols. If you hide inspections symbols that are linked to a defined element type (e.g., dimensions or tolerances), columns that are only relevant for this type will also be hidden.

## Sorting inspection symbols

By clicking on a column header, the inspection symbols are sorted according to this column. By default, the symbols are sorted by inspection symbol number (ascending).

## Highlighting inspection symbols

A click on any cell in an inspection symbol's row highlights the inspection symbol in the drawing.





© 2026 INNEO Solutions GmbH

## 11.2.6 Hiding and unhiding elements

### Hiding Inspect elements




To hide all *Inspect* elements, click the button shown above (1).

All *Inspect* tables and *Inspect* inspection symbols that contain the variable text `LinkedTo` are hidden.

Symbols without the variable text `LinkedTo` are not hidden. This can be the case for inspection symbols created with version of *Inspect* prior to 6.0. The variable text has to be added to these symbols if you want to use the hide functionality.

When hiding *Inspect* elements, linked inspection symbols are unlinked from their target elements. This does not turn them into free symbols – *Inspect* saves the attributes of the target element to the variable text `LinkedTo` to be able to re-link the inspections symbols when they are unhidden.

### Unhiding Inspect elements




To unhide all *Inspect* elements, click the button shown above (2).

All inspection symbols which were linked before hiding are re-linked to their target elements.

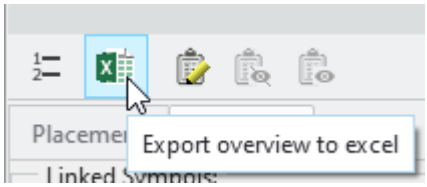
**Warning**: Inspection symbols whose target elements have been deleted are deleted on unhiding!


© 2026 INNEO Solutions GmbH

## 11.2.7 Export to Excel

### Exporting inspection symbols to Excel

To export your inspection symbols to Excel, click on the command *Export overview to Excel* . The dialog Export table to Excel[690] opens, in which you select the Excel template, export file and configuration.




Starting the Excel export can take some time because the Excel template has to be checked. Once the export is complete, the Excel file will open automatically.

**Warning:** Close all Excel windows before starting the export.

### Export dialog

By default, the out-of-the-box template is selected. The file name is generated from the name of the drawing. You can also design an individual Excel template[693].

The selected configuration determines which inspection symbols are exported. Inspection symbols will be exported if the selected configuration refers to their symbol file, i.e. if the configuration contains their inspection symbol type.




### Configuring path to template

The default selection of a template can be changed by using these options:

**gti_excel_template**
Defines the Excel template. Default: *gti_inspection_template_de_en.xlsx*

**gti_folder**
Defines the directory, which contains the Excel template. Default: <*gt_resource_folder*>

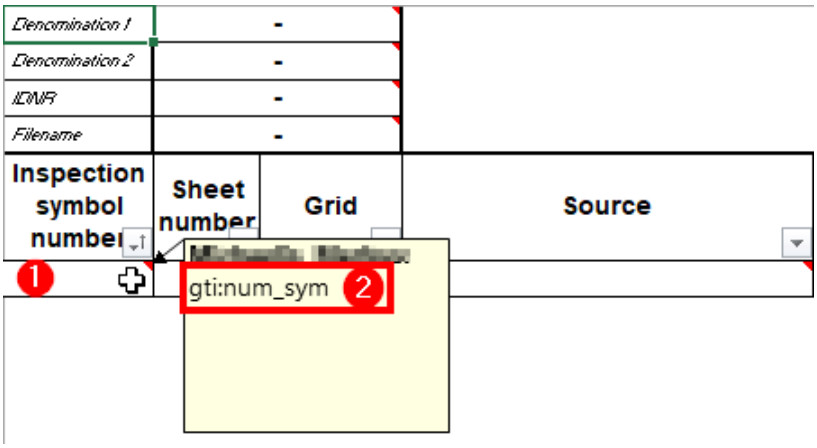
### Customizing the Excel template

If you want to create your own template, you can use the supplied template *gti_inspection_template_de_en.xlsx* from the directory *%gt_resource_folder%\inspect* as a


© 2026 INNEO Solutions GmbH

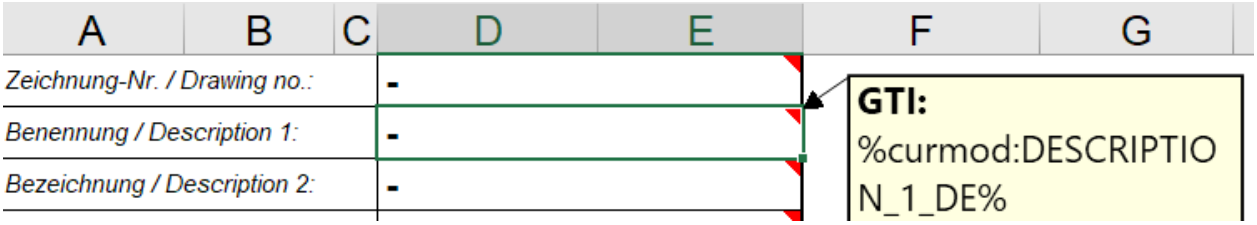
basis and add a comment (2) to the first value cell (1) of the desired column. The chapter Create template[693] describes how to set up an export template step by step.

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.



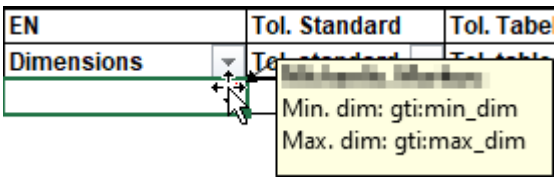

*Comment used for column assignment*

The text of the comment determines which values are to be copied into the column. Head parameters allow variables without module abbreviations, e.g. *%DRAWING_NO%, % curmod:DESCRIPTION_1_DE%.*

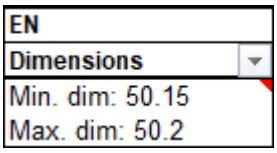



*Head parameter for Inspect*

During table export, all tags in a comment are searched and replaced with the appropriate value. The remaining text in the comment is preserved.




*Comment on the filling of the column with tags and additional text to be displayed.*




*Display in the exported table*


© 2026 INNEO Solutions GmbH

For report parameters, the text in the comment must consist of the component abbreviation *gti:* and a keyword:

| Comment text | Column name |
|---|---|
| gti:<columnName> | Output of additional user-defined parameters |
| gti:gtol_bottom_text | Bottom text |
| gti:val_tol | Boundaries |
| gti:tpe_sym | Creo symbol |
| gti:gtol_datum_references | Datum references |
| gti:descr | Description |
| gti:dim_value_text | Dimension text |
| gti:grd | Grid |
| gti:ipc_dim | Inspection dimension |
| gti:gtol_left_text | Left text |
| gti:low_tol | Lower boundary |
| gti:tpe_main | Main type |
| gti:max_dim | Maximal dimension |
| gti:min_dim | Minimal dimension |
| gti:nme_sym | Name of symbol |
| gti:bse_dim | Nominal dimension |
| gti:gti_note | Note |
| gti:gti_param | Parameter |
| gti:gtol_right_text | Right Text |
| gti:num_sheet | Sheet |
| gti:src | Source |
| gti:tpe_sub | Subtype |
| gti:num_sym | Symbol number |
| gti:tpd_dim | Theoretically precise dimension |


© 2026 INNEO Solutions GmbH

| Comment text | Column name |
|---|---|
| gti:cls_tol | Tolerance class |
| gti:tpe_tol | Tolerance standard |
| gti:mod | Tolerance table |
| gti:gtol_value | Tolerance value |
| gti:gtol_top_text | Top text |
| gti:upp_tol | Upper boundary |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gti*. Thus you can export data from *Inspect 3D* and *Inspect Revision 3D* together, see Export data from several GENIUS-TOOLS-components[701] for an example.

# 11.3 Configuration

This section contains information about the configuration of *Inspect 3D*, which is performed in *Inspect > Inspect Editor*[205].

## 11.3.1 Inspect Editor

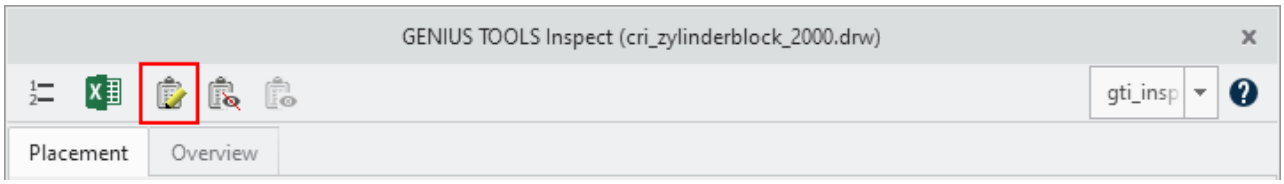
The *Inspect Editor* is used to create and manage settings for *Inspect* and *Inspect 3D*. Settings that are valid only for Inspect or only for Inspect 3D are also only applied in the corresponding module. If a setting for 3D is made in 2D or vice versa, it **cannot** be read, i.e., the setting has no effect.

Multiple groups of settings consisting of inspection symbols, tables, and view settings can be managed. Each configuration is saved as an XML file in the `gti_folder`.

Users can switch between different configurations in the program interface.

## Starting the program

Start the editor  from the command bar of *Inspect* or *Inspect 3D*.





© 2026 INNEO Solutions GmbH

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors 781.

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

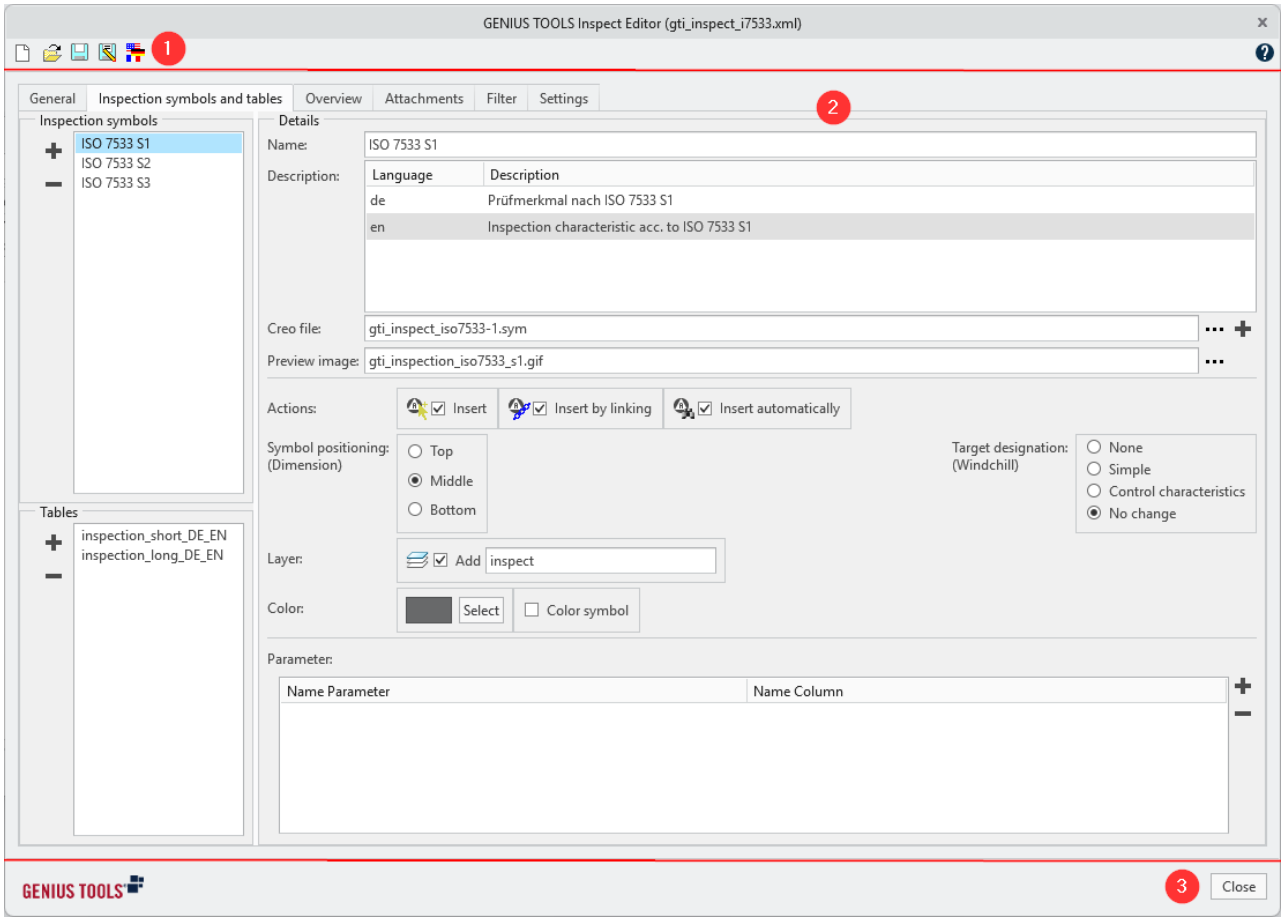
SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

### 11.3.1.1 User interface

The user interface of *Inspect Editor* consists of the following elements:





© 2026 INNEO Solutions GmbH

1. Command bar
2. Detail view of the tab *Inspection symbols and tables*
   The editor is divided into these tabs: Inspection symbols and tables[208], Overview[215], Attachments[215], Filter[216] and GT MBD.
3. Close

## 11.3.1.2 Command bar

The command bar contains the following buttons:

| Symbol | Name | Description |
|---|---|---|
|  | New configuration | Creates a new Inspect configuration[207] (XML). Each configuration can contain different symbol and table definitions as well as view settings. |
|  | Open configuration | Opens an existing configuration from an XML file. |
|  | Save | Saves the current configuration to an XML file. |
|  | Save as | Saves the current configuration under a new name. |
|  | Language dialog | Adds or deletes a language from a configuration. |
|  | Open help | Opens the help. |

## 11.3.1.3 Inspect configuration

A configuration in *Inspect* is an XML file that contains definitions for symbols, tables and display settings. Create new configurations in the Editor by clicking the  button.

**Inspect directory:** Configurations are by default saved in the inspect directory in the resource folder *(%gt_resource_folder%inspect)*.

**Find inspect directory:** The path to the directory can be edited by the option `gti_folder`.

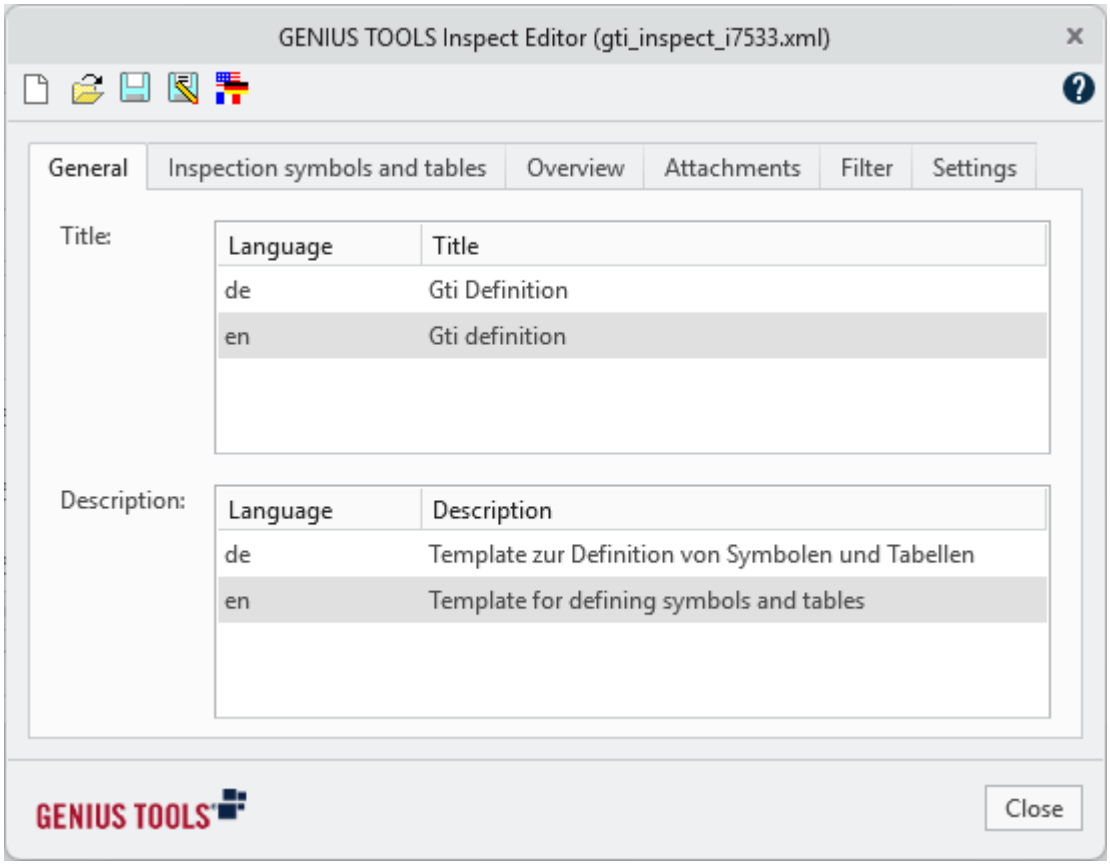
**Save configuration:** The name of a configuration file must not exceed 18 characters.

**Warning:** Configuration files that have been manually created with names longer than 18 characters cannot be opened by the editor. They must be edited manually.

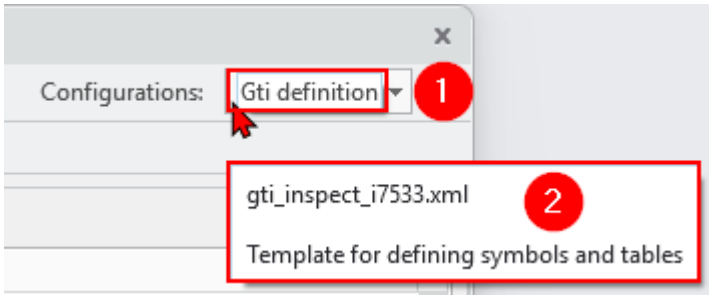

© 2026 INNEO Solutions GmbH

## 11.3.1.4 General

In the *General* tab, you can edit the configuration data in multiple languages.




1. **Title**: The name of the title is displayed in the selection menu.
2. **Description**: Tooltip that is displayed for the title.

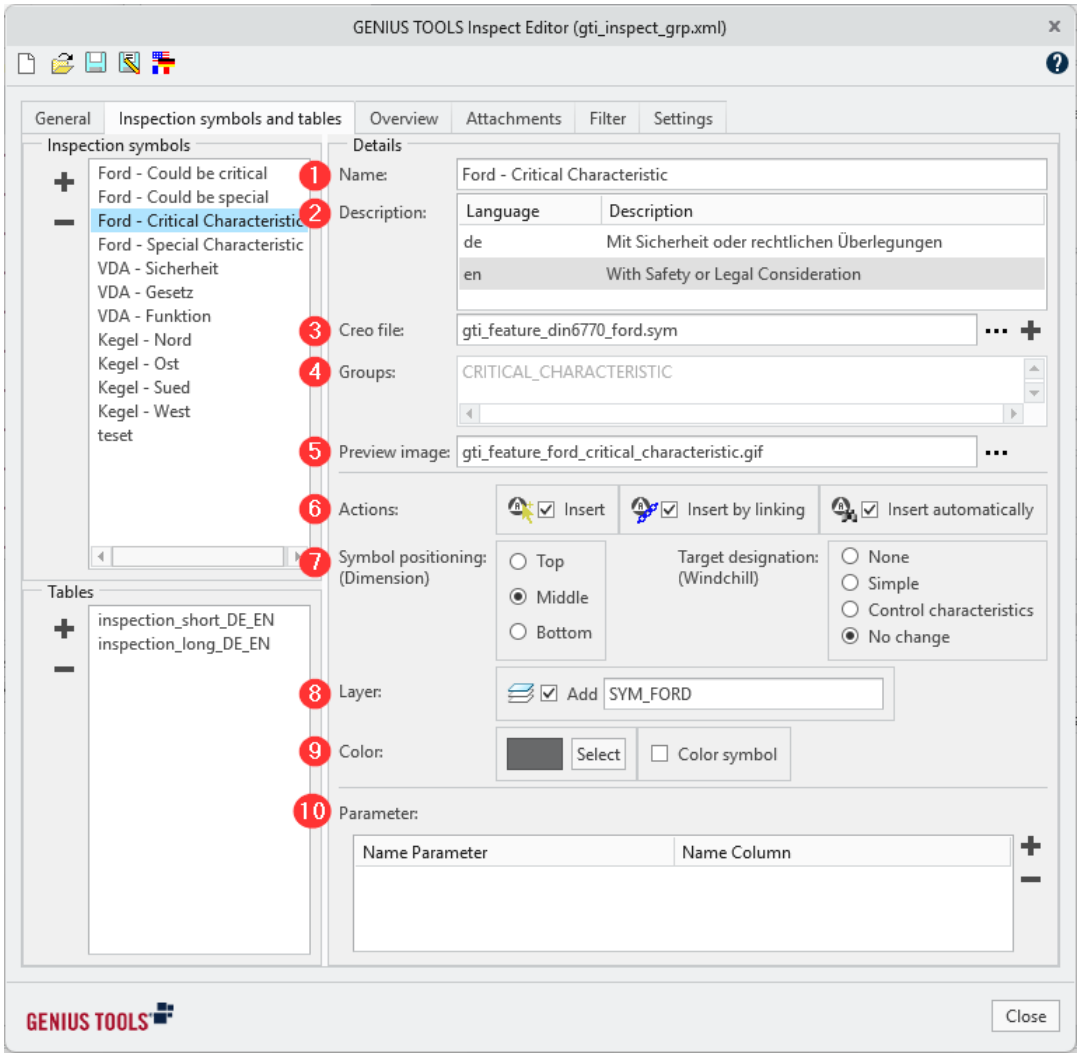



## 11.3.1.5 Editing inspection symbols and tables

In the tab *Inspection symbols and Tables*, the different sets of settings for inspection symbols and tables are configured. Use the (+/-) buttons to add or remove elements.


© 2026 INNEO Solutions GmbH

The order of the inspection symbols and tables determines their order in the user interface of *Inspect*. An elements position can be changed by drag-and-drop, dropping the element below the target element.




Click on an inspection symbol or table in the element selection to view the details.

## Inspection symbols

1. **Name**
   Specifies the displayed name of the inspection symbol.

2. **Description**
   A localized description of the inspection symbol. Use the localization button in the command bar to manage the languages.

3. **Creo-File**
   Specifies the path to a SYM file. Default is *%gt_resource_folder%\inspect*.


*© 2026 INNEO Solutions GmbH*

**Warning:** Do not use the same symbol in different Inspect configurations if you are using numbering similar to DIN 6770. Otherwise, you may experience numbering conflicts.

4. **Grouping**
   If you are using a symbol that contains groupings, create a separate inspection symbol for each symbol variant that you want to place using *Inspect*. If you select a symbol file that contains groupings, the first grouping found is selected for the inspection symbol. To select the required symbol variant for the inspection symbol, proceed as follows.
   - Open a drawing that contains the required symbol variant, or place the required symbol variant on a drawing.
   - Open Inspect Editor and create the new inspection symbol. Select the symbol file under Creo-File using the Browse button (...).
   - Click on the plus symbol next to the Creo-File input field. You are asked to select an element.
   - Select the required symbol in the drawing. The grouping settings of this symbol are written to the inspection symbol definition.
5. **Preview image**: Specifies the path to a preview file of an inspection symbol.
6. **Actions**: Defines the actions that can be used on an inspection symbol: free insertion, linked insertion and linked, automatic insertion.
7. **Symbol positioning (Dimension)**
   Defines the positioning of the symbol in relation to the dimension: Top, Middle, Bottom. Defines whether and in what form the data is displayed in Windchill: None, Simple, Control characteristics, No change
8. **Layer**: Defines the layer on which the symbols are placed. Enter the layer under *Name*.

**Please note**: When you hide the layer, only inspection symbols that have been freely placed will be hidden with it. Linked inspection symbols can only be hidden using the *Inspect* hide and unhide functionality, see Hide and unhide 201.

9. **Color**
   Specify the color to be used for the selected inspection characteristic. Use the checkbox to toggle the coloring.
10. **Parameter**: You can add additional information from drawing or model parameters to an inspection symbol. To do this, a symbol must have variable text, in which the necessary drawing or model parameter is defined. See Creating variable text 221 and Adding parameter values to a symbol 224 for more information.

Under *Name Parameter,* enter the name of the variable text. Under *Name Column*, enter a name for the table column that displays the values from the variable text. This column is added to the symbol, which you can check in the *Overview* 198 tab in *Inspect*.


© 2026 INNEO Solutions GmbH

**Warning:** The following terms are used by *Inspect* and cannot be used for naming parameters and their columns, both upper and lower case.

| | | | |
|---|---|---|---|
| bse_dim | gti_param | max_dim | tpe_sub |
| cls_tol | linkedto | min_dim | tpe_tol |
| creo_id | nme_sym | mod | val_tol |
| descr | num | revision | src |
| grd | num_sheet | separator | tpe_sym |
| gti_note | num_sym | tpe_main | tpe_tol |
| ipc_dim | | | |

To output the additional parameters in the Excel export as well, you have to assign a column in the Excel template to each parameter. In the Excel template, specify the column name (**Name Column**) in the format *gti:<columnName>*, see also Export[202].

## Tables

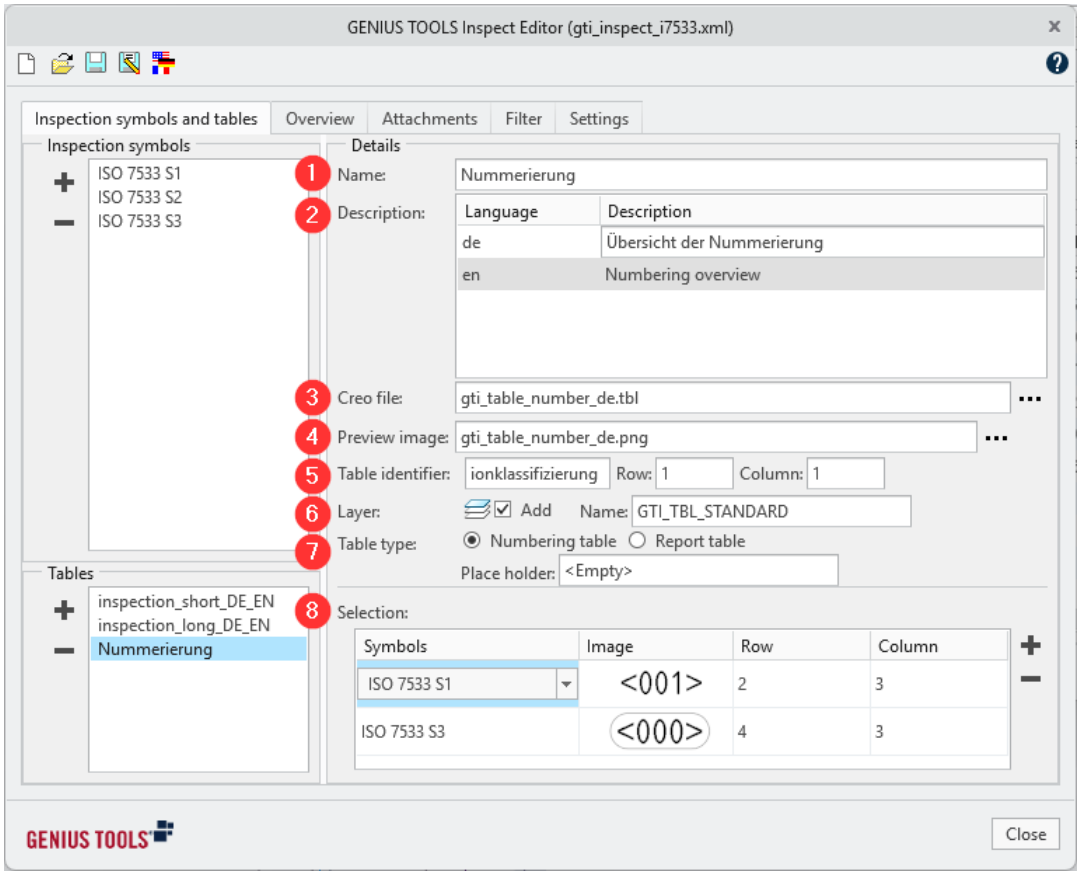
*Inspect* comes with two out-of-the-box tables, one numbering table and one report table. The configuration for the settings *Table identifier*, *Table type* and *Selection* differs according to the table type. The Creo table used also has some different characteristics.

– **Numbering table:** lists inspection symbols with the symbol image, a description and a list of numbers.

– **Report table:** displays detailed information on individual inspection symbols.

For both table types, you can customize the standard template or create your own table templates. Please refer to Configuration for numbering tables[213] and Configuration for report tables[214].


© 2026 INNEO Solutions GmbH

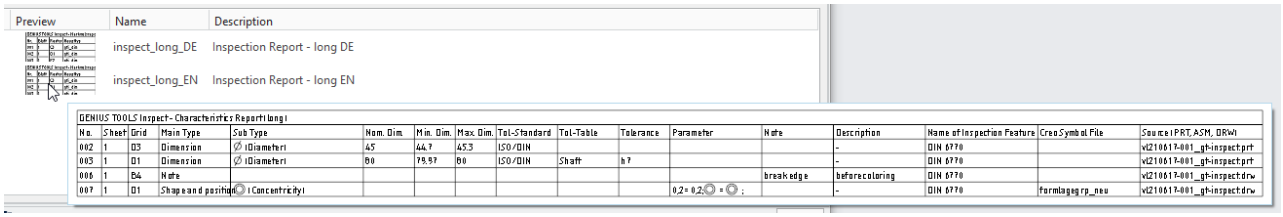



1. **Name**
   Specifies the displayed name of the table.

2. **Description**
   A localized description of the table. Use the localization button in the command bar to manage the languages.

3. **Creo-File**
   Specifies the path to a TBL file. The default path is *%gt_resource_folder%\inspect*. Numbering tables [213] and report tables [214] differ.

4. **Preview image**
   Specifies the path to a preview file of a table image. The tooltip displays the image in its original size, so a large image may improve the legibility of the preview.




5. **Table identifier**
   Used to identify a table. There are differences between numbering tables [213] and report tables [214].


© 2026 INNEO Solutions GmbH

6. **Layer**
   Defines the layer on which the placed table will be placed when *Add* is set. Enter the layer name under *Name*.
7. **Table type**
   The table type is either *Numbering table* or *Report table*. In addition, a place holder for empty cells can be entered for a *Numbering table*, e.g. <Empty>.
8. **Selection**
   This section is displayed only for numbering tables[213].

## Configuring numbering tables

**Creo-File**: Specifies the path to a TBL file. The default path is `gt_resource_folder`. The TBL file for a numbering table specifies symbols and descriptions line by line. For each symbol, a cell is provided for the list of numbers used. The table cell for the numbering is specified in the section *Selection* under *Row* and *Column*.

| Function related classification | | Inspection No. |
|---|---|---|
| Symbol | Critical Feature | 1 |
|  | Hazardous to human beings | |
| Symbol | Main feature | <Empty> |
|  | Restricted functionality, reduced output/performance | |
| Symbol | Standard feature | all other quality parameter |
| | no direct effect on funktionality | |

*Example of a Creo table for a numbering table*

**Table identifier / Row / Column**: The table identifier is used to identify a table and must be unique. The table identifier is *Function related classification* for the example numbering table. If you want to define your own tables, use a unique table identifier.

Enter a row and column to specify where the table identifier is located in the table. By default, the table identifier is located in the first cell of the first row.

**Selection**: The position of the displayed inspection symbols is defined here. Click on the displayed name of an inspection symbol to display a drop-down list of symbols not yet included. Use the (+/-) buttons to add or remove rows.

Under *Row* and *Column*, specify the table cell in which to display the list of numbers used. For summarized cells, specify the top row.

**Please note**: Numbering tables cannot contain multiple inspection symbols that use the same Creo symbol file. Only the last row defined for each symbol file is saved in the table definition.


© 2026 INNEO Solutions GmbH

# Configuring report tables

**Creo-File**: Specifies the path to a TBL file. The default path is `gt_resource_folder`.

For report tables, the second row of the table defines which data to write to each column. This is done by specifying the abbreviation for the desired column. The available column codes are listed in the section on adapting the Excel export template for *Inspect* under Export 202.

The first cell to be filled with data contains the text *Wertezeile* (value row).

If you insert a report table with *Inspect*, the table is filled automatically and the first two rows are hidden, making the table identifier and column types invisible.

**Please note:** There is no way to display the top two rows of the table once you have inserted it using *Inspect*. If you want to make changes to the TBL template, you have to edit it in Creo without using *Inspect*.

| gt_tbl_gti | | |
|---|---|---|
| num_sym | tpe_main | tpe_sub |
| GENIUS TOOLS Inspect Tabelle | | |
| Number | Main type | Subtype |
| Wertezeile | | |

*Example of a Creo table for a report table*

**Table identifier / Row / Column**: The table identifier is used to identify a table and must be unique. The table identifier is *gti_tbl_report_long* for the example report table. If you want to define your own tables, use a unique table identifier.

Enter a row and column to specify where the table identifier is located in the table. By default, the table identifier is located in the first cell of the first row. If you insert a report table with *Inspect*, the table is filled automatically and the first two rows are hidden, making the table identifier and column types invisible.

**Please note:** Using the configuration option `gti_lang` you define - not depending on the system language - the language used for the contents of the report table.

# Customizing tables

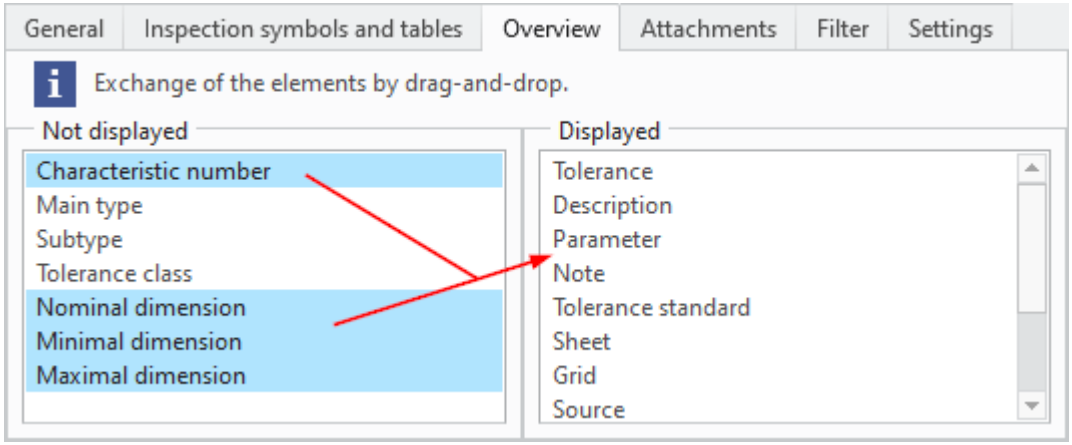
For instructions on creating a customized report table see the chapter Creating a custom report table 229.


© 2026 INNEO Solutions GmbH

## 11.3.1.6 Editing column display

The *Overview* tab is used to manage the display of the *Linked Symbols* table.

Drag and drop elements to be displayed in the table into the desired position. Remove items that you do not want to display in the same way.

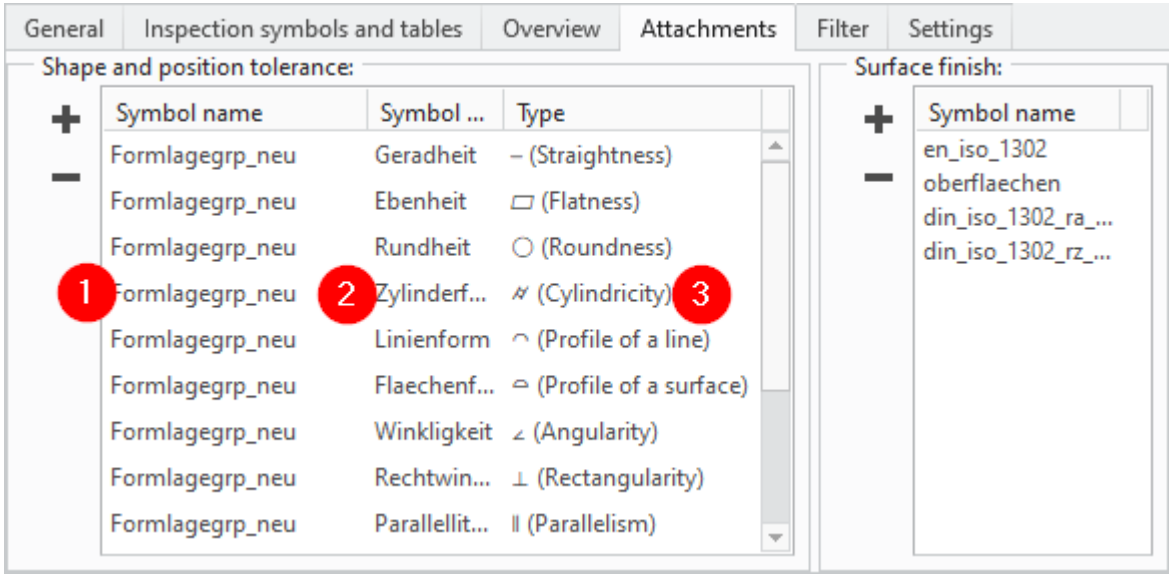



*Use Drag and Drop to compile the overview table*

## 11.3.1.7 Assigning shape and position tolerances / surface finish symbols

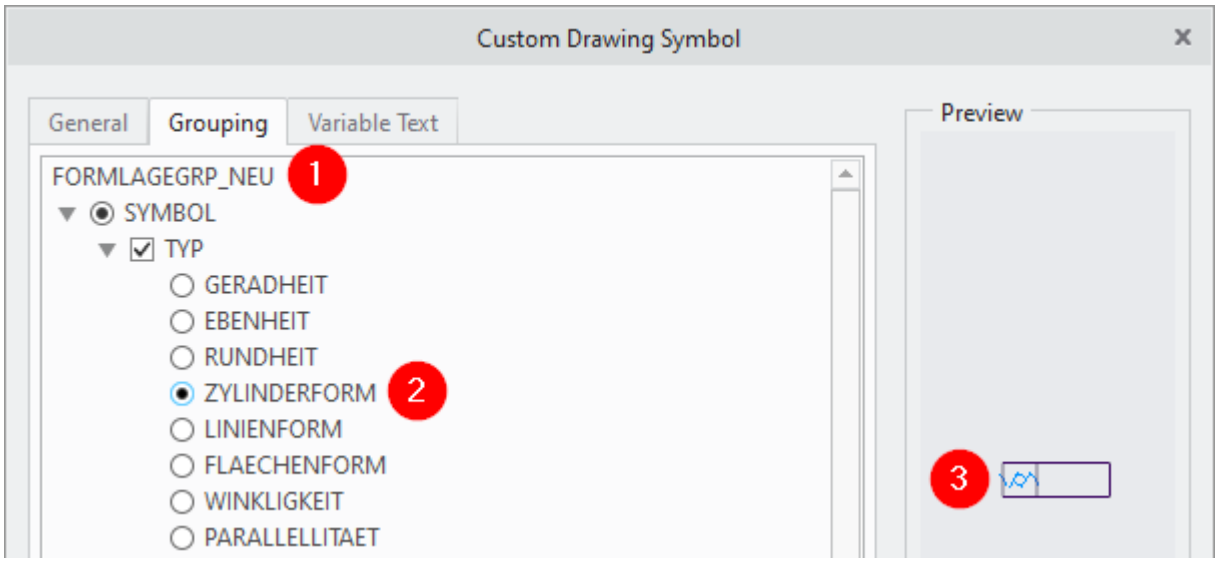
The *Attachments* tab manages the assignment of various shape and position tolerances or surface quality symbols to user-defined symbols.

This makes it possible to filter for inspection symbols linked to defined types of elements.




*Entries in the shape and position tolerance table or surface quality table correspond to properties of symbols in the symbol library.*


© 2026 INNEO Solutions GmbH




*Creo dialog for groupings in drawing symbols*

Enter the file names of the custom drawing symbols (as defined in the Creo dialog box) in the corresponding tables to assign them to the correct main types. In the case of shape and position tolerances, the main type (1) and subtype (2 and 3) can be determined.

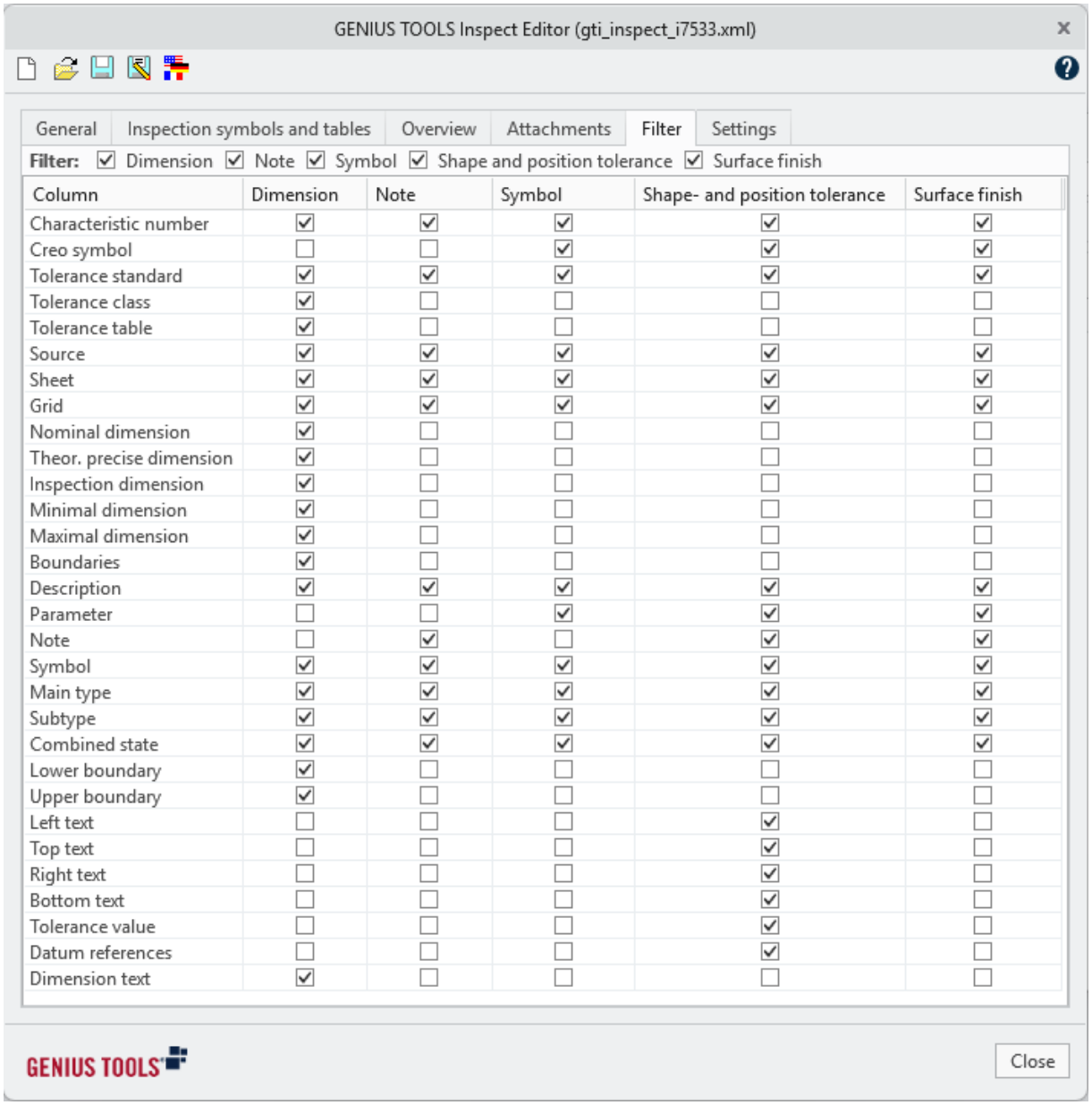
## 11.3.1.8 Filtering displayed inspection symbols

The *Filter* tab defines the view of the usable filters for the *Linked Symbols* overview. The configuration is stored per definition.

First, specify which filters (dimension, note, symbol, shape and position, surface) are to be activated by default.

In the second step, configure the columns to be displayed. Then save the configuration. The overview table *Linked Symbols* will always use the filters configured here when applying the definition.


© 2026 INNEO Solutions GmbH

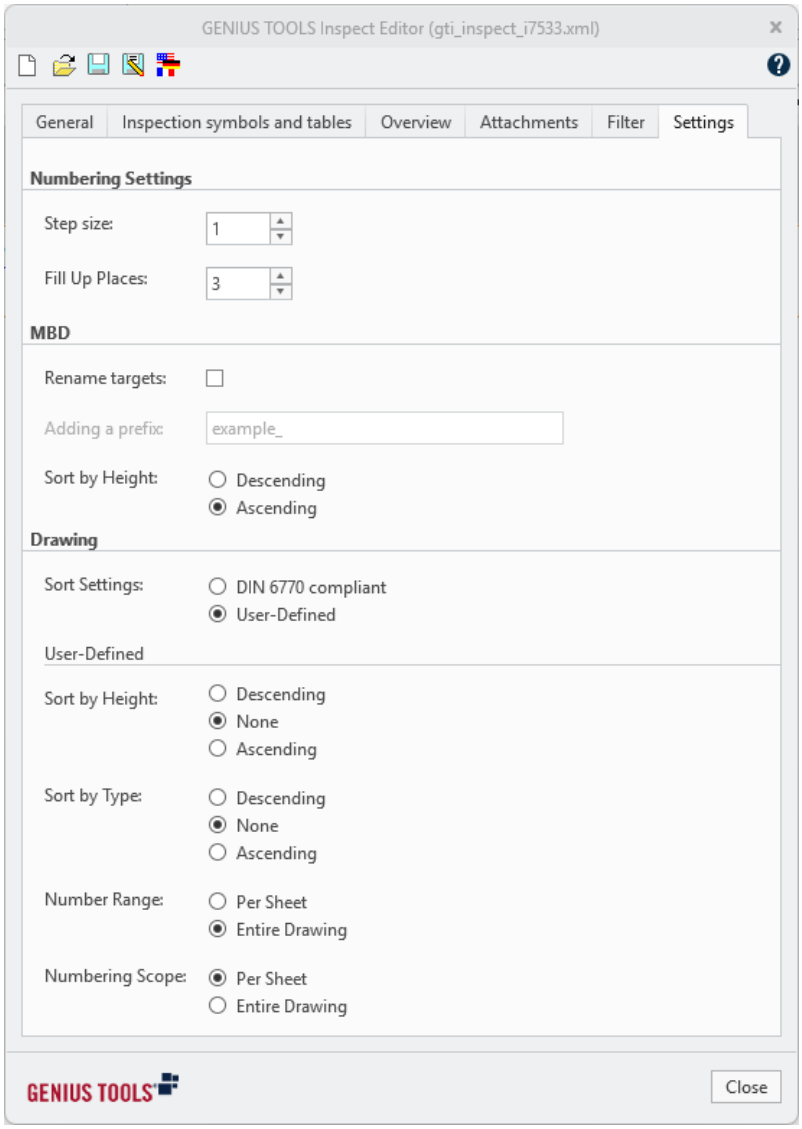



*In this example, the Tolerance class column will only be displayed, if the Dimension filter is active, i.e. if inspection symbols linked to dimensions are displayed.*

## 11.3.1.9 Settings for 3D models

**Please note:** In the tab *GT MBD*, settings can only be made for 3D models that are executed via *Inspect 3D*[243]. In drawing mode, the settings made here are not read and have no effect.


© 2026 INNEO Solutions GmbH




### Numbering Settings

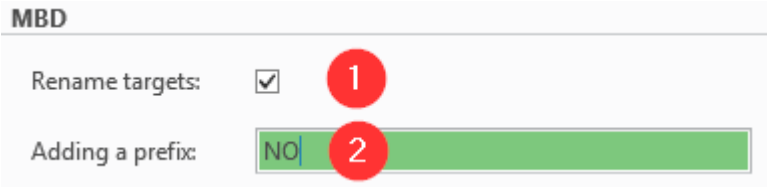
Step size: Set the step size in which symbols are to be numbered for each configuration.

Fill Up Places: Fill with leading zeros up to a certain number of digits.

### MBD

Inspection characteristics can be displayed in the annotation tree with an additional description. The inspection characteristics are displayed in the model as they were specified in the *Editor*. A prefix can be displayed in the annotation tree, e.g. NO.

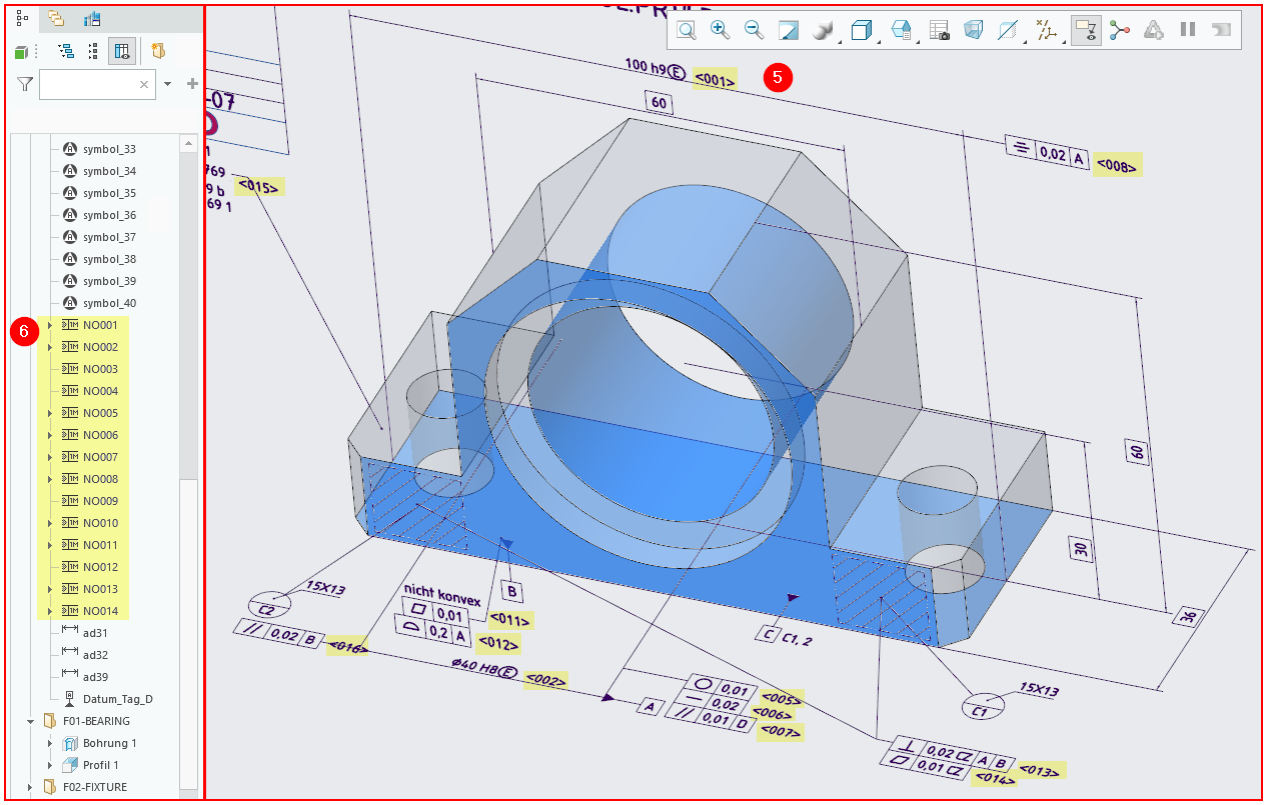
To obtain this display, set the following settings:




1. Select the checkbox for *Rename targets?*


© 2026 INNEO Solutions GmbH

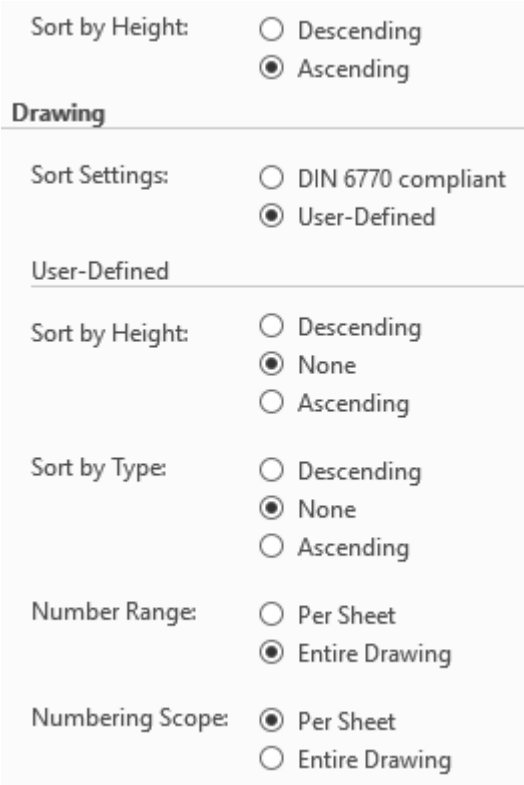
2. Enter the prefix you want to use, here: NO for numbering.
3. Save settings in *Inspect Editor* and close the Editor.
4. Click on *Update inspection features*.
5. You receive the following display:
   - In the model, the inspection characteristics are displayed as defined, e.g. <001> (5).
   - In the annotation tree, the same inspection characteristics are displayed with the defined prefix, e.g. NO001 (6).




## Drawing

The numbering can be defined either in accordance with DIN 6770 or user-defined.


© 2026 INNEO Solutions GmbH




## 11.3.2 Creating a Creo symbol for Inspect

This section explains how to create a new symbol for use as an inspection symbol.

**Please note:** The symbols delivered with *Inspect* use the font *ISONORM LT Regular* based on ISO 3098-5. If you experience issues with displaying the out-of-the-box symbols, please check whether this font is available in your system.

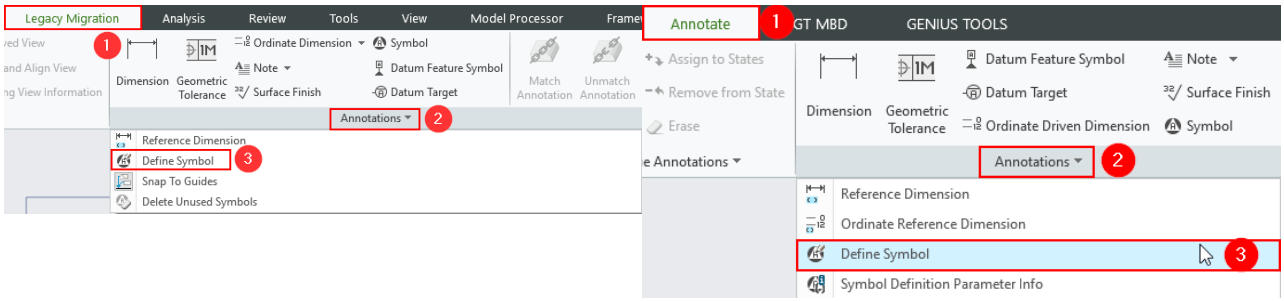
## 1. Sketching the symbol

Open the command *Define Symbol*.

For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:

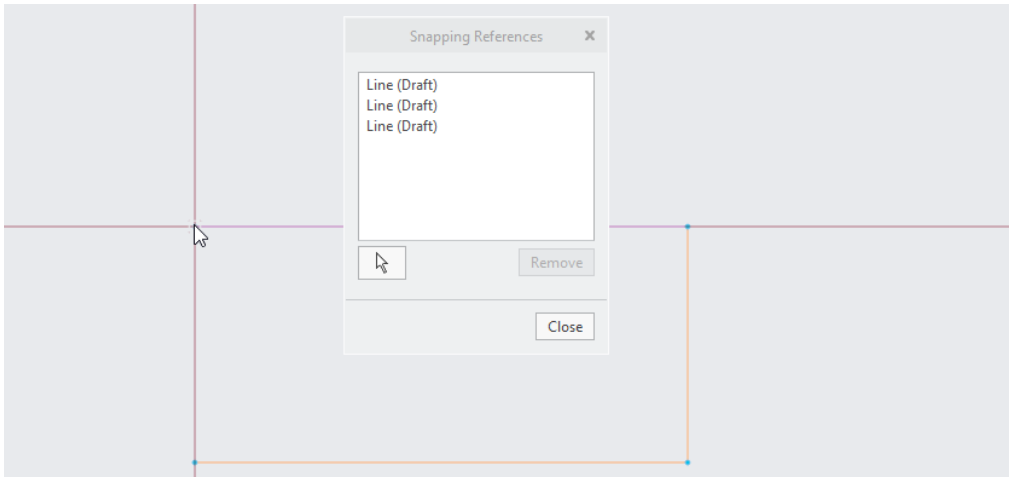
For *Inspect 3D* in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:


© 2026 INNEO Solutions GmbH




In the Menu Manager in the section SYM DEFINITION click *Define* and enter a name for the symbol.

To define your own symbol, create a Creo Parametric symbol in the sketcher.



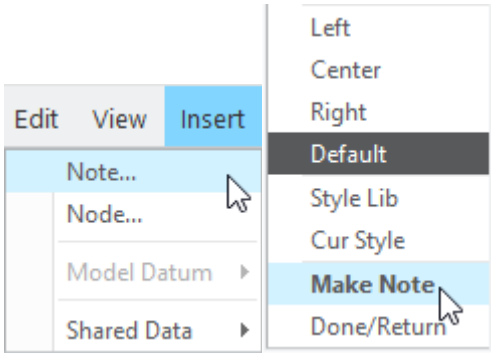

*Sketching a symbol by lines.*

# 2. Creating variable text

Variable text is data that is saved on a symbol as a note.

## 2.1. Create a note

Click *Insert* > *Note* and in the Menu Manager that opens click *Make Note* to create and place a new note.




Next you will be asked for the note name.

To ensure compatibility with *Inspect*, you have to create three notes. Enter the following names:


© 2026 INNEO Solutions GmbH

– \num\ to save the number of the inspection symbol

– \descr\ to save the description

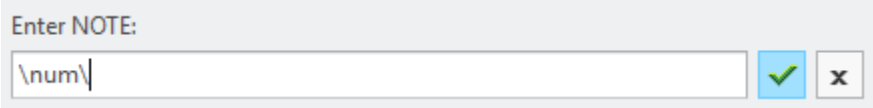
– \LinkedTo\ to save the linked drawing element for the symbol

**Warning:** The following names must not be given to a note – both in upper and lower case –, as they are in use by the program.

| | | | |
|---|---|---|---|
| bse_dim | gti_param | min_dim | tpe_sub |
| cls_tol | nme_sym | mod | tpe_tol |
| creo_id | num_sheet | revision | val_tol |
| grd | num_sym | separator | src |
| gti_note | max_dim | tpe_main | tpe_sym |

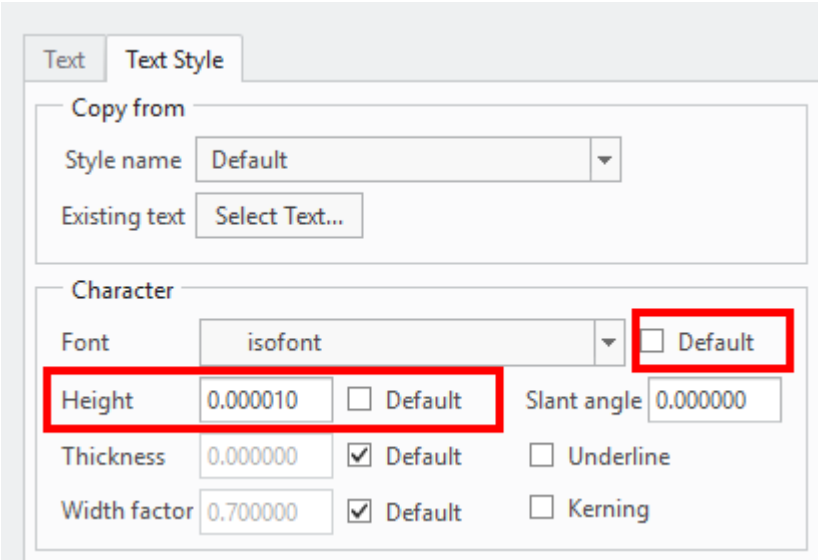
Enter the note name, confirm by clicking on the green arrow, then exit the input dialog by clicking on the X button.

The next notes can be created in the same way.




### 2.2. Change the size of the notes

Since the notes *descr*\ and *LinkedTo*\ should not appear when the drawing is printed, their size has to be reduced. You can change it via the context menu of the note by right clicking on it. Select *Properties* > *Text Style* to change the font and size.




*Text style of a note*

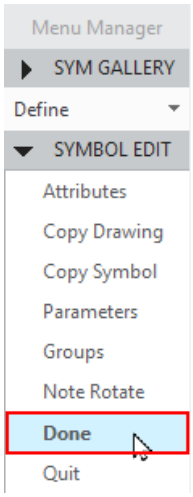

© 2026 INNEO Solutions GmbH

– Remove the checkmark from the properties for *Height* and set a very small value, e.g. 0.00001.

– Remove the checkmark from *Font*.

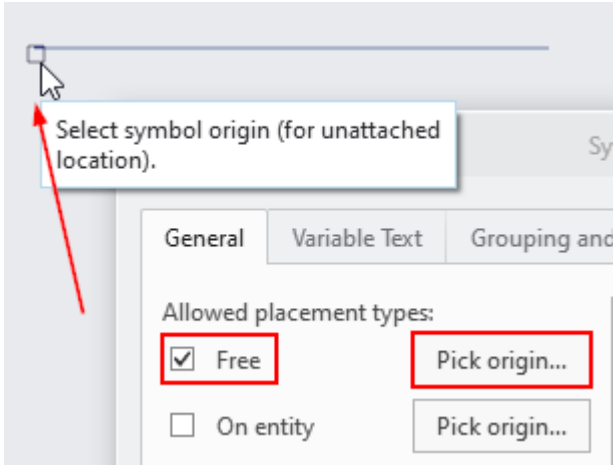
## 3. Define attributes for the symbol

To finish creating the symbol, you need to define the placement, symbol origin and predefined values of the variable texts. The dialog for these settings opens when you click *Done* in the *Menu manager*.




### 3.1. General attributes

Check *Free* and select the lower left corner as origin.




*Choosing the placement and origin*

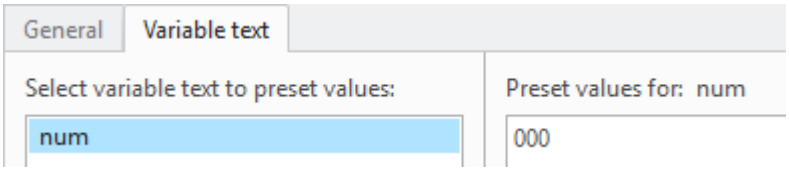
### 3.2. Variable text

In the second tab *Variable text* enter in the section *Preset values for* the following characters:

| Name | Value |
|---|---|
| num | 000 |


© 2026 INNEO Solutions GmbH

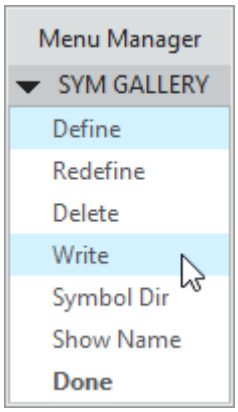
| Name | Value |
|---|---|
| descr | - |
| LinkedTo | -1,-,-1,- |




*Predefined value for /num/.*

## 5. Saving the symbol

As the last step, the symbol has to be saved. You can do this by clicking *Write* in the *Menu manager*. Enter the path, or leave it empty to use the current symbol directory for saving.




## 11.3.3 Adding parameter values to a symbol

You can assign values of drawing or model parameters to an inspection symbol. Parameter values can be either copied or linked to the symbol.

- Copy: The parameter value is copied to the inspection symbol when it is placed on the drawing and is static thereafter. (See Copying parameter values[225] .)
- Link: The parameter value is updated on the inspection symbol when the drawing or the model is opened. (See Linking parameter values[226] .)

For general information on parameters consult the chapter Variables.[916]

Parameters are written into the variable text in the dialog *Symbol Definition Attribute*.

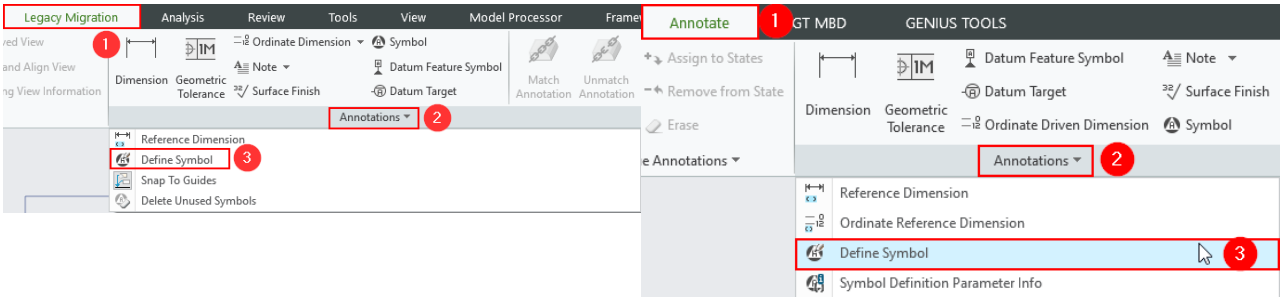
## Defining a symbol

1. Open the command *Define Symbol*.

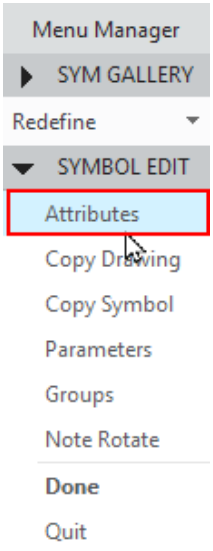

© 2026 INNEO Solutions GmbH

For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:

For *Inspect 3D* in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:




2. In the Menu Manager in the section SYM GALLERY click *Redefine*.
3. Click on the symbol. The Creo symbol editor is opened.
4. In the Menu Manager click on *Attributes*. The dialog *Symbol Definition Attributes* is opened.



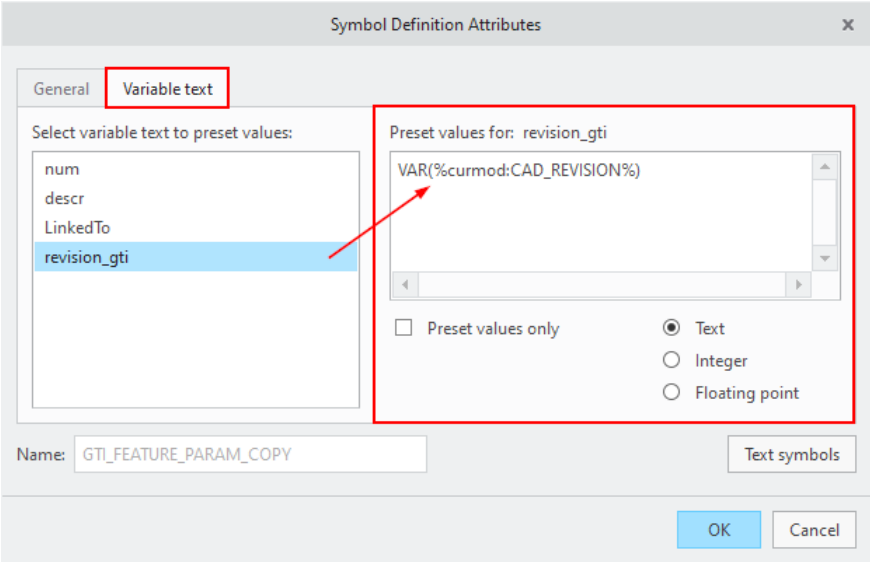

## Copying parameter values

Enter the parameter name in the variable text on the symbol in the Symbol Definition Attributes[224] dialog. In the tab *Variable text* > field *Preset values for* type:

– for a revision parameter from the model: *VAR(%curmod:Parameter%)*

– for a revision parameter from the drawing: *VAR(%Parameter%)*


© 2026 INNEO Solutions GmbH




*Preset value: Copy value from parameter CAD_REVISION*

For a step-by-step description of how to copy parameter values into symbols, see the example Creating Change Symbols[226].

## Linking parameter values (Referencing)

The parameter value is updated at the inspection symbol when the drawing is opened. In the Symbol Definition Attributes[224] dialog > *Variable text* tab > *Preset values for* field, type the parameter name in the following format:

– for a parameter from a model: *&Parameter*

– for a parameter from a drawing: *&Parameter:D*

**Please note:** The revision parameter CAD_REVISION should not be used for change symbols because the revision information on the drawing would change with each new revision.

## Redefine parameters

Existing parameters can be redefined in variable text in the drawing mode. Enter the changes for the variable text in the Symbol Definition Attributes[224] dialog.

## 11.3.4 Creating change symbols and tables

Use change symbols to mark changes to elements. Revision information can be used for this.

To display changes in a drawing, the revision must be copied to the change icon at the time of placement.


© 2026 INNEO Solutions GmbH

With the variable *revision_gti*, *Inspect* provides an interface that copies values of drawing or model parameters to an inspection symbol. It can also be used in an Inspect table.

For a revision value to be copied into a change symbol, the following preset values must be assigned in the variable text *revision_gti:*

– for a revision parameter from the model: `VAR(%curmod:Parameter%)`

e.g.: `VAR(%curmod:CAD_REVISION%)`

– for a revision parameter from the drawing: `VAR(%Parameter%)`

e.g.: `VAR(%CAD_REVISION%)`

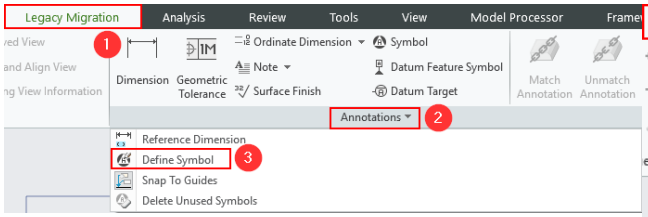
The step-by-step procedure is described below.

# Creating change symbols

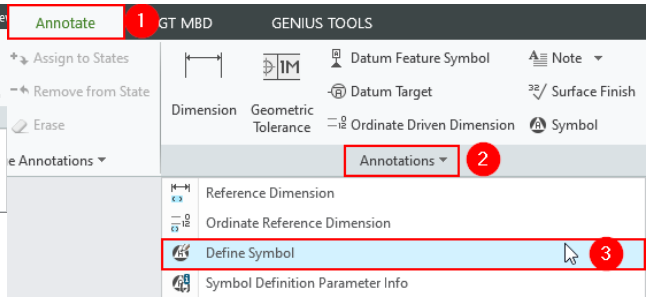
**Procedure for copying the values of a revision parameter to a symbol**

1. Open a drawing which includes the symbol in question or create a new symbol[220] first.
2. Open the command *Define Symbol*.

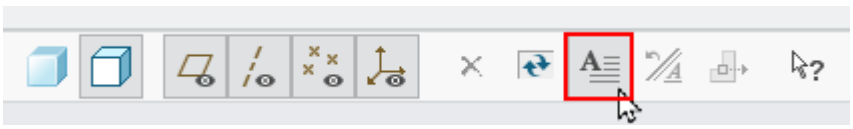
For Inspect in drawing mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Legacy-Migration (1)* and via the menu item *Annotations (2)*:




For Inspect 3D in part mode or assembly mode, you can access the command *Define Symbol (3)* via the Creo ribbon menu *Annotate (1)* and via the menu item *Annotations (2)*:



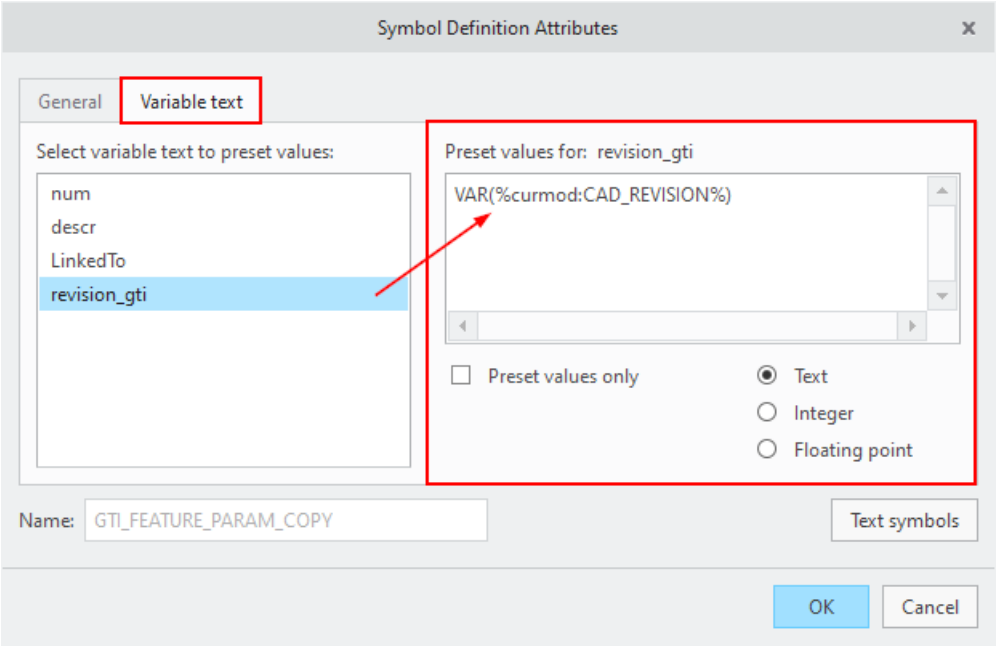

3. In the Menu Manager / SYM DEFINITION, go to *Redefine* and *Pick Inst.*
4. Click on the symbol in the drawing.
5. The Creo symbol editor opens. Go to the *Create Note* icon in the menu bar:



6. In the Menu Manager, go to *Notes* > *Create Note*. Select the style of the note and click *Make Note*.
7. The *Select item* dialog opens. Select a point on the sketch where the note should be located.


© 2026 INNEO Solutions GmbH

8. Enter this name for the note: *revision_gti*\ and click on the green arrow and then on the cross beside it.
9. Click *Done* in the Menu Manager.
10. In the Menu Manager, go to *SYMBOL EDIT* > *Attributes*.
11. This opens the *Symbol Definition Attributes* dialog. Specify the revision parameter for *revision_gti* in the tab *Variable Text*:
   – for a revision parameter from the model: *VAR(%curmod:Parameter%)*
   – for a revision parameter from the drawing: *VAR(%Parameter%)*




*Copy parameter values for the variable text revision_gti*

## Creating change tables

Changes can be displayed on a drawing by inserting a table with a column that shows the copied value of the revision parameter from the icon. You can either use the file *gti_table_param_en.tbl* or create the required column yourself in a table as follows.

1. The Creo symbol must have a variable text – here: *revision_gti* – and the revision parameter must be copied into it. (See step 11 above.)
2. This variable text must be defined for the symbol in *Inspect Editor*. To do this, go to the *Inspect Characteristics and Tables > Parameters*[208] tab and type the name of the variable text under *Name Parameter* – here: *revision_gti* – and the column name under *Name Column* – here: *Revision*.

   This column is now added to the symbol, as you can check in *Inspect* on the *Overview* tab.
3. Now create your own report table as in the example in the Creating a custom report table[229] chapter and perform step 9 and 10 (naming new column) as follows.


© 2026 INNEO Solutions GmbH

– write in the second line the same name as you have defined in step 2 under *Name column* – here: *Revision* (this line will become invisible when the table is placed using *Inspect*)

– write in the fourth line *Revision* (this is the displayed line in the table)

4. Save the table (TBL file) in the Caddepot directory of the synchronization server in the *Inspect* folder.

# 11.4 Inspect Revision 3D

*Inspect 3D Revision* allows you to create a snapshot of all inspection symbols in a combined view / in a model at a given time. Using a revision parameter you can define the revision level of a model and generate a history of all revisions. Revision histories can be exported to Excel.

## 11.4.1 Fundamentals

### Glossary

**Revision**
Reviewed state of a model.

**Revision status**
Change of a revision identified by a number, letters and / or date.

**Revision parameter**
Parameter indicating the revision status of a model.

**Snapshot**
A snapshot shows all changes to inspection-relevant properties. This allows you to see whether, for example, the values of dimensions with inspection characteristics have been changed.

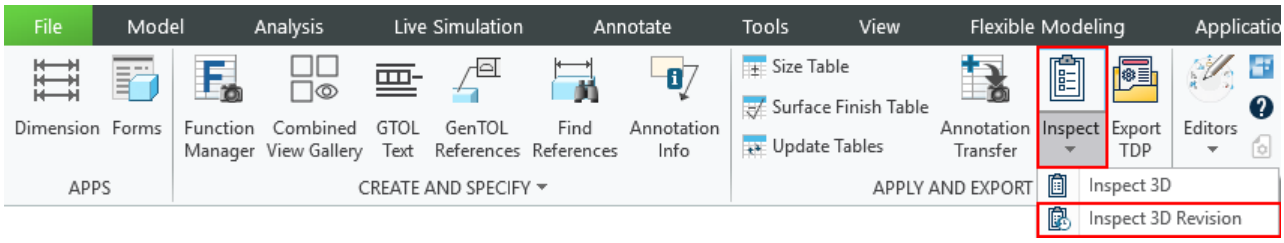
**Snapshot history**
Overview of all taken snapshots. Snapshot history can be exported.

## 11.4.2 Starting the program

### Starting the program

To start *Inspect 3D Revision* , click *GT MBD* in the ribbon menu and go to *APPLY AND EXPORT* section.

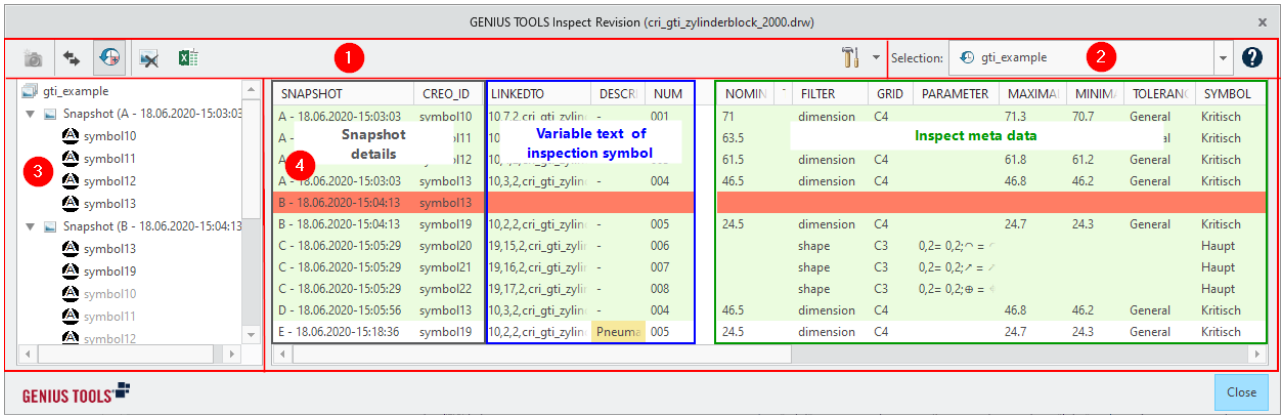

© 2026 INNEO Solutions GmbH




*Starting via the GT MBD tab in the ribbon menu*

## 11.4.3 User interface

The user interface of *Inspect Revision* / *Inspect 3D Revision* contains the following elements:




1. Command bar[232]
2. Select configuration file[233]
3. Symbol / snapshot tree[234]
4. Overview of snapshots with
   - Snapshot details:

     Drawing revision parameter[238],
     Time of creation of snapshot and
     Creo-ID of modified symbol. (Creo-ID is the identity number assigned by Creo.)
   - Variable text of the inspection symbol.

     Variable text[221] is data stored in the SYM file of the symbol.
   - Inspect metadata:

     All data extracted by *Inspect* / *Inspect 3D*.

## 11.4.4 Command bar

The command bar contains these control elements:


© 2026 INNEO Solutions GmbH

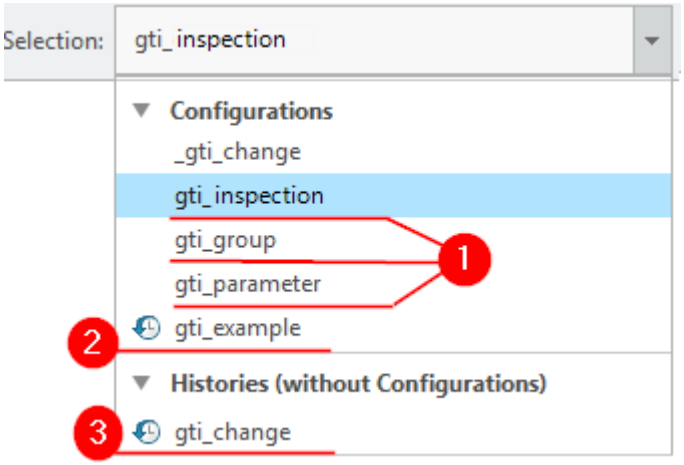
| Icon | Name | Description |
|---|---|---|
|  | Create snapshot[235] | Displays a snapshot of all inspection symbols on a drawing |
|  | Switch tree representation[234] | Switches display between symbol tree and snapshot tree |
|  | Show complete state | Shows all existing inspection characteristics at the time of the respective snapshot |
|  | Delete latest snapshot | Deletes the latest snapshot |
|  | Export history to excel[240] | Exports snapshot history as XLSX file |
|  | Tools | Contains supportive functions:<br>- Save values to XML |
|  | Configuration / history[233] | Select configuration file and snapshot history |
|  | Help | Opens help page for *Inspect Revision / Inspect 3D* |

## 11.4.5 Selecting a configuration file

The command bar contains a field for selecting a configuration. A configuration is an XML-file that contains symbols, table definitions and display settings and that is created in *Inspect Editor*[205]. The clock symbol next to the name indicates whether the configuration file contains snapshot data, i. e. a history.

Snapshot data is data stored in a Creo file (DRW, PRT, ASM). They can be exported from the XML file but also as stand-alone data without data of the XML file. Thus you may send a model with an inspection history without revealing the configuration settings.


© 2026 INNEO Solutions GmbH




*Selection of configurations in Inspect Revision*

**Configuration without history (1)**
No data in Inspect Revision dialog
**Configuration with history (2)**
Snapshot history displayed in Inspect Revision dialog
**History without configuration (3)**
Snapshot cannot be created

## Save history as XML file

A history of snapshots can be exported without the corresponding configuration settings by saving a separate XML file in *Tools* > *Save history as XML file*.

## Configuring the selection field

The configuration file is by default set to *gti_inspection*. The displayed default file can be changed in the configuration option `gti_start_file`.

# 11.4.6 Selecting the tree view

You can expand the snapshot / symbol tree by clicking on the arrow symbol and switch between the two tree displays by clicking on the symbol :

1. **Snapshot view:** Lists all snapshots and the corresponding inspection symbols
2. **Symbol view:** Lists all inspection symbols and the corresponding snapshots

The selection in the snapshot / symbol tree display area determines how snapshot data is displayed. These are the possibilities:

– Select the configuration file: displays complete history

– Select a snapshot (e.g. Revision A): displays a single snapshot


© 2026 INNEO Solutions GmbH

- in snapshot tree: with all modified inspection symbols
- in symbol tree: of selected inspection symbol

– Select an inspection symbol (e.g. symbol 24): displays a single inspection symbol

- in snapshot tree: of selected snapshot
- in symbol tree: with all snapshots

**Please note:** When using *Inspect 3D Revision*, all inspection symbols that are available in the currently selected combined view are displayed. Inspection symbols available in other combined views are grayed out in the snapshot tree.

## 11.4.7 Creating snapshots

The snapshot function  maps all inspection symbols on a drawing / in a model and creates a time stamp. In addition, *Inspect Revision* / *Inspect 3D Revision* can extract values from a Revision parameter[238].

**Tip:** A snapshot corresponds to the revision status, if the revision parameter has been changed for the snapshot.

When a snapshot is taken, all inspection symbols stored in the configuration file are compared to the previous snapshot. A snapshot can be created when at least one inspection symbol has been altered.

The following modifications are included in a snapshot:


© 2026 INNEO Solutions GmbH

| Modification | Color/ example |
|---|---|
| 1. A new inspection symbol was placed (symbol does not yet exist in any snapshot) | green row<br>A - 18.06.2020-15:( symbol12 10,4,2,cri_gti_zylinderblock_ |
| 2. A new value was added to an inspection symbol | green cell<br>General |
| 3. A value was edited in the model | yellow cell in Inspect metadata columns (3)<br>cri_gti_zylinderblo Dimension ISO/DIN |
| 4. A value on an inspection symbol was altered (e. g. in variable text) | yellow cell in Variable text columns (2)<br>10,3,2,cri_gti_zylin Pneumatic length measuring 004 |
| 5. An inspection symbol was deleted (symbol is in snapshot but not in the drawing / in the model) | red row<br>B - 18.06.2020-15:04:13 symbol13 |
| 6. An unchanged symbol from an older snapshot  | greyed out text<br>A - 09.02.2021-07:( symbol12 |

# Recorded inspection symbols

A snapshot unhides all hidden inspection symbols. This means that symbols without a target will be deleted in the process, see also Hide and unhide 201.

A snapshot does not capture all inspection symbols and will blend in all hidden symbols as follows:

| State of inspection symbol | Captured by snapshot? |
|---|---|
| Unnumbered inspection symbols (000 symbols) | no |
| Inspection symbols whose target was deleted | no<br>**Warning:** Inspection symbols, whose target was deleted, will be deleted. |


© 2026 INNEO Solutions GmbH

| State of inspection symbol | Captured by snapshot? |
|---|---|
| Hidden symbols | |
| – with a linked target | yes, are shown again in the drawing / mode |
| – without a target | no<br>**Warning:** Inspection symbols, whose target was deleted, will be deleted. |

## Example: Creating a snapshot of a revision level

**Steps:**

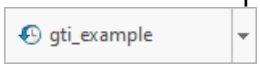
1. Place and number the needed inspection symbols.
2. Assign a new value to the revision parameter in *Tools* > *Parameter*. In the example: CAD_REVISION = D
3. In the command bar click the button *Create snapshot*  .

**Result:**

The snapshot will be displayed in the left section of the dialog (symbol- / revision tree). In the example: Snapshot (D - 15.06.2020 - 13:37:41)

## 11.4.8 History of snapshots and symbols

After the first snapshot is taken, a clock icon appears next to the configuration file.

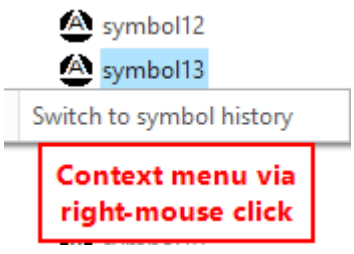



Configurations with a clock have a snapshot history, which is the sum of all snapshots. This history can be exported.[240]

**Tip:** You can get a history of revisions by mapping them with a snapshot after completing a new revision.

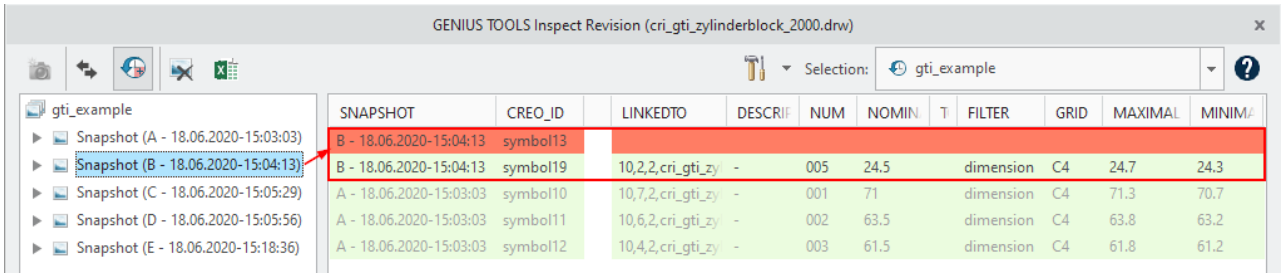
You can view a history by clicking on a snapshot or an icon.

You can use the  button to switch to the symbol tree or click directly on a symbol to switch to its history using the context menu.

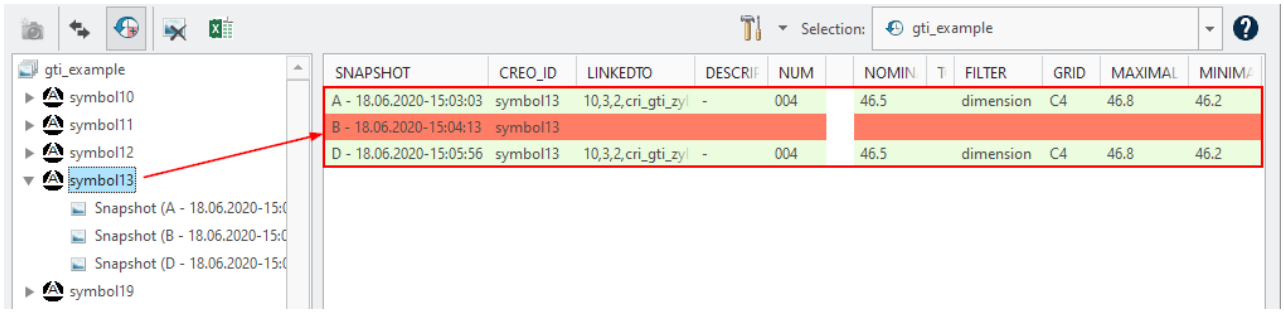




© 2026 INNEO Solutions GmbH




*History of snapshot of revision B*




*History of symbol 13*

## Revert symbol

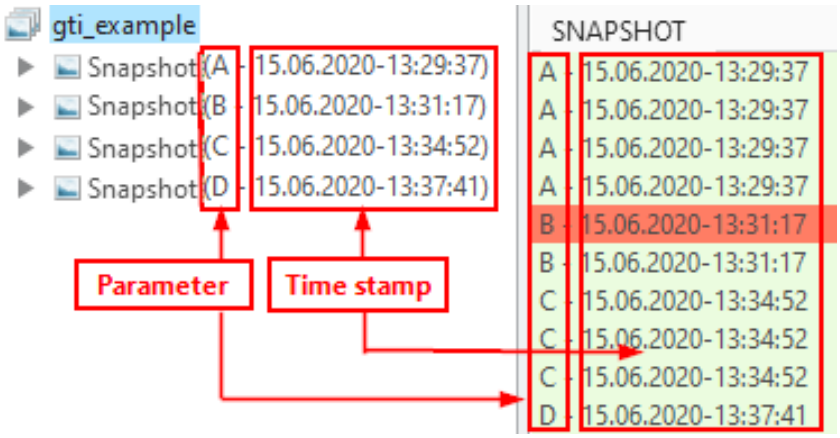
If you have removed a symbol from the drawing, you can restore it. The removed symbols are marked in orange in the table. Right-click on the corresponding row and then click *Revert Symbol* to restore the symbol.




## 11.4.9 Revision parameter

The revision parameter informs about the revision level of a drawing / model. A new parameter value should be assigned after any test-relevant change to the inspection symbols or their values. (In the example below: A - D.) If you do not work with a revision parameter, you can still distinguish snapshots by their time stamp.

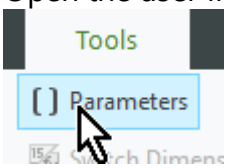

© 2026 INNEO Solutions GmbH



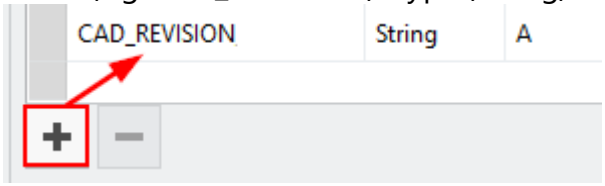

*Details of snapshots in the snapshot tree display*

There are different ways to use a revision parameter in *Inspect* / *Inspect 3D*:

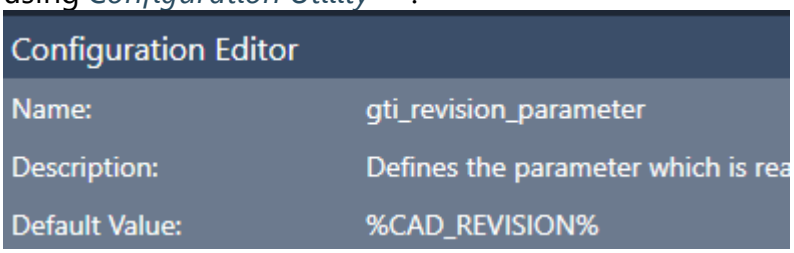
1. **Using a revision parameter from Startup TOOLS**
   If you work with *Startup TOOLS*, you use a preconfigured start part. The start part contains the revision parameter CAD_REVISION. The configuration option `gti_revision_parameter` is set to this by default. If you are creating inspection symbols on the drawing / in the model for the first time, CAD_REVISION is automatically set to A and then read.
2. **Using a revision parameter from Windchill**
   A revision parameter is provided by Windchill and must be specified in the configuration option `gti_revision_parameter`.
3. **Generating a revision parameter yourself**
   **a.** Open the user interface *Parameter* via the Creo ribbon menu *Tools*:




   **b.** Add a new parameter using the *Plus* Button (*Add new parameter*). Enter the parameter name (e.g. CAD_REVISION), Type (String) and Value (A):




   **c.** Enter the new parameter name in the configuration option `gti_revision_parameter` using *Configuration Utility*[765]:





© 2026 INNEO Solutions GmbH

## 11.4.10 Export snapshot history

The history of snapshots can be saved to an XLSX file using a template.




*Excel button in Inspect Revision*

The button opens the dialog Export table to Excel[690] which allows you to select the Excel template, the file to be exported and the snapshot history.

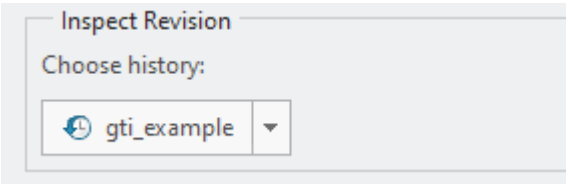
The Excel template is checked when starting the Excel export and may take some time.

**Warning:** Close all Excel windows before starting the export function.

## Export dialog

The template file `gti_revision_template_de_en.xlsx` (in the directory *gt_resource_folder\inspect)* is entered by default as is the name of the drawing file.

Select a configuration – this defines the inspection symbols whose snapshot history is to be exported.




## Configuring export

The excel export is set by using the following configuration options:

`gti_revision_excel_template` defines the name of the default Excel template for export. (Default: *gti_inspection_template_de_en.xlsx*)

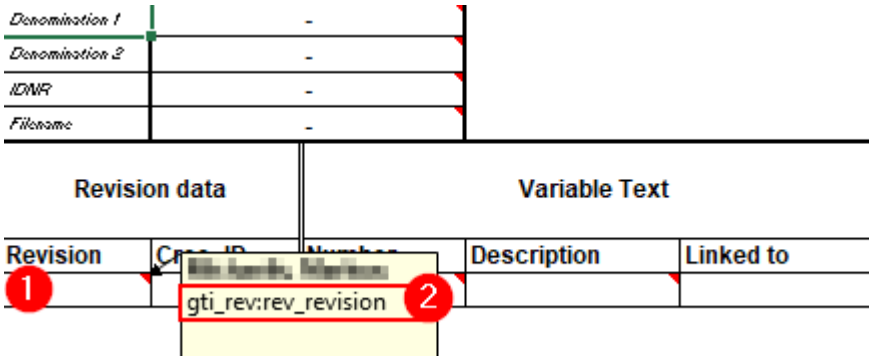
`gti_revision_folder` defines the folder in which the default Excel template is searched for. (Default: *%gt_resource_folder%/inspect*)

`gti_revision_excel_coloring` defines whether data is exported with the coloring from the snapshot data segment of the UI.

## Customizing an export template

You can customize a template by taking the template file *gti_revision_template_de_en.xlsx* from the directory *gt_resource_folder%/inspect* as a basis and adding a comment (2) to the first line (1) of the column you wish to adapt. The chapter Create template[693] describes how to set up an export template step by step.


© 2026 INNEO Solutions GmbH




*Acronym gti_rev: in the comment that assigns a column*

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.

The code in the comment defines the values to be copied into a column. It consists of the component acronym *gti_rev:* and the fill command. Use the following codes:

| Command code | Column name |
|---|---|
| gti_rev:rev_revision | Revision |
| gti_rev:rev_id | Creo_ID |
| gti_rev:var_num | Number |
| gti_rev:var_descr | Description |
| gti_rev:var_LinkTo | Linked to |
| gti_rev:num_sym | Number of inspection symbol |
| gti_rev:num_sym | Symbol number |
| gti_rev:tpe_main | Main type |
| gti_rev:tpe_sub | Subtype |
| gti_rev:cls_tol | Tolerance class |
| gti_rev:bse_dim | Nominal dimension |
| gti_rev:min_dim | Minimal dimension |
| gti_rev:max_dim | Maximal dimension |
| gti_rev:val_tol | Tolerance |
| gti_rev:descr | Description |


© 2026 INNEO Solutions GmbH

| Command code | Column name |
|---|---|
| gti_rev:gti_param | Parameter |
| gti_rev:gti_note | Note |
| gti_rev:tpe_tol | Tolerance standard |
| gti_rev:num_sheet | Sheet |
| gti_rev:grd | Grid |
| gti_rev:src | Source |
| gti_rev:mod | Tolerance table |
| gti_rev:nme_sym | Name of symbol |
| gti_rev:tpe_sym | Creo symbol |
| gti_rev:var_<parametername> | Output of additional user-defined parameters |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gti_rev:*. Thus you can export data from *Inspect* and *Inspect Revision* together, see Export data from several GENIUS-TOOLS-components[701] for an example.


© 2026 INNEO Solutions GmbH

# 12 Library

Available in the ribbon menu GENIUS TOOLS

*Library* provides a convenient way to open, copy and insert objects from a library and to customize actions – such as opening, copying or inserting to a model – for each object individually.

Supported Creo object types are:

- Part, assembly, drawing and sketch
- UDF (user-defined feature)
- Drawing table, drawing frame, drawing symbol, drawing text

*Library* is available in different languages with the following functions:

1. Searching for Creo objects
   - independently from object storage
   - with visibility control: visible, invisible, preferred use
2. Fast search of complete library
   - intuitively
   - advanced search: type, status, parameter, dimensions
3. Configuration options
   - for selection tables
   - for defining actions (copy and insert by copying)
   - for object creation with *Forms*
   - for using UDFs from *UDF Forms*
4. graphical interface to manage libraries: *Library Editor*
5. automatic check with Windchill with *Library Data Importer*

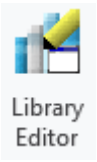
The library objects can be uploaded from hard drive or Windchill. Other PDM/PLM systems can be integrated on request.

## Components of Library




*Library* (Library browser) Searching for Creo objects


© 2026 INNEO Solutions GmbH

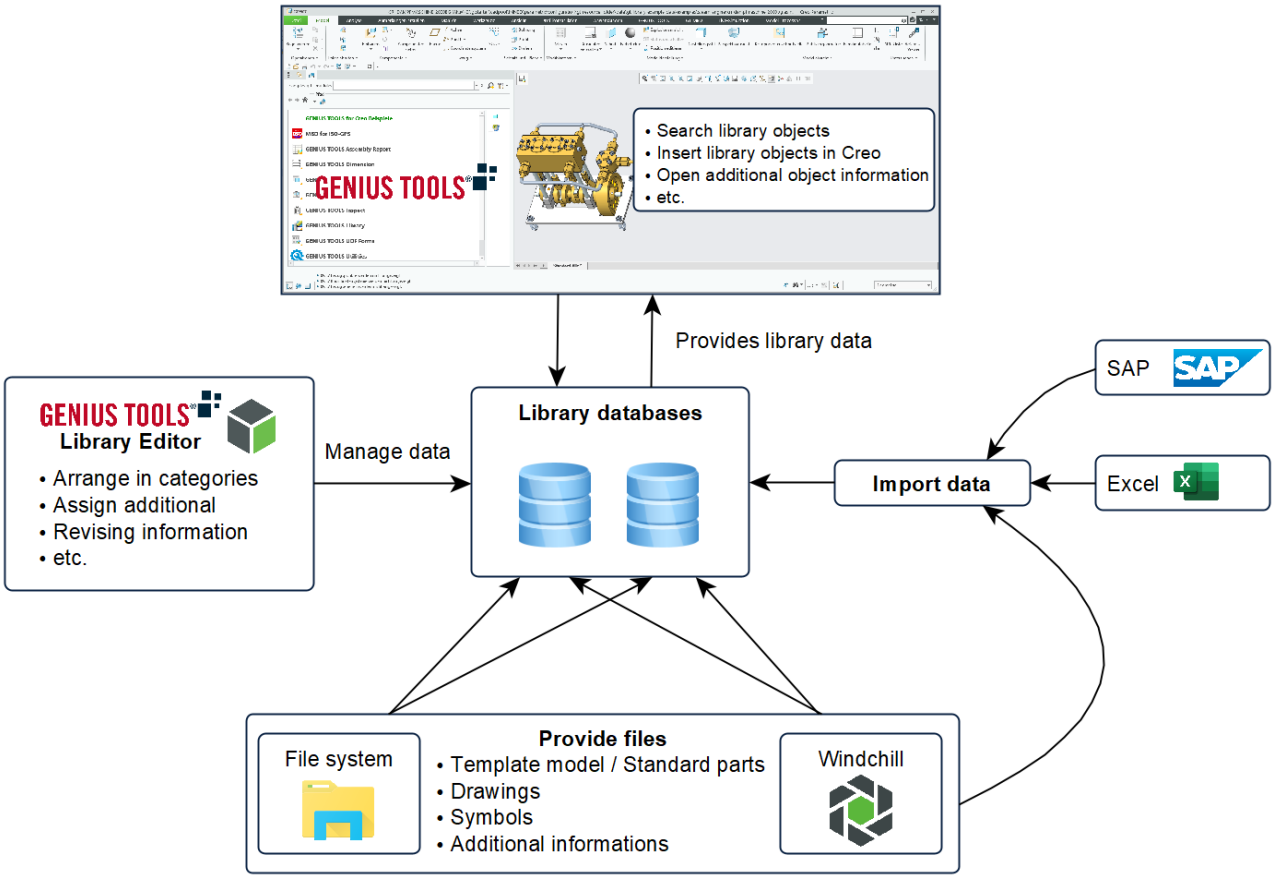



*Library Editor* Managing libraries and library contents

# 12.1 Fundamentals

This introductory section gives you an insight into the operating principles of *Library* and deals with possible use cases. In addition, you will find a glossary of important terms here for a better understanding.

*Library* provides you with database-supported library management systems. It consists of the Editor (*Library Editor*) for managing library databases and the *Library Browser* for use in Creo Parametric.




## 12.1.1 Glossary

**Category tree**

The category tree in the editor and in the library area of *Library* contains all categories and library objects included in a library database. The visibility of the library objects in Library depends on the status set.


© 2026 INNEO Solutions GmbH

**Library actions**
Library actions are actions that can be executed on library objects e.g. “Open” or “Insert into assembly”. The availability of actions in *Library* depends on the object type and the configuration in the *Library Editor*.

**Library browser**
The Library browser is the user interface of *Library*, see User interface 298.

**Library categories**
Library categories are virtual organizational elements in the libraries of *Library*. Categories can be named as desired and can contain any objects. They are used to structure a library logically. Library categories can also represent real directories and can be used to quickly switch the working directory.

**Library databases**
contain additional information on the managed Creo models, and being centrally available, enable all users to have the same information and to work on the same data.

The actual Creo models are not stored in the databases. When importing Creo model information or manually entering library objects, only the required information such as type or location of the models is captured.

**Library objects**
can be enhanced with metadata (additional information). This could be descriptive parameters of a Creo model, the object type (part, assembly, drawing etc.), or the current status of a library object, for instance. This metadata allows easy finding of library objects in the *Library Browser* via search or a status filter. Also other additional information such as selection tables, locations of info documents, a list of models to be copied as well, or how a model should be opened or assembled, can be deposited for library objects.

**Object types**
Each library object has a type. It specifies possible properties and actions of the object.

**Object links**
Library objects can be linked within a library. This way, no copy of the object will be created in the database. Any changes to the object will affect all links of the object. This way, the same object can be available via different paths of the category tree at the same time.

**Selection table**
A selection table is a table that consists of several library objects. In a selection table, these objects can be grouped by parameters and dimensions. A quick selection can be made on the basis of the defined criteria via the detailed view in *Library*.


© 2026 INNEO Solutions GmbH

## 12.2 Usage

This section contains information on using *Library* . It describes the general structure of the program.

Use *Library* to;

– Find library objects via the full-text search

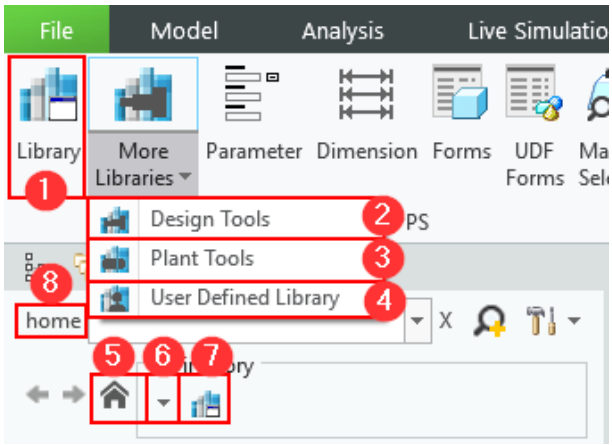
– Access library objects any time

– View detail information for any object

– Use single parts and assemblies as templates

### 12.2.1 Starting the library: in all Creo modes

You can assign different libraries to buttons in the ribbon menu for easy access:




1. Library: `gtl_favorite_button1_liblink` The stored default name for this library is *company*. In SUT: *sut_int_de_creo*

You can access up to three additional libraries by clicking the button *More Libraries:*

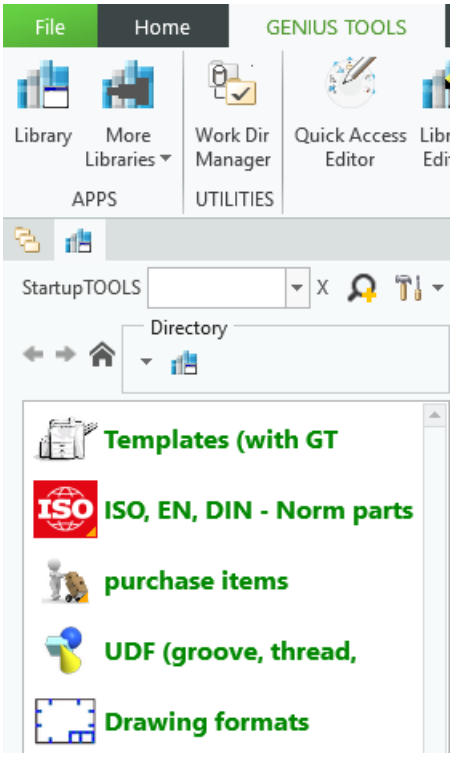
2. Library: `gtl_favorite_button2_liblink` The stored default name for this library is *designtools*. In SUT: *designtools_tbx*
3. Library: `gtl_favorite_button3_liblink` The stored default name for this library is *planttools*.
4. Library: `gtl_favorite_button4_liblink` The stored default name for this library is *user*.
5. Home library: The house icon is visible, if a database is defined under `gtl_home_db`.
6. List of all retrievable libraries under `gt_resource_folder\library`.
7. Displays the start categories of the current library.


© 2026 INNEO Solutions GmbH

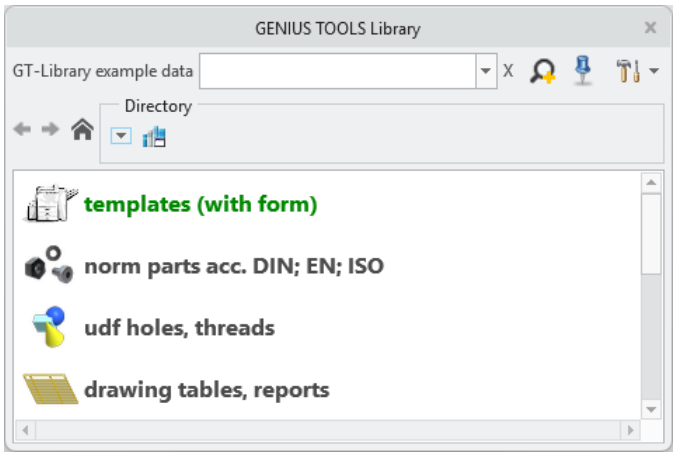
8. Display name of the current library: here: *home*. When you open Creo Parametric, the default library, `gtl_start_db`, opens. If no library is specified under this configuration option, the first library listed under 6 will appear.

## Configuring the display window

*Library* can be configured to be displayed in a separate window. In this case, *Library* opens in a Creo window that cannot be closed. This display is controlled by the configuration option `gtl_run_mode`.




*Library in the Creo navigation area*




*Library as seperate Creo window*

– Each opened Creo window has its own library browser.
– When the Creo model tree is detached, the library browser is displayed in the same window.
– The library browser as an independent Creo window is a library for all Creo windows and can be positioned freely. This setting is a useful choice for multi-monitor setups.
– Use the pin icon to attach the standalone window to its position.

## Open library from another library

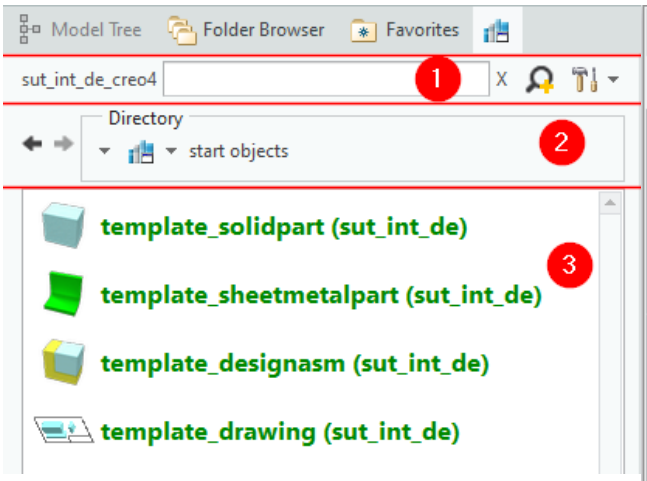
Create an object *Link* in a library. In the object property, enter the opening link under *Object Source*. You can copy this opening link to the clipboard in the target library via the right-click menu of the category tree, see Creating a new library object 369.


© 2026 INNEO Solutions GmbH

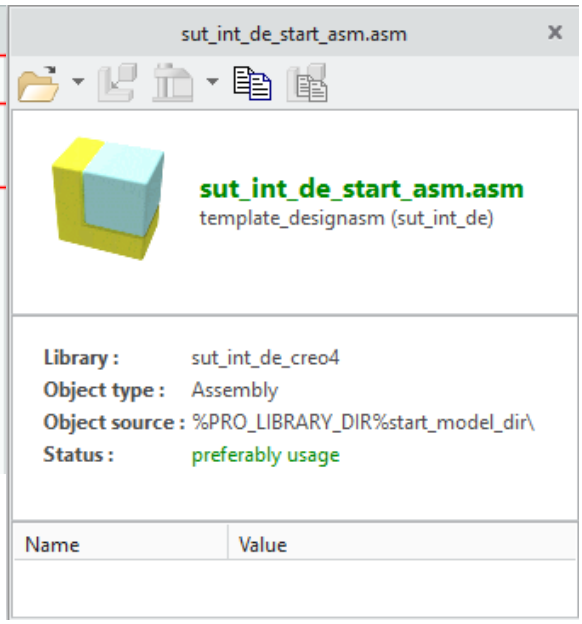
**Example**
sut_int_de_creo | (Opens the library sut_int_de_creo on the top level)

## 12.2.2 User interface

The user interface of *Library* consists of the following elements:




*User interface of Library (library browser)*




*Details window*

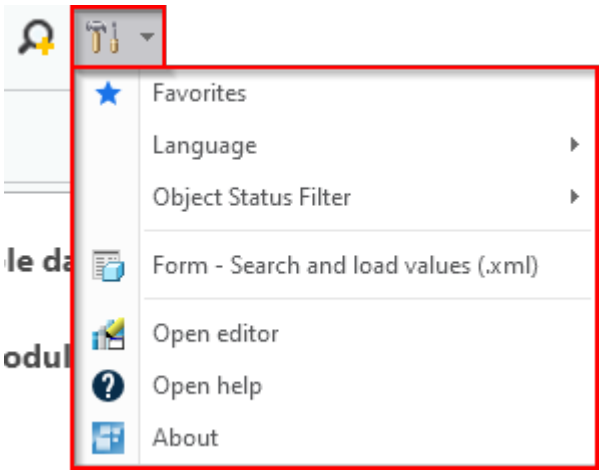
1. Search and filter[304] with tool menu[298]
2. Navigation bar with library selection[310]
3. Object display

The display of library objects is controlled through the search- and filter area or via the path option. Click on a library object (here: template_designasm) to open the detail window[312] with information about and actions on the library object.

## Tools Menu

Access the following functions via the tools menu


© 2026 INNEO Solutions GmbH




| Function | Description |
|---|---|
| **Favorites** | Show or hide the favorites bar, see Managing favorites[317]. |
| **Language** | Change the language in which library objects are displayed. |
| **Object Status Filter** | Filter library objects by their status, see Searching and filtering library objects[304]. |
| **Form – Search and load values (.xml)** | Import XML files containing form values, see Searching for forms and import values from XML[296]. |
| **Open editor** | Opens the editor to edit one or more libraries, see Library Editor[322]. |
| **Open help** | Opens this help. |
| **About** | Opens the user interface that displays the configuration options for GENIUS TOOLS for Creo, see Viewing the configuration[763]. |

## 12.2.3 Using library objects

### Assembling a standard part with selection table

**Prerequisites:**

– an assembly to insert a standard part into

– a properly configured library database (actions, status)

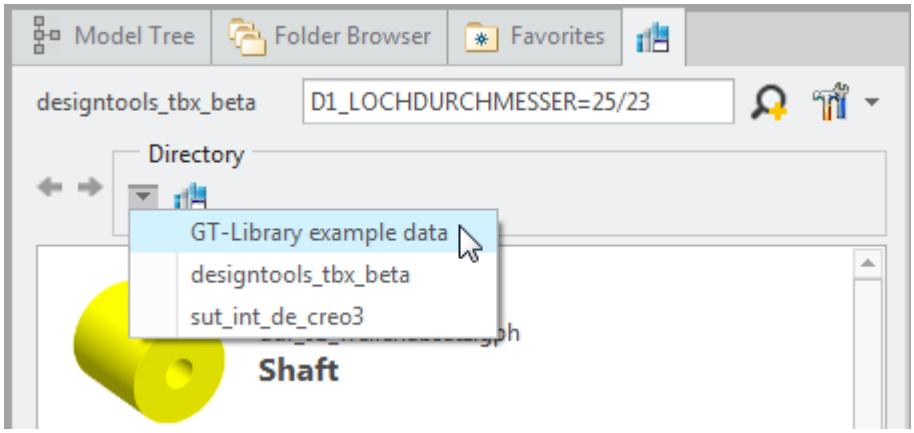
– standard parts with selection tables in the database

**Steps: 1 – Library browser**

1. In Creo Parametric, switch to the assembly to insert the standard part. Make sure not to have a mode such as component placement or sketch mode activated.

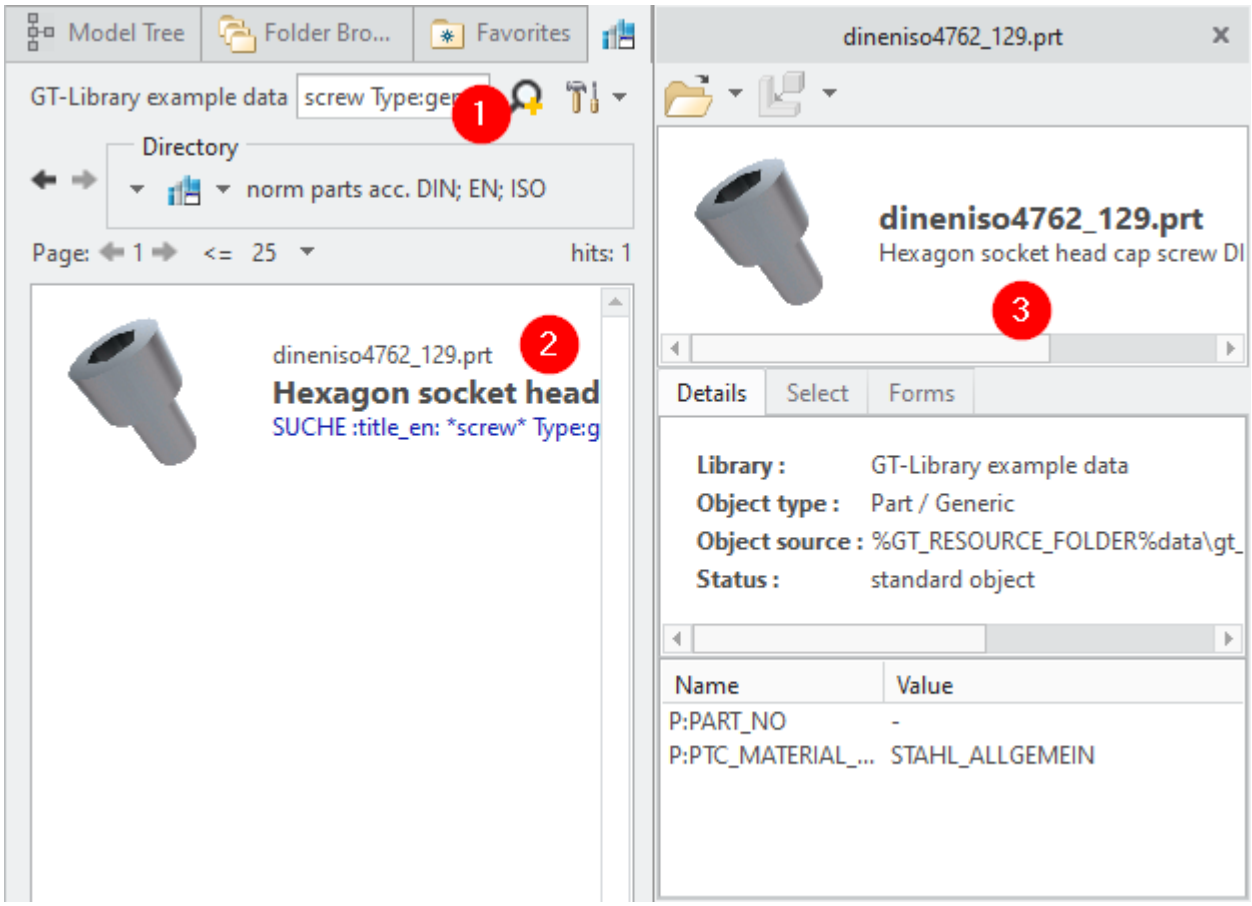

© 2026 INNEO Solutions GmbH

2. Navigate to the Library tab in the navigation area or switch to the Library window.
3. Open the library containing the standard parts needed.




*Opening the library that contains the standard parts needed*

4. Use the search field (2) to find the standard part or navigate the library structure.
5. Click the standard part in the object display (2) and switch to the Details window (3).



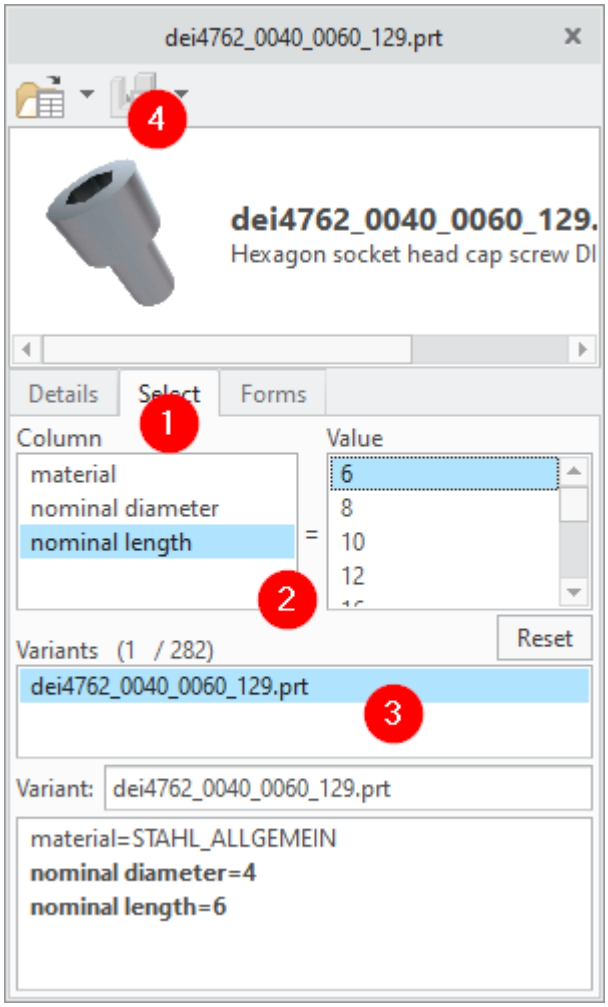

*Standard part in library browser (left) and in details window (right)*

**Steps: 2 – Details window**

6. Open the *Select* tab (1).
7. Select the standard part using the selection table (2).


© 2026 INNEO Solutions GmbH

8. Select the desired variant (3).
9. Click the *Insert selected variant* button (4).
10. Insert the part into the assembly as usual in Creo.




*Details window*

## Assembling a new part with a Form

**Prerequisites:**

– an assembly to insert the part into

– a properly configured library database (actions, status)

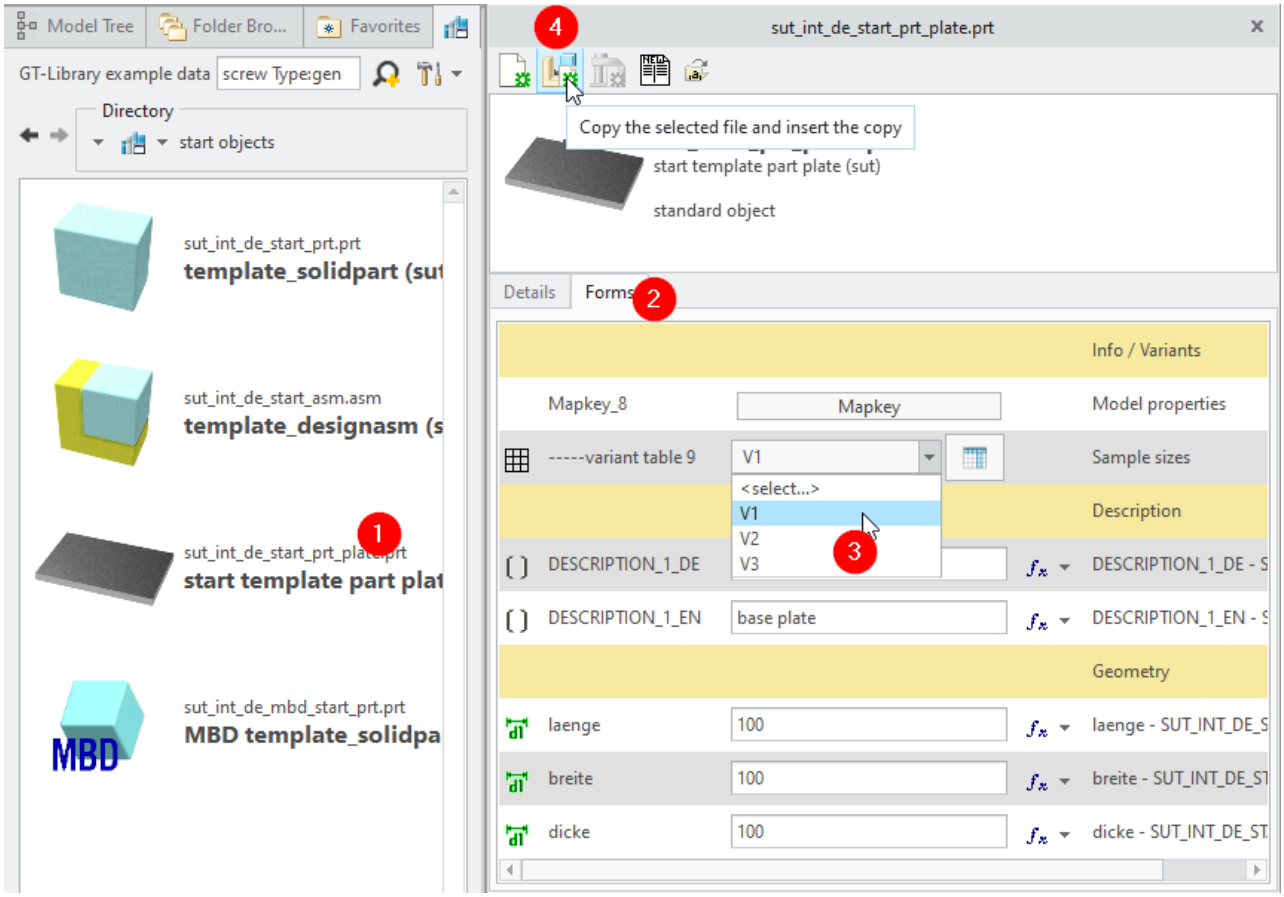
– a part with an external form in the database

**Steps:**

1. Switch to the assembly to insert the part in Creo. Make sure not to have a mode such as component placement or sketch mode activated.
2. Navigate to the Library tab in the navigation area or switch to the Library window.
3. Open the library containing the part with the form.

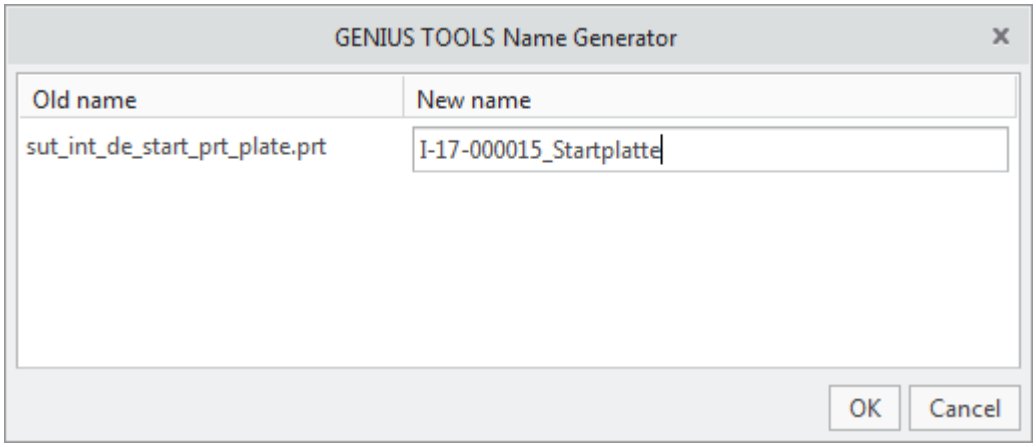

© 2026 INNEO Solutions GmbH

4. Use the search to find the desired part or navigate the library structure.
5. Click the part in the object display (1) and switch to the Details window.
6. Open the *Forms* tab (2).
7. Configure the part in the form (3).
8. Click the button *Copy the selected file and insert the copy* (4).




*Object display in library browser (left) and in details window (right)*

9. Select a Name Generator in the following dialog and confirm your selection with *Next*.
10. Complete the new part name in the last dialog and click *OK*.




*Dialog box of the Name Generator*


© 2026 INNEO Solutions GmbH

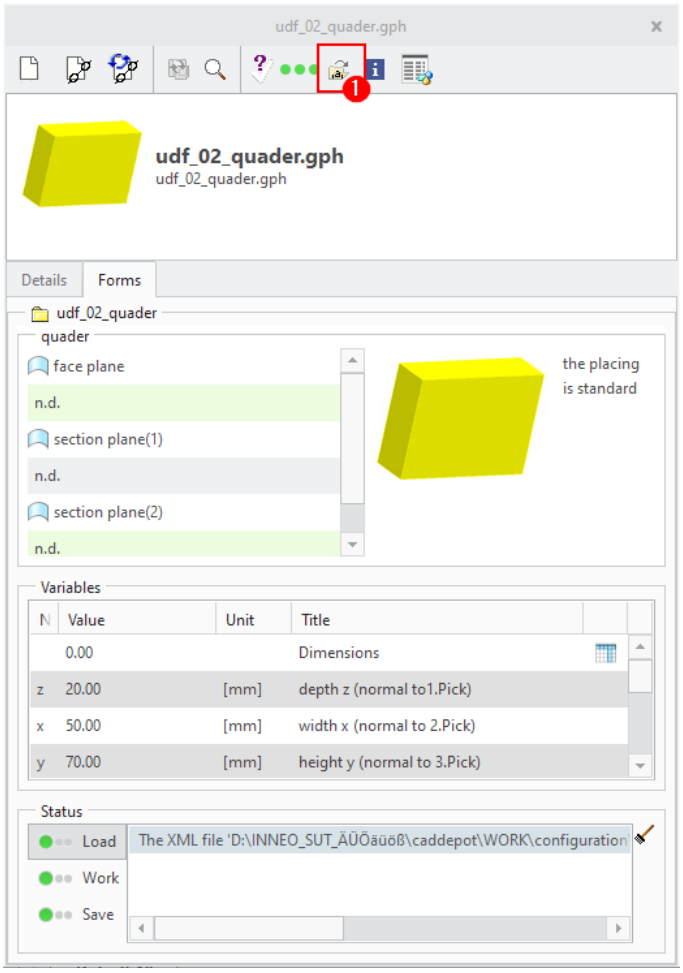
11.Insert the part into the assembly as usual in Creo.

## 12.2.4 Working with templates

### Part modeling – sheetmetal and design part mode

Create new parts from existing library objects with *Library*. Use *Forms* for this. These objects can be assembled with Creo standard capabilities.

Alternatively, you can create *UDF Forms*[601] to make it easier to incorporate user defined features that can be edited later.




*A UDF Form in the details window*

Use library objects like sketches to create elements in Creo Parametric using the sketch mode.

### Completing assemblies – design assembly

Complete your assemblies with parts and subassemblies stored in your libraries. There are three methods for this:


© 2026 INNEO Solutions GmbH

1. Assemble models such as standard or purchased parts without changes
2. Assemble other models using the template method
3. Create new models and drawings, with new rule-based file names, with minimal clicks

## Completing drawings – drawing mode

Use library objects such as drawing frames, symbols, tables and notes to complete drawings with little effort.

## Using templates

When utilizing single parts and assemblies with copy rules, you will be able to specify which component, drawings and additional object should be copied. In addition, you can define name rules that are applied automatically to copied objects. This way, assembly templates can be set up faster, ready for use in new projects, including the required manufacturing documents.

# 12.2.5 Searching and filtering library objects

Search the current library by keywords or use additional search criteria in the advanced search, see also List of search operations 308.

The search and filter area consists of the following elements:




1. Selected database
2. Search field for direct search input 304
3. Remove entries from the search field
4. Advanced search 305 by object type, parameter, dimension
5. Tool menu 298 with Object status filter 306

## Search field for direct search input

The following options are available when using the search field:

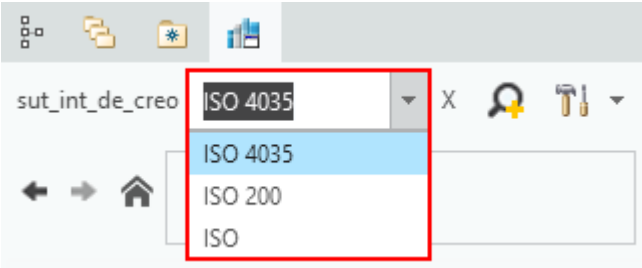
– **Search in current category:** Enter a search term in the search field and press Enter. The currently selected category will be searched for the term, and the results are displayed in the object view.


© 2026 INNEO Solutions GmbH

– **Search the entire library:** Pressing Enter again after searching the current category expands the search to the entire library.
To start the search immediately in the entire library rather than the current category, set the configuration option `gtl_search_tree_dependent` to 0.

Changing the search term will restart the search in the selected category.

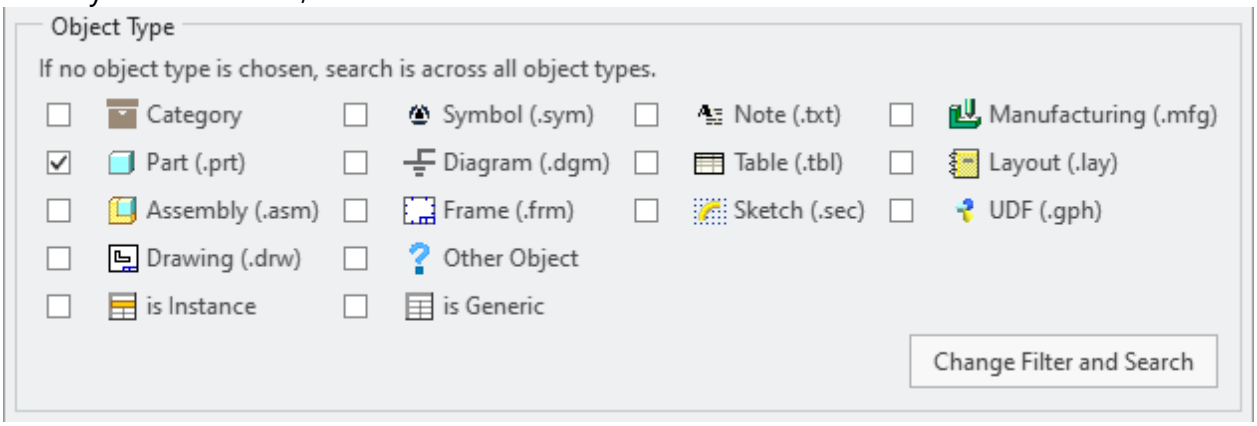
All searched terms of a Creo session are now visible in a dropdown list:




## Advanced search

The magnifying glass button opens the advanced search dialog. With the advanced search, a search can be narrowed down by object types, parameters and dimensions.

– **Object type**
Check the boxes next to the types you want to search, then click *Change Filter and Search*. The filter will be applied immediately to the search field, in addition to any terms you have already entered. Then, the search will be executed.



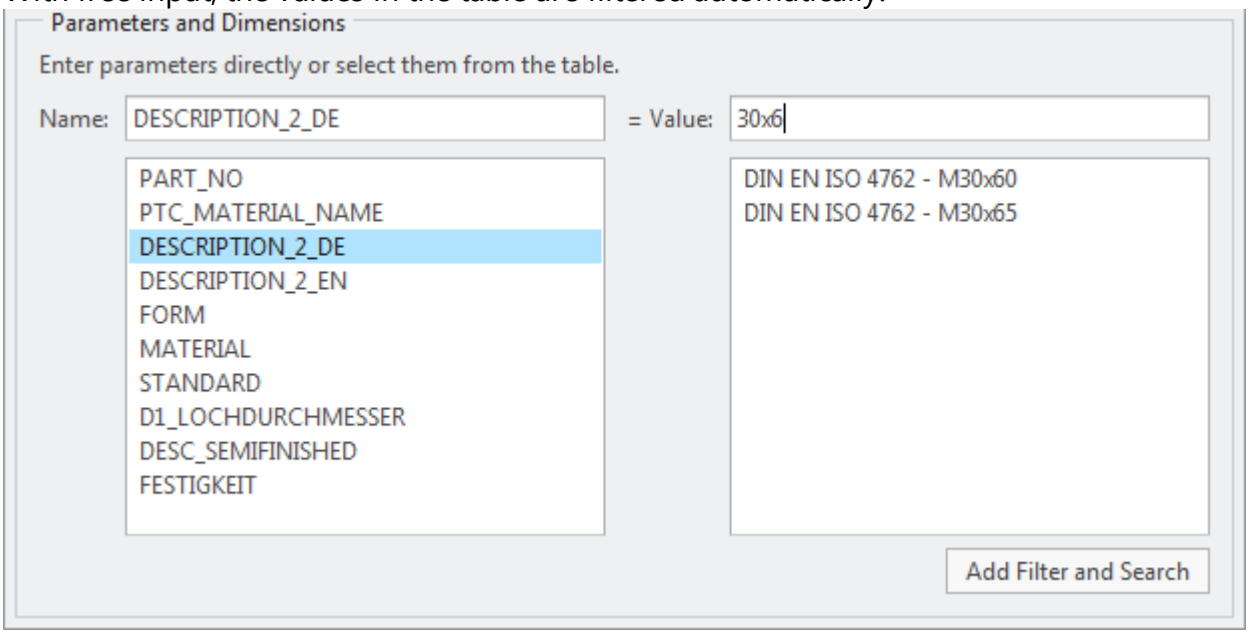

Object types are combined with OR. Exceptions are the object types *is Instance* and *is Generic*: they are combined with AND. For example:

– *Part* and *Assembly* are activated in filter: A library object must be a part or an assembly.

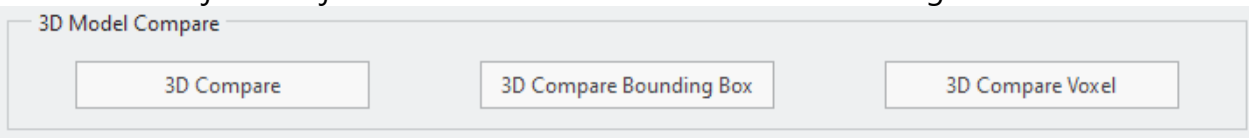
– *Part* and *is Instance* are activated in filter: A library object must be a part and an instance.


© 2026 INNEO Solutions GmbH

– **Parameters and Dimensions**
Parameters and dimensions must be added to a library database to be accessed here. Select a parameter or dimension from the list. Then enter a value from the value list or enter a value freely. Refresh the search using the *Add Filter and Search* button.
With free input, the values in the table are filtered automatically.




– **3D Model Compare**
The section for searching for 3D-like objects only appears if the selected library contains 3D-indexed objects. Objects are indexed in the Batch Mode[358] dialog.



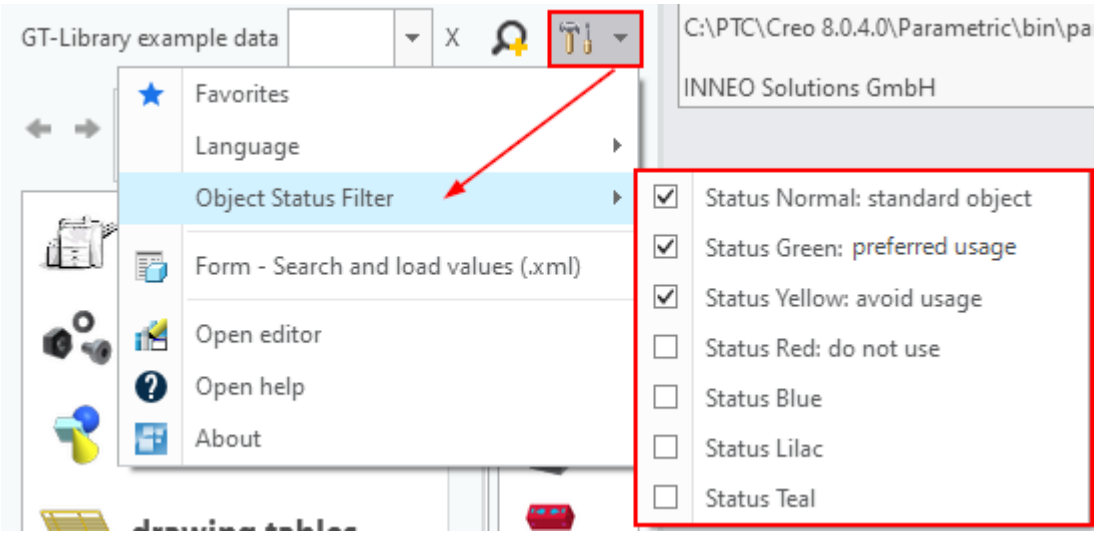

The search buttons correspond to the search operators[308] *mdl3d, mdl3d_bb or mdl3d_vo* which you can alternatively enter directly in the search field.

**Exclude from search**
Be aware that all object data, including parameter values, are searched through. You can exclude individual parameters and dimension from the search index, see Customizing the search function[363].

## Status Indication by Color

In the Library Editor[322], a color-coded status can be assigned to a library object. These colors can be changed using the configuration options `gtl_*_color`. Use the *Object Status Filter* to filter library objects based on their status.


© 2026 INNEO Solutions GmbH



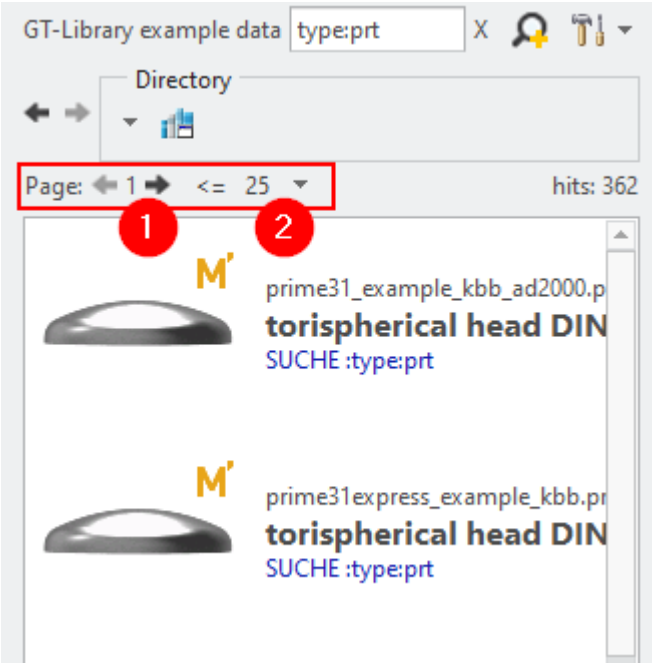

Open the tools menu[298] and select under *Object Status Filter* the statuses you want to display All library objects with the selected statuses will be displayed in the object view.

The object filter does not apply to categories; i. e., an object with a green status will be displayed even if it is located in a category with a red status.

**Please note:** The object status filter is applied in addition to the direct search entry and the advanced search. If these settings conflict, the direct search entry overrides the object status filter.

## Displaying search results

After a search item has been entered an additional bar appears which displays information on pages (1) and hits (2).





© 2026 INNEO Solutions GmbH

## 12.2.5.1 List of search operations

You can enter the following search operations in the direct search field.

| Search operation | Term | Explanation | Input example |
|---|---|---|---|
| Object type* | type: | Finds objects of a specific category, file extension (.prt, .asm etc.) as well as instances and generic parts<br>**Note:** Don't insert a space character between multiple terms | type:prt<br><br>type:prtm,asm |
| Object status | status: | 1=normal, 2=green, 3=yellow, 4=red, 5=blue, 6=lilac, 7=teal (turquoise) | status:2 |
| Parameters and value | parameter-name=value | Finds objects with the specified parameter and its value | material=wood |
| Dimensions and value | maß=wert | Finds objects with the specified dimension and its value | d19=10 |
| Literal search | quotation mark: " " | Finds objects that contain the exact phrase with all words. There is no case sensitivity. | "Hexagon regular nut DIN EN ISO 4032" |
| BOOLEAN AND | AND | Finds objects that contain each search term | Hexagon regular nut AND Plate |
| BOOLEAN OR | OR | Finds objects that contain one of the search terms | Hexagon regular nut OR Plate |
| Parenthesize search terms | Brackets: ( ) | Joins two or more search items<br>**Note:** You must type a space character between a value and a bracket | ( m6 OR m10 ) type:prt |


© 2026 INNEO Solutions GmbH

| Search operation | Term | Explanation | Input example |
|---|---|---|---|
| Exempt from search | exclamation mark: ! | Does not display the objects that are searched for after the exclamation mark (e.g.: Find all objects except of the type part) **Note:** You must type a space character after the exclamation mark | ! type:prt |
| Hidden objects | !tree | Displays objects that have not been added to the model tree | !tree |
| Hidden objects | !selection | Displays objects that have not been added to the selection | !selection |
| Similar objects ** | mdl3d | Displays objects similar to the one in use | mdl3d |
| Similar objects – bb** | mdl3d_bb | Displays objects with similar bounding borders | mdl3d_bb |
| Similar objects – vo** | mdl3d_vo | Displays objects with similar voxels (grid points) | mdl3d_vo |

*this search item can also be generated by the object type filter in the dialog box Advanced Search

**for this search an object needs to be active (opened)

## Examples

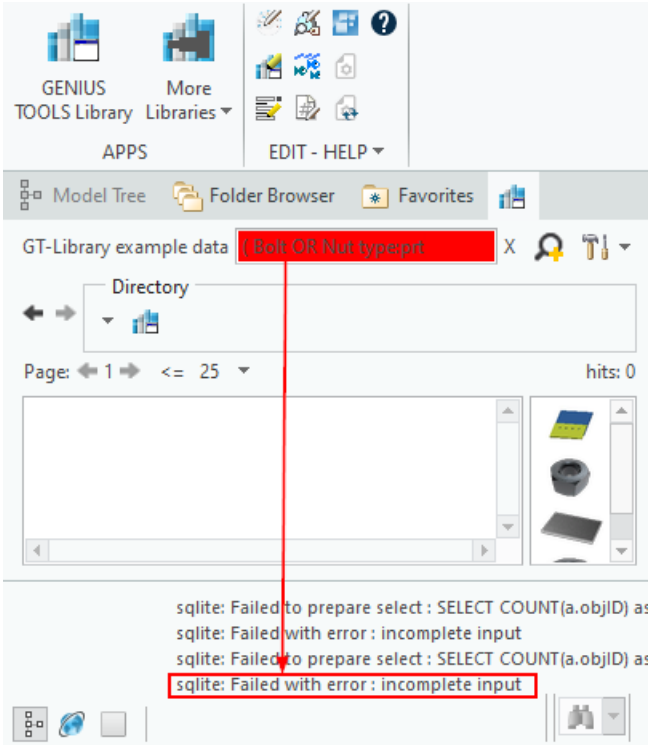
1. The search entry `type:prt,asm,drw` searches for all parts (.prt), assemblies (.asm) and drawings (.drw)


© 2026 INNEO Solutions GmbH

2. The search entry "*material=gold*" searches for all objects with a parameter “Material” and the parameter value “Gold”.
3. The search entry `( status:2 OR status:3 ) material=gold ! type:asm` searches for all objects with the status Green and all with status Yellow that have a material parameter with the value “gold” and are not assemblies. Take care to set a space character after and in front of the brackets.
4. The search entry `! status:3 ! status:4` searches for all objects that neither have the status Green nor Yellow.
5. The search entry `drill bush ! “DIN 179”` searches for all objects which contain the term “drill bush” but not the term “DIN 179”.

### Examples for incorrect search entries

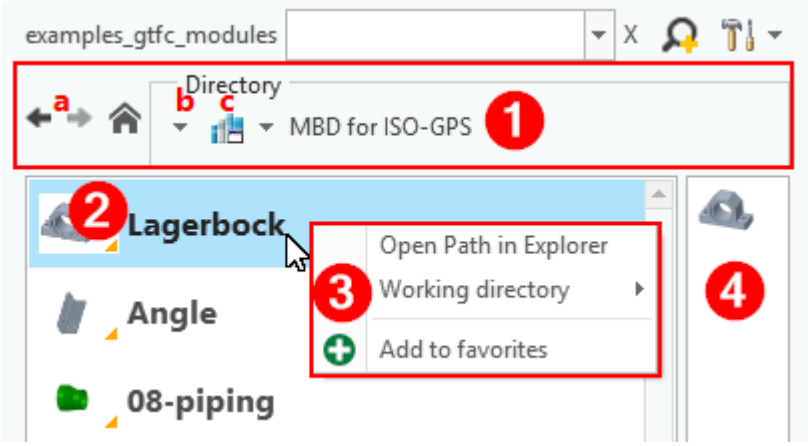
1. The search entry `status:2 status:3` is a Boolean AND, which does not produce any results since objects cannot have two statuses. Use a Boolean OR if you wish to search for all objects that either have status 2 or status 3.
2. The search entry `( Bolt OR Nut type:prt` generates an error message because a bracket is missing:




## 12.2.6 Displaying library objects

Below the search and filter area is the navigation area and the object display of the library browser.


© 2026 INNEO Solutions GmbH




1. Navigation area (with the category name displayed when a category has been selected)
   a. Forward and Back Buttons
   b. Selection of the library
   c. First level of the selected library
2. Object display
   The library objects and categories are displayed with preview and description. The status is indicated by the text color, see Object details[330]. The detail window[312] is displayed as soon as the selected object is a library object.
3. Context menu[311]
4. Display Favorites[311]

**Configuration options**
To open library objects and categories with a double click instead of a single click, set the configuration option `gtl_list_use_with_double_click` to 1.

To highlight categories in the object display, you can attach a small triangle to the category by setting the configuration option `gtl_mark_category_images` to 1. This configuration option must be set before you import the image for the category.

The size of displayed preview images (40 or 100 pixels) is set by the configuration option `gtl_img_size`.

# Context menu

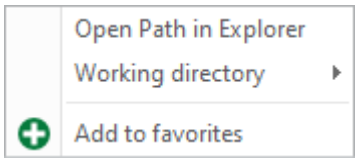
Right-click on an object to open the context menu. Whether you have selected a category or a library item, you always have the following option:

**Add to favorites**: Adds the current category or library object to the favorites bar.

## Context menu for categories

The structure of a context menu for categories depends on the category's configuration, see Category view[327].


© 2026 INNEO Solutions GmbH




**Info**: Opens a linked information document. The function is only displayed when a document has been linked to a category.

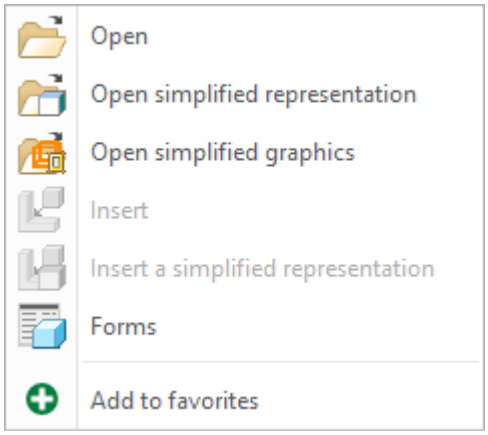
**Open Path in Explorer**: Opens a deposited path in Windows Explorer.

**Change working directory to selected folder**: The Creo working directory is set to the deposited path.

**Please note**: *Open Path in Explorer* and *Change working directory to selected folder* are displayed only when library categories correspond to a folder path. Usually, there is no object source specified when using library categories only for logical structuring, and the functions cannot be used.

### Context menu for library objects

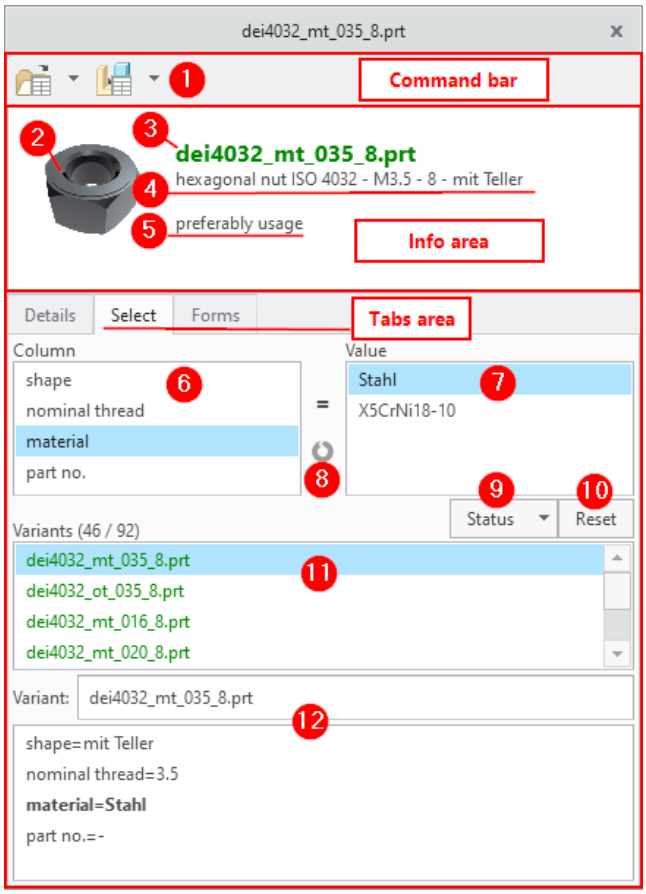
The context menu for library objects depends on the object type and configured actions, see Actions on library objects 348.




## 12.2.6.1 Detail window

The details window displays information about the currently selected library object.


© 2026 INNEO Solutions GmbH




*Example of a library object with instances*

## Command bar

Displays all actions that are available for the selected library object. The active tab can also affect the displayed actions.

1. Actions buttons in the command bar depend on object type, configuration of a library object, and on the current Creo mode, see also Actions on library objects[348].

## Information area

This area displays a preview image and information on the object. Selected objects in the *Select* tab affect the displayed information.

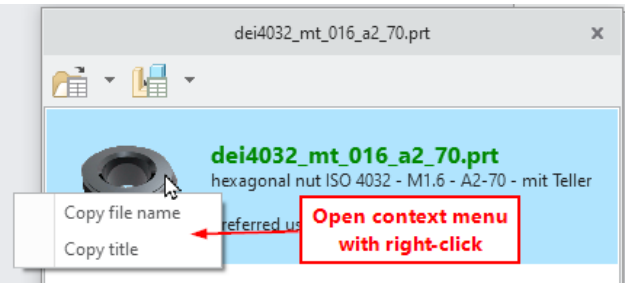
The displayed information can be configured.

2. Detail image: Corresponds to the preview image or can be replaced with a larger, more detailed image, see Working with images[346].
   If a separate detail image[346] has been added to the preview, the file name and object title can be hidden by setting the configuration option `gtl_detail_window_image_show_title=0`.
3. File name


© 2026 INNEO Solutions GmbH

4. Object title (description): Can be hidden with the configuration option `gtl_detail_window_hide_details`.
5. Status information: Can be displayed with the configuration option `gtl_detail_window_show_status_in_head_area`.

Right-click to open the context menu for copying the file name and title to the clipboard.




# Tabs

The tabs displayed depend on the selected library object.

By default, the *Forms* tab is active.

- for parts, if a form is available, otherwise the *Selection* tab is active: The active tab can be changed with the configuration option `gtl_detail_window_preselected_tab`.
- for UDF-Forms: The active tab can be changed with the configuration option `gtl_detail_window_preselected_tab_gph`.

## Details tab

Displays general information on the library object: library, object type, object source and object status[339].

## Select tab

The tab *Select* displays a table of all instances of the library object and allows you to search by the following criteria:

6. Feature selection: Select variants of a library object based on its properties.
7. Property value: Select values to the corresponding property.
8. Select the next column after value selection: When enabled, the cursor automatically jumps to the next property after a selection is made.
9. Status filter: Filters the instances by object status[333]. The preselection of the checkboxes depends on the selection in the object status filter[306]. However, the selection set here does not affect the object status filter.
10. Reset: Deletes the chosen selection.
11. Variant selection: The available library objects are selected based on the chosen properties.


© 2026 INNEO Solutions GmbH

12. Variant display: The library object variants selected above, along with their selected properties, are displayed in bold.

The content depends on the configuration in the *Library Editor*.

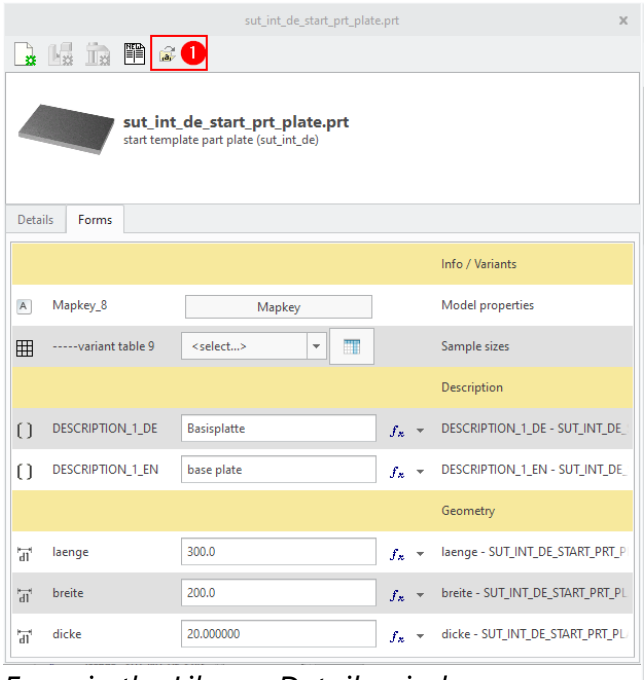
## Forms tab

The Forms tab in the Details window displays Forms and UDF Forms for library objects.

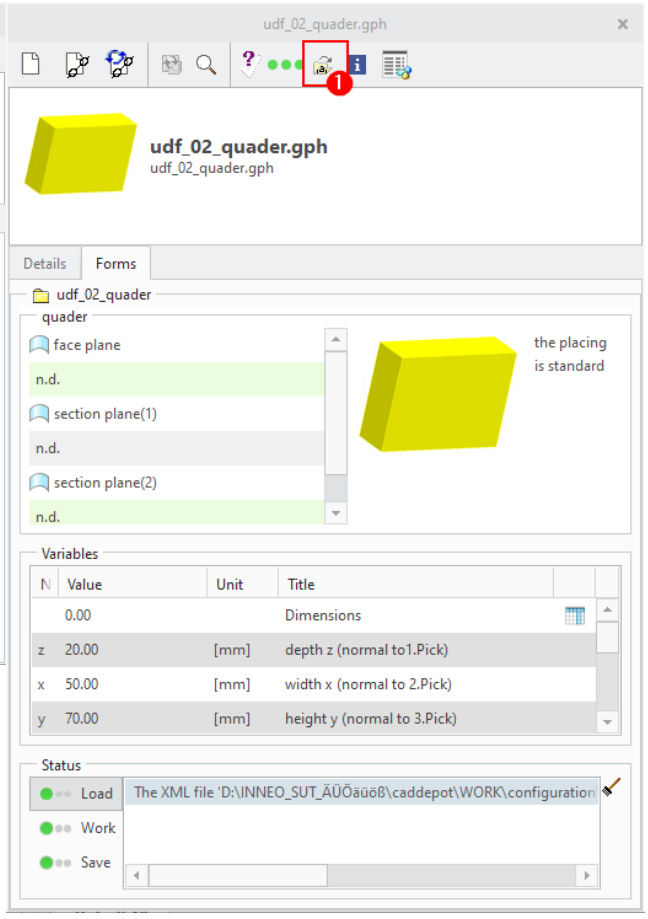
- Displays a Form or UDF Form (created with *Forms* or *UDF Forms*) from the library database matching the library object.
- If a selection table exists for a library object, the Forms tab is always displayed. Individual variants can contain Forms.

**Tip**: Use *Forms*[123] and *UDF Forms*[601] for editing models that have already been created and UDFs that have already been inserted.

Use the tools menu[298] in the Library browser to access the function *Form - Load values (.xml)* if your company uses the XML interface for filling in Forms automatically. For information on how to work with the XML interface, see Forms search and read values from XML[318].




*Form in the Library Details window*




*UDF Form in the Library Details window*


© 2026 INNEO Solutions GmbH

1) The function  *Read values from CSV* searches mask values and reads these values from a CSV-file.

**Forms**
Models such as assemblies or parts can contain a Form. If the Form has also been imported into a library, it is displayed in the Forms tab in the Details window.

Use Forms to customize dimensions and parameters prior to assembling or opening a model, or to control it via value tables.

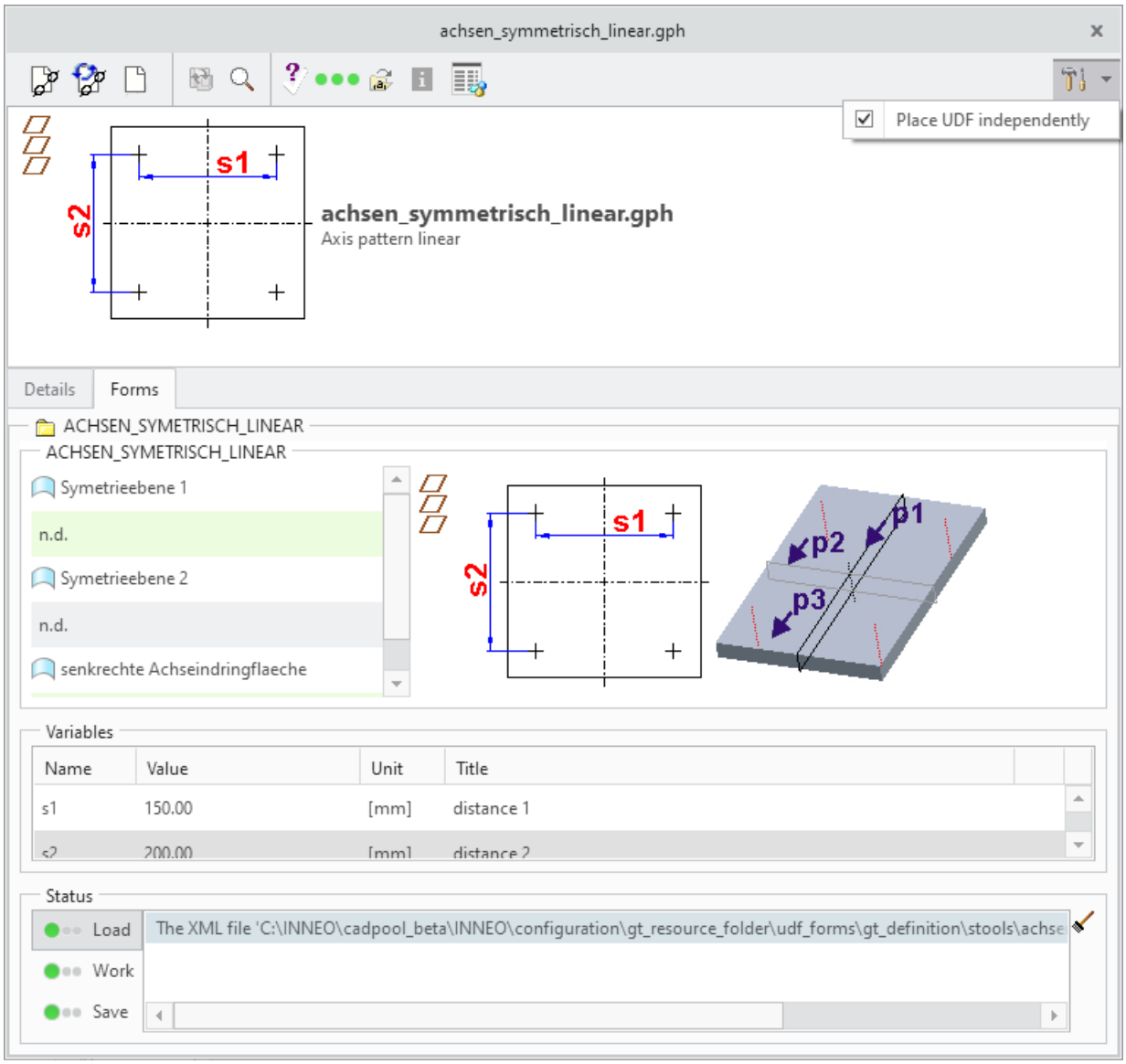
Using Forms in the *Library Details* window is similar to operating *Forms*.

**UDF forms**
If UDFs have been included in a library, they can obtain a UDF form. This will be displayed in the Forms tab in the Details window.

The command bar differs greatly for a UDF Form (i.e. a GPH file). Check the usage in the chapter *UDF Forms*[601] .

Use the UDF form to define placement references and to customize properties prior to assembly, e.g. by placing it dependently[608] or independently.


© 2026 INNEO Solutions GmbH



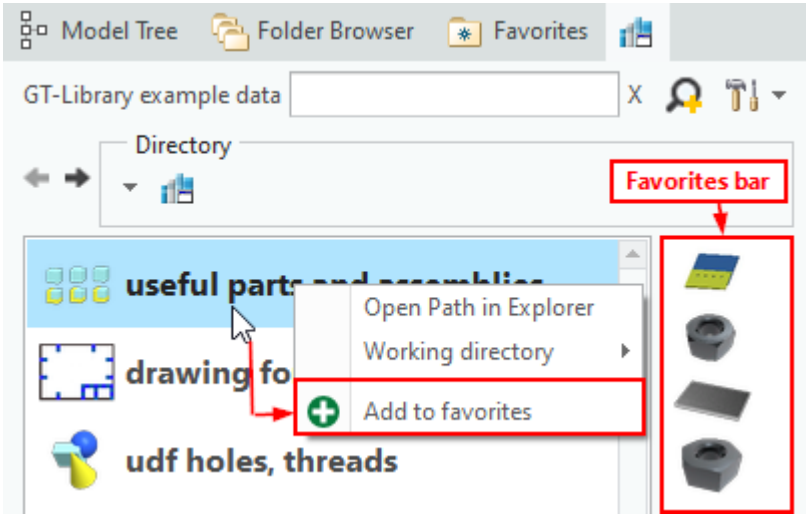

*Detail window for a UDF form*

## 12.2.7 Managing favorites

You can display library objects that you frequently need in a favorites bar in the library browser. You can mark both individual library objects and categories as favorites. A favorites bar is available for each library.


© 2026 INNEO Solutions GmbH

## Adding and removing favorites

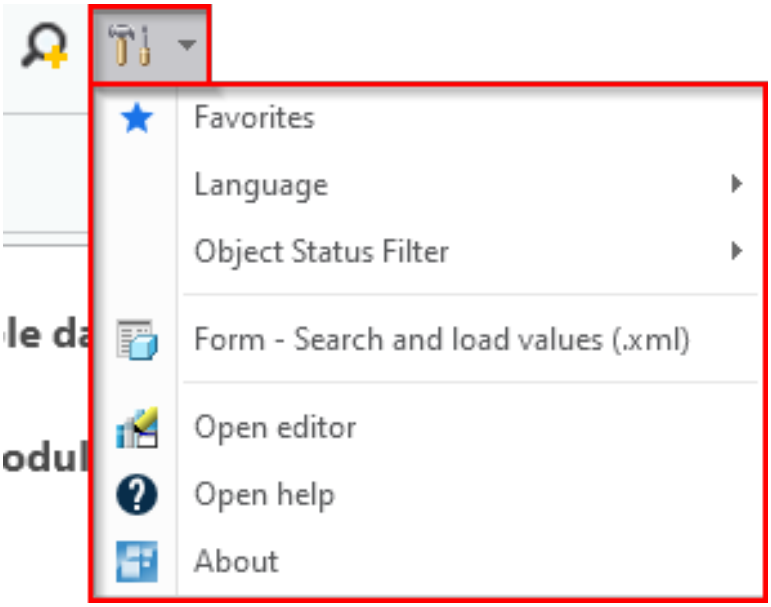



To add an object to the favorites bar, right-click it to open the context menu and select *Add to favorites*. The selected object is displayed in the favorites bar with a small preview image.

To remove an object from the favorites bar, select the object in the favorites bar, then open the context menu and select *Delete from favorites*.

## 12.2.8 Searching for forms and import values from XML

*Library* comes with an XML interface for importing pre-calculated values for Forms. Your company has to define whether this interface is used and for which Forms to create XML files, see XML interface for Form values 364.




To load Form values from an XML file, proceed as follows.


© 2026 INNEO Solutions GmbH

1. Navigate to the Library tab in the navigation area or switch to the Library window.
2. In the tools menu, select *Form - Load values (.xml)*. A file selection dialog is displayed.
3. Select the XML file that contains the required values and click *Open*. The list in the library browser is filtered to the object or objects that contain the Form specified in the XML file. The Form is filled in the background.
4. Click the library object with the Form to open the detail window. The values from the XML file have been set in the Form.
5. Verify the Form values. You can copy the library object now or verify the names for the file copies. The names for the file copies are specified in the XML file together with the Form values.

# 12.3 Configuration

In this section, you will find further information on the folders used in the resources directory. In addition, you will find information on configuring libraries for *Library* in the *Library Editor* tool, and on using multi-lingual libraries.

## 12.3.1 Areas in the resource directory

Library uses the resources directory of *GENIUS TOOLS* for Creo. In the `...\gt_resource_folder\library` folder, you can find the Library-database (file extension: .db) and a folder with the same name for each database.

Resource directory in a *Startup TOOLS* installation: `%GTS_ROOT_DIR%\configuration\gt_resource_folder\`

Resource directory in a *GENIUS TOOLS* for Creo installation: `<GTfCDirectory>\gt_resource_folder\`

The path to the resource directory can be edited with the configuration option `gt_resource_folder`.

These folders contain the preview images for the library objects. The original preview images for library objects are located in the *img* subfolder. The *img_w40* and *img_w100* subfolders contain automatically imagestyleclass="Default" scaled preview images. The optional *img_detail* and *img_tooltip* subfolders contain images for the detail window and images to display as tooltips for library objects.

In addition, you can find backup copies of your databases in the *library* folder. As soon as a database is opened via the *Library Editor*, a backup copy is created. However, only one backup copy per day is created.


© 2026 INNEO Solutions GmbH

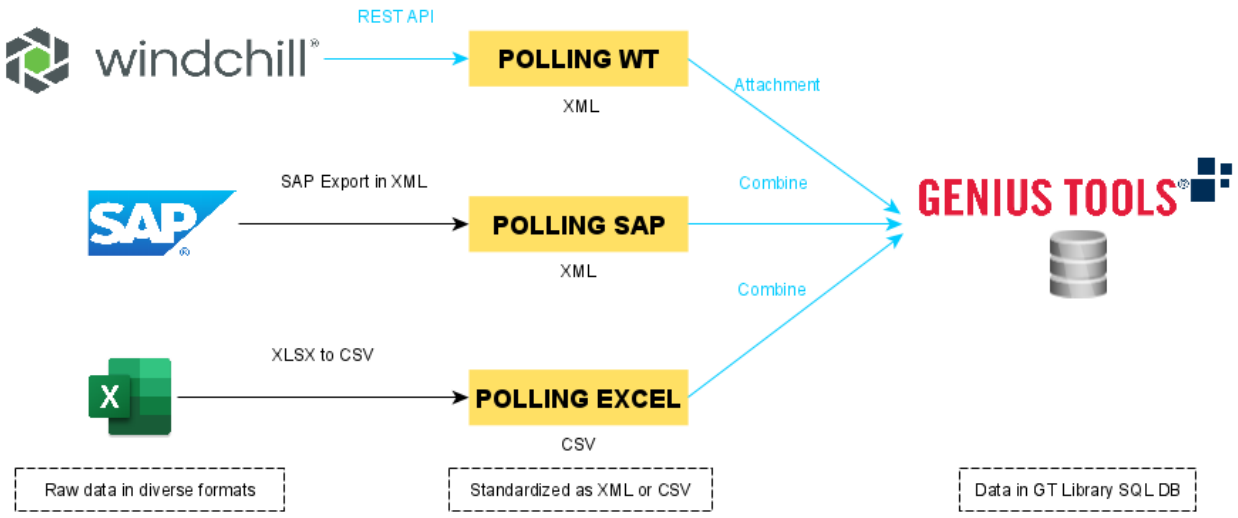
To restore the initial state of a database, unload it from the editor and the Library browser. Then, delete the database. Rename the desired backup copy with the name of the database. The database is now ready for use again.

## 12.3.2 Import data from PDM/PLM

You can import Creo model and meta data from an existing library or other directory structure, e.g. from a PDM, PLM or ERP system, into a library for *Library*. This process is carried out by our module *Library Data Importer*.

In addition, information from other systems can be transferred. The combined information are updated and can be searched.

All processes that import, convert or map can be started in a time-controlled way. Source formats can be CSV, XML (with XSLT) and Windchill REST.




For further information consult the manual *Library Data Importer.pdf* which is located in the *help* directory in your operating environment.

## 12.3.3 Multilingual use

For distributed work at multiple cross-country and cross-language sites, *Library* supports a multi-lingual user interface as well as multi-lingual libraries.

### User interface

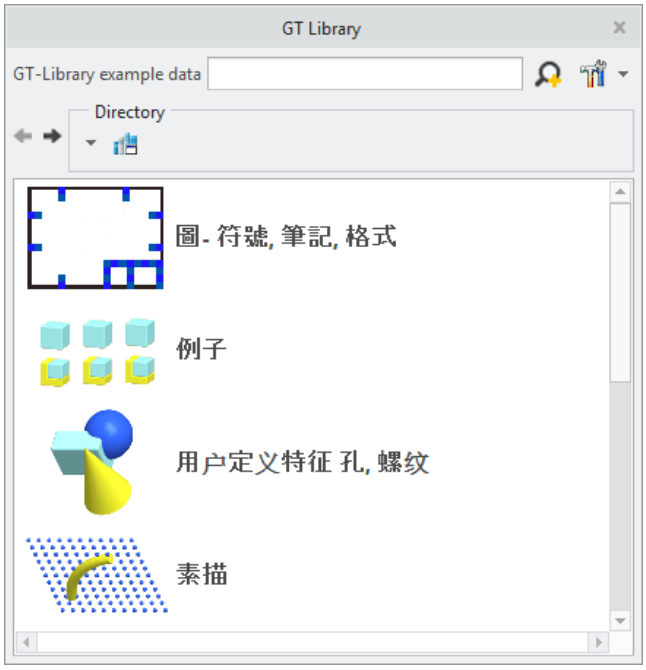
The *Library* user interface depends on the language in which Creo Parametric was started. Currently the languages German and English are available.

When Creo Parametric is started in another language, the *Library* user interface is displayed in English.


© 2026 INNEO Solutions GmbH

# Multi-lingual databases

*Library* databases can contain information in various languages. New languages can be added to library databases via the *Library Editor* using language codes.

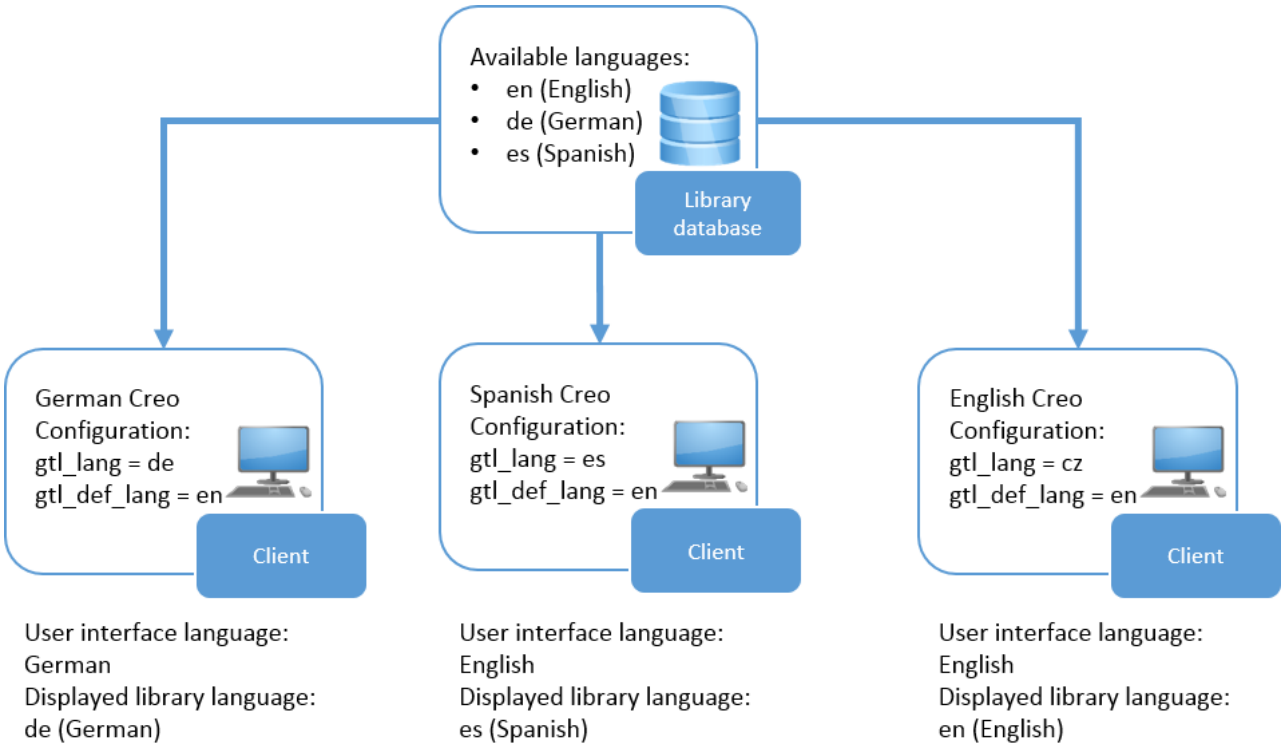



*GENIUS TOOLS Library with Chinese data*

The language displayed in *Library* depends on the client computer configuration. The Configuration option `gtl_lang` specifies (via language code) which language to display.

The Configuration option `gtl_def_lang` specifies a language via language code to be used as a fallback variant if the language specified in `gtl_lang` is not found in a library database.


© 2026 INNEO Solutions GmbH



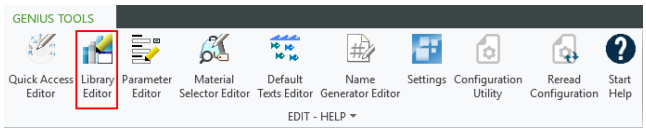

*Displayed localization of GENIUS TOOLS Library and the database contents*

## 12.3.4 Library Editor

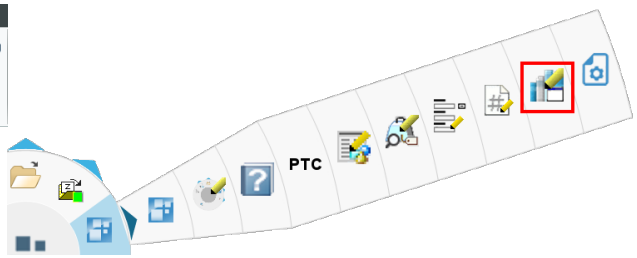
The *Library Editor* allows you to manage your libraries and library content of *Library*.

### Starting the program: Standby mode

To start *Library Editor*, click the *GENIUS TOOLS* tab in the ribbon menu and go to *EDIT-HELP* or by using the *Quick Access* ([<] key).




*Starting via the ribbon menu*




*Starting via Quick Access*

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].


© 2026 INNEO Solutions GmbH

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

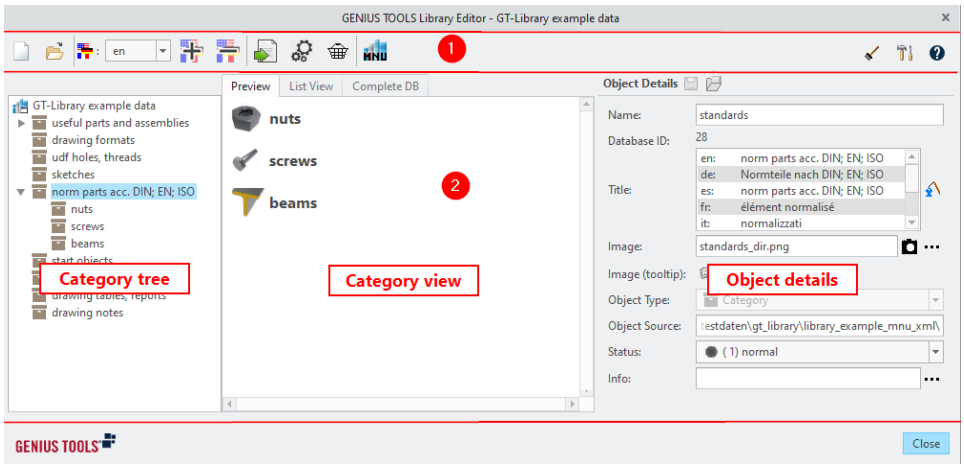
SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## 12.3.4.1 User interface

The user interface of *Library Editor* consists of the following elements:




1. Command bar[324]
2. Library management with
   - Category tree[325]
   - Category view[327] with the tabs Preview, List View, Complete DB
   - Object details[330]


© 2026 INNEO Solutions GmbH

## 12.3.4.2 Command bar

The following buttons are contained in the command bar:

| Icon | Name | Description |
|---|---|---|
|  | Create new database | Creates a new database in the Library resources directory.<br>**Note:** The database opens immediately. Any existing open databases are automatically saved and closed. |
|  | Open database | Opens a database.<br>**Note:** Any database that has already been opened is automatically saved and closed. |
|  | Change displayed language | Changes the displayed language of library elements in the editor. |
|  | Add/remove a language | Adds an additional language to a library or removes an existing one.<br>**Note:** Languages set via the configuration (gtl_lang and gtl_def_lang) cannot be deleted from the database. |
|  | Import objects into current category[359] | Imports additional objects into the currently selected category of the library. These objects are not copied, but rather represented by the corresponding library objects. |
|  | Configure and execute batch mode[356] | Starts a batch run across any library objects.<br>Preview images, parameters or family table information are automatically read or generated. |
|  | Create MNUs[359] | Opens the MNU export dialog. |


© 2026 INNEO Solutions GmbH

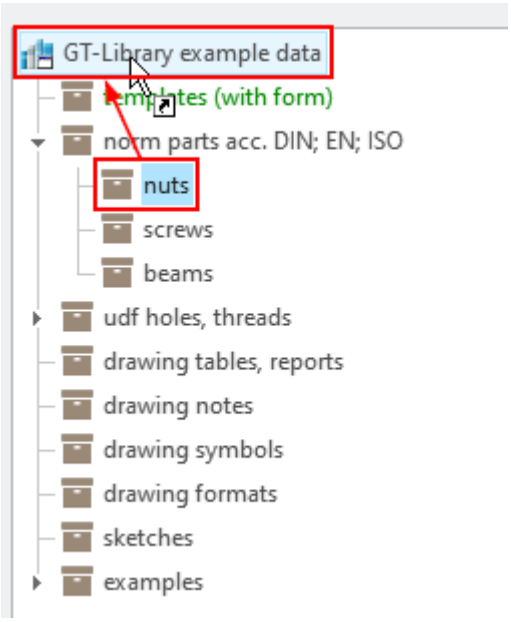
| Icon | Name | Description |
|---|---|---|
|  | Open collector[360] | Opens the collector for batch mode of library objects. |
|  | Clean up data[363] | Opens the dialog for cleaning up library databases, e.g. deleting images. |
|  | Restrict search function[363] | Opens a list that allows you to exclude parameters and dimensions from the search index. |
|  | Help | Opens the Help. |

## 12.3.4.3 Category tree

The category tree shows all categories of the current library.

**Please note**: The editor always opens the last library that was used first.

Use drag-and-drop to rearrange categories. To display them in the first level again, drag categories back to the root node.




*Drag a category and drop it on the root node to display it on first level again*


© 2026 INNEO Solutions GmbH

## Adding objects to the category tree

To add an object to the category tree, find the object using the search function in the *All objects* tab, then use drag-and-drop to place it in the required categories.

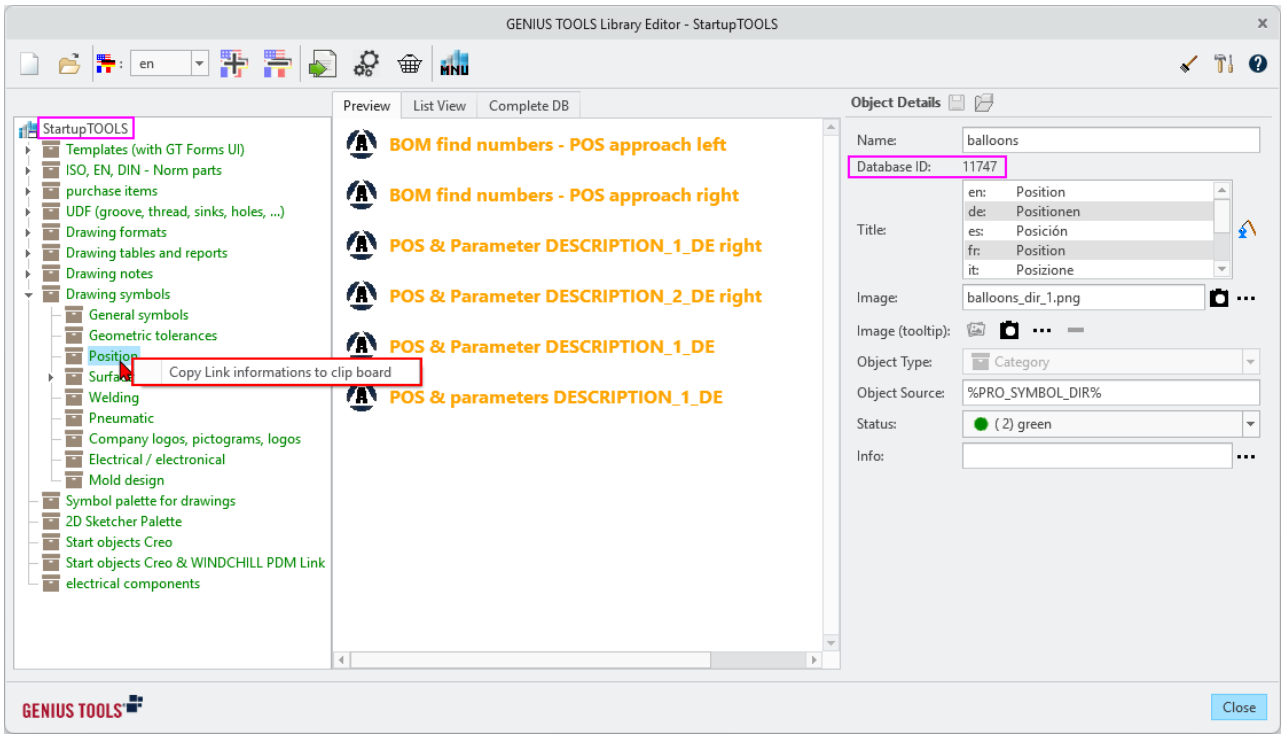
The *All objects* tab has a dedicated function for finding all objects that cannot be reached via the category tree .

## Linking library categories

Categories can be linked to other categories or to other libraries. Library objects of the linked category are displayed in the other library or category. Alternatively, categories can be linked several times in the same library.

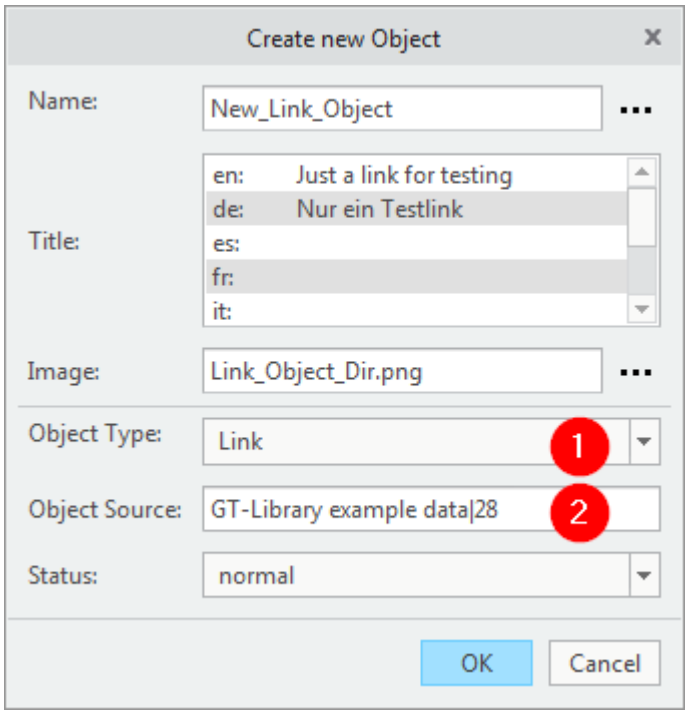
Proceed as follows to link a library category:

1. Open the library, containing the category you want to link.
2. Open the context menu in the category tree by right-clicking, then click *Copy link information to clipboard*.




3. Open the library in which you want to create the link.
4. Create a new object (1) of the object type *Link*.

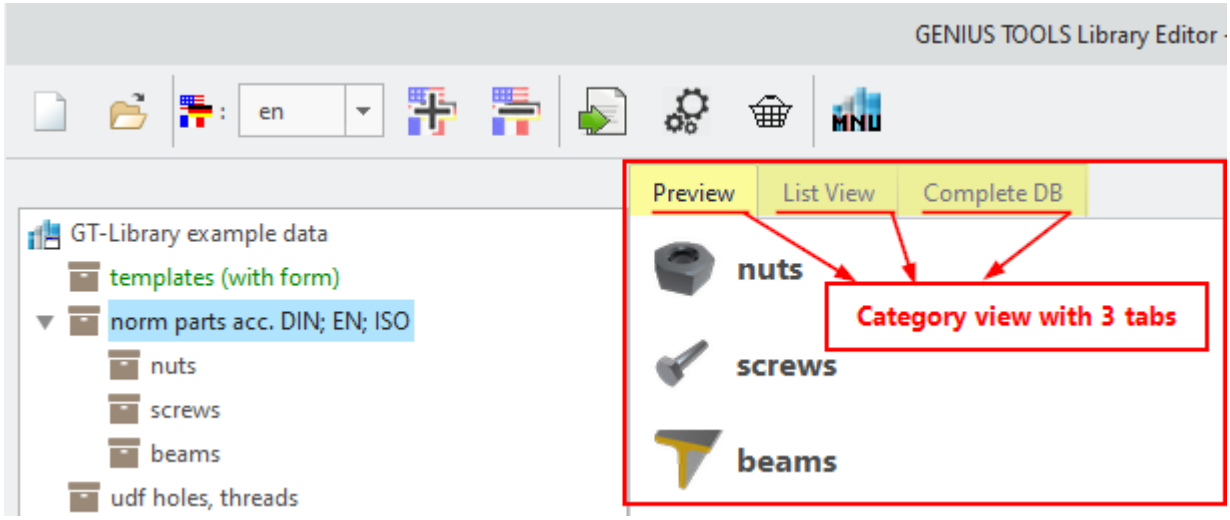

© 2026 INNEO Solutions GmbH




5. Add the link information to *Object Source* (2) and create the object.
6. Click on the link in the details window to open the database.

## 12.3.4.4 Category view

The Category view displays the library objects that are currently selected in the category tree. Click a library object to view additional information in the Details area.




Each of the three tabs shows different information to an object:

**Preview:** Library objects and categories are displayed with preview image and description text.

**List view:** Library objects are sorted by type and internal name. In this view, displayed columns can be changed with the context menu.


© 2026 INNEO Solutions GmbH

All library elements are displayed regardless of their status.

**All objects:** The tab allows a search over all library objects in the current library.

Advanced search for objects that cannot be found in the category tree.

## Sorting objects

Use drag-and-drop to rearrange the library objects in the *Preview* area. The order defined in the *Preview* area is used for displaying library objects in the library browser.

Use drag-and-drop to assign library objects to categories.

If you drag objects from the tabs *Preview* or *List view* from the current category to another one, the selected objects will be moved.

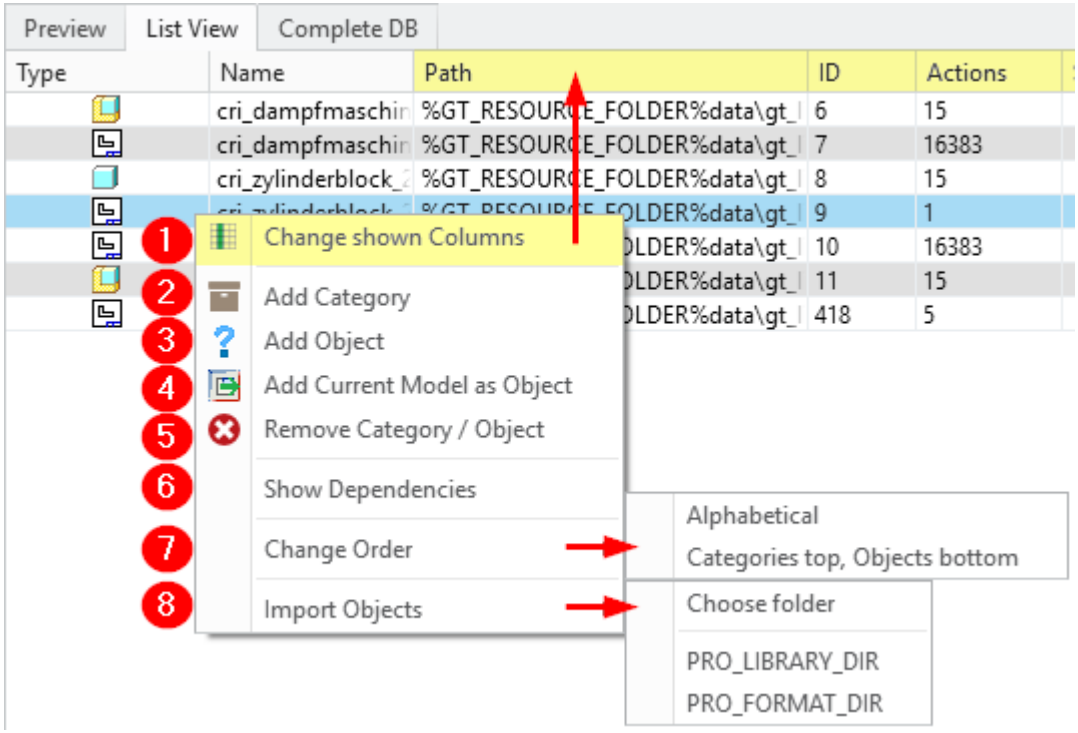
From the *All objects* tab, you can link objects to multiple categories.

You can also use drag-and-drop to place objects in the object collector dialog.

Use the tabs above the Category view to display different views of the library items.

## Further actions in the context menu

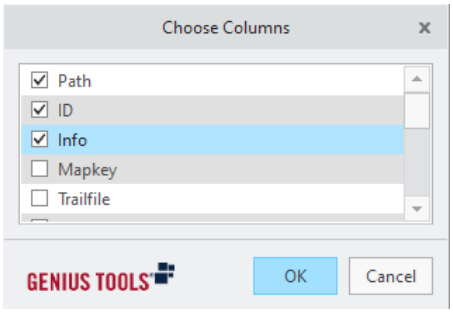
Use the context menu to perform further actions.




The following options are available from the context menu in category *List View*:


© 2026 INNEO Solutions GmbH

### 1. Change shown columns

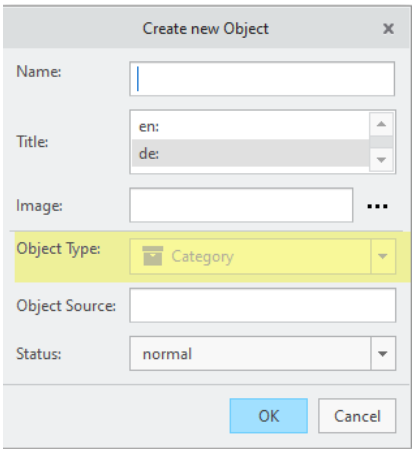



Functions in tabs List View and Complete DB

Select the columns that are to be displayed by checking the relevant boxes. The columns may differ in the list view and the all-objects views. Press and hold the Ctrl key to select multiple entries in the dialog box.

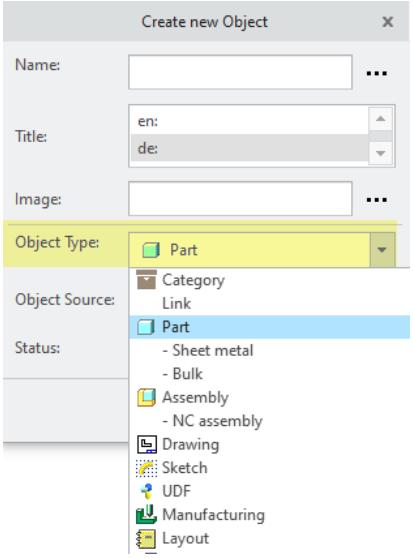
The selected columns will be taken over to the next session.

### 2. Add category




Adds a new (sub)category into the selected category.

### 3. Add object




Adds a new library object to the selected category.

### 4. Add current model as object


© 2026 INNEO Solutions GmbH

Inserts the currently open model into the selected category as a library object.

**5. Remove category/object**



Removes the selected object/category from the library (with subcategories and objects). For objects with family tables, the instances will be deleted as well.

**Tip:** As an alternative to deleting a category or an object, the database reference to an object or a category can be removed instead. The category or object is still in the database, but is not included in the tree structure.

**6. Show dependencies**

Displays parent and child objects of the selected library object.

**7. Change order**

Sorts the displayed elements. Select the options several times to combine the two possible views.

**8. Import objects**

Imports library objects into the selected category.

## 12.3.4.5 Object details

The *Object Details* section displays information on the selected element. The command bar contains the following buttons:

| Icon | Name | Description |
|---|---|---|
|  | Save object details | Saves modified object details into the database. |
|  | Open model in Creo | Opens the current library object in Creo. |


© 2026 INNEO Solutions GmbH

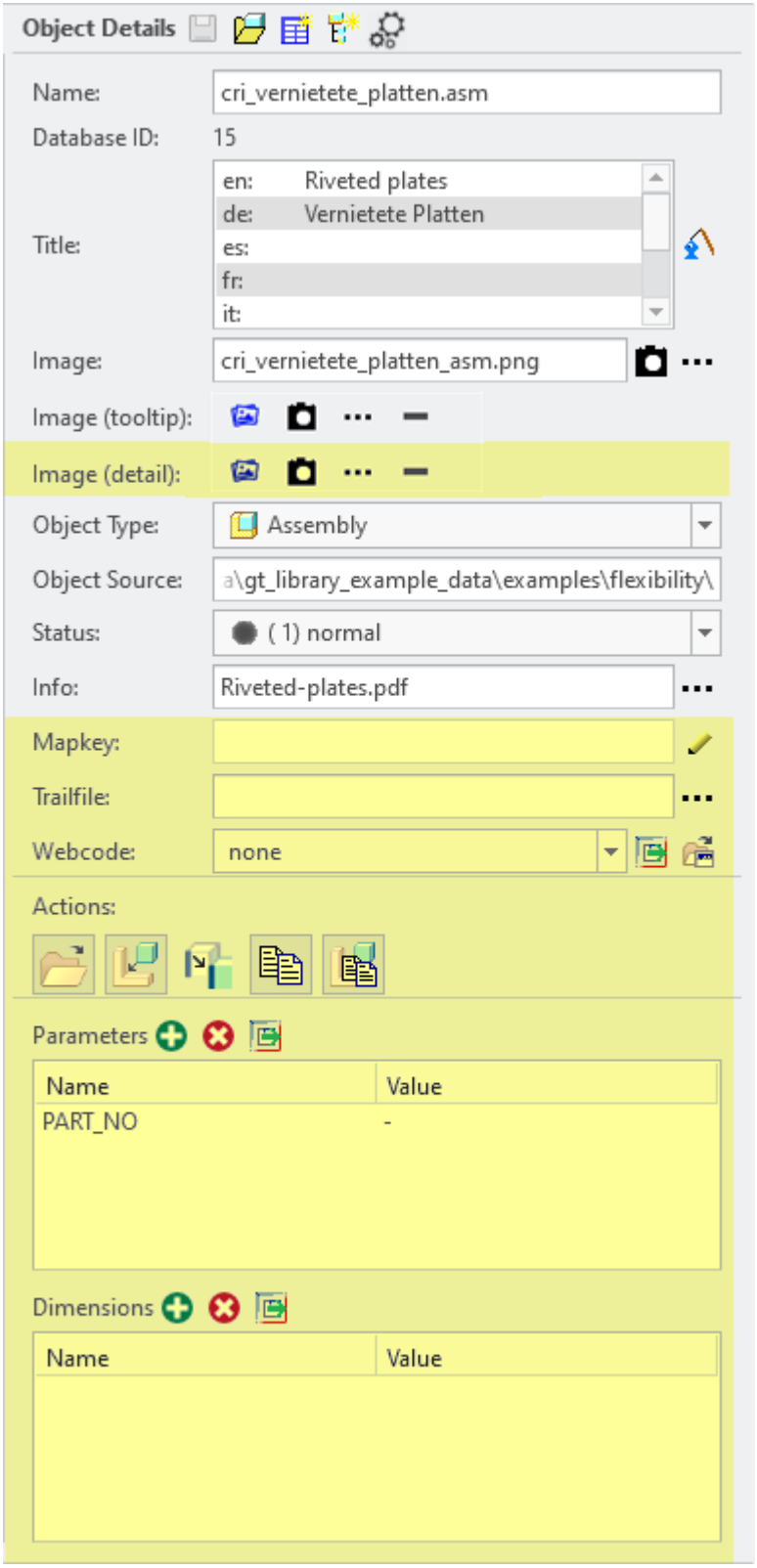
| Icon | Name | Description |
|---|---|---|
|  | Edit selection list[335] | Opens the editor to customize selection lists for the current library object. (Not available for categories.) |
|  | Edit model list for copy purpose[337] | Opens the editor to customize copy rules for the current library object. (Not available for categories.) |
|  | Run objects through batch mode[357] | Opens directly the dialog *Batch Processing* > *Settings* without the intermediate step of selecting the object. This is useful when adding individual models to the database. |

**Tip**: Remember to save any changes you make to a library object.

The information displayed differs for

1. Categories and
2. Library objects


© 2026 INNEO Solutions GmbH




*Detail area for categories and library objects (yellow)*

## 1. Details for both categories and library objects

The following object details are editable:


© 2026 INNEO Solutions GmbH

**Name**: Name of the selected object. After an import, the name corresponds to the file name for library objects; for a category, the name corresponds to the folder name.

**Database ID**: The internal ID of the object in the library database (only for information).

**Title**: The localized names of a library object. Standard texts can be used via the button *Select default text*  , see Description of the standard text selection dialog 678.

**Image**: Chose a preview image for a category or library object or create it, see Working with images 344.

**Image (tooltip)**: An image for display as a tooltip for a category or library object, see Working with images 344.

**Object type**: The current type of a library object, see Object overview 334.

Always check the type of library objects. The type defines how a library object is opened by Creo and what actions can be performed on it. For example, manufacturing assemblies (MFGs) have the same file extension as normal assemblies (.asm), but must be treated differently by Creo.

**Please note**: The object type “Category” cannot be changed after it has been created.

**Object source**: Path of the library object. For categories, the folder name must be omitted.

Example for a category in a library that corresponds to a path:

– Full path of the category: `D:\Parts\StandardParts`

– Object source: `D:\Parts`

– Name: `StandardParts`

Use categories to open paths in the Library browser in Windows Explorer.

**Warning**: Check the validity of the path for categories that are not solely used for logical structuring. The path of a category is composed of the object source and the name.

**Status:** Select one of the pre-defined status colors. (Meaning and choice of colors can be edited, see Defining object status 339.)

**Info**: Document assigned to a library object or category with additional information. For a stored document, the info button is displayed in the Detail window of a Library object. For categories, documents can be accessed via the context menu of the library browser.

**Please note**: Information documents must be available on the client computers so that they can be opened. Viewing applications are controlled by the client computer.

## 2. Additional details for library objects

**Image (detail)**: An image for the display in the detail window of a library object, see Working with images 344.


© 2026 INNEO Solutions GmbH

**Mapkey**: A mapkey for a library object. The mapkey is executed after clicking the mapkey button in the details window.

**Trailfile**: A trailfile for a library object. The Trailfile is executed after clicking the Trailfile button in the detail window.

**Webcode**: Webcode of a Form to be linked with the library object. For information on imports read chapter XML-interface for Form values 364.

**Actions**: Specifies different actions that can be executed on the object in the *Library*. Details window. Different actions can be executed on the individual object types see Actions on Library objects 348.

**Parameters** and **Dimensions**: These two tables are used to add parameter-value pairs or dimension-value pairs to the library object. The parameters or dimensions given here are displayed in the *Library Details* window as additional information and can be searched for.

Use the ⊕ and ⊗ buttons to add or remove value pairs. Value pairs can also be adopted from the current model using the *Read* button.

**Please note**: The parameter tables and dimension tables can be filled automatically for generic models. Use the function in *Read family table cell information* in the batch processing.

Parameters can contain multiple character-separated values. Library objects that contain multiple character-separated values in one parameter will be found in a search for either of the parameter values. The separator character is defined in the configuration option `gtl_parameter_multiple_value_separator`.

## Object overview

The following object types are available:

| Icon | Name | Icon | Name |
|---|---|---|---|
| | Category | | Manufacturing |
| | Link | | Layout |
| | Part | | Diagram |
| | Sheet metal | | Format |
| | Bulk | | Table |
| | Assembly | | Symbol |
| | NC Assembly | | Note |
| | Drawing | | Selection |


© 2026 INNEO Solutions GmbH

| Icon | Name | Icon | Name |
|---|---|---|---|
| | Sketch | | Undefined object (ClvObject) |
| | UDF | | |

Special object types are:

– **Category:** Library categories are virtual organizational elements, see Glossary[295].

– **Link:** You can jump to an object inside and outside the library, see Special objects – Creating a connection to the database[342].

– **Selection:** Selection table for selecting objects. Different object types are possible.

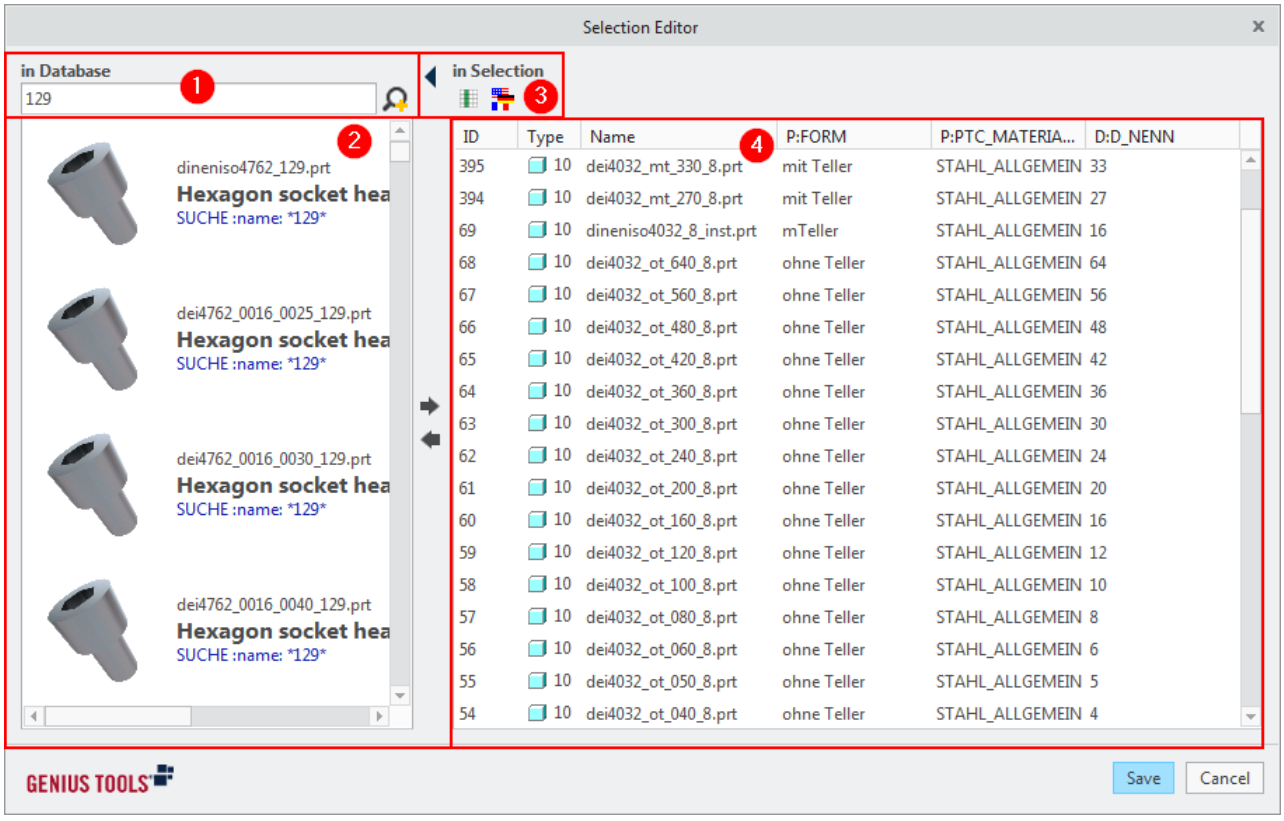
– **Undefined object (ClvObject):** It has no actions. Help documents or mapkeys can be started, see Special objects - Creating a link to a document[342].

# Selection list editor

Selection lists for library objects are created in a special editor. Library objects are compiled in selection lists. They are available in the detail window in the Selection tab as variants of a library object.

In *Library Editor*, select a library object and click the *Edit selection list* button under Object Details.

The editor is divided into the following areas.





© 2026 INNEO Solutions GmbH

1. Search
2. Library browser
3. Options
4. Selection list

**Tip**: Selection lists can be filled automatically for generic parts with family tables. Use batch mode.

Use the search (1) to search the current database. The search is identical to the search of GENIUS TOOLS Library.

On the left side of the dialog is the Library browser (2). It displays library objects of the current database. Next to the Library Browser are the options (3). The library objects of the selection list (4) are displayed on the right.

Drag and drop library objects into the selection list or use the arrow buttons to add and remove library objects. Library objects in categories are added *recursively* (all contained library objects) or *explicitly* (only library objects that are directly in a category - without subcategories).

Click into table cells to edit empty or incorrect parameters and dimensions manually.




*Click into the table cells to edit their content manually*

Parameters and dimensions changed in selection lists are automatically changed in the corresponding library objects.

Modified values are not transferred back to models.

Use the tab key to navigate within selection lists and localization dialogs.

## Options

Use the button *Add columns with the object details* to show or hide columns.

Use the button *Change translations for Columns* to edit translations for columns. You can use the options for default texts or for automated localization.

If library objects in the selection list are generics, use the third button: *Merge selection tables*. Nested family tables are read out by batch processing and processed for the selection list. In addition, parameters can be inherited from generics and the generics can be removed from the selection. This functionality is useful for consolidating nested family tables.

You can also merge selection lists manually. To do so, use the context menu to search the selection list for


© 2026 INNEO Solutions GmbH

– all instances
(instance information will only be written when importing or when using batch mode)

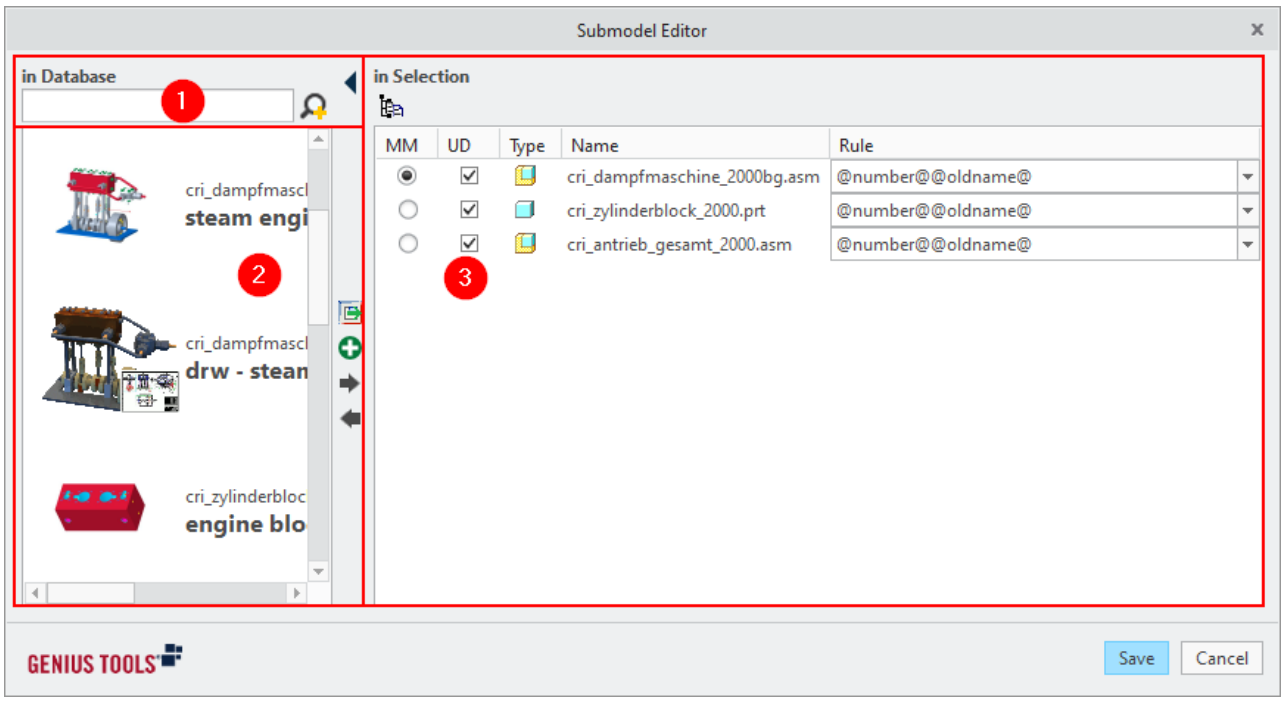
– all selection objects
(that is, all objects that are part of the selection list for the selected object)

If the selection table meets your requirements, click *Save*.

## List of models to be copied

This list contains copying instructions for models to be copied together. Models linked in this way will be copied into the work directory together with the selected object if you execute a *Copy* action.

There is a separate editor dialog for models to be copied together. To open this, select an library object in *Library Editor* and click *List of models to be copied together* .




The editor contains:

1. Search field
2. Library Browser
3. List of library objects in the selection: objects to be copied

Use the Search field (1) to search the current database. The search function is identical to the search function used in *Library*.

The left pane of the dialog shows the Library Browser (2), listing the library objects in the current database. The right pane shows the library objects in the current selection (3).

Drag-and-drop library objects into the current selection, or use the arrow buttons to move library objects to or from the current selection.


© 2026 INNEO Solutions GmbH

The following buttons are available.

| Icon | Name | Description |
|---|---|---|
|  | Copy submodel definition into selection | Use this dialog to define replacement rules that will be inherited by all instances of the selected model. |
|  | Add dependent file that is not in library | Add dependent models for the current Creo model to the copying list<br>**Please note**: The model has to be the currently active Creo model! |
|  | Add file that is not in library | Add any model to the copying list |
|  | Add to current selection / remove from selection | Use the arrow buttons to add or remove library objects |

## Creating copy definitions in the selection

1. In the list of selected objects, define the main model (MM) for the copy definition. The main model determines which form will be used and which model will be opened after copying.
2. Next, define whether instances that you have added to the selection should be unlinked from their generic model on copying (UD = undock).
3. Last, define the copying rule under *Rule*. Add an individual copying rule for each model, or use the dropdown list to select already configured copying rules.

## Variables in copying rules

You can use all GENIUS TOOLS for Creo variables[913]. Additionally, there are two variables specifically for copying rules:

`@oldname@`: returns the old file name of a library object.

`@copyno:<model_name>@`: applies the copying rule of the model given in the variable to the current model.

`@number@`: Returns a number from the *Name Generator* with filter from the configuration option `gtl_gtng_standard_db_filter_for_file_copy` (fallback: `gtng_db_name_filter` > empty filter), see Integrating name configurations in Quick Access[454].

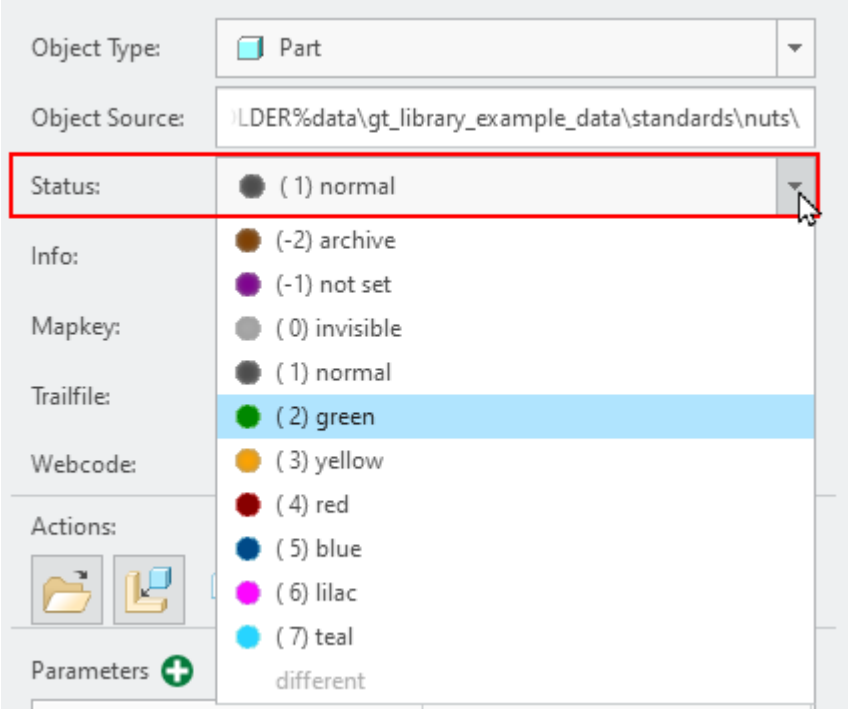
`@number:filter@`: Returns a number from the *Name Generator* with filter *filter*.

**Please note**: All changes are saved to the object when closing the editor.


© 2026 INNEO Solutions GmbH

# Defining object status

You can assign an object status for library objects and for categories, in the Object details[333] section in *Library Editor*.



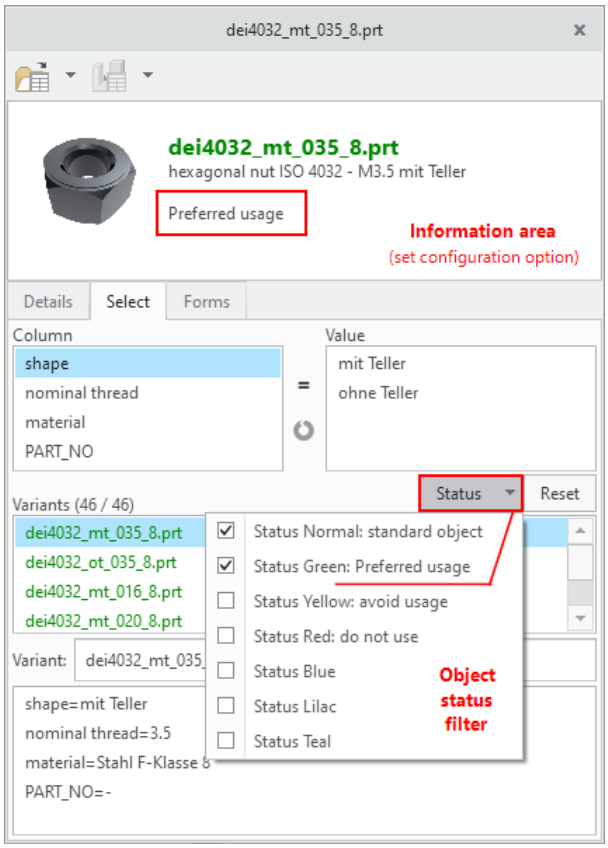

*Selecting status in the section Object details*

Objects can be filtered by this status

– with the object status filter in the library browser[306] and

– with the object status filter in the selection tab of the details window.[314]


© 2026 INNEO Solutions GmbH




### Available status

The status values *Normal*, *Green*, *Yellow*, *Red*, *Blue*, *Lilac* and *Teal* can be assigned freely.

In the start settings the following descriptions are assigned to these colors.

| Status | Description (EN) | Description (DE) |
|---|---|---|
| Normal | Standard object | Standardobjekt |
| Green | Preferred usage | Vorzugsweise verwenden |
| Yellow | Avoid usage | Verwendung vermeiden |
| Red | Do not use | Nicht mehr verwenden |

### Modify status description

You can change the text for the description of these status using the configuration options `gtl_<StatusColor>_text`, e.g. `gtl_filter_yellow_text = obsolete`. These changes can only be displayed in one language.

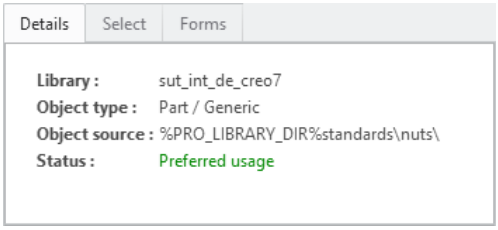
### Configure display

The status of an object is visible in the detail window[312]

– in the upper information area (screenshot above): this display is activated with the configuration option `gtl_detail_window_show_status_in_head_area`.


© 2026 INNEO Solutions GmbH

– in the details tab: always




The respective colors can be changed with the configuration options `gtl_*_color`.

**Tip**: Document the meaning of the freely definable statuses in the work instructions. Assign descriptions using the configuration options `gtl_<StatusName>_text` of *Library*.

The following three status values are administrative and cannot be searched by the object filters:

– Archive: archived library objects are not displayed in the Library browser.

– Not set: library objects without status are displayed in the Library browser and can be searched for.

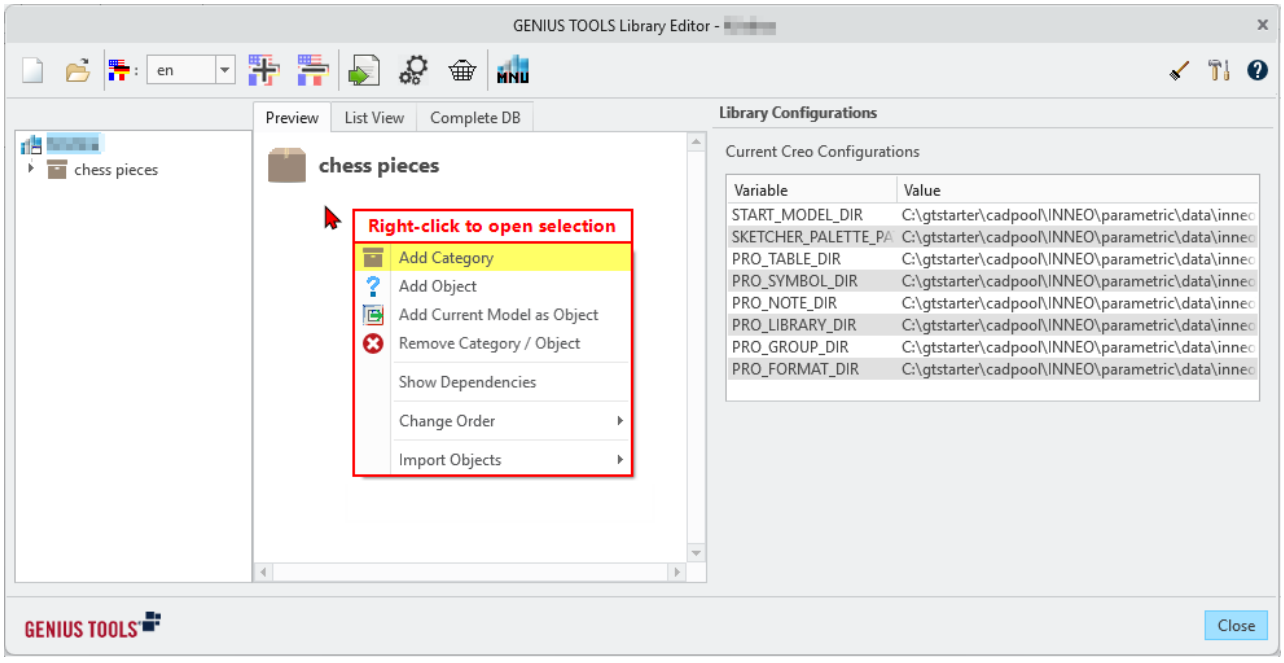
– Invisible: invisible library objects are not displayed in the Library browser and cannot be searched for. Invisible library objects and categories are not displayed in the *Library Editor* preview either, but can be found via the *Lists* view and *All objects* tabs.

# Creating special objects

You can add categories and objects in the preview area. With undefined objects (ClvObject), you can add a document or a Mapkey. With a link, you can create a link to a database within or outside the database.

### Adding a category and object

1. Create a new category, see Creating a new category 368.



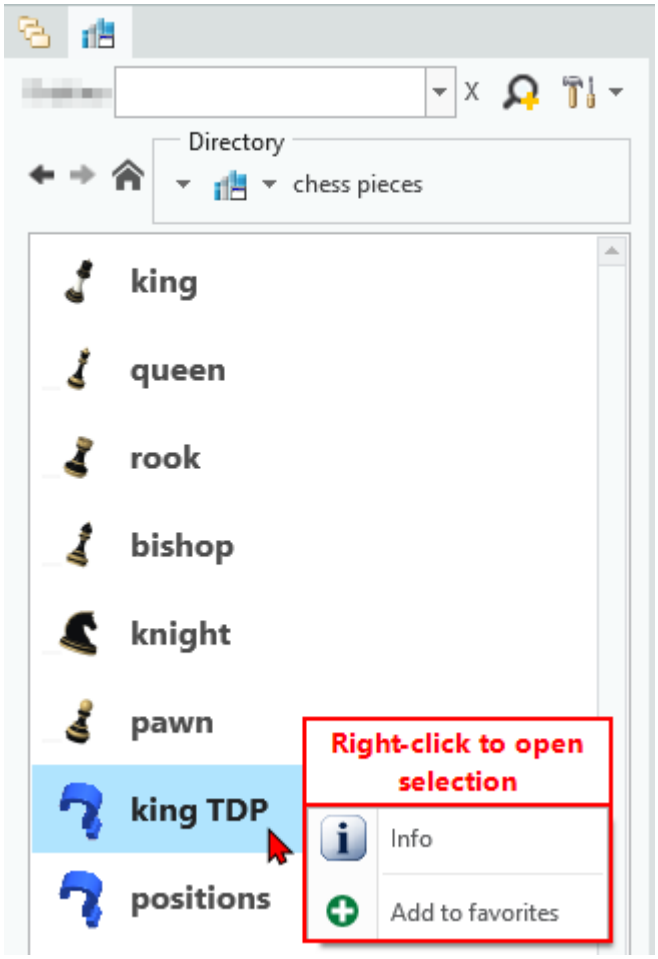


© 2026 INNEO Solutions GmbH

2. Create a new library object, see Creating a new library object 369.

**Creating a link to the document with an undefined object (ClvObject)**
With a *ClvObject*, you can access a document with additional information in the detail window.

1. Follow steps 1 to 3 under *Add category and object*.
2. Fill in the fields in the *Create New Object* window, select *ClvObject* as the *Object Type*, and confirm with *OK*.
3. Click on the *ClvObject* you created in the *Preview* tab.
4. Under *Object Details*, click the three dots next to *Info* and select the document or click on the yellow pencil to add a mapkey.
5. Click *Save Object Details* at the top.
6. Right-click on the link in the details window to click on the *Info* button in the selection menu.



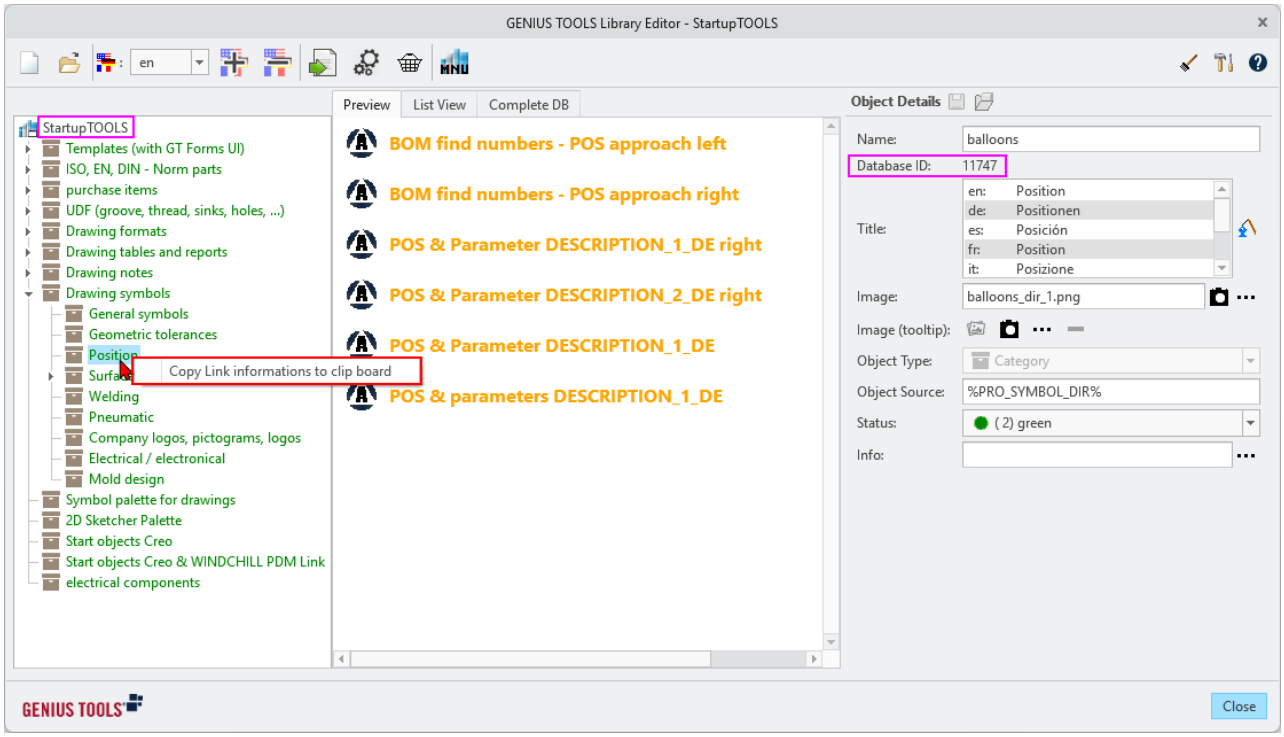

The document will open or the Mapkey is executed.

**Creating a connection to the database with a link**

You can use a link to refer to another database within the database or to another database.

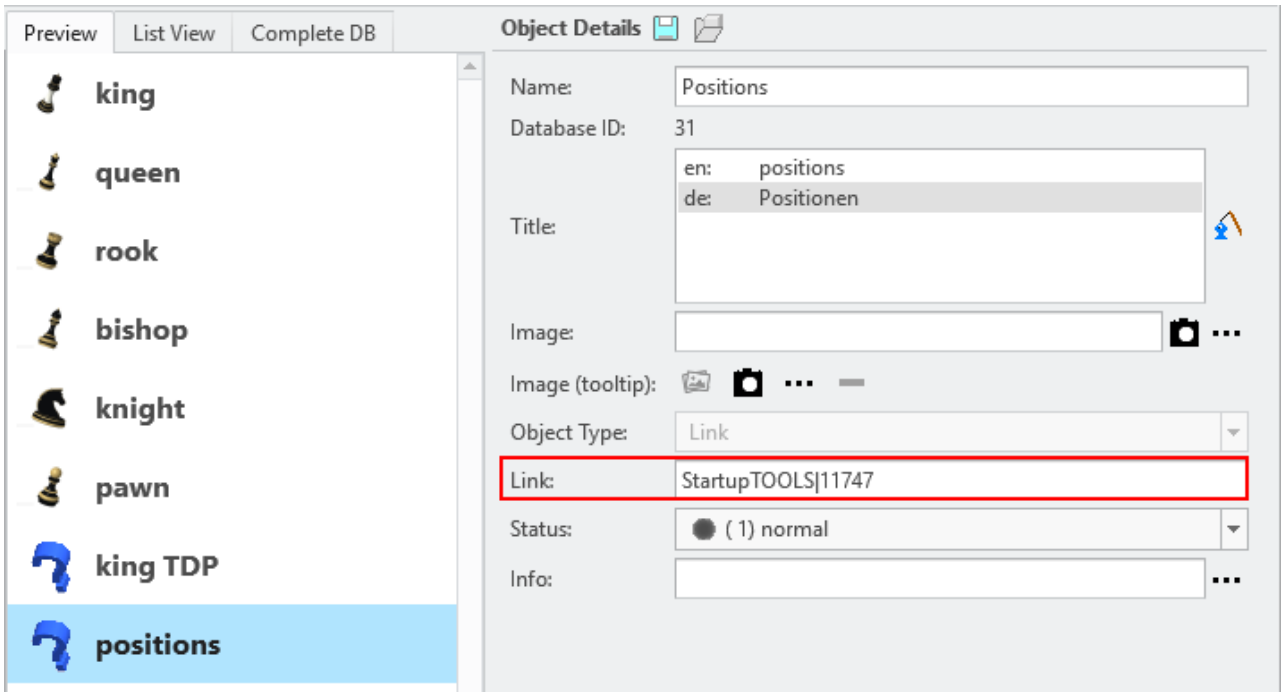

© 2026 INNEO Solutions GmbH

1. Proceed as described in steps 1 to 3 under *Add category and object*.
2. Fill in the fields in the *Create new Object* window, select *Link* as the *Object Type*, and confirm with *OK*.
3. Search for the database you want to link to in the category tree.
4. Right-click on the database and then click *Copy link information to clip board*.




5. Return to the link you created in the *Preview* tab and click on it.
6. In the *Object Details*, paste the link information into the *Link* field, e.g., *StartupTOOLS|11747*.
7. Click *Save Object Details* .


© 2026 INNEO Solutions GmbH




8. Click on the link in the details window.

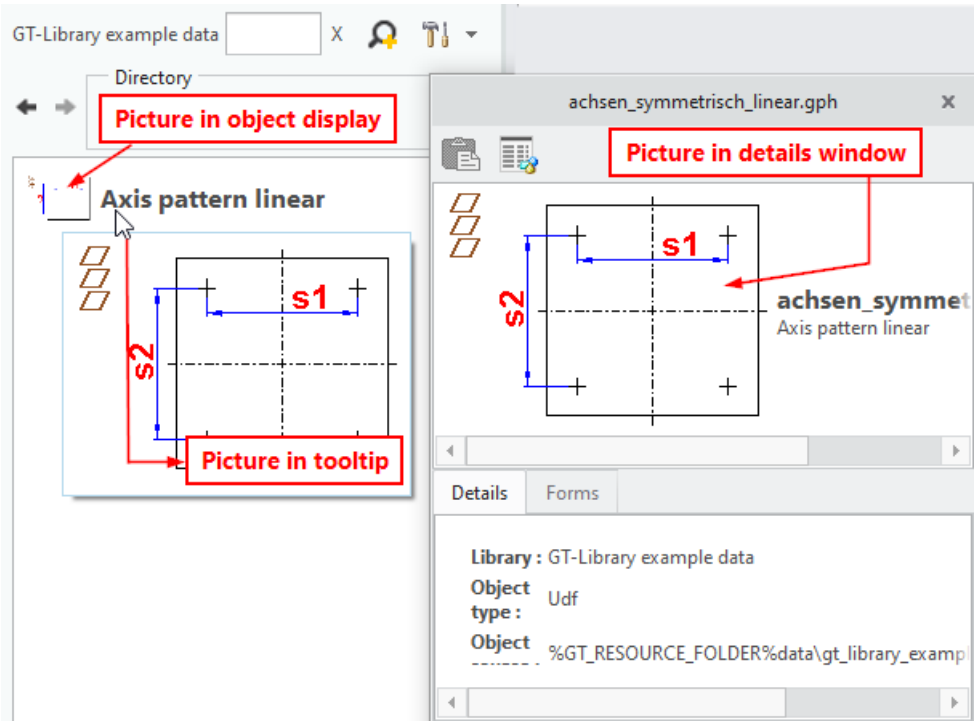
The database will open.

## Working with images

Images can be embedded in *Library*

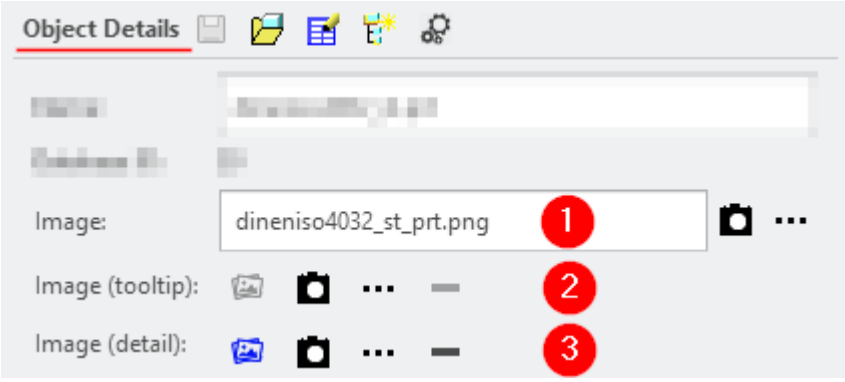
1. in the object display area ("Preview image")
2. in tooltips
3. in the details window[312]


© 2026 INNEO Solutions GmbH




*Three possibilities of using images*

You can add images to the object details in the *Library Editor* or create them directly via snapshot.




Use the following symbols to work with.

| Symbol | Name | Description |
|---|---|---|
| | Open existing image | The symbol turns blue when an image has been uploaded. |
| | Create image | Creates a screenshot of the currently opened model. Replaces an already existing image. |


© 2026 INNEO Solutions GmbH

| Symbol | Name | Description |
|---|---|---|
| ... | Add / replace image | The selected image is automatically copied / converted to the respective folders (*img*, *img_detail*, *img_tooltip*) under the same name and type. |
| ➖ | Delete current image | Removes the assignment of the uploaded image to the object. |

# 1. Images in object display (Preview image)

**Name of image file:** Displays the name of the image from the directory *GT_RESOURCE_FOLDER>/library/<LIBRARY>/img_w40*. Use a language variable, e.g. *gt_lang*, in the path specification for displaying images language-dependently, e.g. *start_bolt_§gt_lang§.png* (See section below.)
**Create image:** Preview images are created by default in the size of 40*30 pixels.

**Please note:** Preview images can be created automatically in batch mode[357]  .

**Add / replace image:** Select an image from a directory.
**Size of image:** The display size can be changed, see Changing the sizes of images[347].

# 2. Images in tooltip

**Open existing image:** Opens the image from the image directory *img_tooltip*. The image must have the same name as the object, using the pattern `<OBJECT_NAME>_<OBJECT_EXTENSION>.png` (e.g., *dei4032_mt_020_8.prt* > *dei4032_mt_020_8_prt.png*). The default directory is *<GT_RESOURCE_FOLDER>/library/<LIBRARY>/img_tooltip* and can be changed with the configuration option `gtl_tooltip_image_folder`. .
**Create image:** Tooltip images are created by default in the size of 200*200 pixels. You can change this size with the configuration option `gtl_img_create_tooltip_size`.
**Add / replace image:** The selected image will be automatically copied / converted with the same file name and type to the *img_tooltip* folder.
**Size of image:** Tooltip images are displayed in original size.

# 3. Images in details window (Detail image)

The details window[312] can either display the image from the object display (Preview) or an image that is especially for this window. If there is an extra image for the details window, you have the possibility to hide the display of the object title with the configuration option `gtl_detail_window_detail_image_show_title`.


© 2026 INNEO Solutions GmbH

**Open existing image:** Opens the image from the image directory *img_detail*. The image must have the same name as the object, using the pattern `<OBJECT_NAME>_<OBJECT_EXTENSION>.png` (e.g., *dei4032_mt_020_8.prt* > *dei4032_mt_020_8_prt.png*). The default directory *<GT_RESOURCE_FOLDER>/library/<LIBRARY>/img_detail* can be changed with the configuration option `gtl_detail_image_folder`.

**Create image:** Details images are created by default in the size of 200*200 pixels. You can change this size with the configuration option `gtl_cimg_reate_detail_size`.

**Add / replace image:** The selected image will be automatically copied / converted with the same file name and type to the *img_detail* folder.

**Size of image:** The default size for displaying detail images is 100*75 pixels. This size can be modified in two ways:

1. The size of all detail images is similar: Use the configuration options `gtl_detail_window_detail_image_height` and `gtl_detail_window_detail_image_width`. to define the height and width of all detail images.
2. The size of the individual image file should be used: Set both the configuration options `gtl_detail_window_detail_image_height` and `gtl_detail_window_detail_image_width` to 0.

Images are retrieved in the following order:

– detail image of the instance

– detail image of the generic

– image of the instance

– image of the generic

**Please note**: Alterations of the detail image (after the initial preview) will be displayed after restarting Creo Parametric.

## Changing the sizes of images

Images are by default available in small size (for object display) and in large size (for detail window). You may set the object display from small to large images and you may also modify the standard size of all images.

Images are stored in format 4:3. The pixel values in the configuration options refer to width.


© 2026 INNEO Solutions GmbH

| Type of image | Standard for display size | Configuration options |
|---|---|---|
| Object display images | small (40*30 Pixel) | `gtl_img_switch_size`: defines the sizes of images in the object display. (Default width: 40 pixels)<br>`gtl_img_size`: Changes size of object display images from small=40 to large=100 |
| Details window images | large (100*75 Pixel) | `gtl_img_detail_size`: defines the size of detail images (Default width: 100 pixels)<br>For larger images: Create separate *Detail* directory |
| Tooltip images | Original size | Original size is used up to 800*800 pixels. |

**Warning:** The configuration options modify the size of all images of an image type and change the name of the storage folder. Therefore, take care not to change sizes of several image types simultaneously, i.e. do not make changes to both configuration options `gtl_img_detail_size` and `gtl_img_switch_size`.

## Display images language dependent

You can display images depending on a language. This can be useful when images contain textual information.

– For detail and tooltip images: Create a folder per language and link to that folder.

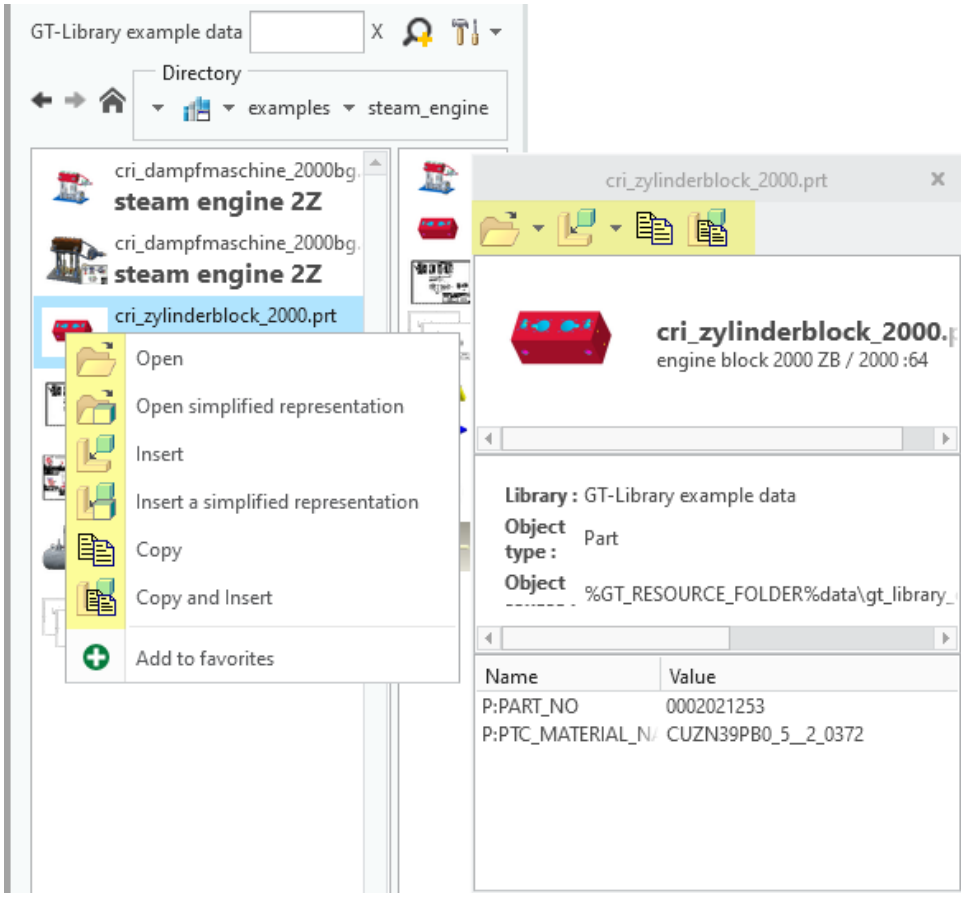
– For preview images: Add a language abbreviation to the image file name, which is output by the variable *gt_lang*, and enter the variable in the notation *§gt_lang§* in the file name (1).

## 12.3.4.6 Actions on library objects

Library object actions are specified in the *Library Editor*. They are available in a library objects Details window or in the context menu.


© 2026 INNEO Solutions GmbH

Example: The part *engine block 2000* can execute the actions *Open*, *Insert*, *Copy* and *Copy and Insert*.

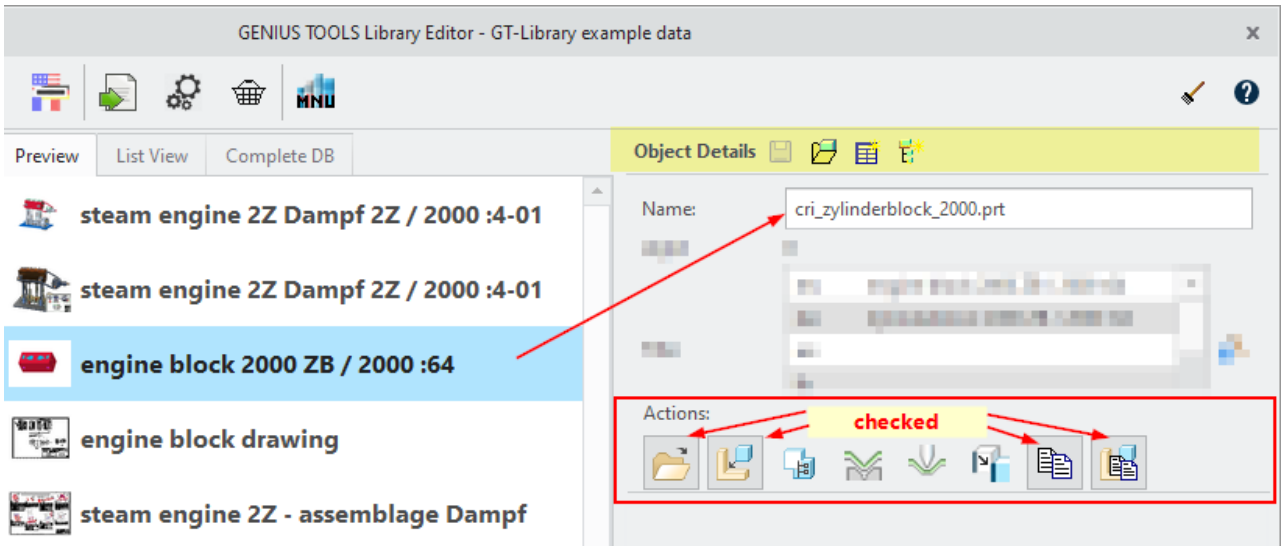



*View in the context menu (left) and details window (right)*

These actions are activated in the *Library Editor* by clicking on the corresponding icons:

– either for multiple objects simultaneously by using the object collector or

– for individual objects in the object details > Actions, see screenshot below.





© 2026 INNEO Solutions GmbH

**Please note**: Different types of library objects support different actions.

**Batch file for models that cannot be retrieved**
You can use the configuration option `gtl_retrieve_run_batch` to specify the path to a batch file. This batch file will be called if a model cannot be retrieved by *Library*. After the batch file has run, *Library* again tries to retrieve the model data. In the batch file, the variables *@name@* for the model name and *@path@* for the model path can be used.

**Open** 
Allows to open models or to open models as simplified representations. Assemblies can also be opened with simplified graphics.

Open with defined Representation open the model with the representation that is defined by the configuration option *gtl_action_open_defined_simp_rep*.

You can use the following terms as option values:

Explicitly specified : e.g. LAYOUT or FLYER

DEFAULT ENVELOPE REP

AUTOMATIC REP

MASTER REP

DEFAULT

Notes, symbols and selections cannot be opened.

**Insert** 
Allows to insert library objects into assemblies, parts, sketches, drawings and reports.

Additionally activates the action **Insert as simplified representation**.

| Library object type | Insertable in |
|---|---|
| PRT | ASM |
| ASM | ASM |
| GPH | ASM, PRT |
| SEC | SEC |
| FRM | DRW |
| TBL | DRW, FRM, PRT |


© 2026 INNEO Solutions GmbH

| Library object type | Insertable in |
|---|---|
| SYM | DRW, FRM, PRT |
| TXT | DRW, PRT |

**Copy models** 
Copies a library object into the current working directory with a new name using *Name Generator*. The copy is opened in Creo.

The action is available for parts and assemblies. This action copies models from the lists of models for copy purpose as well.

If a library object contains a Form, the action *Copy the selected file* will additionally be activated for the library object. This copies the model and values in the form are applied to the model. If you have defined copying rules for the model, these will be applied.

The workflow in combination with *Name Generator* (`gtl_gtng_*`) and *Forms* (`gtl_gtf_*`) can be defined using configuration options. For more information, please refer to configuration[830].

**Copy drawings** 
Copies a drawing with a new name (using *Name Generator*) and without copying the model in the current working directory.

**Insert file with note reference** 
Inserts a TXT file with hint arrow to a drawing (DRW), 3D model or a report (Rep).

**Merge/Inherit** 
Inserts a library object (parts) as an inheritance into another part.

**Die** 
Inserts a library object (parts) as a die for a sheet metal part.

**Punch** 
Inserts a library object (parts) as a punch for a sheet metal part.

**Copy geometry** 
Inserts a library object as a copy geometry for a sheet metal part. If the library object has multiple copy geometries defined, the user is offered a selection.

**Execute mapkey** 
Executes a mapkey deposited in the library object.


© 2026 INNEO Solutions GmbH

### Trail file

Executes a trail file deposited in the library object.

## List of all actions by object type

Add an object type and assign actions. Actions cannot be added to the object types *Category*, *Link*, and *Selection*.

| Object Type | Open the file | Insert the file | Insert the file | Merge/Inheritance | Die | Punch | Copy Geometry | Copy the selected file | Copy the selected file and insert the copy | Open variant | Insert the file with note reference |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Category | | | | | | | | | | | |
| Link | | | | | | | | | | | |
| Part | X | X | | X | X | X | X | X | X | | |
| Sheetmetal | X | X | | X | X | X | X | X | X | | |
| Bulk | X | X | | X | X | X | X | X | X | | |
| Assembly | X | X | | | | | X | X | X | | |
| NC-Assembly | X | X | | | | | X | X | X | | |
| Drawing | X | | | | | | | X | | | |
| Sketch | X | | X | | | | | | | | |
| UDF | | | X | | | | | | | X | |
| Manufactoring | X | | | | | | | X | | | |
| Layout | X | | | | | | | | | | |
| Diagram | X | | | | | | | | | | |
| Format | X | | X | | | | | | | | |
| Table | | | X | | | | | | | | |
| Symbol | | | X | | | | | | | | |
| Note | | | X | | | | | | | | X |
| Selection | | | | | | | | | | | |
| Clv Object | | | | | | | | | | | |

### Part, Sheetmetal and Bulk

| Icon | Action | Description |
|---|---|---|
| | Open the file | Opens the file |
| | Insert the file | Inserts a part/sheetmetal/bulk into an assembly. Also activates the action *Insert as simplified representation*. |
| | Merge/Inheritance | Inserts a part/sheetmetal/bulk as inheritance into a part. |
| | Die | Inserts a part/sheetmetal/bulk as a die for a sheet metal part. |
| | Punch | Inserts a part/sheetmetal/bulk as a punching die for a sheet metal part. |
| | Copy geometry | Inserts a part/sheetmetal/bulk as copy geometry for a sheet metal part. If multiple copy geometries are defined on the part/sheetmetal/bulk, the user is offered a selection. |


© 2026 INNEO Solutions GmbH

### Part, Sheetmetal and Bulk

| Icon | Action | Description |
|---|---|---|
| | Copy the selected file | Call up the Name Generator and copy the model to the current working directory under a new name.<br>Models from lists of models to be copied are also copied. |
| | Copy the selected file and insert the copy | Inserts a copy of the file |

### Assembly and NC Assembly

| Icon | Action | Description |
|---|---|---|
| | Open the file | Opens the file |
| | Insert the file | Inserts an assembly into an assembly. Also activates the action *Insert as simplified representation*. |
| | Copy geometry | Inserts an assembly as copy geometry for a sheetmetal part. If multiple copy geometries are defined for the assembly, the user is offered a selection. |
| | Copy the selected file | Call up the *Name Generator* and copy the model to the current working directory under a new name.<br>Models from lists of models to be copied are also copied. |
| | Copy the selected file and insert the copy | Inserts a copy of the file |

### Drawing

| Icon | Action | Description |
|---|---|---|
| | Open the file | Opens the file |


© 2026 INNEO Solutions GmbH

### Drawing

| Icon | Action | Description |
|---|---|---|
|  | Copy the selected file | Call up the *Name Generator* and copy the model to the current working directory under a new name.<br>Models from lists of models to be copied are also copied. |

### Sketch

| Icon | Action | Description |
|---|---|---|
|  | Open the file | Opens the file |
|  | Insert the file | Inserts a sketch into an assembly. Also activates the action *Insert as simplified representation*. |

### UDF

| Icon | Action | Description |
|---|---|---|
|  | Insert the file | Inserts a UDF into an assembly. Also activates the action *Insert as simplified representation*. |
|  | Open variant | If a selection table is available, you can select a variant. |

### Manufacturing

| Icon | Action | Description |
|---|---|---|
|  | Open the file | Opens the file |
|  | Copy the selected file | Call up the *Name Generator* and copy the model to the current working directory under a new name.<br>Models from lists of models to be copied are also copied. |


© 2026 INNEO Solutions GmbH

### Layout and Diagram

| Icon | Action | Description |
|---|---|---|
|  | Open the file | Opens the file |

### Format

| Icon | Action | Description |
|---|---|---|
|  | Open the file | Opens the file |
|  | Insert the file | Inserts a drawing frame into an assembly. Also activates the action *Insert as simplified representation*. |

### Table and Symbol

| Icon | Action | Description |
|---|---|---|
|  | Insert the file | Inserts a table or symbol into an assembly. Also activates the action *Insert as simplified representation*. |

### Selection

| Icon | Action | Description |
|---|---|---|
|  | Insert the file | Inserts a drawing, 3D model, or report into an assembly. Also activates the action *Insert as simplified representation*. |
|  | Insert the file with the note reference | Inserts a TXT file with an arrow pointer into a drawing, 3D model, or report. |

### Actions when using a form

| Icon | Action | Description |
|---|---|---|
|  | Copy the selected file | Call up the *Name Generator* and copy the model to the current working directory under a new name.<br>Models from lists of models to be copied are also copied. |


© 2026 INNEO Solutions GmbH

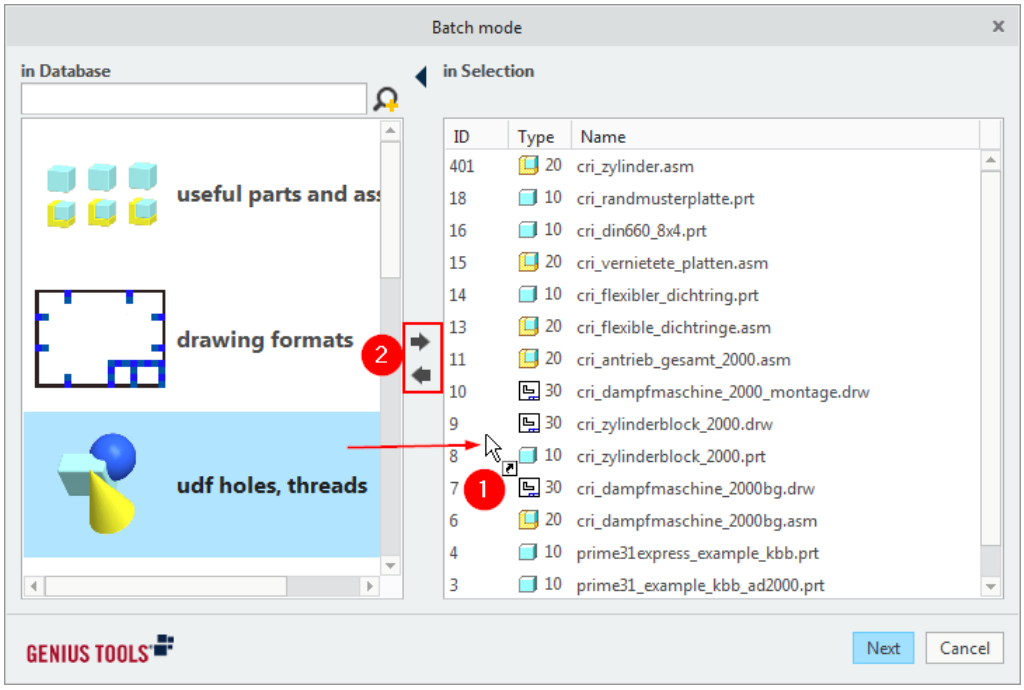
| Actions when using a form[370] | | |
|---|---|---|
| **Icon** | **Action** | **Description** |
| | Copy the selected file and insert the copy | Inserts a copy of the file |
| | Copy the selected variant and insert the copy as fixture | Inserts a copy of the selected variant as fixture |

## 12.3.4.7 Wizard: Batch mode

There is a batch mode wizard guiding you through the configuration of the batch mode. The wizard consists of two dialogs: Object composer and Batch mode configuration.

In the *Library Editor*, click on the *Configure and execute batch revision* button to open the configuration dialog of a batch revision.

# 1. Batch mode dialog: Object composer




*First dialog box of Batch mode: Object selection*

In the first dialog, you can specify the library objects for batch revision by drag and drop (1) or by using the two arrow buttons (2).


© 2026 INNEO Solutions GmbH

The dialog is divided into two parts. The library objects already selected (in Selection) are displayed on the left side. The browser displaying the library objects of the current database is located on the right side.

You can either drag and drop library objects into the selection list, or use the arrow buttons to add library objects to the selection list or to remove them.

When adding library categories to the selection list, select whether the library objects included should be added recursively (all library objects underneath the category and underneath included categories) or only from the selected folder (all objects underneath the category).

Use the Search bar to search in the database. The search function is the same as in the *Library* Search and filter area.

Click *Next* to open the next wizard dialog.

## 2. Batch mode dialog: Configuration

The second dialog configures the batch mode actions. Select the actions to be executed on the library objects.

The following options are available:

The options are divided into options for loaded models and options for models in the database. Options for loaded models require loading the entire model in Creo and are more time consuming. Options for models in database are only executed on the database.

You can also save the configuration for a batch mode job, load it later and run it again or make adaptations to the configuration.

**Saving and loading the configuration**

**Save configuration:** Writes the current settings for batch mode into a configuration file of the type: GTLE_BATCH_CFG. The default storage location for *Library* is preselected.

**Load configuration:** Loads a configuration file of the type GTLE_BATCH_CFG and restores the settings for batch mode stored in the selected file.

**Options for loaded models**

**Create instances as objects:** Specifies whether instances of a generic part are also included in the library database as independent objects.

- Add new instance objects to batch list: Specifies whether newly created library objects are included in the current batch list.
- Add existing instance objects to batch list: Specifies whether library objects being already known instances of generic parts are included in the batch list.
- Do not revise existing generics: Excludes generics from further batch mode.


© 2026 INNEO Solutions GmbH

**Read family table cell information:** Specifies whether family tables are read from parts and added to the database.

**Read parameters:** Specifies the parameters to be read from the library objects. They will be displayed in the *Library* Details window.

**Map parameters to existing language:** Specifies one or more parameters for each available language to be used as the library object title in *Library*.

Specify individual parameters by name. For multiple parameters, use a percent sign before and after each parameter name, for example:

– *parameter1*

– *%parameter1% - %parameter2%*

– *%parameter1% - Generic part*

**Create preview images:** Creates a preview image (image for the object display area) for each library object.

By changing the configuration option `gtl_editor_use_black_on_white_for_screenshots` you can decide whether the preview image will be created with the system colors "Black on White" (1) – i.e. the images will have a white background – or with the current color settings (0).

**Only if no preview image exists:** Excludes library objects that already have a preview image from image creation.

**Index – Form:** Creates an index for the form (bounding box side lengths) of library objects for 3D search[306].

**Index – Voxel:** Creates an index of voxel information (spatial record) of library objects for 3D search[306].

**Options for models in the database**

**Assign info documents from the info folder:** Looks up info documents of the model name in the folder specified in gtl_info_folder and links them to library objects of the same name in the database. Supported file formats: PDF, HTML, DOC, DOCX, XLS, XLSX and XLSM.

**Please note**: Paths are always calculated on the client. Use the configuration option `gtl_info_folder` to specify a path for multiple databases.

**Change path:** When this option is enabled, path segments (left input field) are replaced by an input (right input field). Use this option when Creo files referenced by library objects have been moved and cannot be found via the search paths. If an info document is stored in this Creo model, the corresponding path elements for the link to the info document are also changed. After changing the path, the info document can still be called.
For changing a path to a directory we recommend using the function Replace path substrings by in the Cleaning up data[363] dialog (broom button  ).


© 2026 INNEO Solutions GmbH

**Change status:** Change the status of all library objects in batch mode.

Click *Save configuration* if you want to run a batch list several times. This will save the configuration for later executions.

After configuring batch mode, click *OK* to start.

**Please note**: Depending on the selected options, a batch mode may take a long time.

## 12.3.4.8 Wizard: Import

The Object Import dialog is used to import library objects.

**Warning:** Always check the object type of library objects after importing them.

Click the button *Import Objects in current category* to open the dialog. Alternatively, use the context menu in the Preview area.

The following options are available:

**Folder**: Defines the file system folder that is to be imported.

**Import chosen folder as category**: Defines whether the folder will be imported as a separate library category below the current category.

**Import Folders only**: Imports a folder structure without including the contained data in the library database.

**Import Files and Folders**: Imports both: folders and files.

**Configure MNU import**: Defines the language assignment in MNU files.

**Configure STOOLS XML import**: Defines the language assignment in the XML files of the *Library Viewer* of the TOOLBOX.

**Check file existence**: Checks if all the referenced files are contained within folder structure.

**Warning:** Checking all referenced files takes a long time.

## 12.3.4.9 Wizard: MNU export

There is a dialog for exporting MNU files that guides you through the configuration of the export. In *Library Editor* click on the *Create MNUs* button to open the MNU export dialog.

The following options are available in the dialog:

**PRO_LIBRARY_DIR**: This is the directory in which the file *index.mnu* is created, which contains the content of the first level of *Library*.


© 2026 INNEO Solutions GmbH

**Warning:** Only use the environment variable %PRO_LIBRARY_DIR%, if this directory corresponds to the entry in the configuration file (*config.pro*). If it is not the same, use an absolute path.

**Category-based mode**: An MNU file is created for each category in a library. The files are stored in the path of the category (object source).

**File-based mode**: A folder and the associated MNU file are created for each category of a library. The structure corresponds to the categories of the library. Use this mode, if you have entered an absolute path in the field PRO_LIBRARY_DIR.

**Categories**: Select the categories to be exported here.

**First language/Second language**: Define here the language assignment for the MNU files.

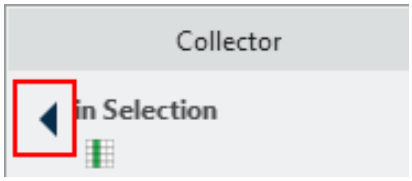
Confirm the dialog with *OK* to start the MNU export.

## 12.3.4.10 Wizard: Object collector

For revising library objects, the Object collector dialog exists. The object collector is used to edit properties such as status, actions or preview images of several library objects simultaneously.

Click the *Open object collector* button in *Library Editor*.

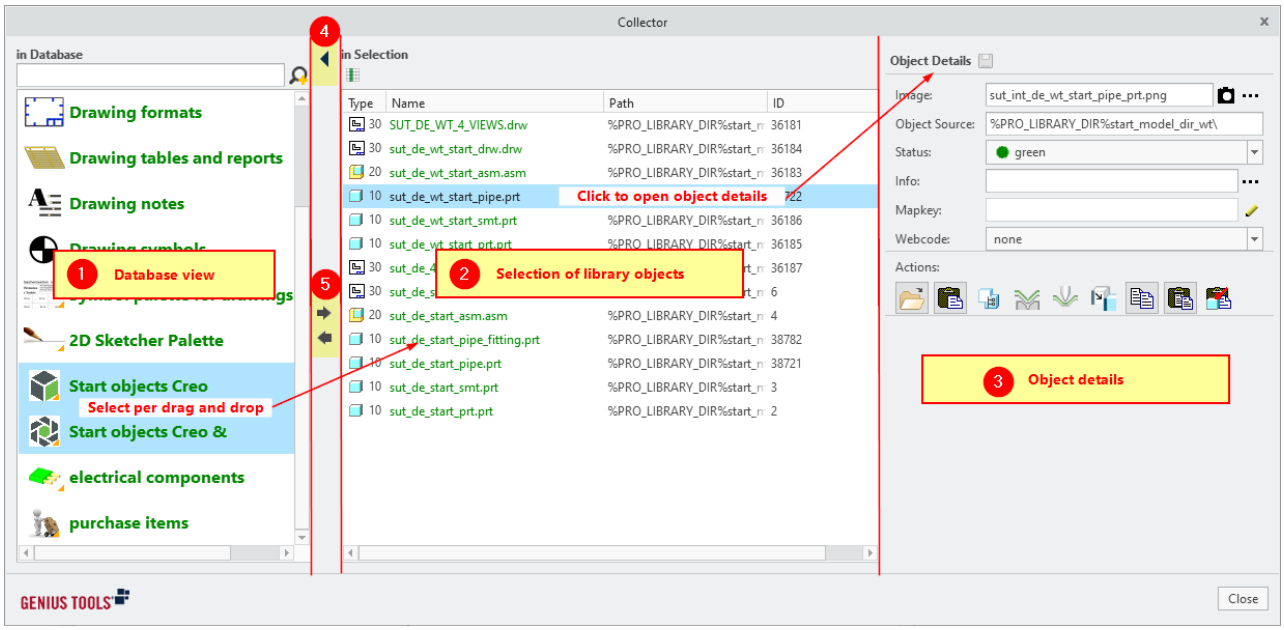
Click on the arrow button to display the database view.




In the database view (1), library objects and categories are displayed as in the library browser.

Add library objects to be revised to the selection list (1) via drag and drop or use the arrow buttons (5). Use the Shift or Control key to select multiple categories or objects.


© 2026 INNEO Solutions GmbH



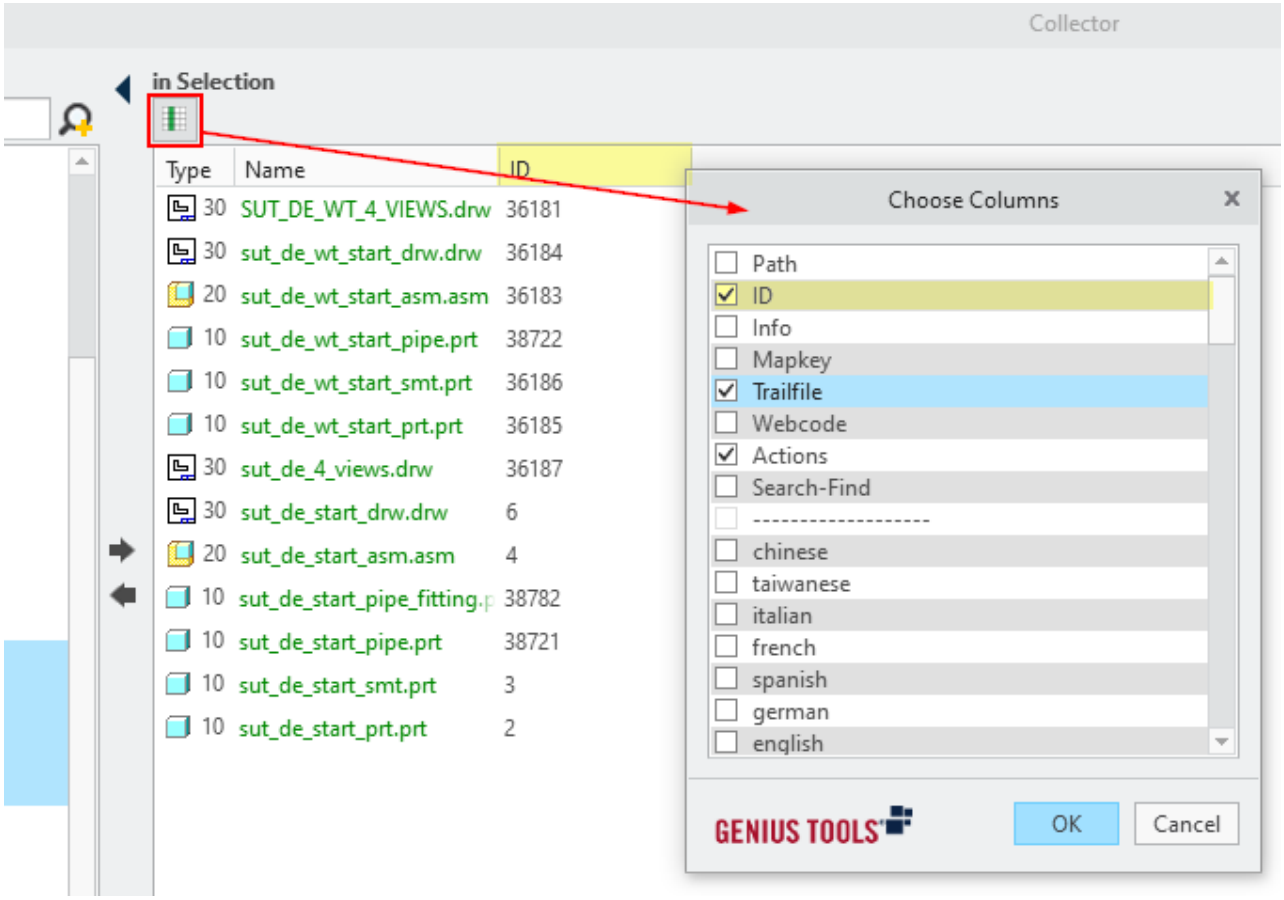

*Collector dialog window*

The object details (3) will be displayed if at least one library object is selected in the selection.

## Library objects in selection and Object details

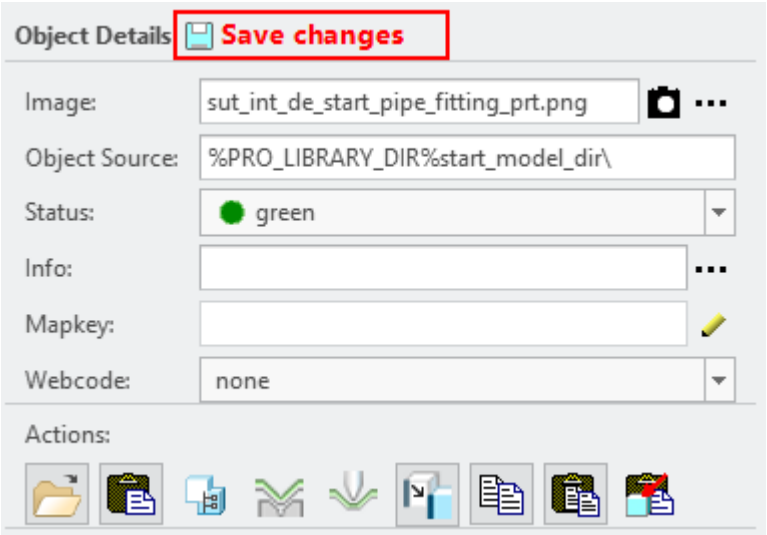
This section displays the library objects that have already been selected.

Use the button *Show Columns* to show additional information and select them by checking the boxes.




*Display additional information in the collector*

Select the library objects to be edited in the selection. Use the keys Shift and Control for multiple selections. To the right, modify object details to be revised.



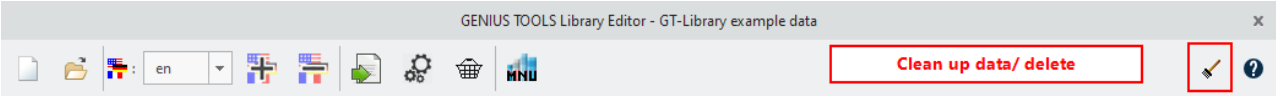

*Object details*

Click the button *Save Object* to save changes in the library database. Reload the library database in *Library* to work with the updated data basis.


© 2026 INNEO Solutions GmbH

## 12.3.4.11 Cleaning up data

The button *Cleanup* data opens the dialog box for cleaning the database. Deleted data cannot be recovered. The following actions are available.

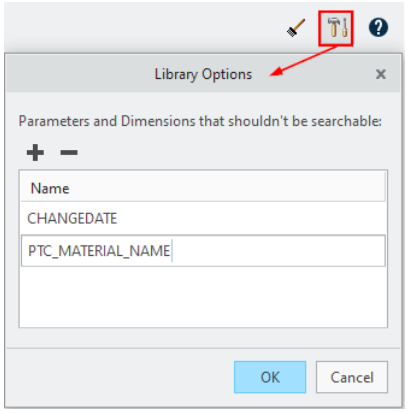



1. Set all instance images to the generic image (does not delete files)
2. Delete unused images from data directories as well as from all image directories (img, img_w40, img_w100)
3. Delete parameter / dimension: **from all models** in the database. This action can be used to delete e.g. incorrect parameters or dimensions.
4. Replace path substrings by
   The default settings in *GENIUS TOOLS*: the path set by the variable *PRO_LIBRARY_DIR* is replaced by *%PRO_LIBRARY_DIR%*.
5. Compress database

## 12.3.4.12 Customizing the search function

You can search all library objects by entering a term in the search field[304]. This can lead to unintentionally many results, e.g. when a search term is found both in the name of the object and in the material parameter.

In the command bar, you can use the tool icon to open a dialog and enter individual parameters and dimensions that should be excluded from the search.




**Please note:** Entries in this dialog will change the search strings in the entire database for *Library* and *Library Importer*. This may take some time.


© 2026 INNEO Solutions GmbH

## 12.3.5 XML interface for Form values

Values determined in advance can be read in for *Forms*[123] via an XML interface. Whether and for which forms this interface is used must be determined on a company-specific basis.

For users, the function *Form -Load values (.xml)* from an XML file is available in the tools menu[298] of the library browser.

An XML file for reading Form values must be in a specific XML format and UTF-8 encoding. This section describes the required XML format.

### Root element

The element `root` has the following attributes:

– `modelname` (optional): name of the model that contains the target Form, without extension

– `webcode`: webcode of the target Form

– `new_name`: name for the copy of the model that contains the target Form

### Form values

Each value to be set in the target Form is defined by an element `element` with the following attributes:

– `name`: name of the Form element

– `id` (optional): ID of the Form element. If no ID is given, the name is used for determining the Form element.

– `value`: value to be set in the Form

`element` is an empty element.

To access the names and IDs of the target Form elements, you can export the Form as an XML file via the tools menu of *Forms*.

### Models to be copied

The list of models to be copied with the model that contains the target Form is bracketed by an element `submodels`.

Each model to be copied with the target model is defined by an element `submodel` with the following properties:

– attribute `new_name`: name for the copy of the model, without extension

– attribute `copy` (optional): This attribute can contain the value `y` for *yes*, be empty, or be absent in order for the model to be copied. The model is not copied if you set `copy="n"`.

– element content: original name of the model to be copied, with extension


© 2026 INNEO Solutions GmbH

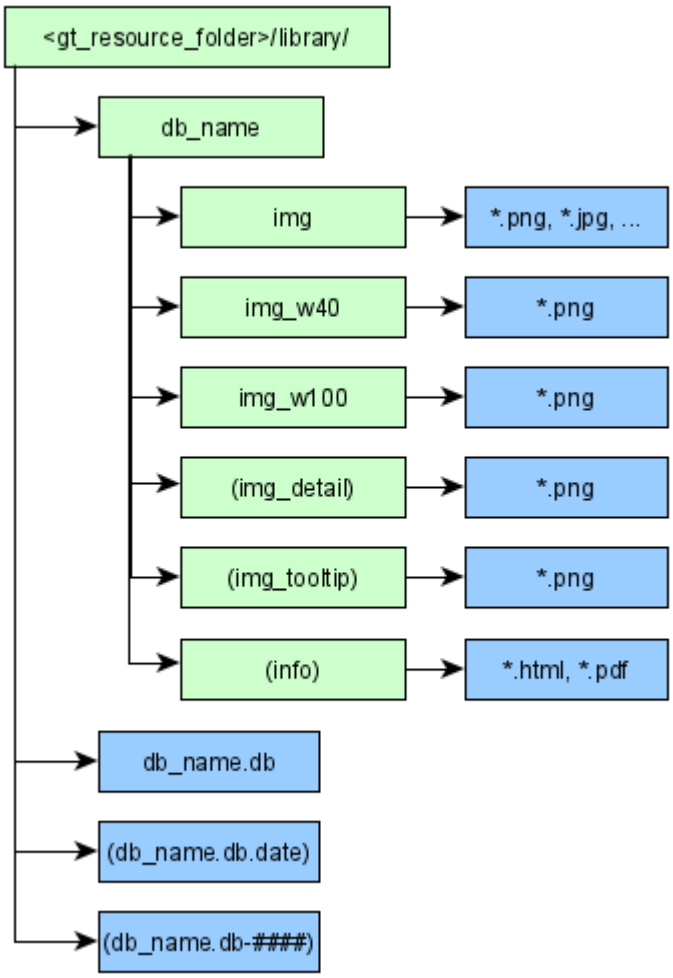
## Example

```
<?xml version="1.0" encoding="UTF-8"?>
<root modelname="SUT_INT_DE_START_CUBE" webcode="sut_int_de_start_cube"
      date="30.10.2018 9:28" new_name="model_copy">
      <element name="DESCRIPTION_1_DE" id="para|DESCRIPTION_1_DE"
      value="Startwürfel" />
      <element name="DESCRIPTION_1_EN" id="para|DESCRIPTION_1_EN"
      value="base cube" />
      <element name="length" id="dim|0" value="20" />
      <element name="width" id="dim|1" value="20" />
      <element name="thickness" id="dim|2" value="20" />
      <submodels>
            <submodel new_name="model_copy_drawing"
            copy="y">sut_int_de_start_cube.drw</submodel>
      </submodels>
</root>
```

## 12.3.6 Data structure

The data that is needed by *Library* is saved inside a folder located in a configurable location, with the default at `<gt_resource_folder>/library`. Every database that is usable inside Library has its own storage structure inside this folder.


© 2026 INNEO Solutions GmbH




*GT Library - Structure tree*

Files are represented in blue and folders in green in the figure. Images that are linked by the database are saved inside the image folders. Folders and files that are shown in brackets can be there, but do not have to be.

## Back up copies

Images in the image folder and back up copies of the database (`db_name.db.####`) can be deleted. This has no impact on Library.

By default a back up copy is created once a day. This can be deactivated by setting the configuration option `gtl_editor_create_db_security_copy_once_a_day`.

## Synchronization

The file *db_name.db.date* is important for synchronization. It needs to be rewritten at every change of the database. This is done automatically by the Editor or GENIUS TOOLS Data Importer. If there is no *db_name.db.date* file the database *db_name.db* will be synchronized on every folder change. In this case, you should deactivate the synchronization, see also Configuration Options 799.


© 2026 INNEO Solutions GmbH

## Images / Convert images

Different sizes of each image are saved pre-formatted in the folders *img_w40* and *img_w100*. If these do not exist when *Library Editor* is started, they will be created and the pictures from the respective other directory will be converted into them. This behavior has been implemented for legacy data, but it can also be used to clean up inconsistencies. It's important to remember that images from the *img_w40* folder cannot be resized to a bigger size at the conversion.

Detail images can be saved in the *img_detail* folder. You can find more information under Library Editor - Object pictures[344].

## Info files

Info files can be saved in the info folder and linked automatically with objects by a batch process. You can find more information about batch modes under Library Editor - Object details[330] and Library Editor - Wizard: Batch mode[356].

# 12.3.7 Setting configuration options

The following options allow you to define further settings. These options can be found in the Library module in *Configuration Utility*[765].

**gtl_gtp_start_gtp_after_model_creation**
Determines whether the *Parameters*[472] dialog box opens automatically after a new file is created. Specify the desired file types separated by commas. Possible file types: prt, asm, drw. Default: None.

The parameter dialog does not open if you create a new file and insert it into an assembly in one step.

# 12.3.8 Use cases

In this section, you will find short instructions on tasks relating to *Library Editor*.

## 12.3.8.1 Creating a new library

Proceed as follows to create a new library:

1. Open *Library Editor*.
2. In the command bar, click *Create new database* .
3. Enter a descriptive name for the new library. Be sure not to use special characters. The database and the required sub-directories are created in the resource directory.


© 2026 INNEO Solutions GmbH

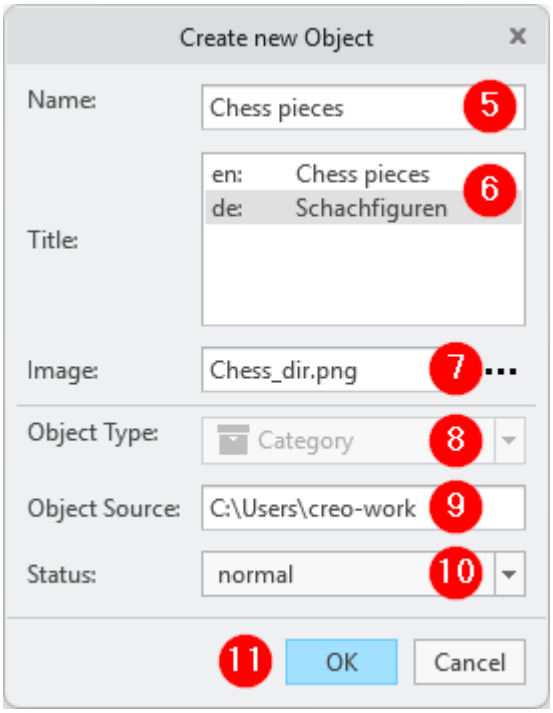
4. In the next step, check the languages set in *Library Editor*. A standard library has German and English language set.
5. Click *Add a language* to add new library languages using language codes. Use the *Remove a language* button to remove the language codes not needed.

**Please note**: The language code en and the language code specified via the `gtl_lang` configuration option cannot be removed.

Your new library is now ready for use. Proceed with adding categories and library objects to your database. They can either be created manually or be imported.

## 12.3.8.2 Creating a new category

1. Open the *Library Editor*.
2. Click *Open database* and select a database to be edited.
3. In the category tree[325], navigate to the position where you want to create a new category.
4. Open the context menu in the Preview area (right mouse button) and click *Add category*.
5. Enter the Name for the new category. Enter the *folder name* as the Name if the category corresponds to a folder.




6. Enter a Title for the category. Fill in the language dependent input fields.
7. If the category is to contain an image[344], choose the appropriate preview image by directly entering its location into the input field or by selecting a preview image via the (...) button.


© 2026 INNEO Solutions GmbH

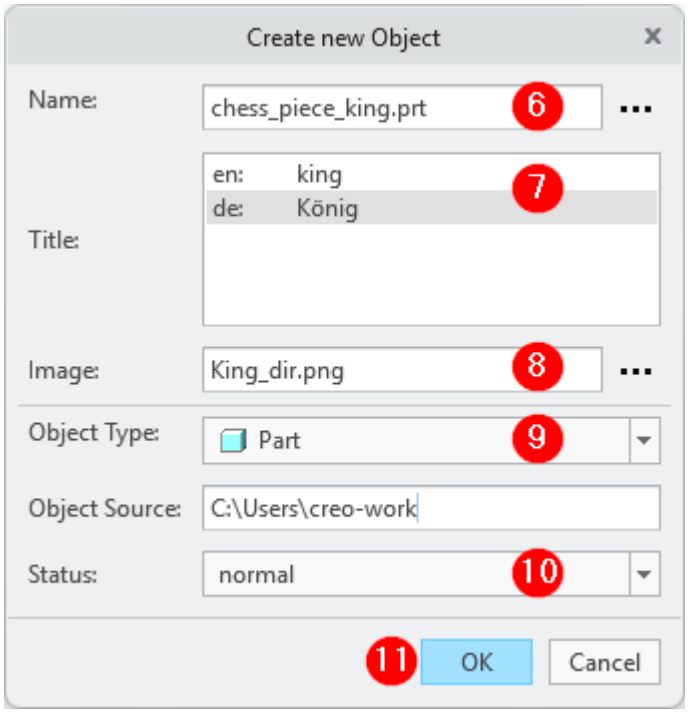
**Tip**: When you create the category, the preview image is copied to the database image folder in the resources directory with a new name. When you open the category in the editor, another image name is displayed.

8. Check if the object type[330] is set to category.
9. Enter the object source. Enter the *path*, without the last folder name, as the object source if the category corresponds to a folder.
10. Specify a status[339] for the category.
11. Click *OK* to complete the category creation.

Your new library category is now ready for use. Proceed by adding library objects to your database. They can be created manually or be imported.

## 12.3.8.3 Creating a new library object

1. Open the *Library Editor*.
2. Click *Open database* and select a database to be edited.
3. Navigate to the location in your database structure where you want to create a new library object.
4. Open the context menu in the Preview area (right mouse button) and click *Add object*.
5. In the next dialog, enter the following information:




6. Enter the name of the object to be created or select a Creo object via the *three dotted button*.


© 2026 INNEO Solutions GmbH

7. Enter a title for the object. Fill in all language dependent input fields.
8. If the library object should contain an image[344], choose the appropriate preview image by directly entering its location into the input field or by selecting a preview image using the (...) button.
   Transferring title information from parameters to library objects and creating preview images can be automated via batch mode.
9. Check the object type[334] and adjust if needed.

**Please note**: Pay special attention to the object type with manufacturing assemblies.

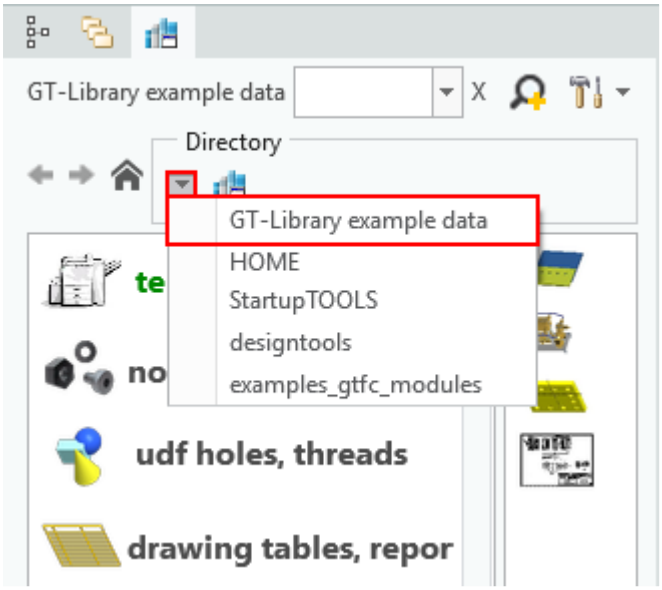
10. Specify a status[339] for the library object.
    Use the preset status *Invisible* as long as you have not completed the revision of your library objects. Invisible library objects are not displayed in the editor preview, switch to list view to see them.
    Setting statuses can be automated via batch mode.
11. Confirm the dialog with *OK*. Your new library object is now created.
12. Check the object details in the *Library Editor* and adjust if necessary.
13. Add an information document to the new object: Click on the *three dotted button* after **Info** and specify a document that can later be opened from the *Library* Details window.
14. Specify a mapkey or trail file with a mapkey for your library object: Click the *Pen icon* after **Mapkey** to enter a mapkey directly in the next dialog (e.g. via Copy and Paste) or click on the *three dotted button* after **Trail file** to select a trail file from your file system.
15. Specify the executable actions on the library object: Click the actions that should be available for the library object.
16. Automatable revisions for your library object: Modifying title information, preview images and status can be automated via batch mode. Instances and family tables can be captured automatically as well.

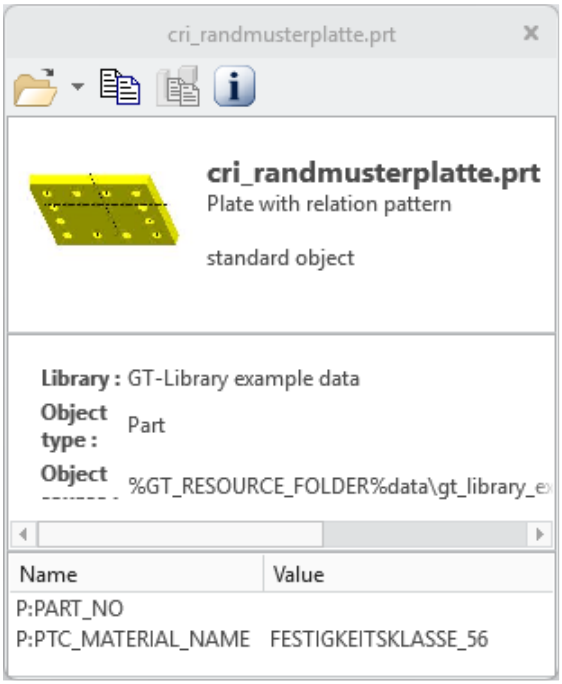
## 12.3.8.4 Creating a library object with a form

This procedure allows you to create variants from a template more quickly and in a more defined manner.

1. Go to *Library* and open the library object *cri_randmusterplatte.prt* in the *GT-Library axample data database*.

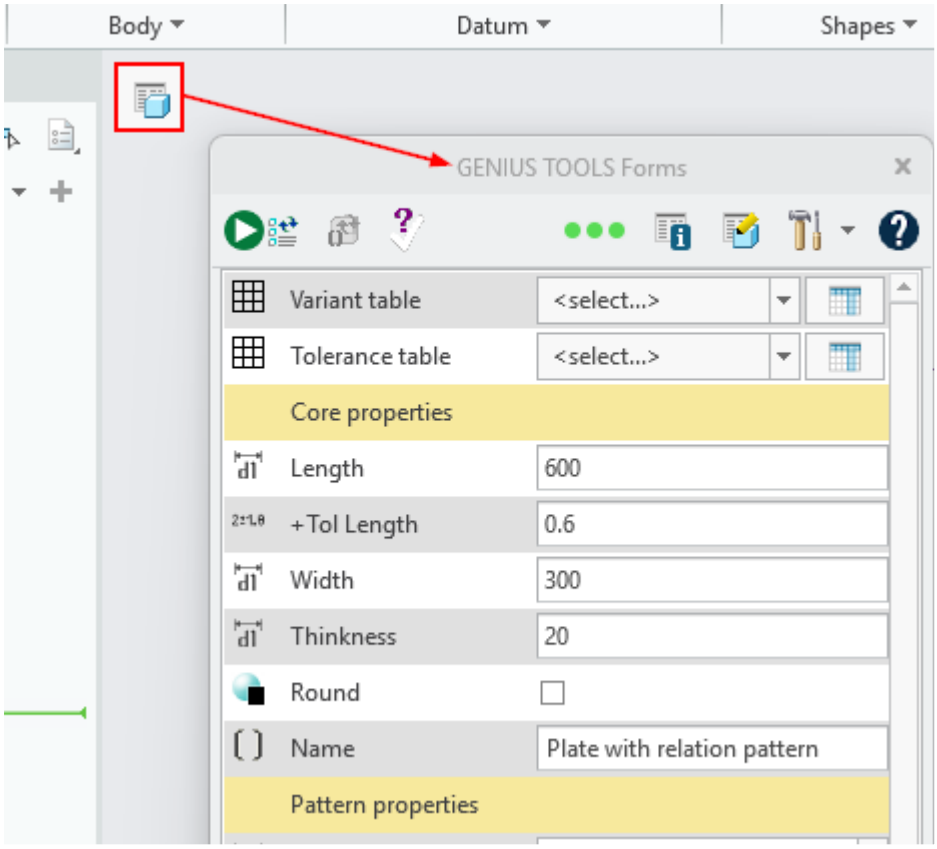

© 2026 INNEO Solutions GmbH



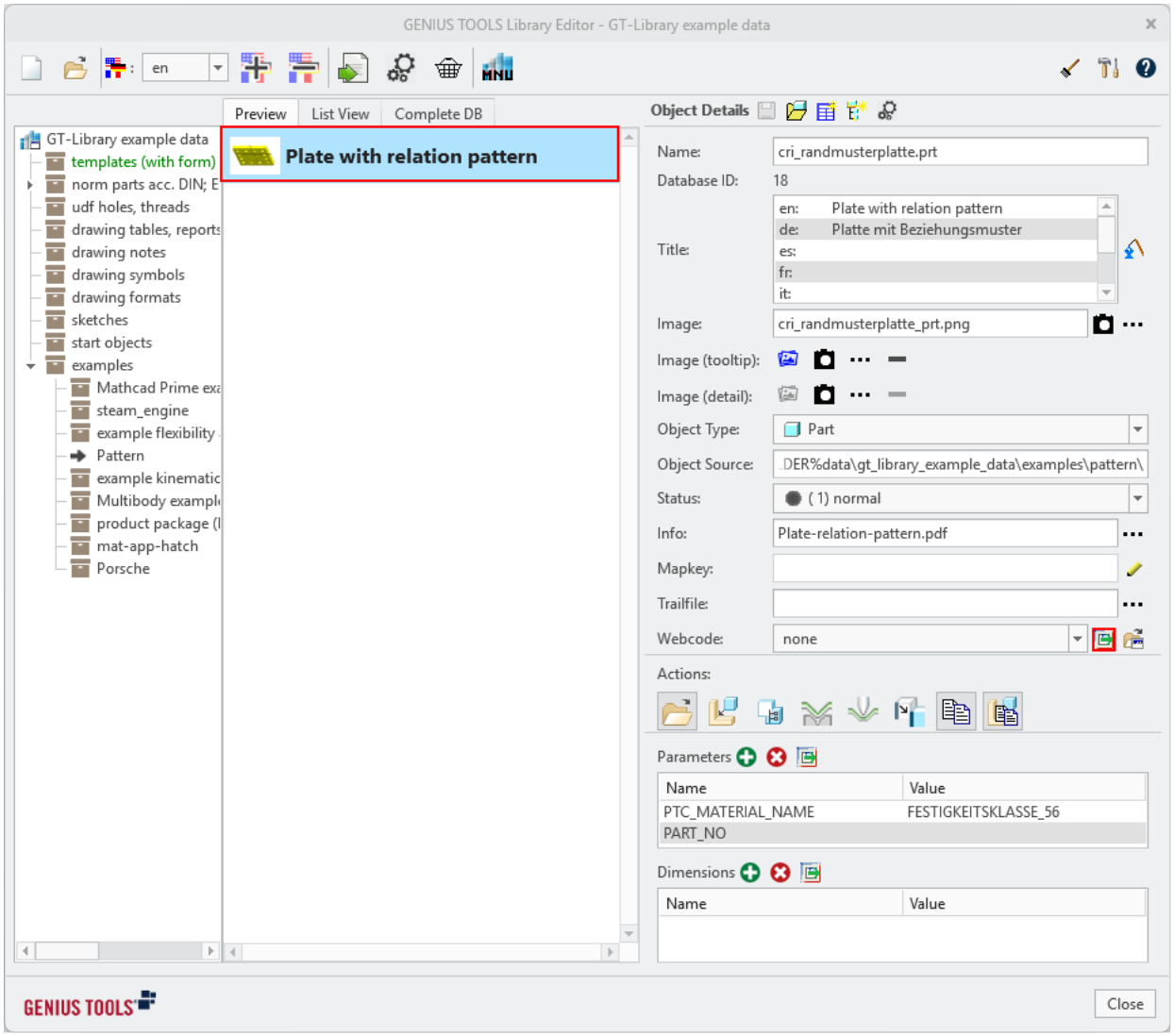

2. This component already contains a *Form*[123]. Open the form and take a look at it. A form like this must be included in this model in order to transfer it to the library.


© 2026 INNEO Solutions GmbH




3. Open *Library Editor* and click on the library object *cri_randmusterplatte.prt* an. In the object details, click *Copy internal Form of the current Model as external Form* .


© 2026 INNEO Solutions GmbH

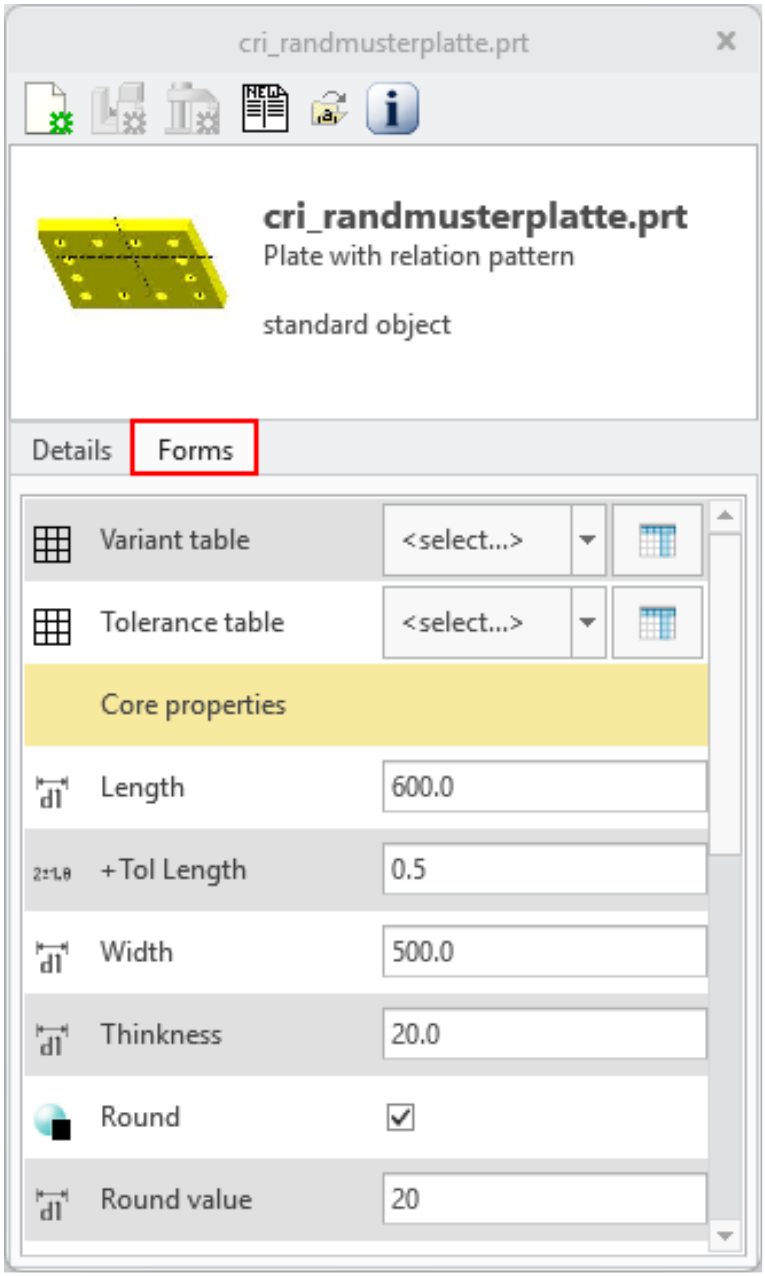



4. Close the current model in Creo Parametric.

**Please note:** To create a new object with a form from a library object, the library object itself must not be in the Creo Parametric session.

5. Go back to Library and click on the library object *cri_randompatternplate.prt*. The Forms tab is displayed.


*© 2026 INNEO Solutions GmbH*




6. In *Library Editor, you can select cri_randmusterplatte.prt* in the object details under *Webcode*.

## 12.3.8.5 Importing family tables

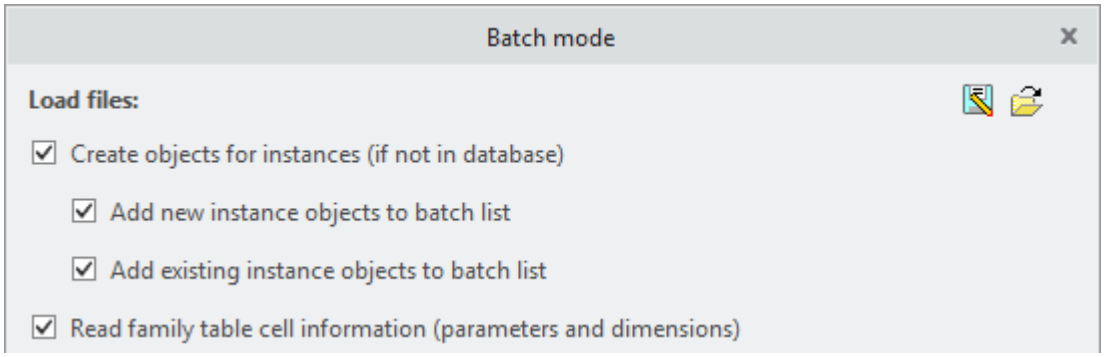
Proceed as follows to import family tables of a generic part:

1. Create a new object in your library structure and select the generic part. The generic part should have previously been saved to the directory using Creo Parametric (*.idx updated). Verifying the family table and setting up a search path guarantee error-free use of the variants in Creo Parametric.


© 2026 INNEO Solutions GmbH

If the instances should have the same preview image as the generic part, specify the preview image prior to the batch mode. Then the image will be inherited by the instances.

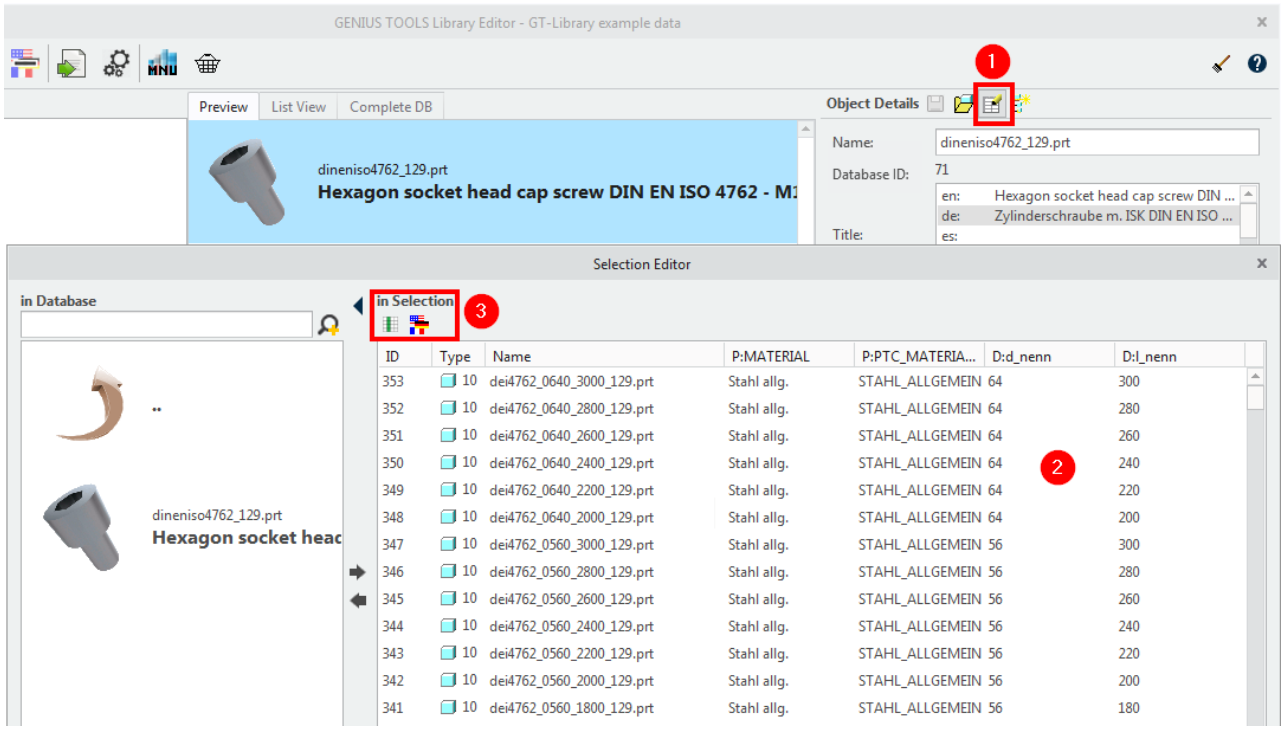
2. Start a batch mode. Add the library object you have just created to the selection. Then, click *Next*.
3. Activate the following options for the batch mode:
   a. Create objects from instances (if not in database)
   b. Add new instance objects to batch list
   c. Add existing instance objects to batch list
   d. Read family table cell information (parameters and dimensions)
   e. If each instance should have an individual preview image, also activate the *Create preview images* option.




*Activate these options to import family tables*

4. Click *OK*. The family table is read. This may take some time.
5. Select the generic part in the editor and click *Edit selection list* in the Object Details area.
6. Check the created selection list. If needed, complete missing information. Confirm the dialog after making changes.

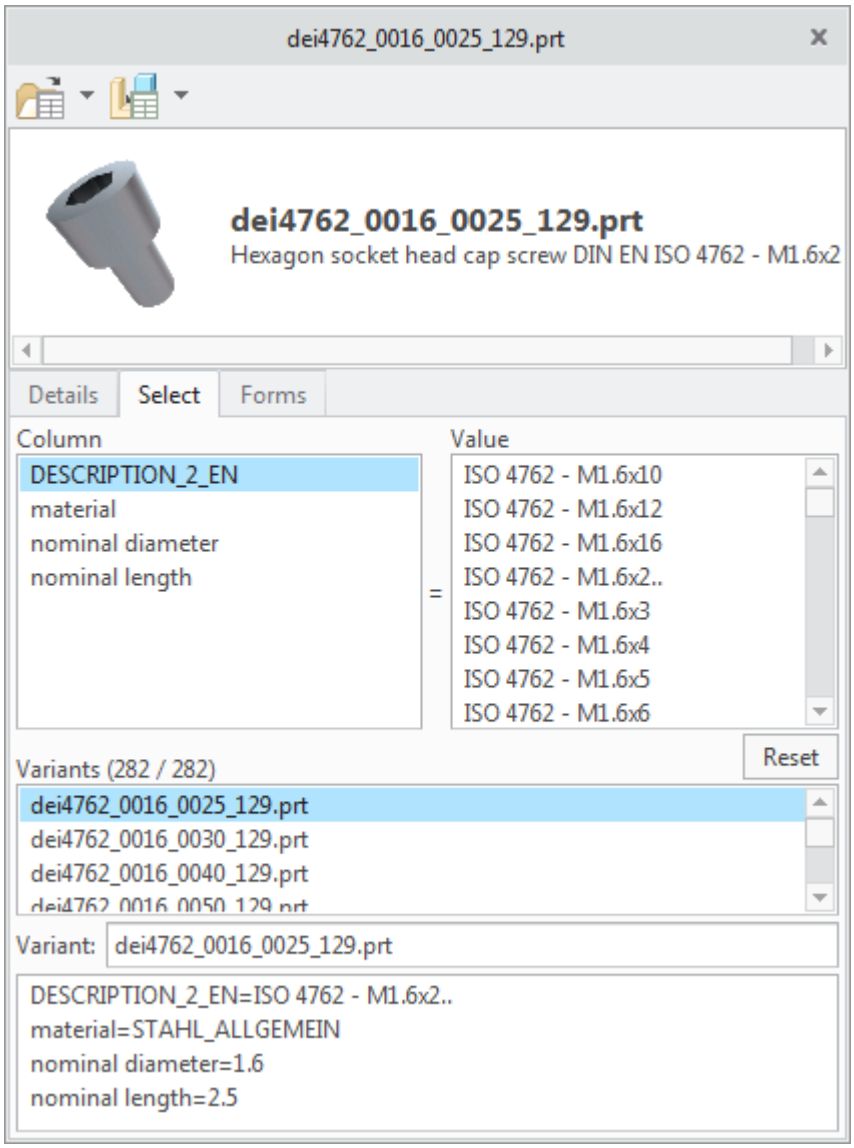

© 2026 INNEO Solutions GmbH




*In the editor, click "Edit selection list" (1). Then check the selection list (2). You can customize the columns display and assign multilingual column names using the command bar buttons (3).*

The family table is now imported completely and is available as a selection table in the library object.


*© 2026 INNEO Solutions GmbH*




*The selection list is ready for use*

## 12.3.8.6 Updating objects

When model information changes, this changed information must also be revised in your library database.

Use the *Library Editor* batch mode to update your library objects.

Using batch mode you can, for example:

– Update modified parameters

– Map modified parameters to multilingual titles

– Update preview images for your library objects

For these three scenarios, proceed as described at Importing additional object information[379]. Existing preview images, titles and imported parameters will be overwritten.

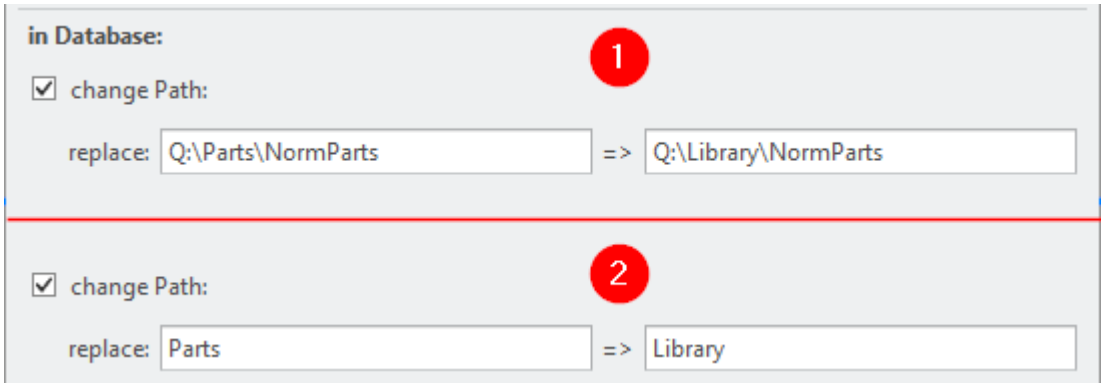

© 2026 INNEO Solutions GmbH

# Updating file paths

File paths can be updated – first, by using the batch mode[356] function (for single library objects) and second, by using the function *Replace path substrings by* in the Cleaning up data[363] dialog.

### Updating file path of single library objects: batch mode

1. Open *Library Editor*.
2. In the command bar click *Execute batch mode* .
3. Select the library elements whose file paths have changed the same way. Confirm your selection by clicking *Next*.
   Example: All standard parts have been moved from `Q:\Parts\NormParts` to `Q:\Library\NormParts`.
4. In the second dialog box deselect all options except for *change Path*.
5. In the first input field, enter the whole file path (1) or the segment of the old file path (2) that should be changed.
6. In the second input field, enter the replacement for the segment.
7. Click *OK* to start the batch mode.




*Segment in the second dialog box of the Batch Revision function*

**Please note**: Entire file paths or just segments of a path can be replaced. Make sure there are no library objects included in the batch mode for which the file path segment should not be replaced.

### Updating file path of all library objects in a directory: Clean up data

1. In the command bar click *Cleanup* data (broom button ).
2. Select the checkbox *Replace path substrings by*
3. In the first input field, enter the segment of the old file path that should be changed.
4. In the second input field, enter the replacement for the segment.
5. Click OK to start the cleanup function.


© 2026 INNEO Solutions GmbH

## 12.3.8.7 Importing additional object information

Use batch mode to import additional information for library objects into the database.

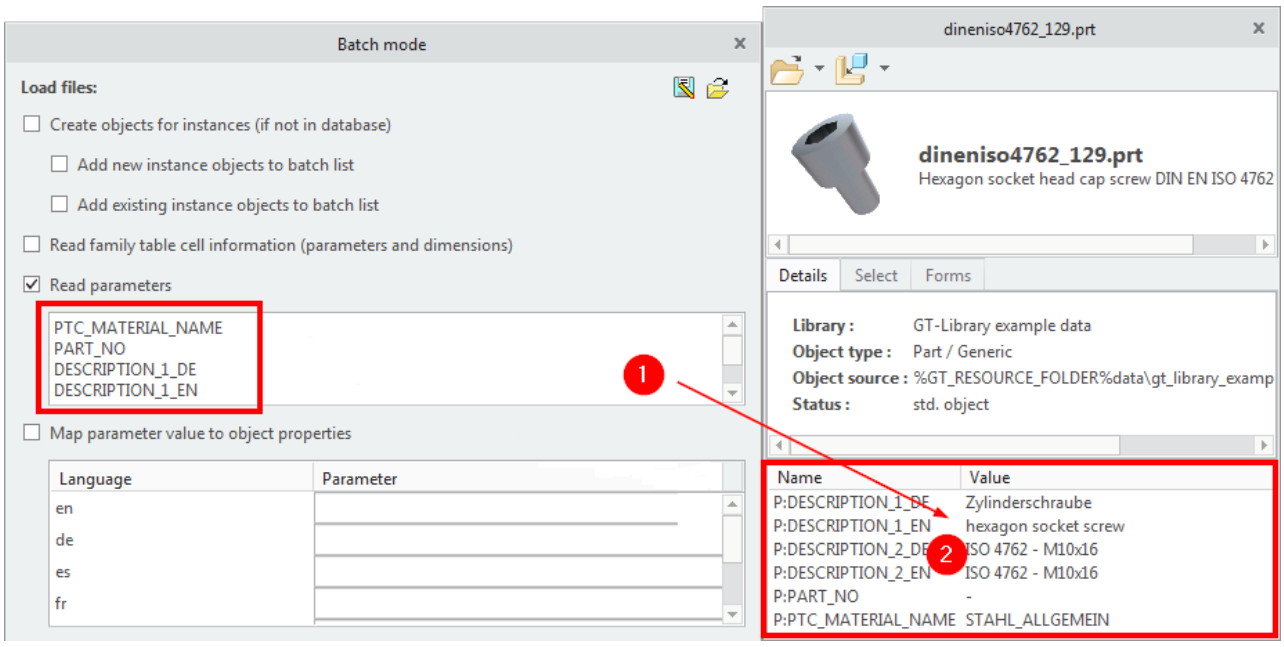
1. Open *Library Editor*.
2. Click *Execute batch mode*.
3. Select the library objects for which the additional information should be imported into the database. (individually or an entire category)
4. Click *Next*.
5. Select the additional information to be processed. In the dialog, deactivate all revision options that are not needed.
6. Click *OK* to start the revision.

## Importing parameters for search optimization

Imported parameters are saved for each library object. They are displayed in the Details window when selecting a library object and can be used in the search, in advanced search and in selection tables.

1. Activate the *Read parameters* option.
2. In the input field below, enter the parameters that should be imported. Enter one parameter per line.

**Please note**: Make sure you spell all parameters correctly. Parameters not found in a library object will be ignored.




*All parameters read into the batch revision (1) are imported into the library objects and are displayed in the Details window, for example (2)*


© 2026 INNEO Solutions GmbH

# Importing parameter as library object titles

Parameters from Creo data can be used to generate multilingual titles for library objects. A library object's title increases retrievability and can be used in the search and in advanced search. You can also search for titles in languages not displayed.

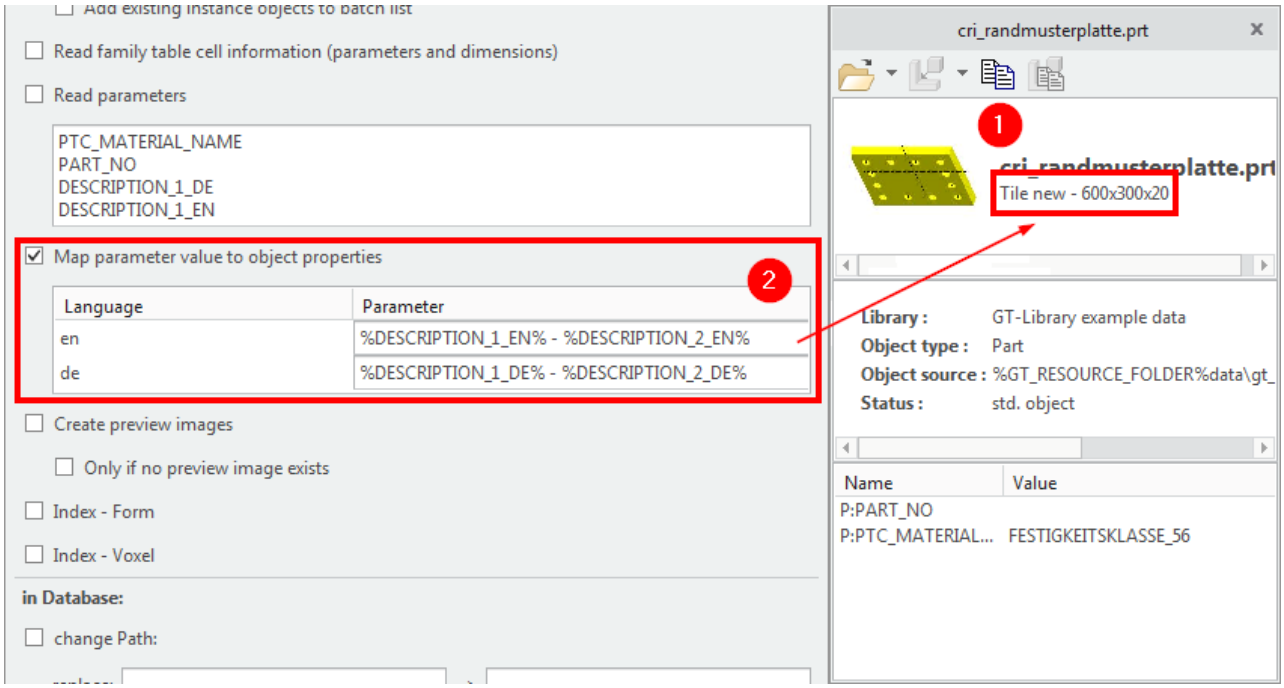
1. Activate the *Map parameter value to object properties* option.
2. In the table, enter the parameter to be used as the title after the individual language codes.

You can also combine parameters and text, and use multiple parameters as the title. Use percent signs (%) to separate parameters from each other and from static text.

**Examples**

```
%parameter1% - %parameter2%

%parameter1% - Generic part
```




*Parameters mapped to a language (1) are available as title in library objects (2)*

# Creating preview images automatically

Preview images increase retrievability of library objects in the Library browser.

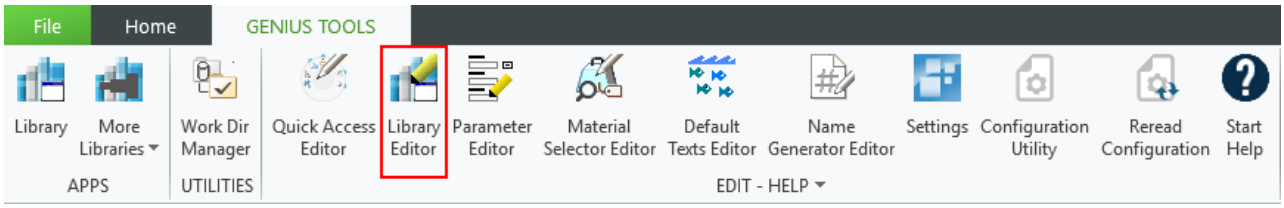
1. Activate the *Create preview images* option.
2. If for some of the library objects preview images do already exist, activate the additional option *Only if no preview image exists*.


© 2026 INNEO Solutions GmbH

# 12.3.8.8 Importing Library Viewer libraries

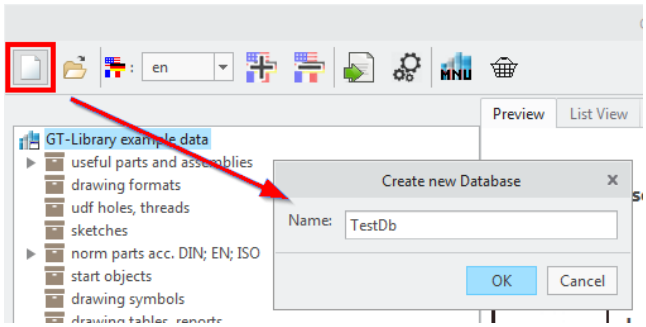
*Library* can import the file-based libraries of the *Startup TOOLS TOOLBOX Library Viewer*. These step-by-step instructions explain how to import a Library Viewer library using an example.

Start the *Library Editor*.




*Start GENIUS TOOLS Library Editor from the ribbon*

Create a new database and enter a descriptive name. The name entered is also the database file name.



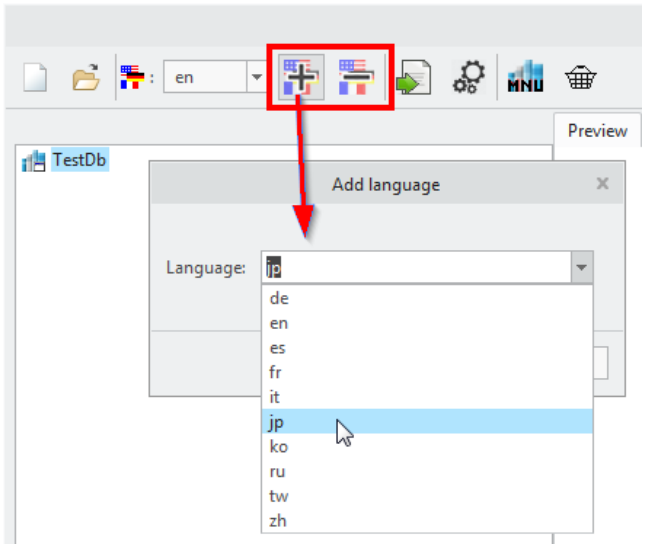

*Create a new database*

**Warning**: Make sure nobody else is working on the database during import when using an existing database.

Create the additional languages needed in the database. Select a language code from the drop-down list or enter any custom language codes. This step ensures that all additional information maintained can be reused in the new database.

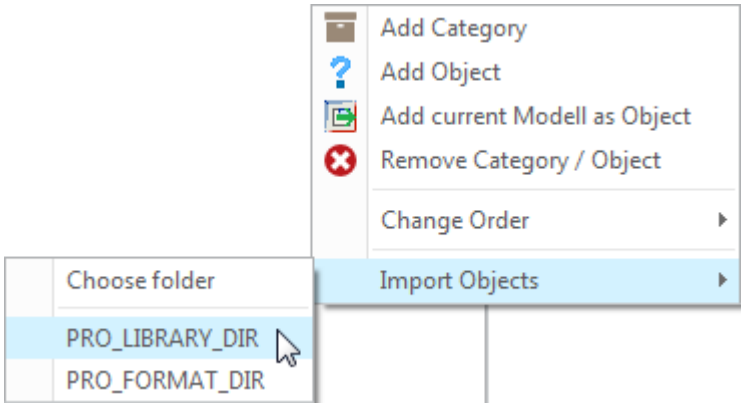
**Please note**: The input does not accept special characters and numbers.


© 2026 INNEO Solutions GmbH




*Manage the included languages from the flag buttons*

Open the context menu in the library preview (center area of *Library Editor*). Select the menu entry *Import Objects* and select *PRO_LIBRARY_DIR*.

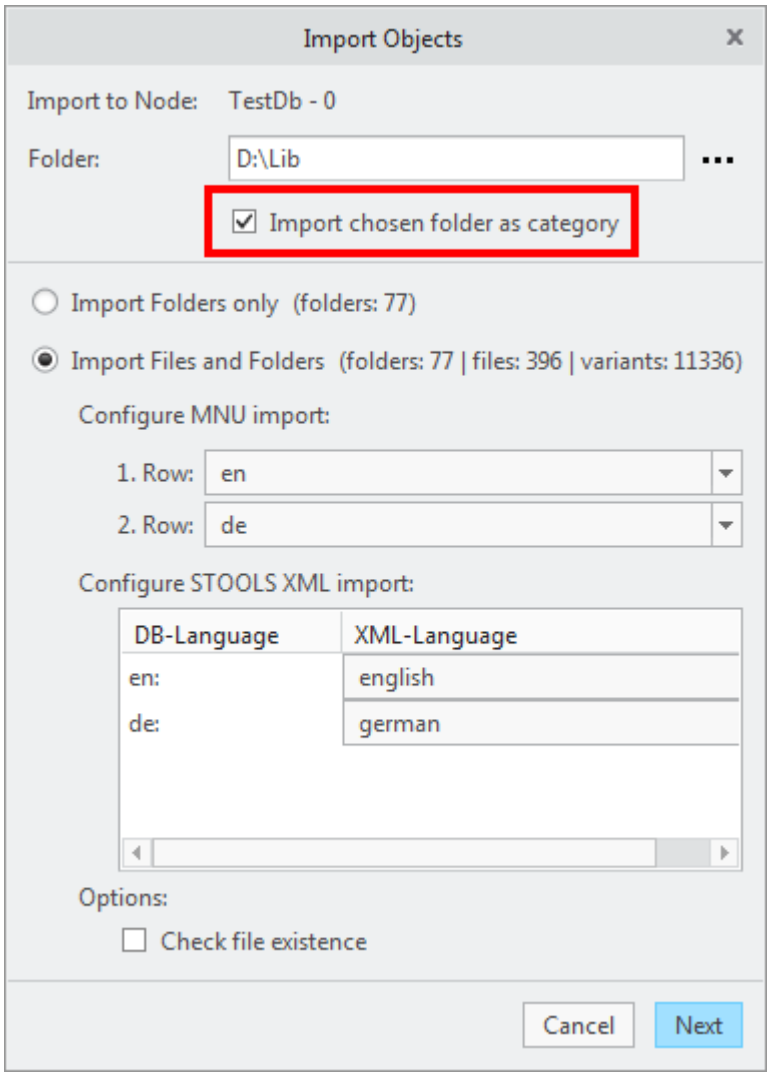



*The context menu in the preview area opens via right mouse button*

Deactivate the option *Import chosen folder as category*. Otherwise, the PRO_LIBRARY_DIR will be displayed as a category above the existing library structure in the Library browser.

Make sure to select the proper library directory when working with Startup TOOLS and multiple projects. To find out which folder is currently referred to, check the config.pro option `PRO_LIBRARY_DIR` in the current Creo session.


© 2026 INNEO Solutions GmbH




*Check the directories and options in the import dialog.*
*Do not forget the language assignments!*

**Please note**: Check the assignment of the language settings.

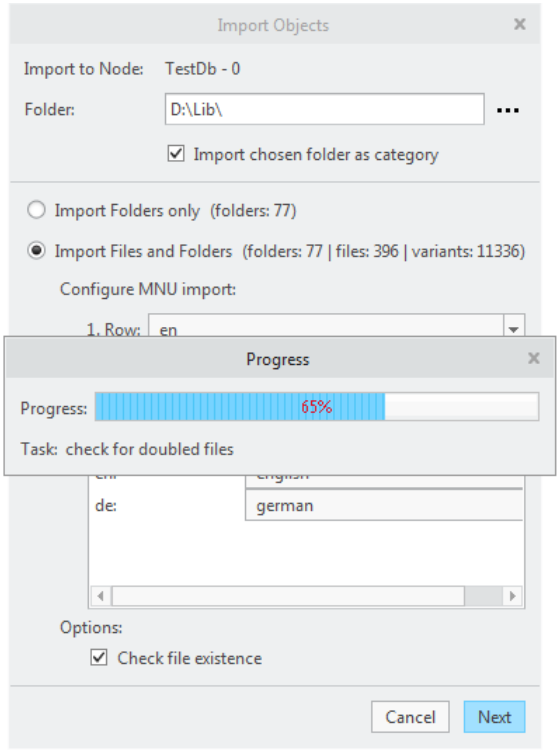
Activate the *Check file existence* option if multiple manual changes have been made in your library. With this option, you can ensure all files exist. This action is executed immediately as you activate it.

Do not use this option with Windchill as the library files will not be found.

Click *Next*.

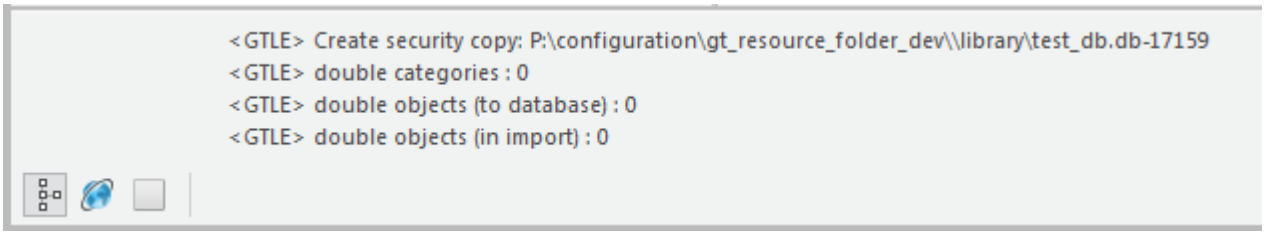
The library directory is checked for duplicate files.


© 2026 INNEO Solutions GmbH




*With bigger libraries, checking for duplicate files takes some time*

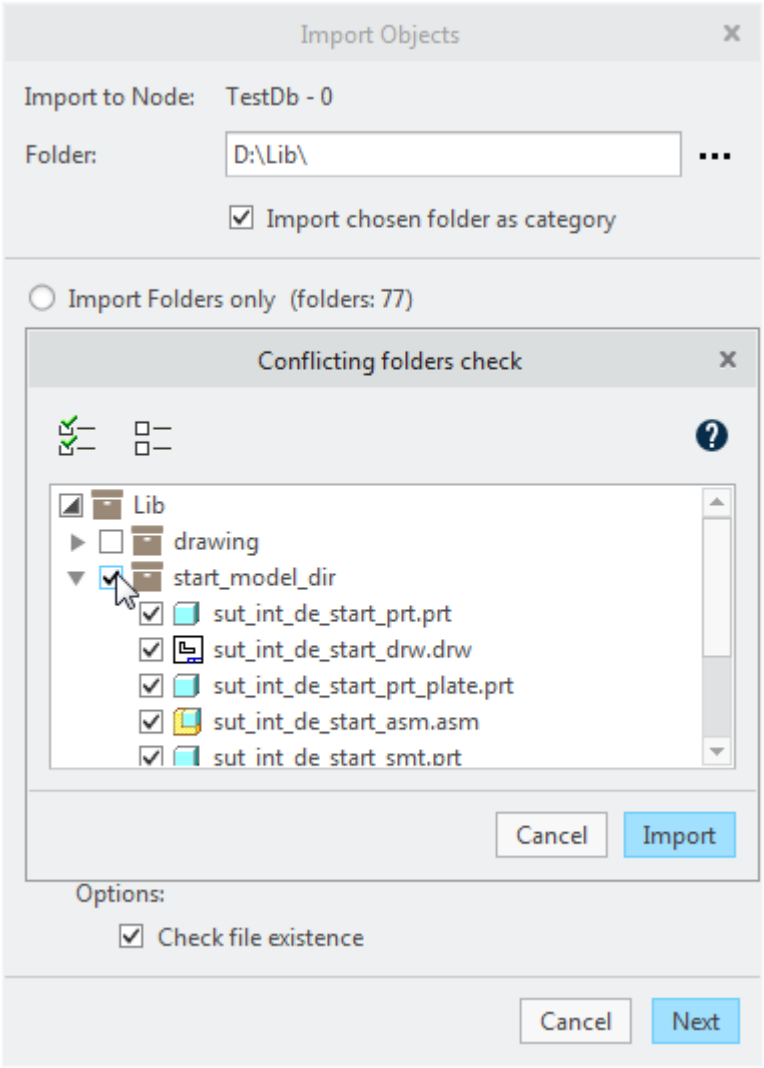
Additional information on the importing directory can be found in the Creo Parametric info area after the check.




*Refer to the Creo info area for additional information*

Select the elements to be imported in the tree structure.


© 2026 INNEO Solutions GmbH



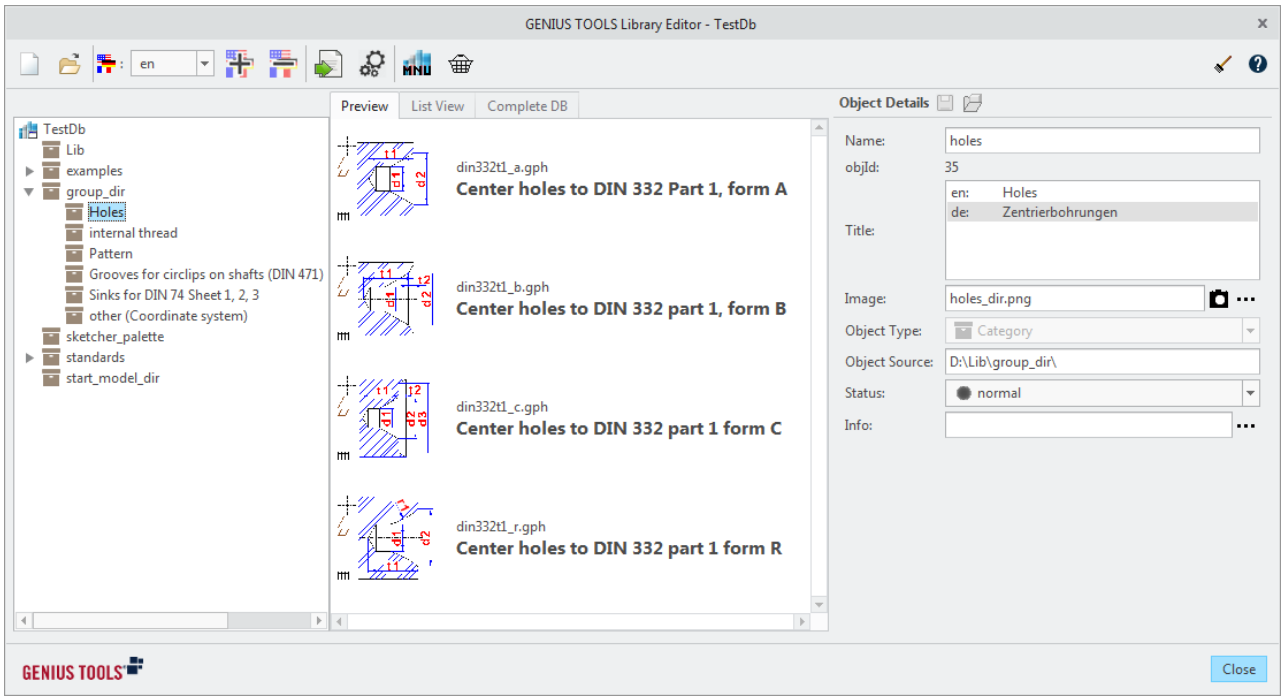

*Select the files and directories to be imported and start the import*

Click *Import*.

**Please note**: Checking and importing may take some time. This depends on the storage medium containing the library data as well as the access speed to the library.

The import of the *Library Viewer* library is now complete. The library functionality (except for ModelUIs) is now established.


© 2026 INNEO Solutions GmbH




*After importing, the library database is ready for use*

To include additional information for searching and selection (parameters, family table values and variants) in the library, execute a batch mode.

## 12.3.8.9 Importing MUI files

Proceed as follows to import MUI files into a library.

The MUI template models have to be present in a *Library* library, e.g., because you imported them. Also, the model for which you want to import an MUI has to be loaded in Creo.

1. Open the library containing the imported template models in the *GT Library Editor*.
2. Search for the required template model (ASM or PRT) for the MUI you want to import. Open the model in Creo.
3. Click *Convert STOOLS MUI* in the object details. A dialog is displayed. Select the MUI and confirm.

## Verifying the conversion

1. Close the *Library Editor* and select the model in *Library*. In the details window, select the tab *Form*.
2. Create a new model using the form.
3. Verify the properties of the newly created model. Were the correct models copied with it (asm, prt, drw) and the correct names assigned?
4. Verify the copying rules in the *Library Editor* and the Library configuration options 830:
   a. gtl_action_copy_set_file_as_common_name


© 2026 INNEO Solutions GmbH

b. gtl_copy_drawings_with_same_name
c. gtl_copy_check_existence
d. gtl_gtf_save_forms_in_model
e. gtl_gtng_overwrite_std_number_definition
f. gtl_gtng_standard_db_filter_for_file_copy

5. Verify the form as displayed in the Library details window
   a. Are all form elements present?
   b. Do all form elements work?

**Please note**: Forms that have been changed have to be re-assigned to their library object.


© 2026 INNEO Solutions GmbH

# 13 Material

Available in the ribbon menu GENIUS TOOLS

The component *Material* is available in part mode with the following features:

- Selection of materials based on material properties
- Assignment of materials to parts and bodies by selecting an MTL file from the material directory
  These files usually have cryptic names, e.g. *10143_s275j0.mtl* for an unalloyed structural steel.
- Adjustment and localization of the material data display in the Editor
- Automatic checking of materials in models, e.g. after their revision

**Please note:** *Material* does not support MAT files. You can use the Utility *Convert Materials*[663] to transform existing MAT files into MTL files.

## Components of Material

The module *Material* includes the following components:

| | | |
|---|---|---|
|  | Material Selector[391] | – assigns materials to a part / bodies of a part<br>– can filter the material list by one or more properties<br>e.g. according to delivery standard, material group or revision parameters<br>– reads material properties from the database *material.db* |
|  | Material Selector Editor[401] | – determines, which materials are available to users in the material selection and which properties are displayed<br>– edits properties in different languages<br>– stores information about each material<br>e.g. standard specifications<br>– writes selected information from the MTL-files of the material directory into the material database *material.db* |
|  | Material Browser[410] | – edits the MTL-files |


© 2026 INNEO Solutions GmbH

**Please note:** In order to provide users with material information, it is necessary to grant access to *Material Selector Editor*.

## Material database

In Creo material properties are stored in a model, i. e. in the file of a part (PRT file). The content of these properties is read from material files (MTL files) in the material directory.

With *Material* material properties are also sourced from MTL files, but are additionally copied into an SQLite database (material.db) in order to make them available to the users in the user component *Material Selector*.

When loading MTL files into the database, the directory structure of the material directory is preserved, i. e. subfolders are taken into account.

The database *material.db* is located in *%GT_RESOURCE_FOLDER%\material*.

The intermediate step of going through a database is necessary to restrict the material selection to Creo users, to display defined material properties in a favored language and to add extra information (PDF or web link). These functions are available in the administrative component *Material Selector Editor*.

In addition, this structure enables material version checking between the material properties stored in the model and the material properties maintained in the material directory.

## Updating the database material.db

Please note that the material database is not updated automatically. Update the material database in *Material Selector Editor* using the button *Update database (*Integrity check405*)*  .

# 13.1 Usage

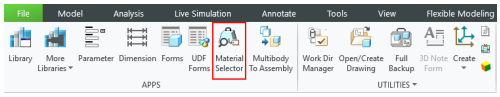
The user component of *Material* is the material selection dialog *Material Selector*. It lists all materials and their properties, which are stored in the material database and which are approved in the editing tool *Material Selector Editor*401. For an easier selection process, materials can be filtered by one or more of their properties and corresponding values.

## Starting the program: in part mode

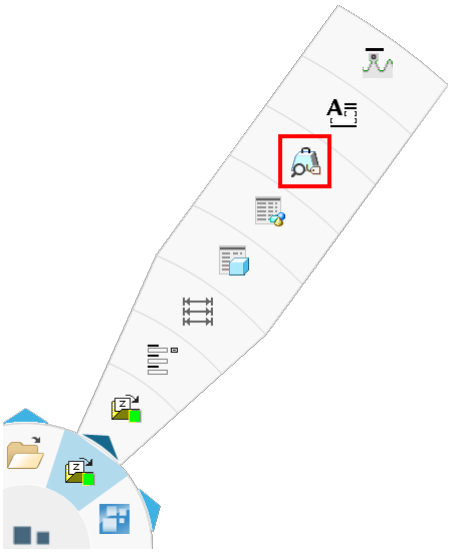
To start *Material Selector*  , click *GENIUS TOOLS* in the ribbon menu and go to APPS section or by using *Quick Access* (key [<]).

The button is available in part mode only since materials cannot be assigned in any other mode (such as assembly or drawing mode).


© 2026 INNEO Solutions GmbH




*Starting via the GENIUS TOOLS in the ribbon menu*

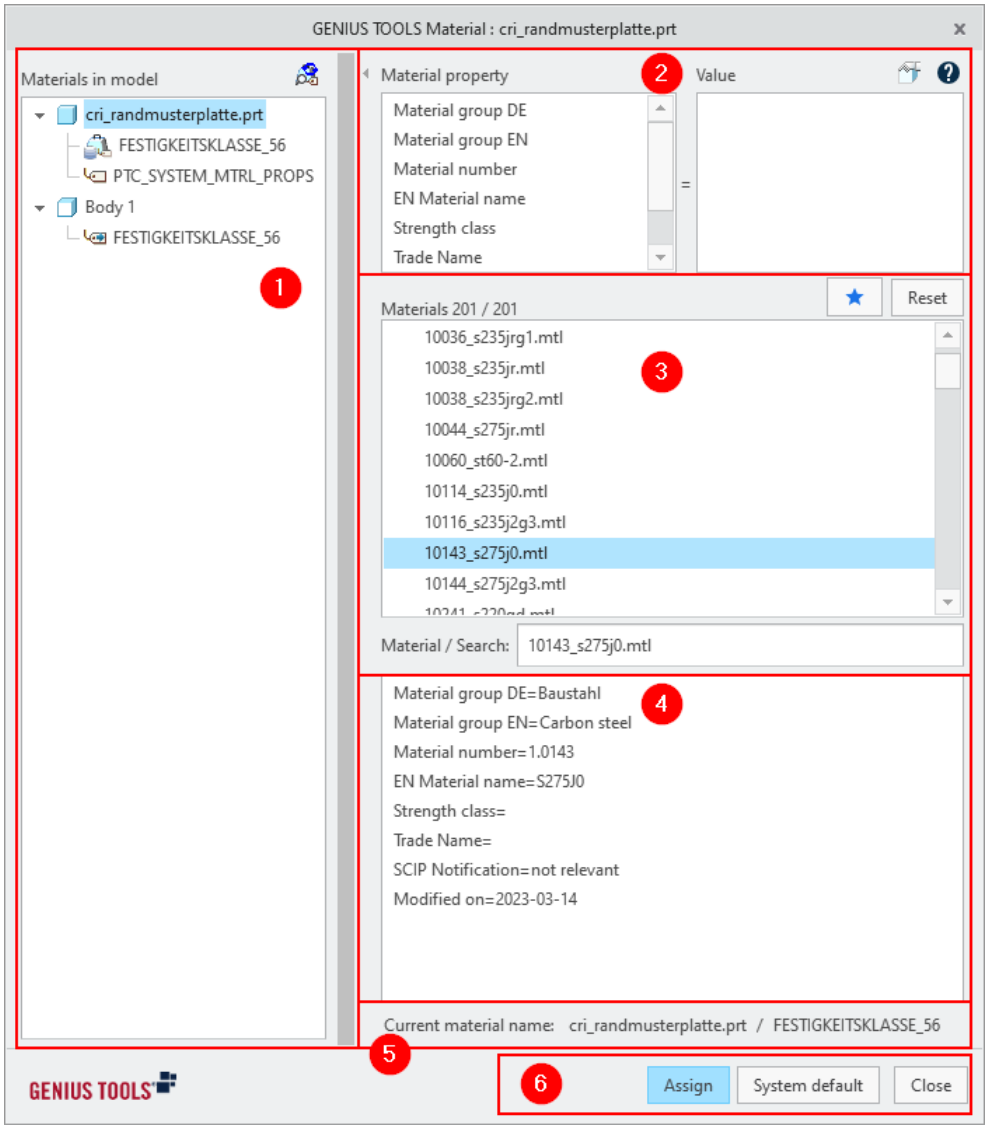


*Starting via Quick Access*


© 2026 INNEO Solutions GmbH

## 13.1.2 User interface

The user interface of *Material Selector* consists of the following elements:




1. Display of all materials in a model[395]
2. Material properties[392] and corresponding values, link to the Creo material dialog and to this help
3. Material list[393] of the material database, Setting and displaying favorites[393], Search for materials[394]

**Please note:** You can only select materials for which the *Is selectable*[404] checkbox is ticked in *Material Editor*.

4. Detail view of the selected material[394]
5. Display of currently assigned (active) material


© 2026 INNEO Solutions GmbH

6. Execute commands: Assign[394], System default[395], Close

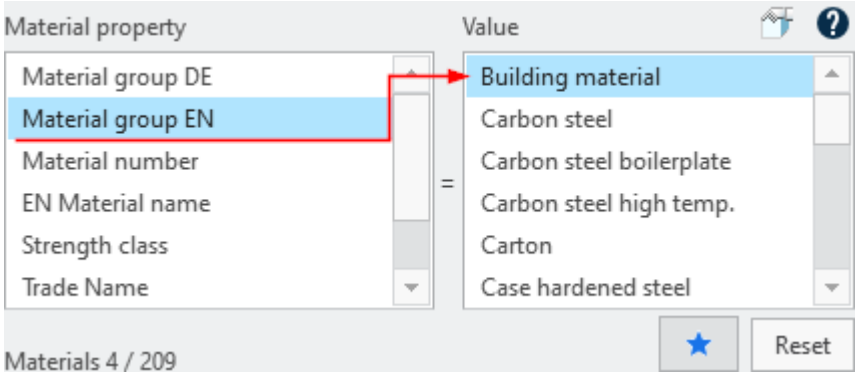
## 13.1.3 Selecting materials

The following commands are available when selecting a material:

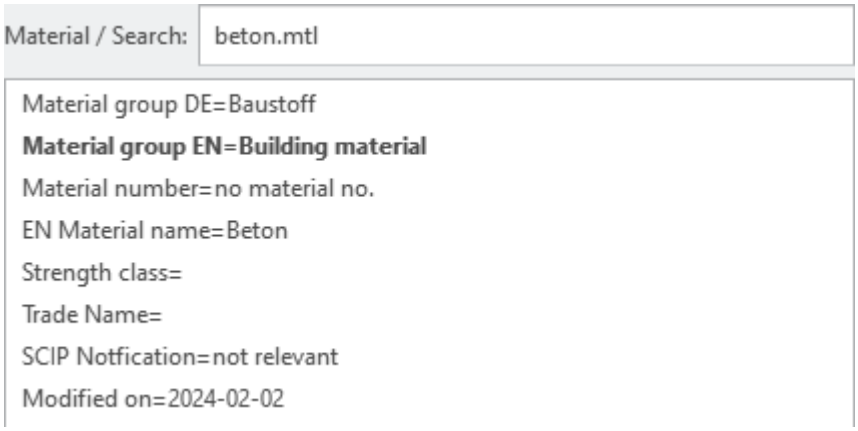
### Filtering materials by property and value

The fields *Material property* and *Value* restrict the materials displayed in the material list.

First, click a material attribute and then select the desired value, e. g. *Material group EN* > *Building Material*. All materials matching the selection are displayed in the material list. Repeat this step with additional *Material property-Value* pairs to further restrict the selection.




The selection of your filters is displayed in the material list[393]. The first material in this list is displayed in the detail view[394]. All properties of a material are listed there. The properties that match the set filters are displayed in bold:




The display of the material properties depends on the language that is set. If no translation is available, the name of the material property in the database is used. Displayed attributes and values are defined in *Material Selector Editor*[401], which manages the material database.

Click *Reset* to reset the selection.


© 2026 INNEO Solutions GmbH

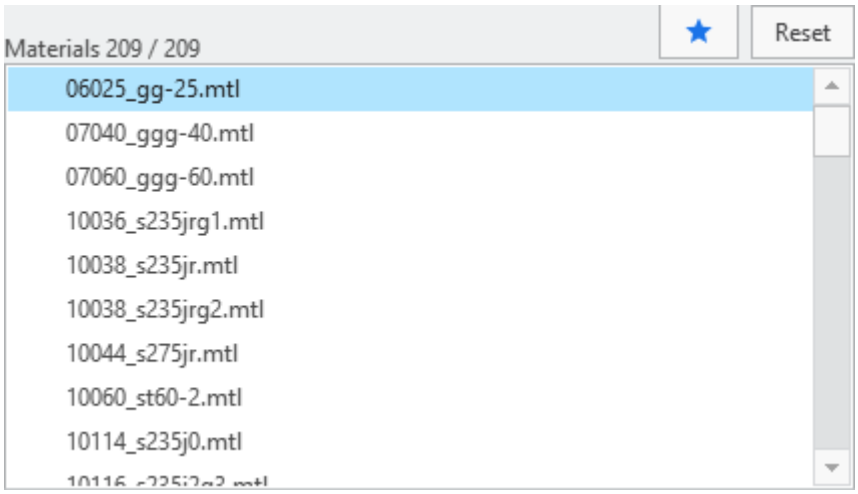
# Displaying available materials

The material list displays the results of the material attribute/value filtering.

The number of results (2) and the number of all available materials is displayed above the material list.

Click on the required material to select it and *Assign* to assign it to a model.

The selected material (3) is displayed below the material list.




**Please note:** You can only select materials for which the *Is selectable*[404] checkbox is ticked in *Material Editor*.

# Create favorites

A material can be defined as a frequently used material (*favorite*) by right-clicking on it. Once a favorite was defined, the star icon appears. Clicking the star icon controls the material display as follows:

| | |
|---|---|
| Button not highlighted (Default) | Display of all materials |
| Button highlighted with dark gray background | Display only the materials marked as favorites |

This does not apply any filter criteria. Instead, the favorite materials displayed are read from a specially created file. By default, the file *filter.txt* is created under *%appdata%\<OperatingEnvironment>\GENIUSTOOLS\for_Creo* and is also edited as soon as favorites are added or deleted. In the configuration option `gtm_favorite_file`, you can specify a different file and thus distribute, for example, a predefined file to specific sites or business units.

The configuration option `gtm_allow_edit_favorites` is used to prevent users from editing this list of favorites. You will not be able to right-click a material to add it as a favorite. (Default is 1 = Allow)


© 2026 INNEO Solutions GmbH

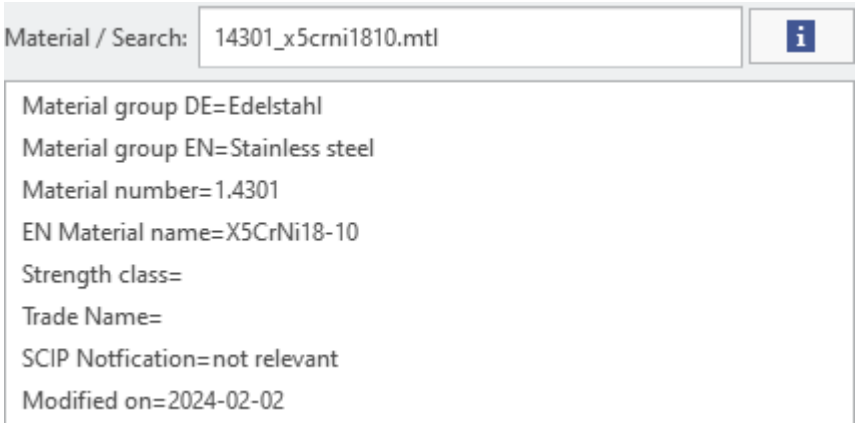
## Displaying current material / Searching material names

The selected material appears below the material list. The name of the current material is displayed in the text box. You can also start a material search from this input field. Click in the input field and enter any search terms. The search includes a text search in the file names and in all values. Press *Enter* to confirm your input. The search results are displayed in the material list.

Without active search, the currently selected material is displayed in the search field. If filters are set, properties corresponding to the set filters are printed in bold[392].

**Please note:** Only materials meeting the current filter are searched. If you want to search all materials, click *Reset* first.

If a material is selected in the *Materials in model* area on the left, it is immediately inserted in the input field and displayed as a result in the material list. This makes it possible to quickly assign a material that is already contained in the model to a body or the entire part.




**Info document**

If a material with an info document is selected, an info icon appears on the right of the material name. Click to open it. Info documents can be assigned to the materials via *Material Selector Editor*[406].

If you want to see more properties, expand the selectable material properties in *Material Selector Editor*.

## 13.1.4 Assigning materials

After clicking the *Assign* Assign button, the dialog box closes. If you want to keep the dialog box opened, e. g. because you assign materials to several bodies, set the configuration option `gtm_close_at_set_material` to 0. To close the user interface manually, use the *Close* Close button.


© 2026 INNEO Solutions GmbH

## Assigning the system default

All parts receive the system material *PTC_SYSTEM_MTRL_PROPS*. The properties of the system material can be edited, but the material itself cannot be deleted in a model. Use the button *System default* 

to assign the system material to a part. Bodies inherit the active material (master) from the part.

## Display of all materials in a model

In the left segment *Materials in model* (1) all parts and bodies with their assigned materials are listed.

The material that is assigned to a part has the icon (active or standard material / master material). All other materials in a model receive the icon . A material in a body that is inherited from the part has the icon .

Additionally, the following icons can be displayed as a result of material checks: - - . You have to activate the material check and a deviation of a material has to be found for one of those symbols to appear. Find a list of the meaning of the symbols in chapter Checking materials in a model.[408]

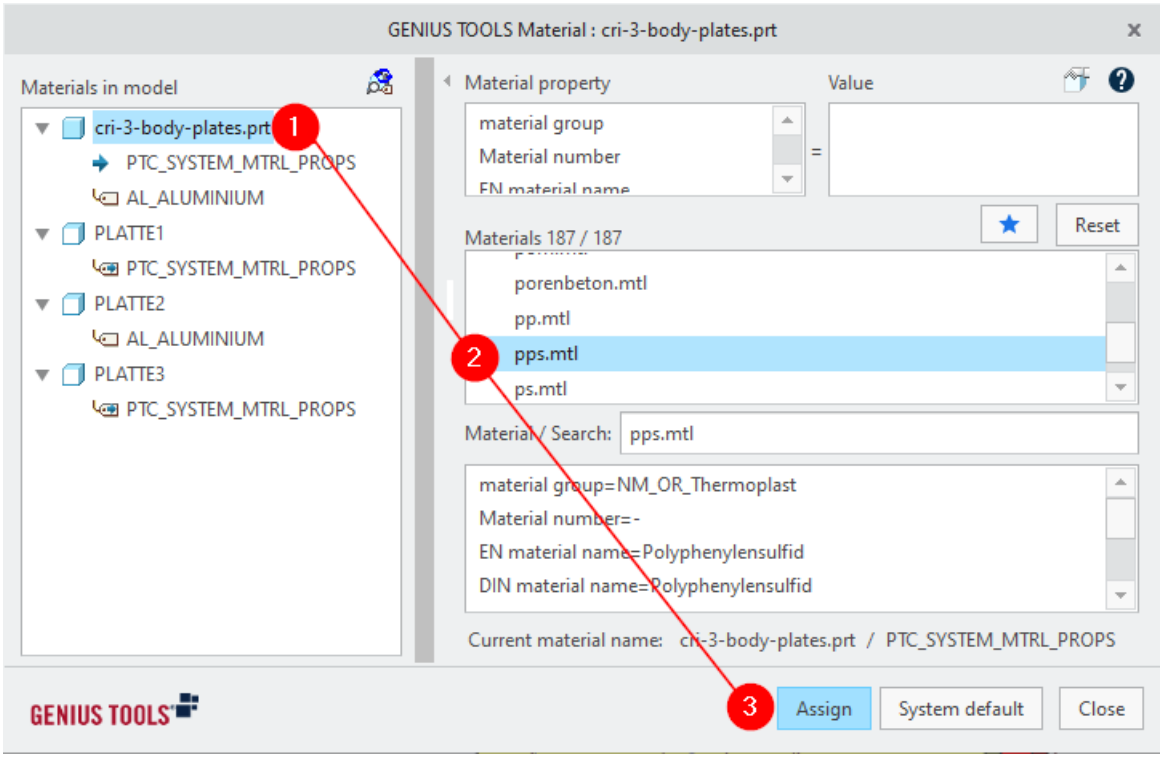
The left segment *Material in model* can be collapsed with the arrow icon . You can define that it is already hidden when opening the user interface by setting the configuration option `gtm_show_body_selection` to 0.

## Assigning a material from the material list

**Please note:** You can only select materials for which the *Is selectable*[404] checkbox is ticked in *Material Editor*.

1. In *Material Selector* select a part or a body in the left segment *Materials in model*.


© 2026 INNEO Solutions GmbH

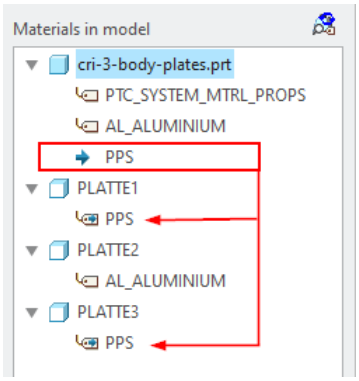



*Selecting the part "cri-3-body-plates" and the material "pps"*

2. Select the required material in the material list[393] on the right.
3. Click *Assign*.

**Results: Part versus body**

**– Assign material to the entire part:**
The newly assigned material PPS is set as the standard / master material for the part (cri-3-body-plates) and gets the icon.

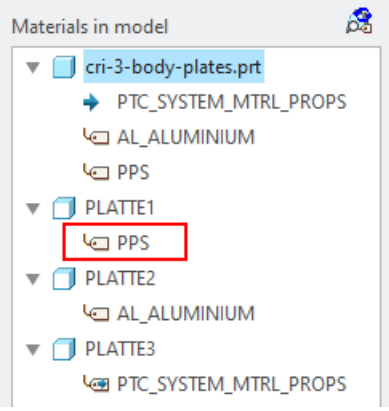



The new standard material is inherited to all bodies where an inheritance has previously been defined, as shown by the icon. (Here: bodies "PLATTE1" and "PLATTE2")

Materials that are no longer active remain in the model and are marked with the icon. (here: PTC_SYSTEM_MTRL_PROPS)


© 2026 INNEO Solutions GmbH

**– Assign material to a body:** The newly assigned material PPS is assigned to the selected body ("PLATTE1") and marked with the icon.

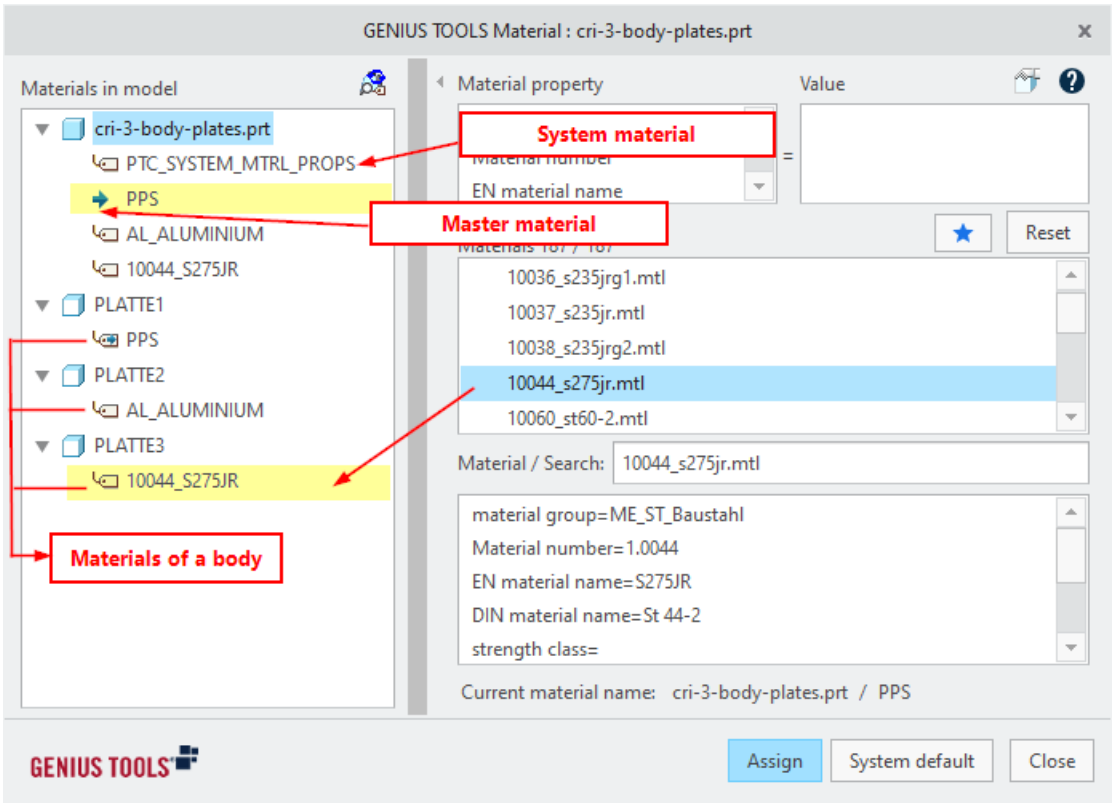



4. To display the newly assigned material in *Material Selector*, it must be explicitly activated in the *Material Selector Editor*, see Display settings of Material Selector dialog 406.

## Assigning material to a body

A part can consist of several bodies and, hence, of different materials. If no material is assigned to a body, the body inherits the material (master) assigned to the part.

To assign material to a body, click on the body in the *Materials in model* segment before selecting a material.

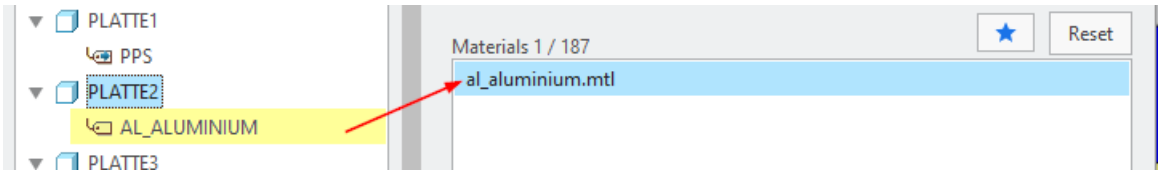



*Material of a part consisting of three bodies*


© 2026 INNEO Solutions GmbH

## Assigning an existing material

If you want to assign a material, which already exists in the part, you can click the desired material in the area *Materials in model* on the left. This will display the material in the material list on the right, so that you do not have to search for the material in the material list.




## Assigning material in Creo Material dialog

You may also assign material in Creo's Material dialog, which opens when clicking the icon on the top right corner or in Creo model tree *(*Right-click on part or body > *Edit Materials*).

## 13.1.5 Deleting material

To delete a material that is in the model, open the Creo Material dialog via the icon at the top right. Select the material to be deleted and delete it by Right-click > *Delete*. This is a default Creo function that checks if there are dependencies in a model. If there are dependencies, the material cannot be deleted.

Delete the material assigned to the model (active material, master), marked with the icon , Creo automatically sets the system material[395] as master.

**Please note:** *Material Selector* will be closed when opening the Creo material dialog.

The notification *No material found* appears with the icon , if no material has been explicitly assigned to the model and the material check[408] is activated.

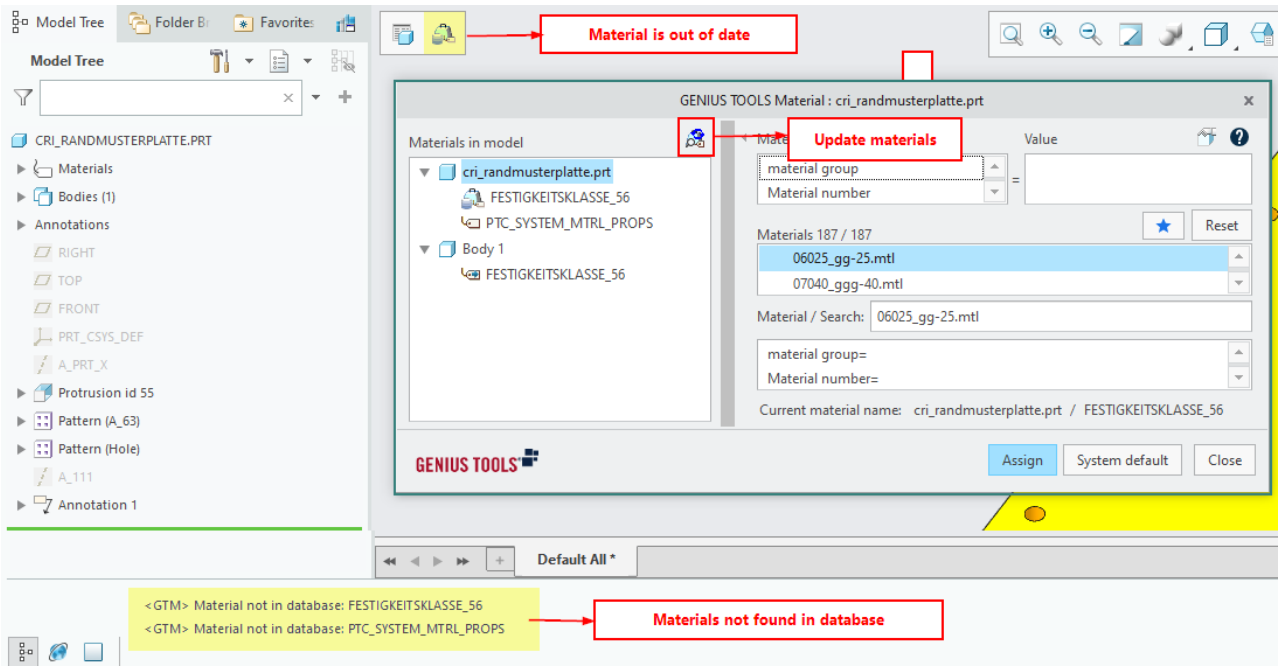
**Warning:** The `gtm_delete_not_current_materials=1` configuration option (default: 0) deletes all unused materials from a model - regardless of valid relations, family tables, bodies or other dependencies. A configuration option set to this value does not check whether such dependencies exist. Deleted materials cannot be restored!
However, it is not recommended to use this configuration option. Make sure to use this configuration option only if you are sure that there are no dependencies in your model.


© 2026 INNEO Solutions GmbH

## 13.1.6 Updating materials

If a model contains obsolete materials, the *Update-materials* button will appear in the area *Materials in model*. This function searches the material database for MTL files of the same name with a newer revision.

1. Click the button to automatically assign these material files.
2. Check if the update was successful: if the assigned material was not found in the database, this will be visible in the main Creo window by the icon as well as in the Creo message bar.




*Materials cannot be updated if they are not in the database.*

## Excluding a material from the update

If you want to exclude a material from the update function, list the material name without file extension in the configuration option `gtm_exclude_material_from_update_all`. This can be important for materials with variable density or for free materials.

## 13.1.7 Checking materials in a model

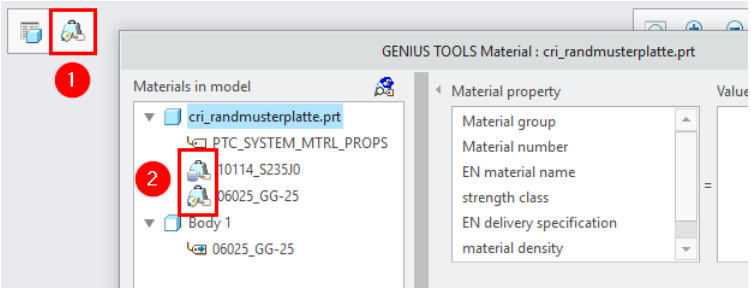
*Material* enables automatic checking of materials in models. This includes whether the material is up to date, whether it corresponds with the information in the database and whether a material has been assigned to a part / body.

The chapter Activating material check 408 describes the configuration of this function.


© 2026 INNEO Solutions GmbH

# Results of a material check

If material check is activated and the checked material versions diverge, an icon will be displayed in the Creo Parametric main window on the left (1). Also, the icon will be displayed next to the material in the *Material* Selector dialog in the left area *Materials in model* (2).




*Icons used to indicate material check*

The results are calculated for all materials in the model and, depending on the result, a corresponding warning symbol is generated. These can be switched off with separate configuration options, see Activating material check 409.

If a model contains several materials, the highest of the following results of the material check is displayed.

| Icon | Message | Meaning |
|---|---|---|
| No icon | – | No error |
|  | Outdated material version | The revision parameter of the material in the model is not the same as in the database. It is not detected that the revision in the model is older, only that it differs. |
|  | Several materials found with the same name (e. g. Subdir) | Several files found in the database, e. g. in subdirectories. |
|  | Material cannot be found in database | |
|  | Material has no revision parameter | |


© 2026 INNEO Solutions GmbH

| Icon | Message | Meaning |
|---|---|---|
|  | Material is Creo system material | This error message can be disabled by the administrator, see Activating material check[408]. (Each part / body that is not assigned a material is assigned the system material[395] PTC_SYSTEM_MTRL_PROPS. The system material can be changed but not deleted). |

## Option: Checking only the current material

If you want to receive a material check message only for the current material in a model in the Creo main window, set the configuration option `gtu_ui_change_check_material_check_only_current_material` to 1.

**Please note:** This configuration changes the display of the symbols in the Creo main window, not in the *Material Selector* window. When opening the *Material Selector* dialog, all materials in the model are always checked.

**Please note:** Do not activate this configuration option if a message for outdated material in a body should be displayed in the main Creo window.

# 13.2 Configuration

In this section you find information on configuring *Material* with the administrative component *Material Selector Editor* to configure the automatic material check and to update material files.

## 13.2.1 Material Selector Editor

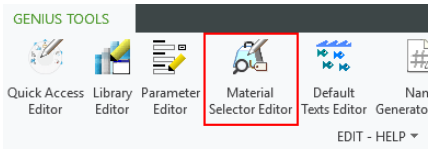
With *Material Selector Editor*, you can manage the number and the display of materials and material attributes in the dialog box of *Material Selector*.

**Please note:** *Material Selector Editor* does not edit material files.

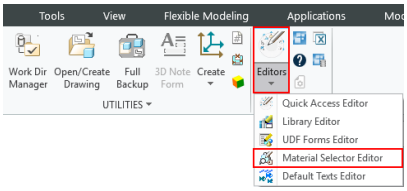

© 2026 INNEO Solutions GmbH

## 13.2.1.1 Starting the program

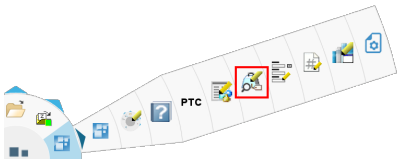
*Material Selector Editor* can be started in the following ways:




*In Creo standby mode*




*In part mode via the ribbon menu GENIUS TOOLS in the section Editors*




*Via Quick Access (key [<])*

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

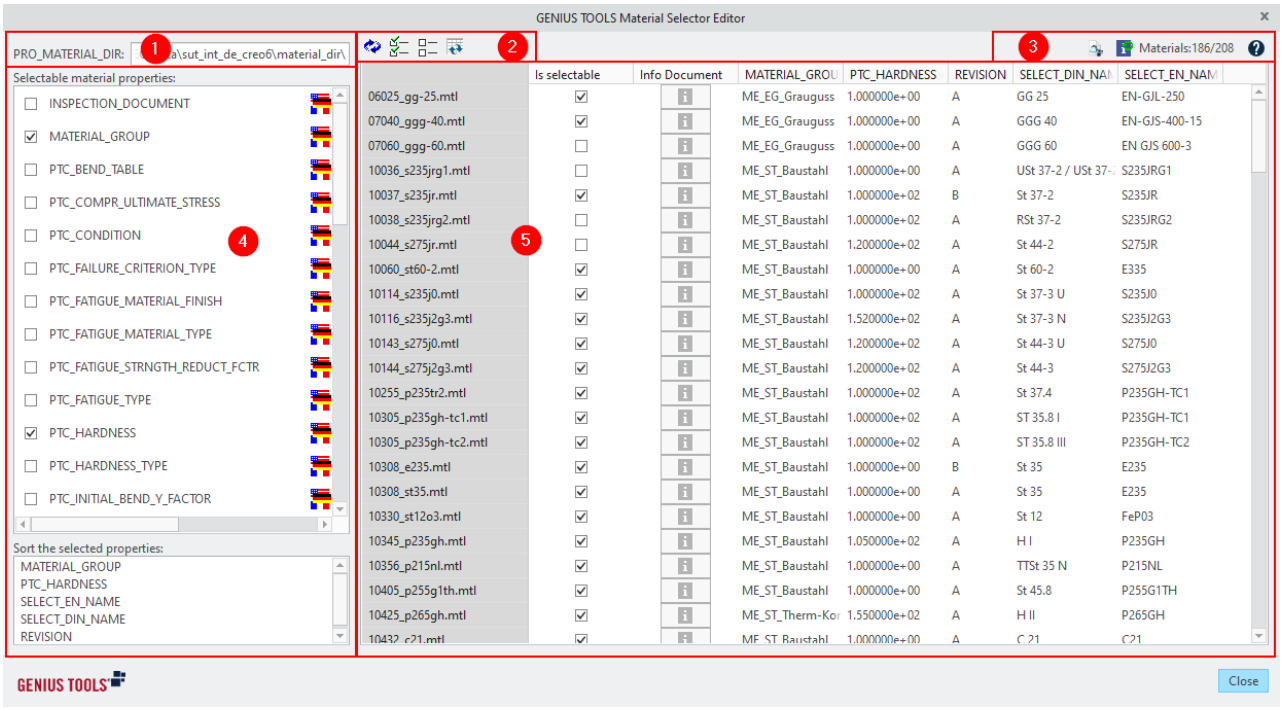
To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.


© 2026 INNEO Solutions GmbH

## 13.2.1.2 User interface

The user interface of *Material Selector Editor* consists of the following elements:




1. Material directory (`pro_material_dir`)
2. Buttons for reloading material files, de-/selecting materials and updating database
3. Additional functions: Material Browser[410], Info documents[406], Number of material files[405], Help
4. Selectable properties and display order
5. List of all materials (MTL-files from the material directory)

## Warning message when opening the editor

If the number of material files in the material directory is less than in the material database, an error message is displayed. Update the material database by clicking on *Check database attributes/ values*  . (See also Updating complete database: Integritycheck[405])

## 13.2.1.3 Material directory

*Material Selector Editor* displays all material files available from the material directory, including sub-directories, after starting. The current directory is displayed on the left of the dialog box.

You can edit the path of the material directory by setting the configuration option `pro_material_dir`.

All files in the material directory are displayed in *Material Selector Editor*.

### 13.2.1.4 Defining selectable materials

In *Material Selector Editor* you define the list of materials, which is available to users as well as the displayed properties. These settings are transferred to the material database and update the database in the corresponding locations. For a complete update of the database - which compares all contents of the database with the contents of the selected materials in the Editor - an integrity check is required, which is started by clicking on the command *Update database* .

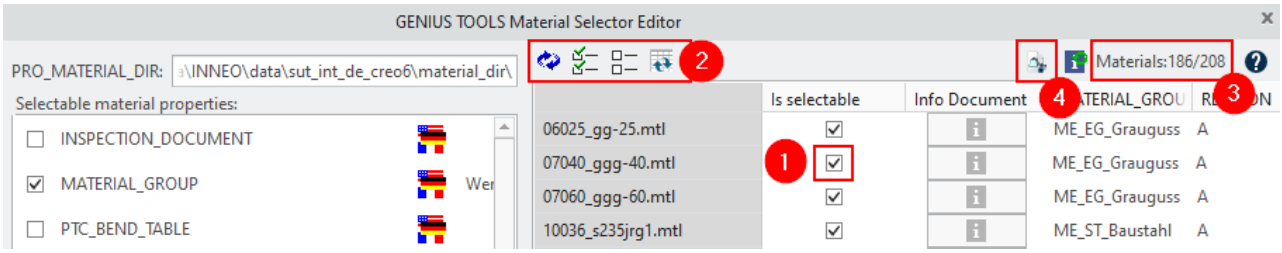
**Please note:** All changes in the editor immediately affect the material database, except for the command *Reload material files* . Use this command only if the material files were changed externally while *Material Selection Editor* was open.

## Selecting material from material list

The segment on the right displays all material files of the material directory (`pro_material_dir`) including sub-directories.

Check the box *Is selectable* (1) to make the material selectable in *Material* . If the box is unchecked, the material cannot be selected. If the box is checked, the selected properties of this file will be read into the material database.

By clicking the buttons *Select all materials* or *Select no material* (2) you can quickly check and uncheck the complete material list.




Materials are displayed with the properties selected in the section *Selectable material properties* on the left.

## Updating material files vs. updating the database (2)

If you make changes to MTL files while working in *Material Selector Editor* (e. g. via *Material Browser*), they will only become visible in *Material Selector Editor* after clicking *Reload material files in Material Selector Editor* . The database must then be updated.


© 2026 INNEO Solutions GmbH

| | |
|---|---|
| *Reload material files in Material Selector Editor* | – Uploads changes in MTL files and new MTL files into *Material Selector Editor*<br>– Equivalent to restarting *Material Selector Editor*.<br>– Does not load content from MTL files into the material database. |
| *Update database (integrity check)* | – Re-reads the material properties and selected MTL files selected in the *Material Editor* into the material database and checks all entries with an integrity check.<br>· Removes MTL files that no longer exist (files that exist in the database but no longer in the material directory).<br>· Copies changes in the MTL files to the database (if these are changes to the selected material properties).<br>If you add material files, reload *Material Selector Editor* with . |

**Please note:** Update the database if material files were deleted or changed. Update the Editor first if material files were added. If material files were edited externally (e. g. with Material Browser[410]) while the Editor was open, update the Editor first .

## Number of displayed material files (3)

The top-right corner (3) displays the total number of materials in the database and the total number of materials in the material directory. If the first number is higher than the second, it means that the database is not up-to-date.

## Editing material files (4)

With the button (4) you can edit the material files directly in the material directory if you work with *Material Browser*[410].

The path to the executable file *GT_Material_Browser.exe* is set in the configuration option `gtm_editor_material_browser_path`.

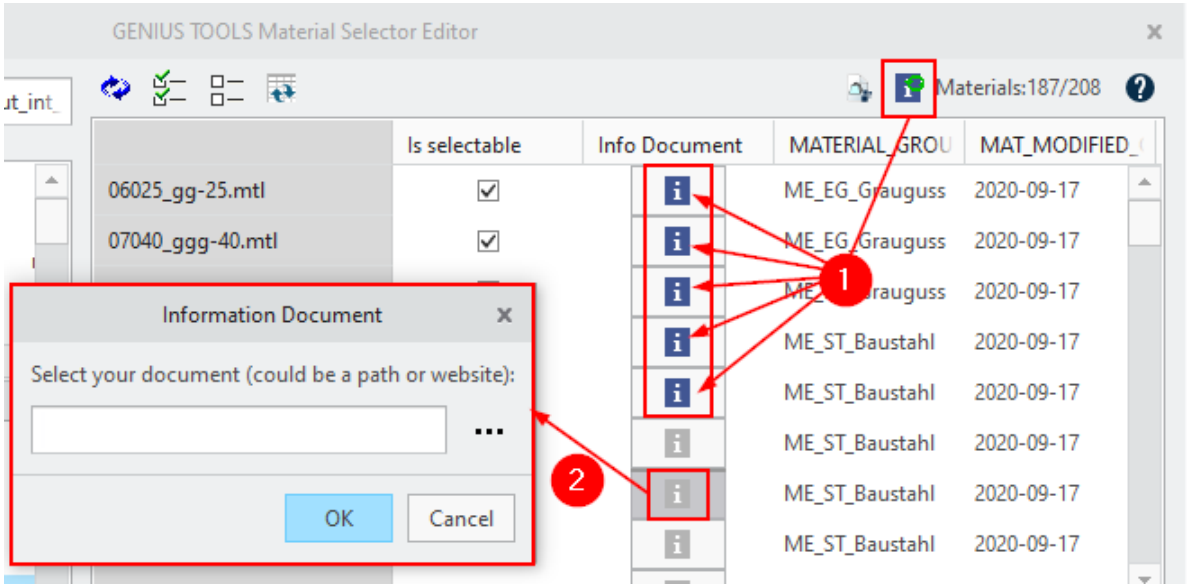
If you are working with the Windchill PDM system, see also Editing material files in Windchill[411].


© 2026 INNEO Solutions GmbH

## Adding info documents

You can add documents to individual materials. The user can access these info documents in the *Material Selector* user interface.

You can upload all available documents automatically (1) or upload a document for a single material (2).




*Providing documents for materials*

(1) Use the button to activate the automatic upload of all documents whose name corresponds to the name of any material file. This function searches for documents of all file types in a specific folder. The path to this folder is set by the configuration option `gtm_infoDoc_folder`. Default: `%gt_resource%\material\info`

(2) To upload info documents individually, click the info icon next to the material and enter a file or web address in the following dialog. To open websites and network documents, you must specify the appropriate protocol (http://, https://, ftp:// etc.).

## 13.2.1.5 Display settings of Material Selector dialog

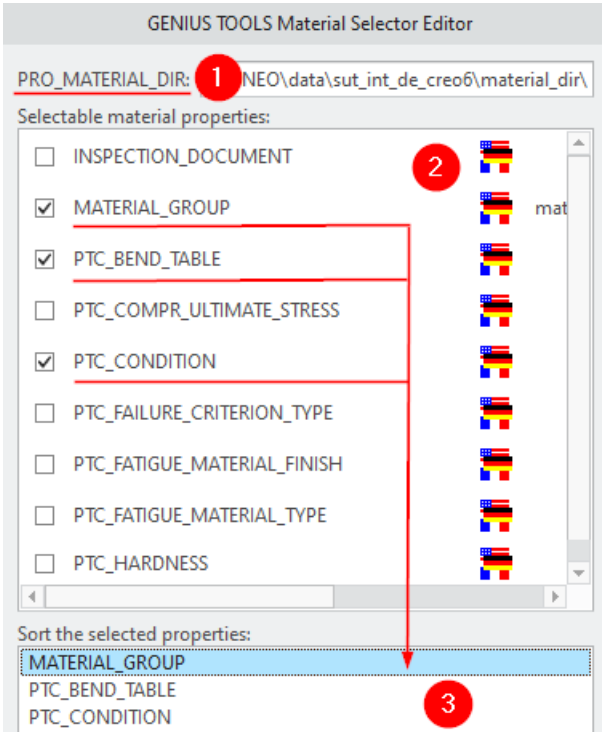
In the following sections you define which material properties are displayed to the user in *Material Selector* as well as the language settings of the material properties.

## Specifying material attributes and arranging their display order

The material attributes which should be displayed to the user are listed in the segment *Selectable material properties* (2). Activate the checkboxes next to the required material properties.


© 2026 INNEO Solutions GmbH

Material properties are read from the parameters in an MTL-file and can be Creo parameters (e. g. PTC_BEND_TABLE) as well as company-specific, manually entered parameters (e. g. SELECT_EN_NAME).




*Selection of properties MATERIAL_GROUP, PTC_BEND_TABLE und PTC_CONDITION*

**Please note**: The material database is rewritten, as soon as a material property is activated. With numerous selected materials, this action may take some time.

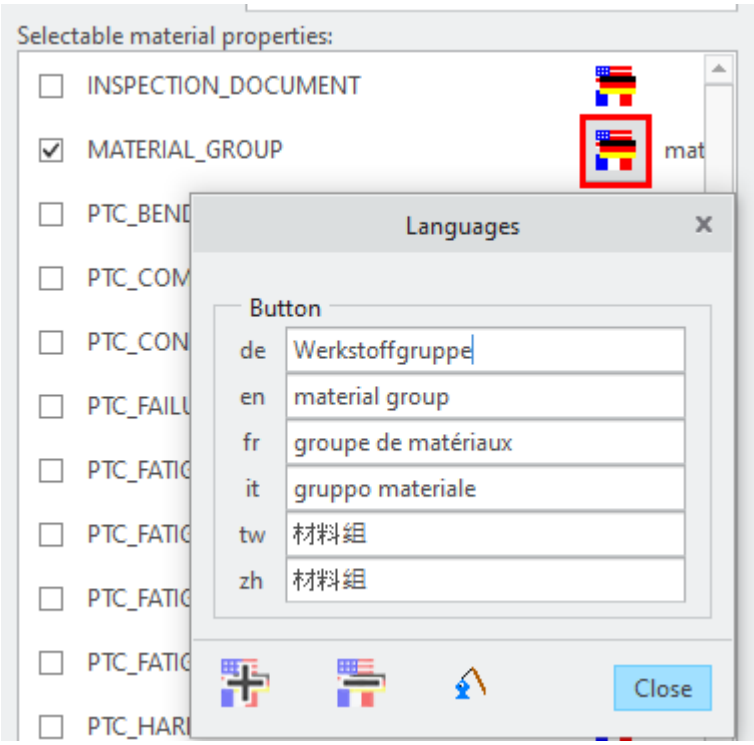
The display order of the selected attributes is displayed below under *Sort the selected properties* (3), i. e. the order in which the properties appear in *Material Selector*. Drag and drop the properties to the required position.

## Translating material properties

Material properties can be localized, i. e. displayed in a specific language. Click the flag icon behind the property terms to open the localization dialog.

You can add or delete languages by clicking on the + or - icon in the localization dialog. With the button you can choose standard text, see Set standard text 678 ..


© 2026 INNEO Solutions GmbH




If no localization exists, the name of the material property (i. e. the parameter) used in the material files is used.

The set language (via configuration option `gtm_db_lang`) determines which localization is used in *Material Selector*.

## 13.2.2 Activating material check

*Material* allows you to automatically check materials in models. The result of the check is displayed as an icon in the top left corner of the Creo Parametric main window and next to the assigned materials in the Material Selector dialog. For a list of possible check results, refer to the chapter Checking materials in a model 399.

**Please note:** Materials in a model are not replaced by this function.

Enable the material check as follows:

1. To display the icon, set the configuration option `gtu_ui_change_check_material_version` to `1` (Default: 0).
2. Create a material parameter in Creo Parametric. Make sure the parameter is of type String.
   For the material check, you need a material parameter that stores the revision of the deposited material. When opening parts, the parameter is read from the material in the model, compared with the material parameter in the material database, and the appropriate notification is issued. The material parameter is controlled by the configuration option `gtu_ui_change_check_material_version_parameter` (Default:

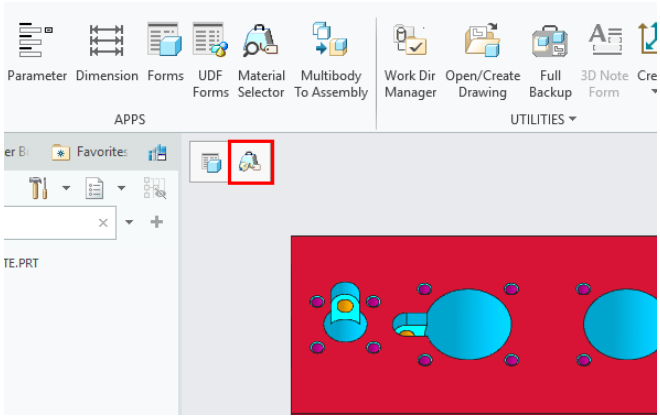

© 2026 INNEO Solutions GmbH

REVISION). Make sure that the material parameter in the configuration option has the same name as in the Creo parameter menu. If this is not the case, the configuration option needs to be adjusted.

a. Maintain this parameter in your materials - either in the individual MTL files in the material directory, via the Creo Material Editor or by using *Material Browser*[410].

3. Save[775] and reload[777] the configuration options.

4. The material check icon appears in the Creo Parametric main window.




## Option: Disabling display of notification "Material is Creo system material"

The error message  indicating that the Creo system material is assigned to a part can be turned off by setting the configuration option `gtu_ui_change_check_material_system_material_is_wrong` to 1.

## Option: Checking only the current material

If you want to receive a material check message only for the current material in a model in the Creo main window, set the configuration option `gtu_ui_change_check_material_check_only_current_material` to 1.

**Please note:** This configuration changes the display of the symbols in the Creo main window, not in the *Material Selector* window. When opening the *Material Selector* dialog, all materials in the model are always checked.

**Please note:** Do not activate this configuration option if a message for outdated material in a body should be displayed in the main Creo window.

## Switching off options for the material check

The following configuration options check materials and generate the according warning icons. The default settings are 1 (on).


© 2026 INNEO Solutions GmbH

| Icon | Message | Meaning, configuration option |
|---|---|---|
| | Outdated material version | The revision parameter of the material in the model is not the same as in the database. It is not detected that the revision in the model is older, only that it differs.<br>`gtu_ui_change_check_material_check_old_material` |
| | Several materials found with the same name (e. g. Subdir) | Several files found in the database, e. g. in subdirectories.<br>`gtu_ui_change_check_material_check_multiple_materials` |
| | Material cannot be found in database | `gtu_ui_change_check_material_check_material_not_in_db` |
| | Material has no revision parameter | `gtu_ui_change_check_material_check_material_has_no_param` |

## 13.2.3 Editing material files

Material files (MTL files) can be located in a material directory locally or on a server or in Windchill.

Without Windchill you can edit material files in two ways:

– Write new material properties as parameters manually into an MTL file, or

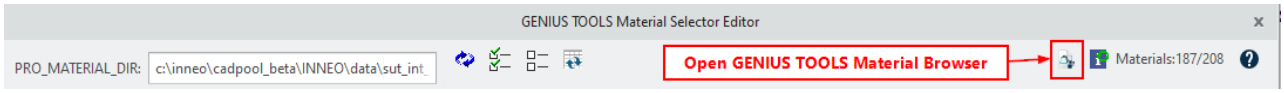
– Edit larger quantities of MTL files with the separate freeware *Material Browser*.

You can open *Material Browser* directly from *Material Selector Editor*, see Edit material files with *Material Browser* 410.

Information on editing material files in Windchill are found under Edit material files in Windchill 411.

### 13.2.3.1 Editing material files simultaneously

You can edit material files (MTL files) directly in the *Material Selector Editor*. The button  opens the separate freeware *Material Browser* and displays all files in the material directory.

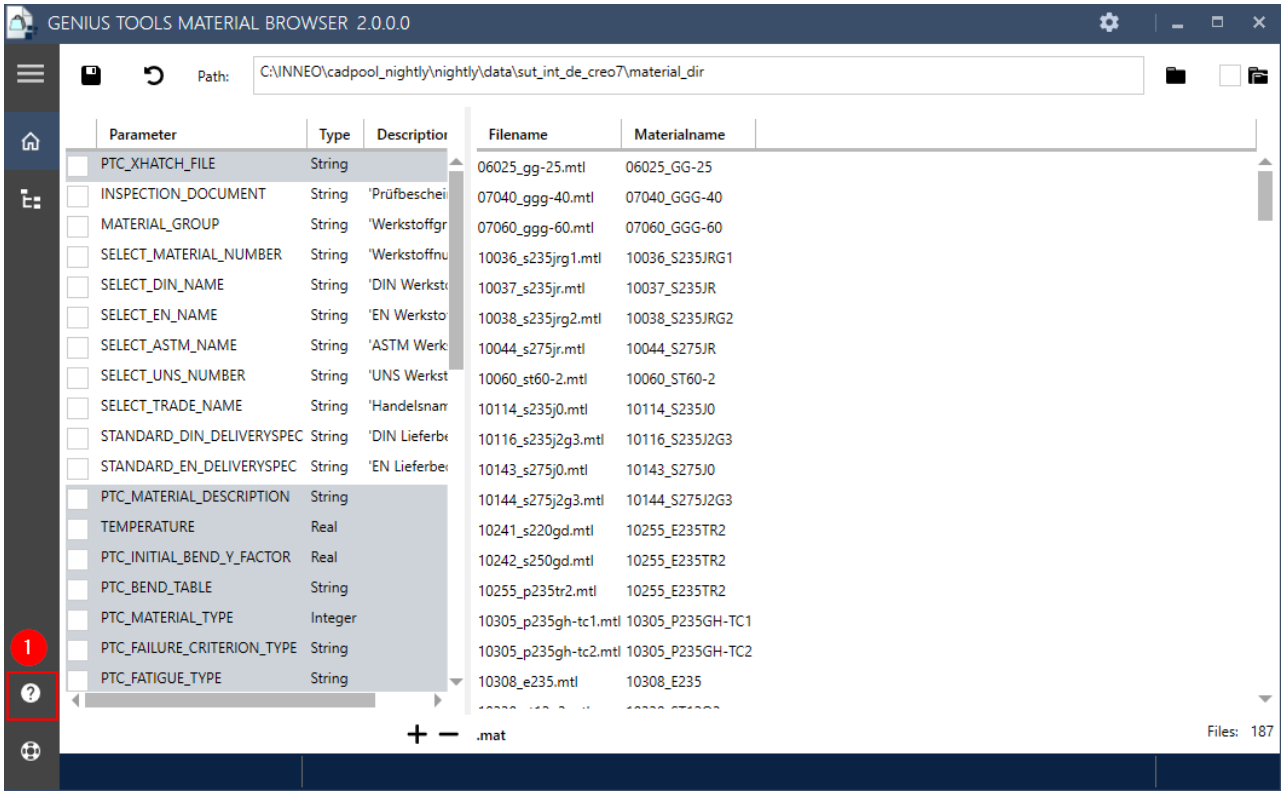




© 2026 INNEO Solutions GmbH

The button  is displayed if *Material Browser* is installed and the path to the executable file *GT_Material_Browser.exe* is set correctly in the configuration option `gtm_editor_material_browser_path`. *Material Browser* opens the material directory with which you work in the current session.

Information about the *Material Browser* can be found in the program via the help icon (1). The application can be downloaded free of charge from the download area of the Inneo website. *(https://www.inneo.co.uk/en/services/technical-support/genius-tools-downloads.html)*




## 13.2.3.2 Editing material files in Windchill

If you use a material directory which is located in Windchill you can edit material files the following way.

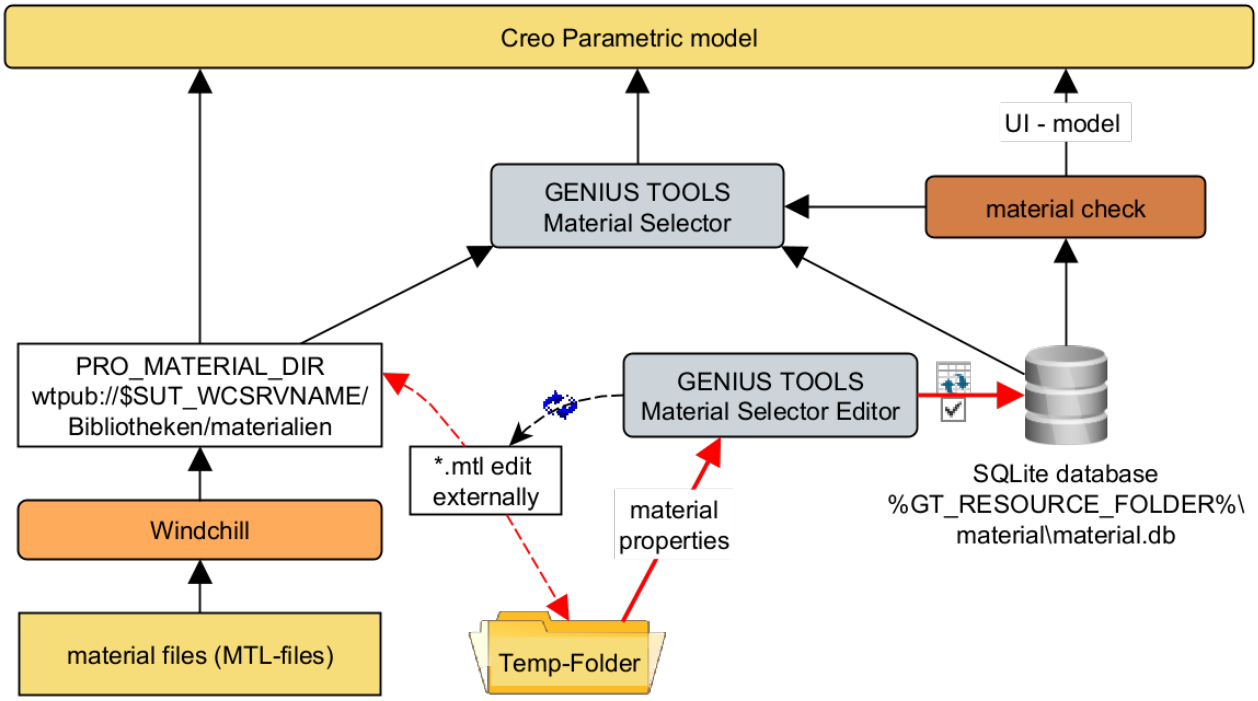
**Prerequisites:**

– an active Windchill server

– Windchill is connected

– the path to the material directory specified by the configuration option `pro_material_dir` is set to a windchill directory, e. g. *wtpub://$SUT_WCSRVNAME/Bibliotheken/materialien*

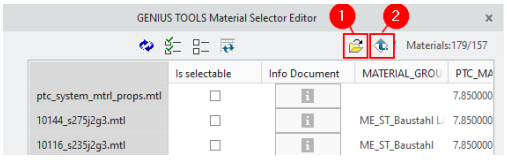

© 2026 INNEO Solutions GmbH




**Procedure:**

**Warning:** Keep *Material Selector Editor* open during the whole process.

1. Open *Material Editor*.
   This will load all MTL-Files of the material directory from Windchill in a temporary directory *Temp folder*. The structure of the material directory is preserved, i. e. any existing subfolders are loaded as well.
2. Open the Temp Folder by clicking *Open temporary folder* (1)




3. In Temp folder edit the MTL files (manually or with *Material Browser*).
4. Return to the open *Material Editor* and upload the edited material files into the workspace by clicking on *Upload edited files* (2)
5. Check in the edited material files to the workspace.
6. Re-read the material files in *Material Selector Editor* by clicking *Reload material files*.
7. Update the material database in *Material Editor* by clicking on *Update database (Integrity check)*[405].


© 2026 INNEO Solutions GmbH

# 14 MBD Info

Available in the ribbon menu GT MBD

The module *MBD Info* is an enhancement of the module *Annotation Info*[642].

The module is used to display information about all types of annotation elements in combined views. This module lists non-visible information to help you find errors and redundancies. Annotation elements can be displayed in parts, assemblies and in multi-body models. It is an information module only and cannot be used to edit annotations.

*MBD Info* is available in part mode and assembly mode with these features:

1. Viewing occurrence and frequency of annotations in a clear and concise table format
2. Finding annotations and thus dependencies that are not visible in any combined state

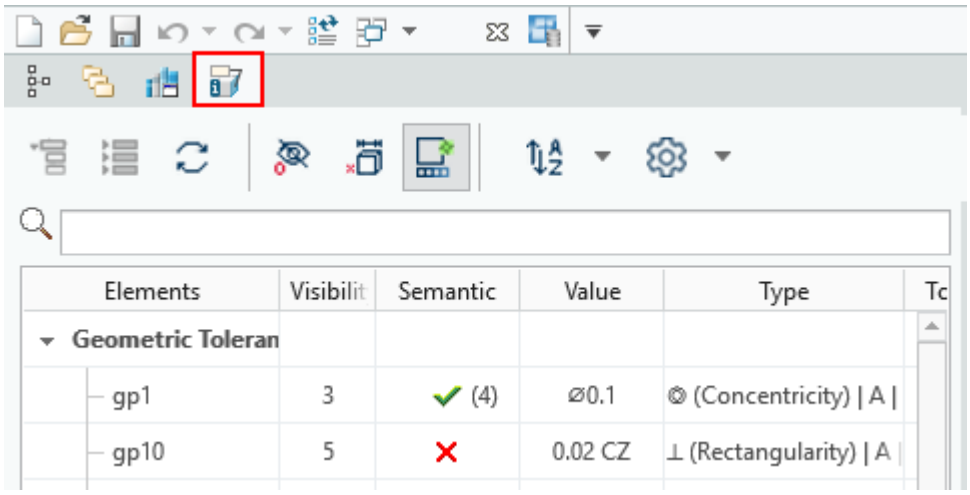
Use the following modules to further process the information obtained with *Annotation Info*:

– to manage combined views, see *Function Manager*[167]

– to transfer annotations, see *Annotation Transfer*[649]

## Starting the program: in part and assembly mode

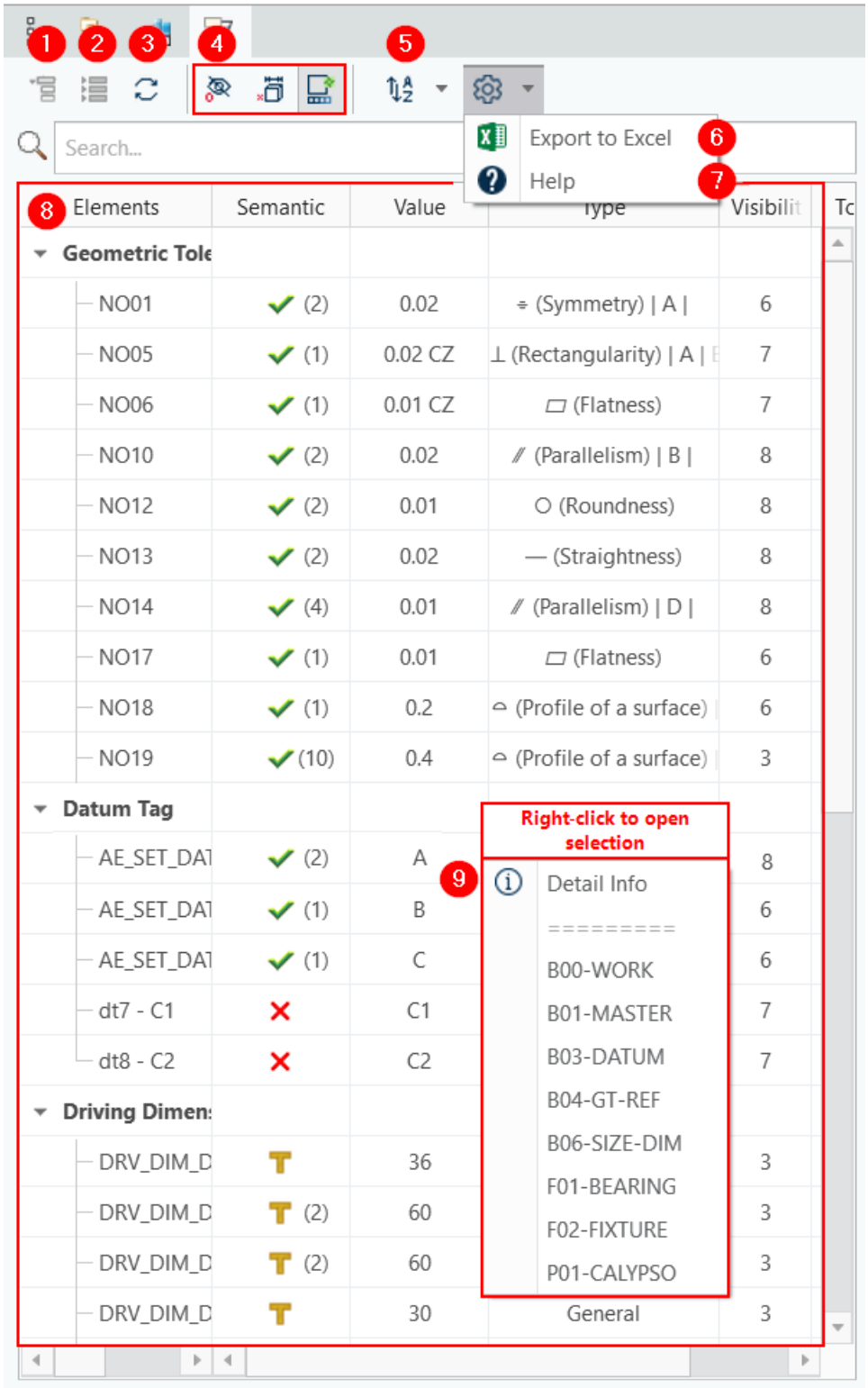
Start *MBD Info* in the navigation area.




## User interface

When opening the navigation area, you see the following user interface:


© 2026 INNEO Solutions GmbH




1. Expanding the table
2. Collapsing the table
3. Update information
4. **Filter**

   *Show only invisible annotations*


© 2026 INNEO Solutions GmbH

Existing annotation elements that are not part of a combined view are highlighted in red. Clicking on such an annotation element displays its position in the model.

 *Show only annotations without semantics*

 *Show only annotations from the active combined view*

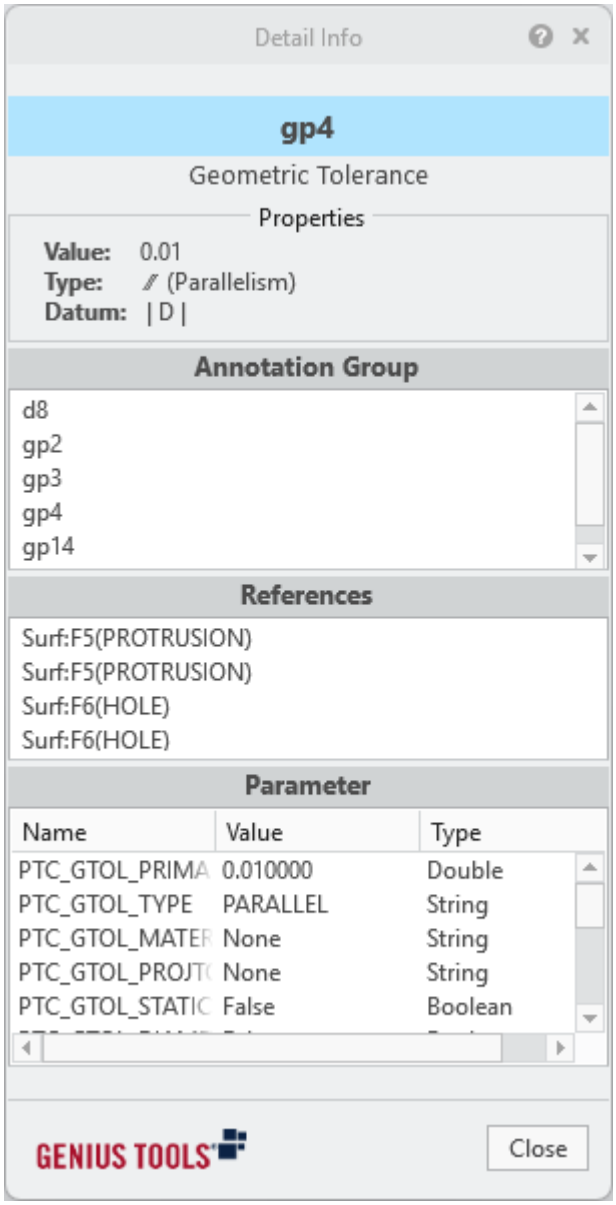
5. **Set the column by which the table is sorted in ascending order**
   By default, the table is sorted by the column *Elements*.
6. Exporting the table, see *Export Table to Excel* 689

**Please note:** The name of the export template is stored in the configuration option `gtmbdinfo_excel_template`. Further settings can be made using the configuration options that begin with the following prefix: `gtu_table_to_excel_`


© 2026 INNEO Solutions GmbH

7. Open the help
8. **Tabular listing of all annotation elements in a model (across all combined views):**
   *Types of annotation elements:*
   - *Geometric Tolerance*
   - *Datum Tag*
   - *Driving Dimension*
   - *Driven Dimension*
   - *Surface Finish*
   - *Symbol*
   - *Note*
   - *Table*
9. **Displaying combined views for an annotation and detail info**
   Left-click in a row to highlight the corresponding annotation element. Right-clicking an annotation opens a sub-menu showing all the combined views in which the annotation is present. Left-click on a combined view to view the model in that combined view. In addition, the *detail info* option includes a summary of the most important properties, as well as information on annotation group, references, and parameters.


© 2026 INNEO Solutions GmbH




# Displaying annotations

By default, the table is sorted by the column *Elements*. The following columns are displayed:

| Column | Remark | Table can be sorted by this column |
|---|---|---|
| Elements | The display is structured according to the following scheme:<br>`Annotation element type (number of these annotation elements in combined views \| number of these annotation elements not present in any combined view)` | ✓ |


© 2026 INNEO Solutions GmbH

| Column | Remark | Table can be sorted by this column |
|---|---|---|
| Visibility | 0 = invisible<br>≥ = visible | ✗ |
| Semantic | Indicates if the annotation element is linked. If there is no link, this may cause problems in subsequent processes based on that annotation element. If the link is missing, this may cause problems in subsequent processes that rely on this annotation element. A number in parentheses appears after the icons, indicating the number of references.<br>✓ = semantic link exists<br>✗ = semantic link does not exist<br>T = TED dimension that is not semantically linked | ✗ |
| Value | | ✓ |
| Type | | ✓ |
| Tolerance | | ✓ |
| Comb View Amount | Combined view containing the annotation element | (✓: table filtered by number of combined views) |

# Displaying automatically generated hole notes

The configuration option `gtmbdinfo_show_hole_notes=1` allows you to display automatically generated hole notes by Creo Parametric. Default: 0=off.

# Displaying automatically generated hole notes

The configuration option `gtmbdinfo_mandatory_ref_number_for_linear_dimensions` allows you to define the minimum number of semantic references a linear dimension must have in order to be marked as semantically correct in the *MBD info*. Default: 2

This applies to:


© 2026 INNEO Solutions GmbH

Driving dimensions

Driven dimensions

Reference dimensions

## Filter for symbol definition

The configuration option `gtmbdinfo_hidden_symbol_definitions_file_path` to specify the folder path to the text file containing the symbol definitions. Symbols with these definitions are ignored by *MBD Info* and are not included in the list.

## Configuration of the ribbon menu

Use the configuration option `gt_start_mbdinfo` to switch off the function, so no button is displayed in the *GT MBD* ribbon menu. (Default is 1=On/not hidden)

# 14.1 Excel to Export

## Exporting annotation elements to Excel

To export annotation elements to Excel, click on the command *Export to Excel* . The dialog Export table to Excel[690] opens, in which you select the Excel template, export file and configuration.




Starting the Excel export can take some time because the Excel template has to be checked. Once the export is complete, the Excel file will open automatically.

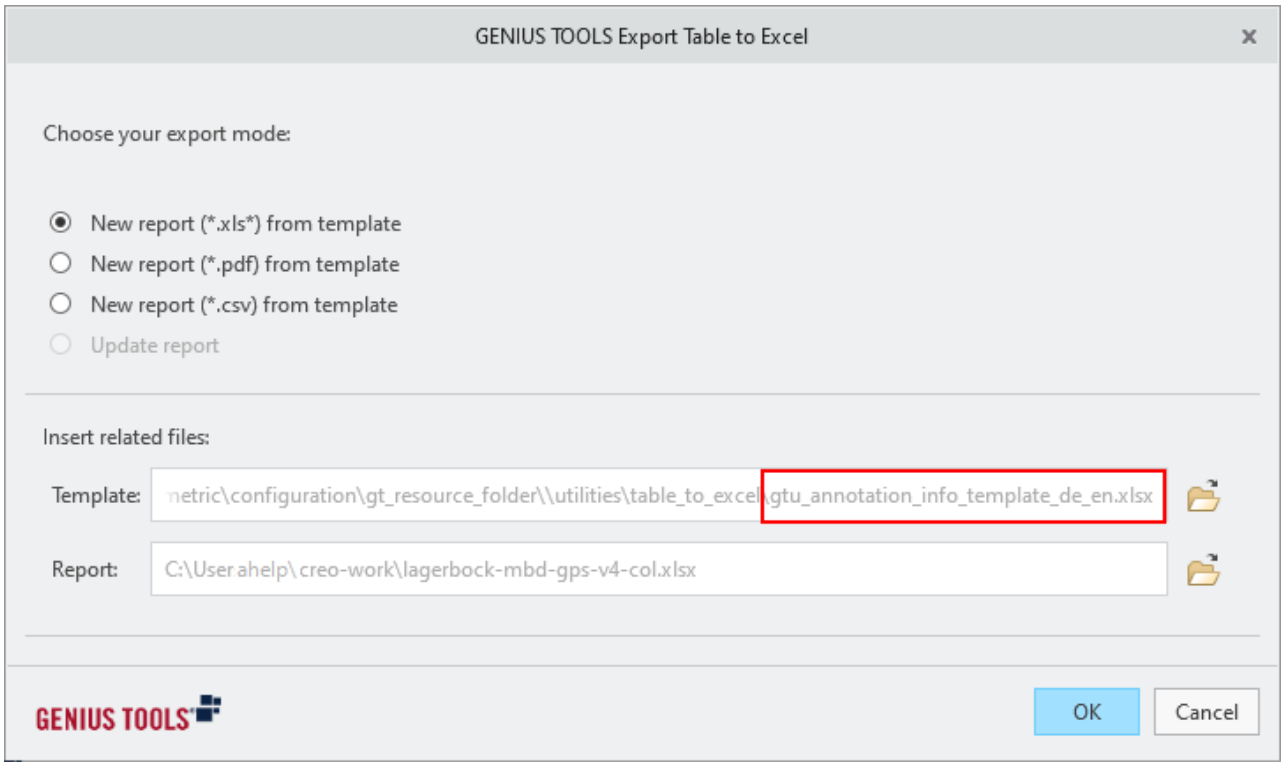
**Warning:** Close all Excel windows before starting the export.

## Export dialog

By default, the out-of-the-box template is selected. The file name is generated from the name of the model. You can also design an individual Excel template[420].

The selected configuration determines which annotations are exported. Annotations will be exported if the selected configuration contains the annotation.


© 2026 INNEO Solutions GmbH




# Configuring path and name of the template

The selection of the template and the name of the template are determined by these configuration options:

**gtu_annotation_info_excel_template**
Defines the name of the basic Excel template. Default: *gtu_annotation_info_template_de_en.xlsx*

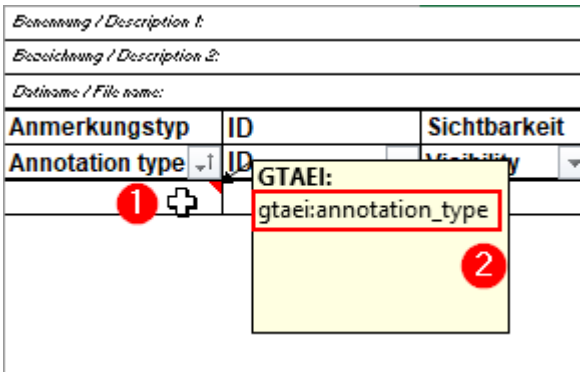
**gtu_table_to_excel_template_folder**
Defines the path to the template directory for "Table to Excel" (.xlsx-files). Default: *%gt_resource_folder%\utilities\table_to_excel*

# Customizing the Excel template

If you want to create your own template, you can use the supplied template *gtu_annotation_info_template_de_en.xlsx* from the directory *%gt_resource_folder%\utilities\table_to_excel* as a basis and add a comment (2) to the first value cell (1) of the desired column. The chapter Create template[693] describes how to set up an export template step by step.

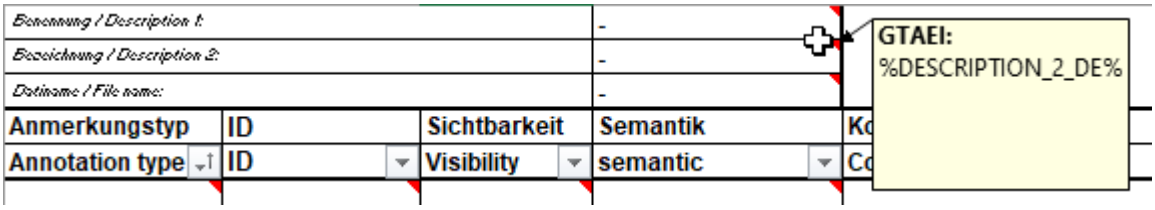
**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.


© 2026 INNEO Solutions GmbH




*Comment used for column assignment*

The text of the comment determines which values are to be written into the column. Head parameters allow variables without module abbreviations, e. g. *%DESCRIPTION_1_DE%*.




*Head parameter forAnnotation Info*

The text in the comment must consist of the component abbreviation gtaei: and a keyword:

| Comment text | Column name |
|---|---|
| gtaei:<columnName> | Output of additional user-defined parameters |
| gtaei:annotation_type | Annotation type |
| gtaei:combined_views | Combined views |
| gtaei:semantic | Semantic |
| gtaei:tolerance_value | Tolerance value |
| gtaei:type | Type |
| gtaei:value | Value |
| gtaei:visibility | Visibility |
| gtaei:comment | Comment |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gtaei*. This allows you, for example, to export data from *MBD Info* and *Inspect 3D* together, see Export data from several GENIUS-TOOLS-components[701] for an example.


© 2026 INNEO Solutions GmbH

# 15 MBD Tables

Available in the ribbon menu GT MBD

With the module *MBD Tables*, you can create and manage tables for parts. The module is divided into submodules that allow you to create different types of tables: Size Table[422] and MBD Surface Finish Table[429].

**Please note:** The module can only be used from Creo version 11.

## 15.1 Size Table

With the submodule *Size Table*, you can place and manage size tables for parts.

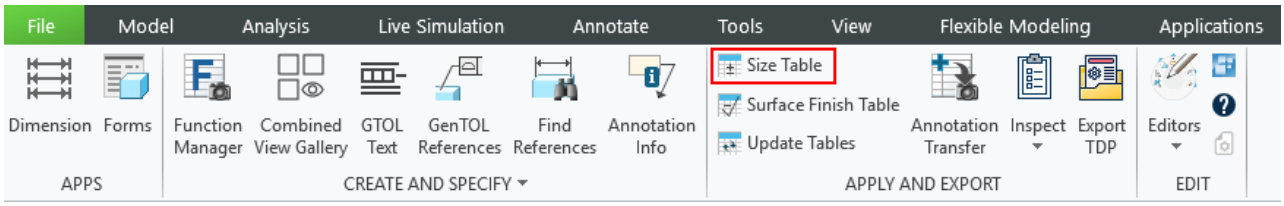
**Please note:** The submodule can only be used from Creo version 11.

### 15.1.1 Usage

In this section you will find information about the use of *Size Table*. The general structure of the program is explained.

#### Starting the program: in part mode

To start *Size Table*, click *GT MBD* in the ribbon menu and go to *APPLY AND EXPORT* section.




*Starting via the GT MBD in the ribbon menu*

#### Functionality

All dimensions in the model are displayed with their tolerances in a table. The following dimensions are ignored:

– ISO tolerance table to General

– ISO tolerance table to None and tolerance to nominal value

– TED dimensions

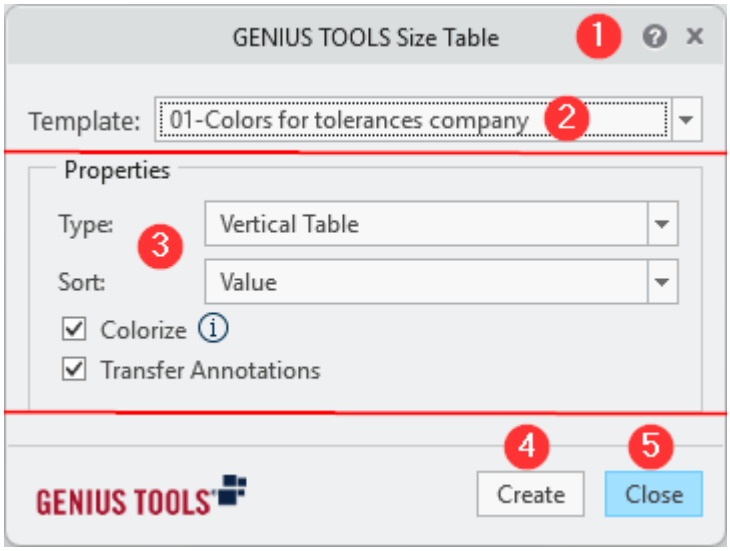

© 2026 INNEO Solutions GmbH

## Configuration

`gtmtab_display_all_size_dimensions` defines whether the Size Table includes all toleranced dimensions (1), or just the fit dimensions (0). Default setting: 1

### 15.1.1.1 User interface

The user interface of *Size Table* consists of the following elements:




1. Help
   Opens this help.
2. Template
   Select a saved XML template[425].
3. Settings[424]
4. Create
   A table is created with the entered settings, see User interface[424]. If a table already exists, the command is called *Update*. In this case, no new table is created, but the existing one is updated. You can also use the Update function directly from the ribbon menu *GT MBD*, see MBD Update Tables[439]. After creating or updating a table, the user interface is automatically closed.

**Please note:** Tables are automatically placed at the zero point in the model. To move the table, mark the table by clicking the left mouse button and then click on the move handle. The table is attached to the mouse and can be moved to any location in the model.

5. Close
   The user interface is closed without a table being created. Previously changed settings are not saved.


© 2026 INNEO Solutions GmbH

## Result

You will see a table in which the coloring of each size is indicated by an icon in a color column / color row. If you deselect the option *Colorize* in the user interface, the color column / color row will disappear. If you select the option again, the color column / color row will reappear after clicking *Update*.

| | Dimension | Fitsize | µm |
|---|---|---|---|
| X | ⌀12 | 0.500<br>-0.500 | 1000 |
| ◆ | ⌀40H8Ⓔ | 0.039<br>0.000 | 39 |
| - | ⌀50 | 0.200<br>0.100 | 100 |
| ◆ | 100h9Ⓔ | 0.000<br>-0.087 | 87 |

*Example Table*

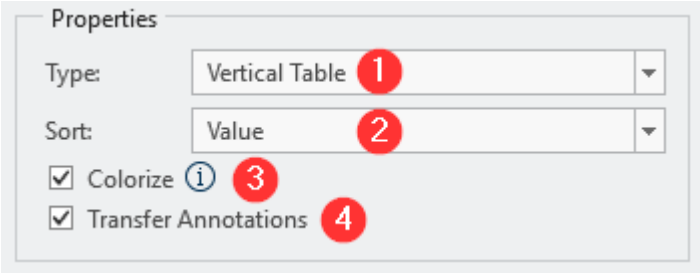
In a table, the following types of icons can appear:

| Icon | Description |
|---|---|
| ◆ (Diamond in the color in which a size was colorized) | Assigned references are colored. |
| X | The value of the size is within the range of values for a defined color, see used templates[427]. However, the size has no semantic references that can be colored. All affected sizes that are found without semantic reference are reported in the Creo message log.<br>If no references are found, either there are no references or the size is obsolete and needs to be updated or converted in the model tree. |
| - | The value of the size is outside the value range for a defined color, see used templates[427]. The size contains references. |

## 15.1.1.2 Formatting tables

In the following area of the user interface, you see the settings for specifying the selected template:


© 2026 INNEO Solutions GmbH




1. **Type**
   Define the alignment of the table with the following options:
   *Horizontal Table*: Fitsize table
   *Vertical Table*: Maximum table

2. **Sort**
   Setting whether the size symbols are displayed in sorted or unsorted order:
   *Value*: Table is sorted by values.
   *Tolerance*: Table is sorted by tolerances.
   *None*: Table is not sorted.

3. **Colorize**
   Overwrites the appearance state in the active combined view.

**Please note:** An info symbol ⓘ appears when this command is activated because it cannot be undone.

4. **Transfer Annotations**
   All annotations present in the model that will be visible in the table are transferred to the combined view in which the table is created.

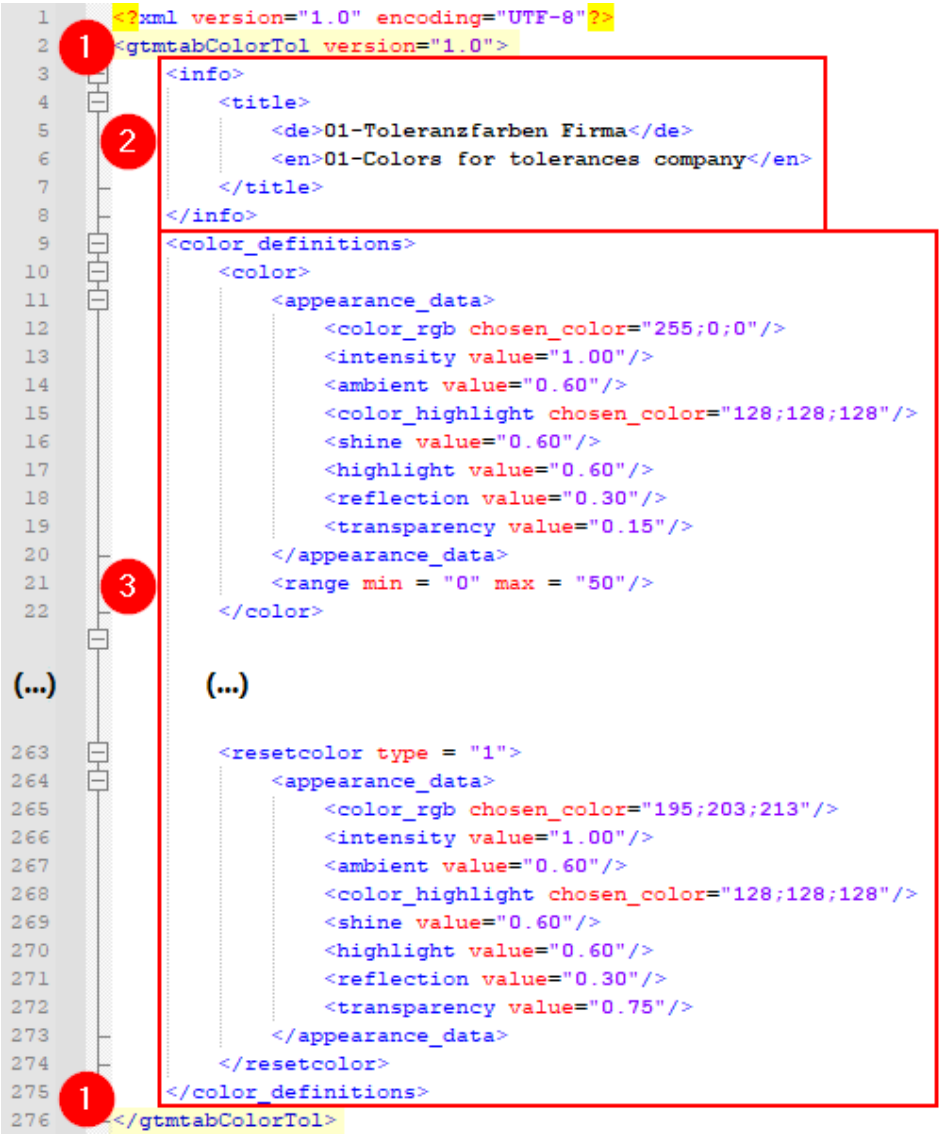
## 15.1.2 Configuration

In this section you will find detailed information about the structure of the module *MBD Tables* and how to create table templates.

### 15.1.2.1 Editing templates for Size Tables

The illustration shows an example of a table template for displaying sizes. This example is used to explain the different sections of a template.


© 2026 INNEO Solutions GmbH



```
<?xml version="1.0" encoding="UTF-8"?>
<gtmtabColorTol version="1.0">
    <info>
        <title>
            <de>01-Toleranzfarben Firma</de>
            <en>01-Colors for tolerances company</en>
        </title>
    </info>
    <color_definitions>
        <color>
            <appearance_data>
                <color_rgb chosen_color="255;0;0"/>
                <intensity value="1.00"/>
                <ambient value="0.60"/>
                <color_highlight chosen_color="128;128;128"/>
                <shine value="0.60"/>
                <highlight value="0.60"/>
                <reflection value="0.30"/>
                <transparency value="0.15"/>
            </appearance_data>
            <range min = "0" max = "50"/>
        </color>

(...)       (...)

        <resetcolor type = "1">
            <appearance_data>
                <color_rgb chosen_color="195;203;213"/>
                <intensity value="1.00"/>
                <ambient value="0.60"/>
                <color_highlight chosen_color="128;128;128"/>
                <shine value="0.60"/>
                <highlight value="0.60"/>
                <reflection value="0.30"/>
                <transparency value="0.75"/>
            </appearance_data>
        </resetcolor>
    </color_definitions>
</gtmtabColorTol>
```

1. `<gtmtabColorTol>...</gtmtabColorTol>`
   Frame for the template: This XML file shows a defined template for a size table. You can define multiple colors for coloring sizes in an XML file.
   - `version`=Specification of a number for versioning the template
2. Editing templates for Size Tables[426]
3. Editing templates for Size Tables[427]

## Information about the template

Each template requires the following tags:


© 2026 INNEO Solutions GmbH

```
<info>
    <title>
        <de>01-Toleranzfarben Firma</de>
        <en>01-Colors for tolerances company</en>
    </title>
</info>
```

4. `<info>... </info>`
   Tags to frame the information section

5. `<title>... </title>`
   Title in English `<en>` and German `<de>`

## Colors for coloring sizes

Colors for coloring sizes are defined with the following properties:

```
<color_definitions>
    <color>
        <appearance_data>
            <color_rgb chosen_color="255;0;0"/>
            <intensity value="1.00"/>
            <ambient value="0.60"/>
            <color_highlight chosen_color="128;128;128"/>
            <shine value="0.60"/>
            <highlight value="0.60"/>
            <reflection value="0.30"/>
            <transparency value="0.15"/>
        </appearance_data>
        <range min = "0" max = "50"/>
    </color>
    (...)
    <resetcolor type = "1">
        <appearance_data>
            <color_rgb chosen_color="195;203;213"/>
            <intensity value="1.00"/>
            <ambient value="0.60"/>
            <color_highlight chosen_color="128;128;128"/>
            <shine value="0.60"/>
            <highlight value="0.60"/>
            <reflection value="0.30"/>
            <transparency value="0.75"/>
        </appearance_data>
    </resetcolor>
</color_definitions>
```

6. `<color_definitions>...</color_definitions>`
   Frame for the part which contains all color definitions.

7. `<color>...</color>`
   Frame for one color definition.


© 2026 INNEO Solutions GmbH

8. `<appearance_data>...</appearance_data>`
   Color definition with the following attributes. Enter the properties of the color definition in quotes:
   - `color_rgb chosen_color`=Color specification in RGB format
   - `intensity value`=Intensity of the color (value between 0.00 and 1.00)
   - `ambient value`=Ambient light (value between 0.00 and 1.00)
   - `color_highlight chosen_color`=Specification of the highlight color in RGB format
   - `shine value`=Shine (value between 0.00 and 1.00)
   - `highlight value`=Highlight color (value between 0.00 and 1.00)
   - `reflection value`=Reflection (value between 0.00 and 1.00)
   - `transparency value`=Transparency (value between 0.00 and 1.00)
9. `<range .../>`
   Value range in which this color is assigned to a size:
   `min`=Minimum (including this value)
   `max`=Maximum (excluding this value, i.e. in this case all values up to $0{,}04\bar{9}$ are colored with this color)

**Please note:** Specify a value range so that the color can be applied. If a color is not assigned to a value range, the color cannot be applied. However, if only one value is not specified, it is automatically assumed to be 0.

10. Repeat steps 6 through 9 for each color you want to use. You can enter as many colors as you like.

**Tip:** For clarity, it is recommended that you insert the colors in the template in ascending or descending order of the value range.

11. `<resetcolor>...</resetcolor>`
    Frame for defining the color of a fallback color. There is only one fallback color per template.
    `type`=Application mode for the fallback color. Setting options:
    `1`=Default mode: First, the fallback color is applied to the entire model. Then the references are colored according to the colors defined in this template.
    `0`=All colors remain as they are, reference colors are applied on this basis, i. e. only references are colored.
    `2`=Same as 1 except that the default color of the model is applied instead of the fallback color.
12. `<appearance_data>...</appearance_data>`
    Color definition of the fallback color with the following attributes. Enter the properties of the color definition in quotes:


© 2026 INNEO Solutions GmbH

- `color_rgb chosen_color`=Color specification in RGB format
- `intensity value`=Intensity of the color (value between 0.00 and 1.00)
- `ambient value`=Ambient light (value between 0.00 and 1.00)
- `color_highlight chosen_color`=Specification of the highlight color in RGB format
- `shine value`=Shine (value between 0.00 and 1.00)
- `highlight value`=Highlight color (value between 0.00 and 1.00)
- `reflection value`=Reflection (value between 0.00 and 1.00)
- `transparency value`=Transparency (value between 0.00 and 1.00)

# 15.2 Surface Finish Table

With the submodule *Surface Finish Table*, you can place and manage surface finish tables for parts.

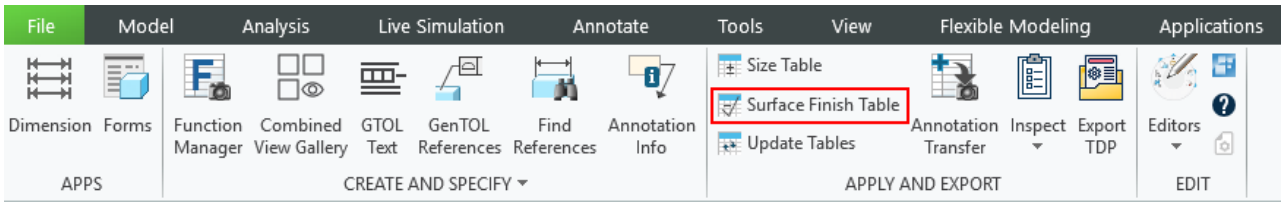
**Please note:** The submodule can only be used from Creo version 11.

## 15.2.1 Usage

In this section you will find information about the use of *Surface Finish Table*. The general structure of the program is explained.

### Starting the program: in part mode

To start *Surface Finish Table*, click *GT MBD* in the ribbon menu and go to *APPLY AND EXPORT* section.

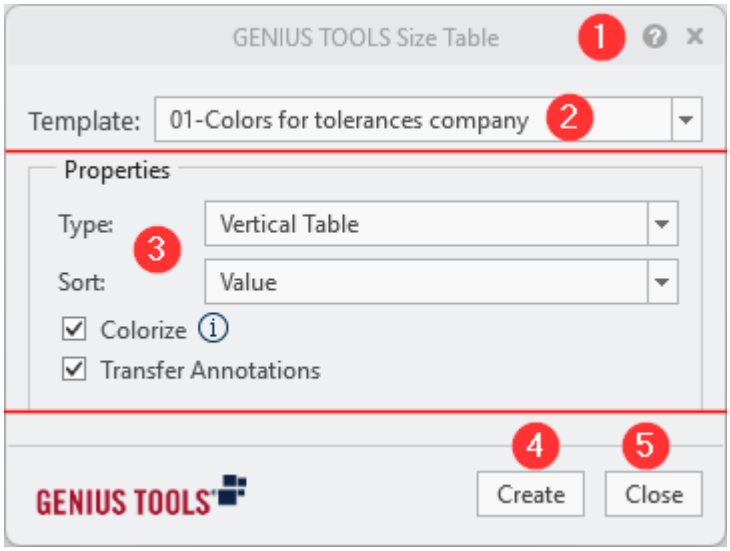



*Starting via the GT MBD in the ribbon menu*

#### 15.2.1.1 User interface

The user interface of *Surface Finish Table* consists of the following elements:


© 2026 INNEO Solutions GmbH




1. Help
   Opens this help.
2. Template
   Select a saved XML template[435].
3. Settings[431]
4. Create
   A table is created with the entered settings, see User interface[430]. If a table already exists, the command is called *Update*. In this case, no new table is created, but the existing one is updated. You can also use the Update function directly from the ribbon menu *GT MBD*, see MBD Update Tables[439]. After creating or updating a table, the user interface is automatically closed.

**Please note:** Tables are automatically placed at the zero point in the model. To move the table, mark the table by clicking the left mouse button and then click on the move handle. The table is attached to the mouse and can be moved to any location in the model.

5. Close
   The user interface is closed without a table being created. Previously changed settings are not saved.

## Result

You will see a table in which the coloring of each surface finish is indicated by an icon in a color column / color row. If you deselect the option *Colorize* in the user interface, the color column / color row will disappear. If you select the option again, the color column / color row will reappear after clicking *Update*.


© 2026 INNEO Solutions GmbH

| Allgemein | Spez. 1 | Spez. 2 |
|---|---|---|
| Rz 22 | Rz 3.2 | Rz 6.3 |
| ◆ | - | X |

*Example Table*

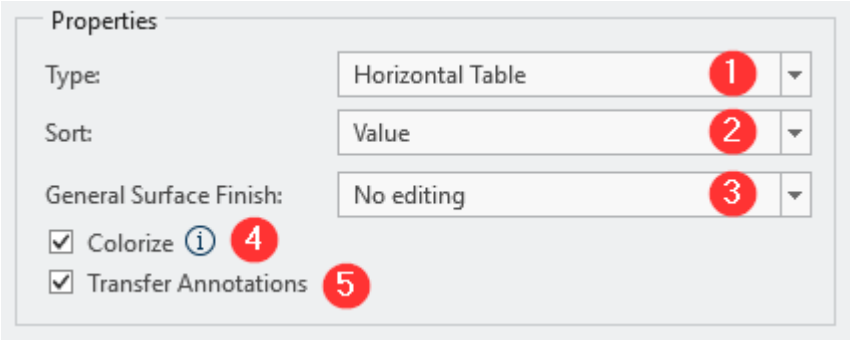
In a table, the following types of icons can appear:

| Icon | Description |
|---|---|
| ◆ (Diamond in the color in which a surface finish was colorized) | Assigned references are colored. |
| X | The value of the surface finish is within the range of values for a defined color, see used templates[427]. However, the surface finish has no semantic references that can be colored. All affected surface finishes that are found without semantic reference are reported in the Creo message log.<br>If no references are found, either there are no references or the surface finish is obsolete and needs to be updated or converted in the model tree.<br>**Exception:** Driven dimensions cannot be updated because outdated annotations cannot be referenced. |
| - | The value of the surface finish is outside the value range for a defined color, see used templates[427]. The surface finish contains references. |

## 15.2.1.2 Formatting a table

In the following area of the user interface, you see the settings for specifying the selected template:


© 2026 INNEO Solutions GmbH




**Please note:** To ensure that the proportions in the table are correct, set the *Symbol instance height* (see Creo window *Symbol Definition Attributes* > *General* > *Symbol instance height*) to the same setting for all symbol definitions used in the model.

**Please note:** The two symbols *3d_iso_21920_2021_surf_fin_en.sym* and *3d_iso_21920_einfach.sym* need to be in the *PRO_SURFACE_FINISH_DIR* directory in order for the supplied examples to work. You will find the two symbols in the folder path *\gt_resource_folder\mbdTables\surfFinTable*.

1. **Type**
   Define the alignment of the table with the following options:
   *Horizontal Table*: Table header in the first row with a line-by-line listing of the surface finishes
   *Vertical Table*: Table header in the first column with a column-by-column listing of the surface finishes
2. **Sort**
   Setting whether the surface finish symbols are displayed in sorted or unsorted order:
   *Value*: Table is sorted by values.
   *None*: Table is not sorted.
3. **General Surface Finish**
   Enter the general surface finish. All surfaces that are not assigned to a surface finish are defined by the general surface finish.
   From the drop-down list, you can select the symbols that are saved in User interface[433] .
   Special mode *None*: Colors are applied in the order defined in the XML template. Color ranges are ignored.

**Please note:** A warning icon ⚠ appears if the previously selected surface finish is not loaded in Creo. Set the path to the surface finish symbols in User interface[433] . If no *general surface finish* is selected, undefined surfaces are not assigned a *general surface finish*. The table can also be created without a *general surface finish*. However, in this case the table is not complete.


© 2026 INNEO Solutions GmbH

**Please note:** It is possible to manually adjust the *General Surface Finish* after the table is created. When you update the table with the previous properties, all information except the *General Surface Finish* is updated. If you use symbol definitions with stored default parameters, this behavior cannot be guaranteed (see Creo window *Symbol Definition Attributes* > *Grouping and Variable Text Parameters*).

4. **Colorize**
   Overwrites the appearance state in the active combined view.
5. **Transfer Annotations**
   All annotations present in the model that will be visible in the table are transferred to the combined view in which the table is created.

## 15.2.2 Configuration

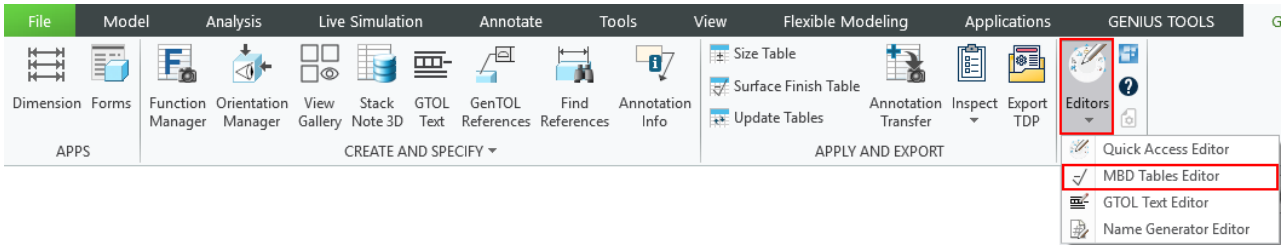
In this section you will find detailed information about the structure of the module *MBD Tables* and how to create table templates.

### 15.2.2.1 MBD Tables Editor

In this section you will find information about the use of *MBD Table Editor*. With this submodule, you can store various general surface finishes that you can select in User interface[429].

## Starting the program: in part mode

Start *MBD Table Editor* from the ribbon menu *GT MBD*. Click on the button to open the user interface.

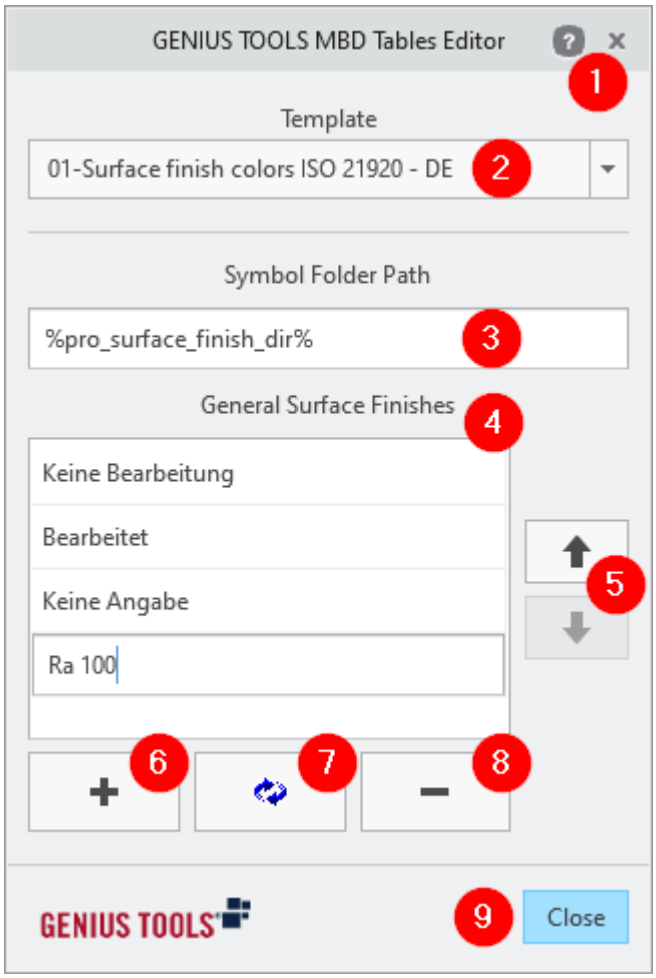



*Starting via the GT MBD tab in the ribbon menu*

## User interface

The user interface of *MBD Table Editor* consists of the following elements:


© 2026 INNEO Solutions GmbH




1. Help
   Opens this help.
2. Template
   Select a saved XML template[435].
3. Symbol folder path
   An explicit storage location must be specified, where the symbol created under 4. is saved. This is necessary so that the symbol can be loaded into the table.
4. General Surface Finishes
   List of symbols that can be selected as the general surface finish.

**Tip:** Hover over a line to see the information stored in the XML definition for this symbol.

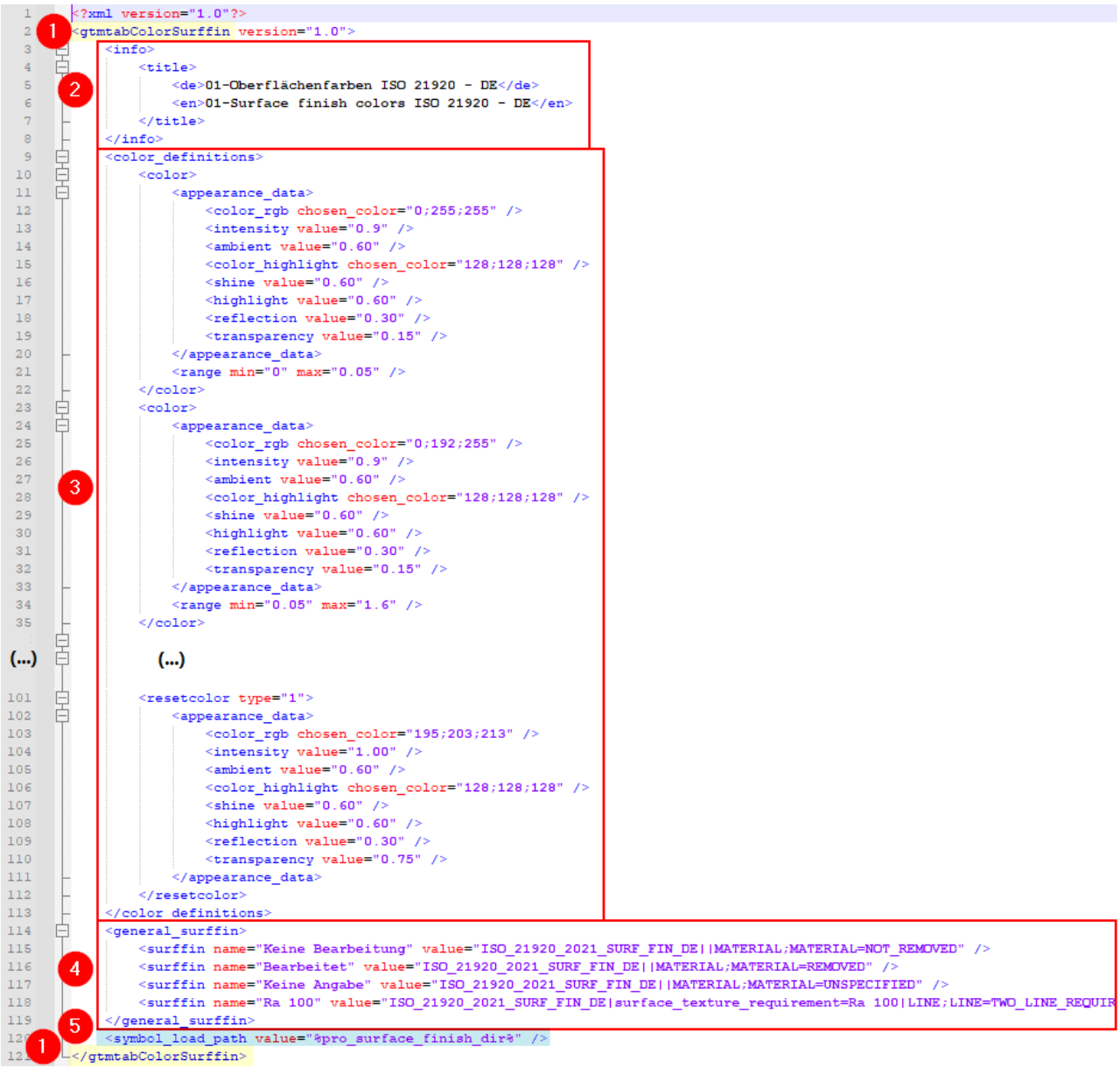
5. Moving a symbol up ⬆ or down ⬇ in the list
6. Adding a symbol to the list
   After clicking the button, you can select a surface finish symbol in the model, which will be added to this line. After selecting a symbol, a new line is added in which you can enter a name for this symbol. The symbol can be selected in Formatting a table[431] under this name. Newly created symbols are stored in the symbol folder path (see 3).


© 2026 INNEO Solutions GmbH

7. Updating the XML definition for a symbol
   A previously created symbol is being modified.
8. Deleting a symbol from the list
9. Close
   The user interface is closed. Changes are saved automatically.

## 15.2.2.2 Editing templates for Surface Finish Tables

The illustration shows an example of a table template for displaying surface finishes. This example is used to explain the different sections of a template.




```
<?xml version="1.0"?>
<gtmtabColorSurffin version="1.0">
    <info>
        <title>
            <de>01-Oberflächenfarben ISO 21920 - DE</de>
            <en>01-Surface finish colors ISO 21920 - DE</en>
        </title>
    </info>
    <color_definitions>
        <color>
            <appearance_data>
                <color_rgb chosen_color="0;255;255" />
                <intensity value="0.9" />
                <ambient value="0.60" />
                <color_highlight chosen_color="128;128;128" />
                <shine value="0.60" />
                <highlight value="0.60" />
                <reflection value="0.30" />
                <transparency value="0.15" />
            </appearance_data>
            <range min="0" max="0.05" />
        </color>
        <color>
            <appearance_data>
                <color_rgb chosen_color="0;192;255" />
                <intensity value="0.9" />
                <ambient value="0.60" />
                <color_highlight chosen_color="128;128;128" />
                <shine value="0.60" />
                <highlight value="0.60" />
                <reflection value="0.30" />
                <transparency value="0.15" />
            </appearance_data>
            <range min="0.05" max="1.6" />
        </color>
(...)       (...)
        <resetcolor type="1">
            <appearance_data>
                <color_rgb chosen_color="195;203;213" />
                <intensity value="1.00" />
                <ambient value="0.60" />
                <color_highlight chosen_color="128;128;128" />
                <shine value="0.60" />
                <highlight value="0.60" />
                <reflection value="0.30" />
                <transparency value="0.75" />
            </appearance_data>
        </resetcolor>
    </color_definitions>
    <general_surffin>
        <surffin name="Keine Bearbeitung" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=NOT_REMOVED" />
        <surffin name="Bearbeitet" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=REMOVED" />
        <surffin name="Keine Angabe" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=UNSPECIFIED" />
        <surffin name="Ra 100" value="ISO_21920_2021_SURF_FIN_DE|surface_texture_requirement=Ra 100|LINE;LINE=TWO_LINE_REQUIR
    </general_surffin>
    <symbol_load_path value="&pro_surface_finish_dir&" />
</gtmtabColorSurffin>
```

1. `<gtmtabColorSurffin>...</gtmtabColorSurffin>`
   Frame for the template: This XML file shows a defined template for a surface finish table. You can define multiple colors for coloring surface finishes in an XML file.

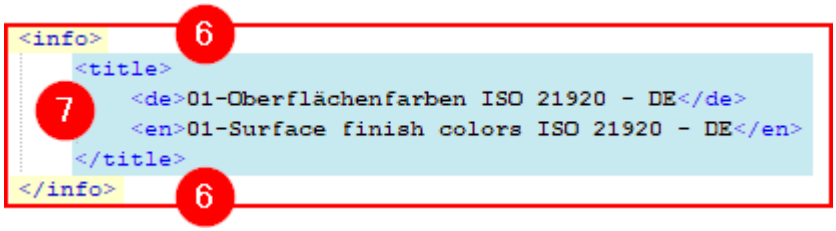

© 2026 INNEO Solutions GmbH

- `version`=Specification of a number for versioning the template

2. Editing templates for Surface Finish Tables[436]
3. Editing templates for Surface Finish Tables[436]
4. Editing templates for Surface Finish Tables[438]
5. Symbol folder path
   The symbol definitions are searched for in this folder path if they are not found in the table when the general surface finish is inserted. This folder path can also be set in the Editor, see User interface[434] .

## Information about the template

Each template requires the following tags:

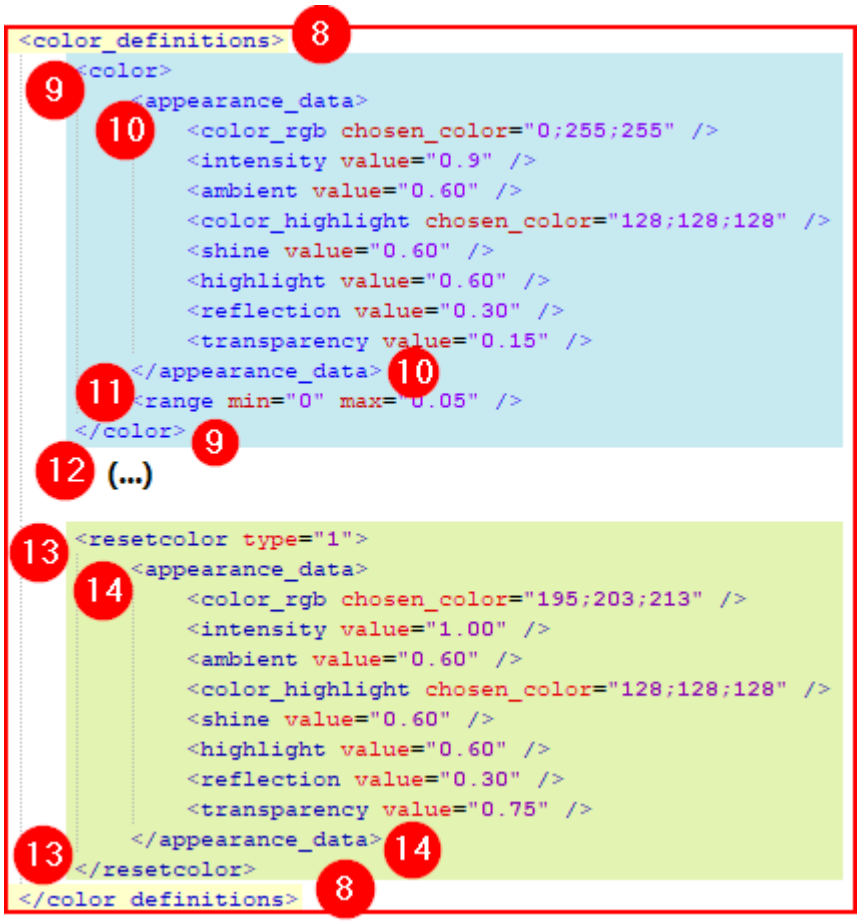



6. `<info>... </info>`
   Tags to frame the information section
7. `<title>... </title>`
   Title in English `<en>` and German `<de>`

## Colors for coloring surface finishes

Colors for coloring surface finishes are defined with the following properties:


© 2026 INNEO Solutions GmbH




8. `<color_definitions>...</color_definitions>`
   Frame for the part which contains all color definitions.

9. `<color>...</color>`
   Frame for one color definition.

10. `<appearance_data>...</appearance_data>`
    Color definition with the following attributes. Enter the properties of the color definition in quotes:
    - `color_rgb chosen_color`=Color specification in RGB format
    - `intensity value`=Intensity of the color (value between 0.00 and 1.00)
    - `ambient value`=Ambient light (value between 0.00 and 1.00)
    - `color_highlight chosen_color`=Specification of the highlight color in RGB format
    - `shine value`=Shine (value between 0.00 and 1.00)
    - `highlight value`=Highlight color (value between 0.00 and 1.00)
    - `reflection value`=Reflection (value between 0.00 and 1.00)
    - `transparency value`=Transparency (value between 0.00 and 1.00)

11. `<range .../>`
    Value range in which this color is assigned to a surface symbol:


© 2026 INNEO Solutions GmbH

`min`=Minimum (including this value)
`max`=Maximum (excluding this value, i. e. in this case all values up to $0{,}04\bar{9}$ are colored with this color)

**Please note:** Specify a value range so that the color can be applied. If a color is not assigned to a value range, the color cannot be applied. However, if only one value is not specified, it is automatically assumed to be 0.

12. Repeat steps 8 through 11 for each color you want to use. You can enter as many colors as you like.

**Tip:** For clarity, it is recommended that you insert the colors in the template in ascending or descending order of the value range.

13. `<resetcolor>...</resetcolor>`
    Frame for defining the color of a fallback color. There is only one fallback color per template.
    `type`=Application mode for the fallback color. Setting options:
    `1`=Default mode: First, the fallback color is applied to the entire model. Then the references are colored according to the colors defined in this template.
    `0`=All colors remain as they are, reference colors are applied on this basis, i. e. only references are colored.
    `2`=Same as 1 except that the default color of the model is applied instead of the fallback color.
14. `<appearance_data>...</appearance_data>`
    Color definition of the fallback color with the following attributes. Enter the properties of the color definition in quotes:
    - `color_rgb chosen_color`=Color specification in RGB format
    - `intensity value`=Intensity of the color (value between 0.00 and 1.00)
    - `ambient value`=Ambient light (value between 0.00 and 1.00)
    - `color_highlight chosen_color`=Specification of the highlight color in RGB format
    - `shine value`=Shine (value between 0.00 and 1.00)
    - `highlight value`=Highlight color (value between 0.00 and 1.00)
    - `reflection value`=Reflection (value between 0.00 and 1.00)
    - `transparency value`=Transparency (value between 0.00 and 1.00)

## Surface finishes

Surface finishes are stored in the template in the following form:

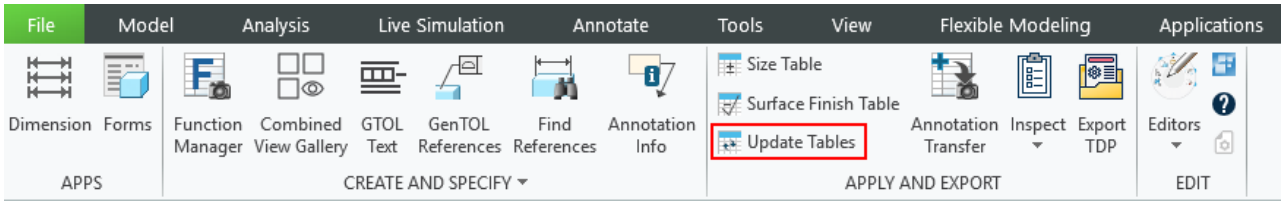

© 2026 INNEO Solutions GmbH

```
<general_surffin>
    <surffin name="Keine Bearbeitung" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=NOT_REMOVED" />
    <surffin name="Bearbeitet" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=REMOVED" />
    <surffin name="Keine Angabe" value="ISO_21920_2021_SURF_FIN_DE||MATERIAL;MATERIAL=UNSPECIFIED" />
    <surffin name="Ra 100" value="ISO_21920_2021_SURF_FIN_DE|surface_texture_requirement=Ra 100|LINE;LINE=TWO_LINE_REQUIR
</general_surffin>
```

15. `<general_surffin>...</general_surffin>`
    Frame for defining each surface finish. Enter the properties of the surface finishes in quotes.
    - `surffin name`=General surface finish available for selection in the *MBD Surface Finish Table*
    - `value`=Detailed information about this surface finish. Displayed as meta information in this template and as a tooltip in *MBD Table Editor*.

## 15.3 MBD Update Tables

Start the function *Update Tables* via the ribbon menu *GT MBD*. Click on the button to perform the update for all existing MBD tables. No new window is opened during the update. The model is regenerated with the update and a corresponding message appears in the Creo message log.




*Starting via the GT MBD tab in the ribbon menu*


© 2026 INNEO Solutions GmbH

# 16 Name Generator

Available in the ribbon menus GENIUS TOOLS | GT MBD

Creo Parametric is only to a certain extent able to ensure the continuous naming of individual objects. Especially if no product data management solution is in use in a company, it is difficult to achieve consistent and logical numbering.

*Name Generator* is an intuitive tool for sequentially assigning names with a numbering for file names of individual parts, sheet metal parts and assemblies. However, the program can also be used for any other files and objects for the construction. In this way, you ensure unique assignment and enable sequential numbering in accordance with working guidelines.

Name Generator can be used individually on stand-alone workstations (locally) as well as in a network (globally) for all Creo workstations.

*Name Generator* is available with following features:

1. Usable in intelligent Mapkeys in the Modules *Quick Access*[544], *Library*[293], *Forms*[123] and *Parameter*[465].
2. Manual application in various design steps when
   - assigning names for family table variants,
   - creating parts in assemblies.

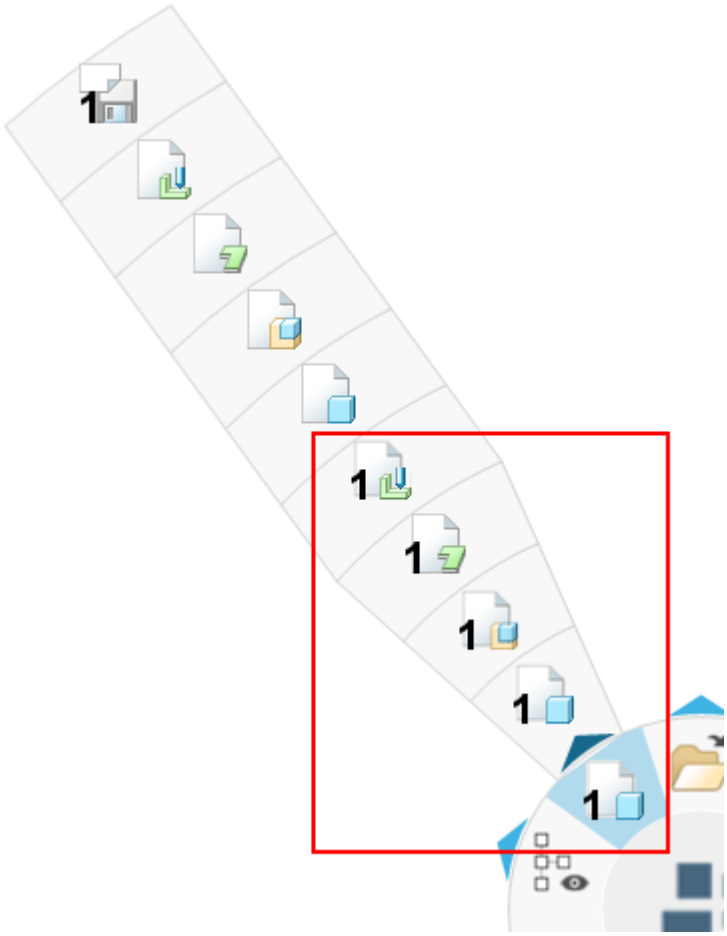
## 16.1 Usage

### Starting the program for creating new models

The main application of the *Name Generator* is the generation of filenames for new assemblies and parts.

In *Quick Access* ([<] key), the buttons for creating models with *Name Generator* are already active in standard configuration. You can identify those buttons by the small number One.

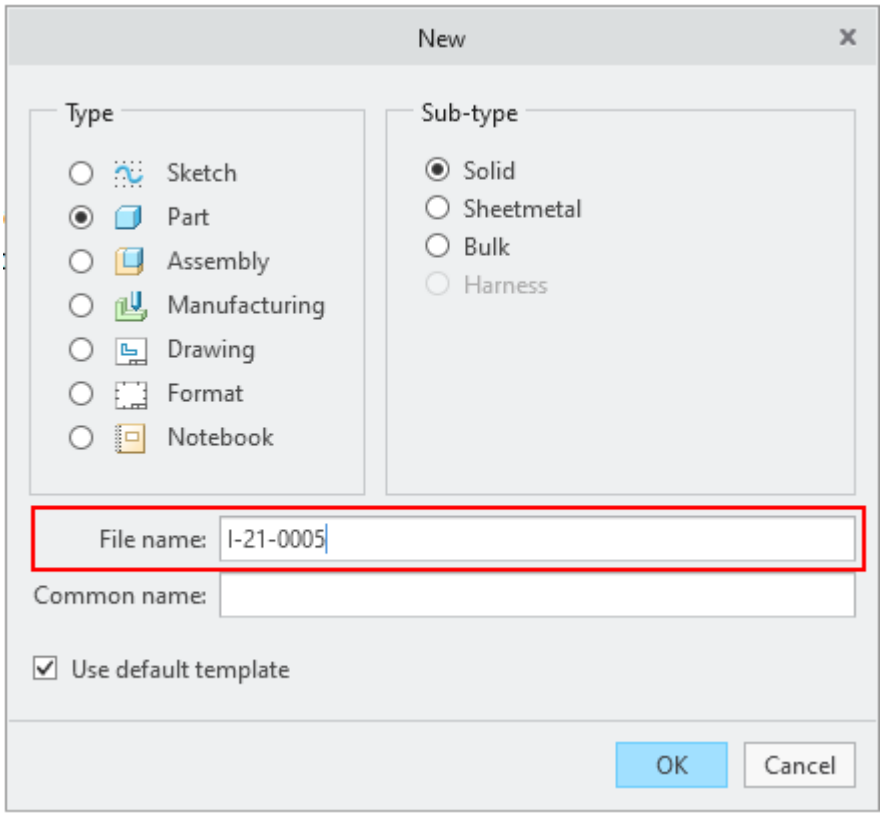

© 2026 INNEO Solutions GmbH



*Starting via Quick Access*

One click creates models of the specified type with the preset name configuration. The Creo default dialog for creating new models is opened. Instead of the typical Creo proposal a generated name proposal is pre-entered.


© 2026 INNEO Solutions GmbH




**Please note**: Make sure not to change the model type afterwards. (Additions within the scope of maximum name length may be reasonable.) Different name configurations for generating the model names can be deposited.

## Automated application in other GENIUS TOOLS modules

*Name Generator* is embedded in the *GENIUS TOOLS* for Creo standard configuration for the following modules and can be used automatically for the creation of new parts, assemblies or drawings.

– In *Quick Access* and *Forms* the *Name Generator* is used with customized smart mapkeys.

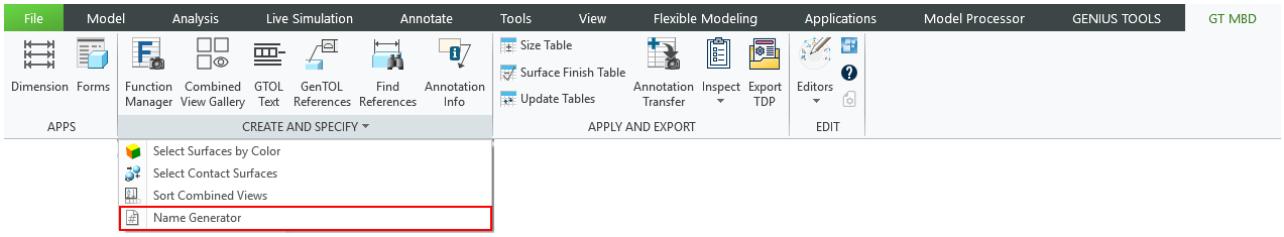
– In sample data of *Parameter* the *Name Generator* is included.

– Via the *Library* sample libraries, new parts are created with *Name Generator*.

## Special case: Starting the program from the ribbon menu: in part / assembly mode

To start *Name Generator* , click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section. This is necessary if you generate a number independent from the module and want to paste it via the clipboard. For example to paste the generated name in family tables.

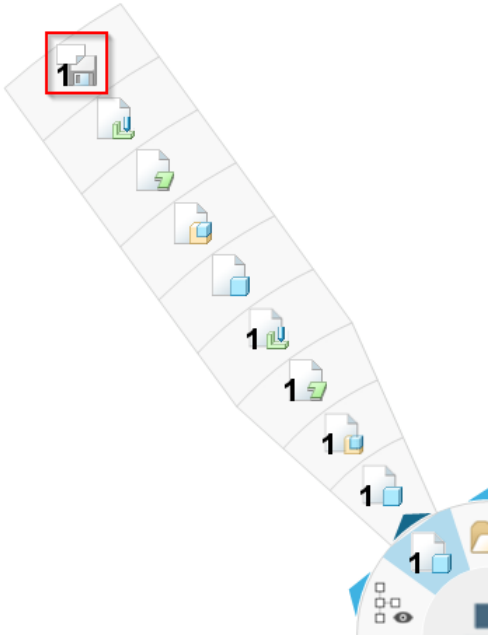

© 2026 INNEO Solutions GmbH



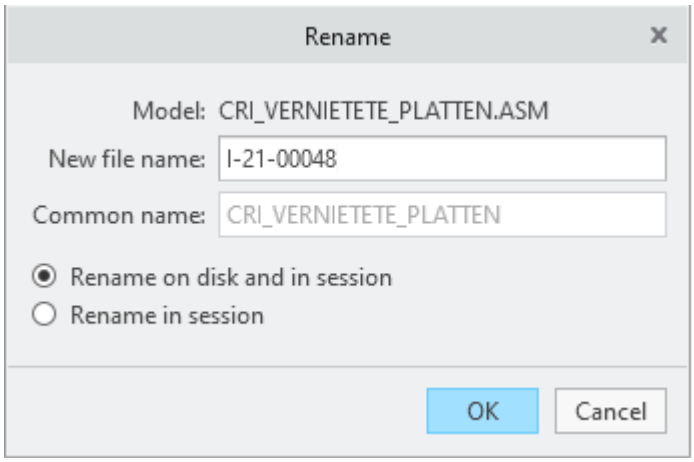

*Starting via the GT MBD in the ribbon menu*

# Renaming models using Quick Access

To rename the current model use the button *Rename current model* in the *Quick Access* menu.



Clicking this button opens the *Name Generator*.



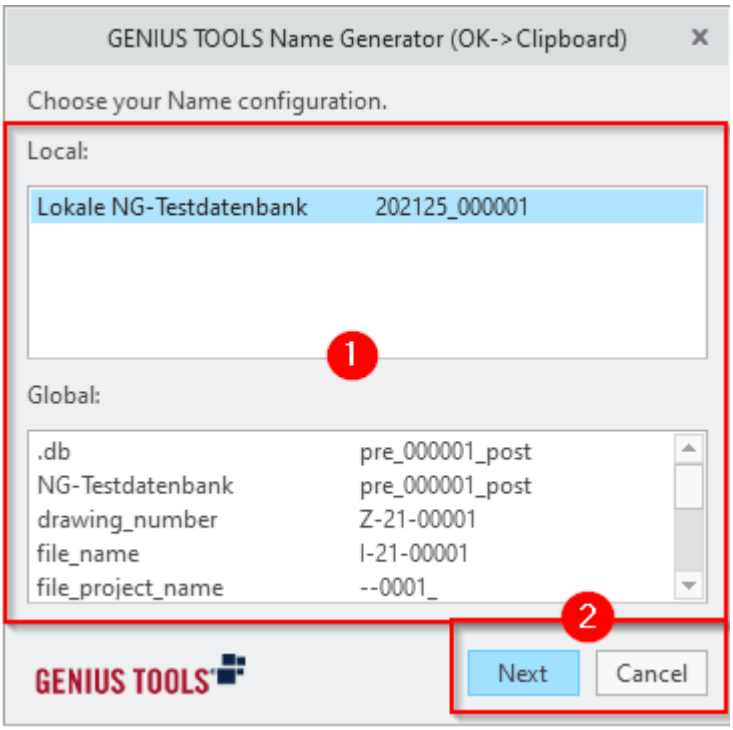

Select the required name configuration and confirm the dialog. The Creo default dialog for renaming is opened with the newly generated file name pre-entered.


© 2026 INNEO Solutions GmbH

## 16.1.1 User interface

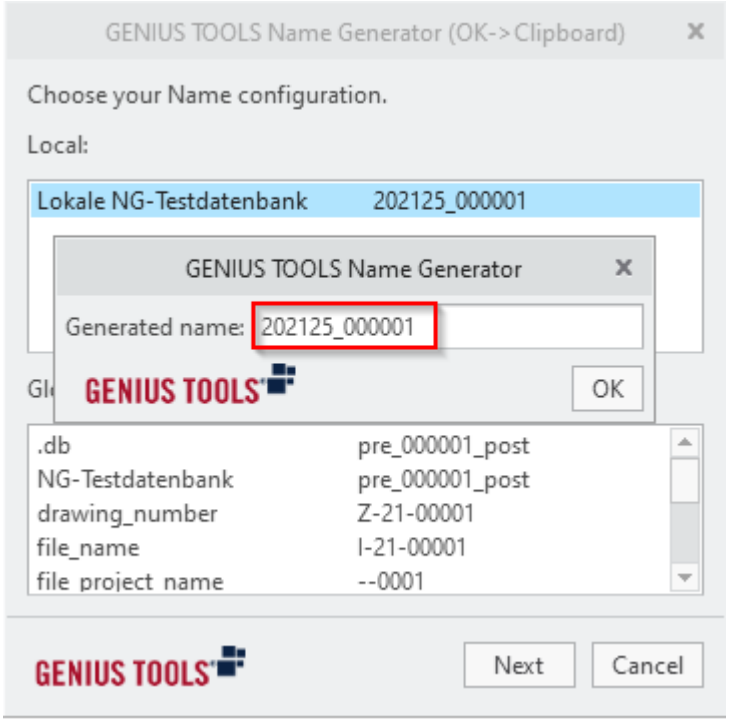
The user interface of *Name Generator* consists of the following elements:

1. Local and global name configurations
2. Buttons for *Next* and *Cancel*.




A generic preview is displayed behind each name configuration. Select one of the available name configurations (1) and confirm your selection (2). Customize the generated name with your own entries in the next dialog and confirm the dialog.

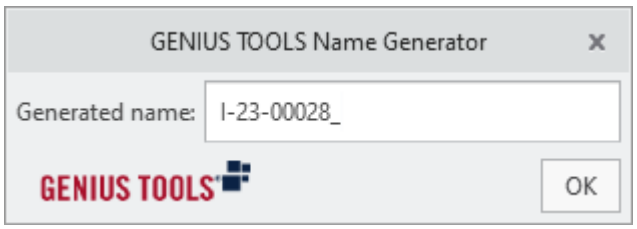

© 2026 INNEO Solutions GmbH




Customize the generated name if necessary. After clicking *OK*, the name is in the buffer memory. Use the generated name via CTRL+ V or the context menu (*right-click -> Paste*).

## Controlling the user interface display

The appearance of the user interface depends on the `gtng_db_name_filter` configuration option. This configuration option allows you to specify values by which name configurations are searched. Then only the name configurations that match the stored value range are displayed. If you have stored a value in the `gtng_db_name_filter`, only the name configurations that match or correspond to this value are displayed. If exactly one name configuration corresponds to the value range, this name is already suggested. The user interface then opens in this form:




# 16.2 Configuration

In this section you will find information about the configuration of the *Name Generator* module, which is done in *Name Generator Editor*[446].


© 2026 INNEO Solutions GmbH

## 16.2.1 Name Generator Editor

*Name Generator Editor* is used to create name configurations and to save them either locally or globally.

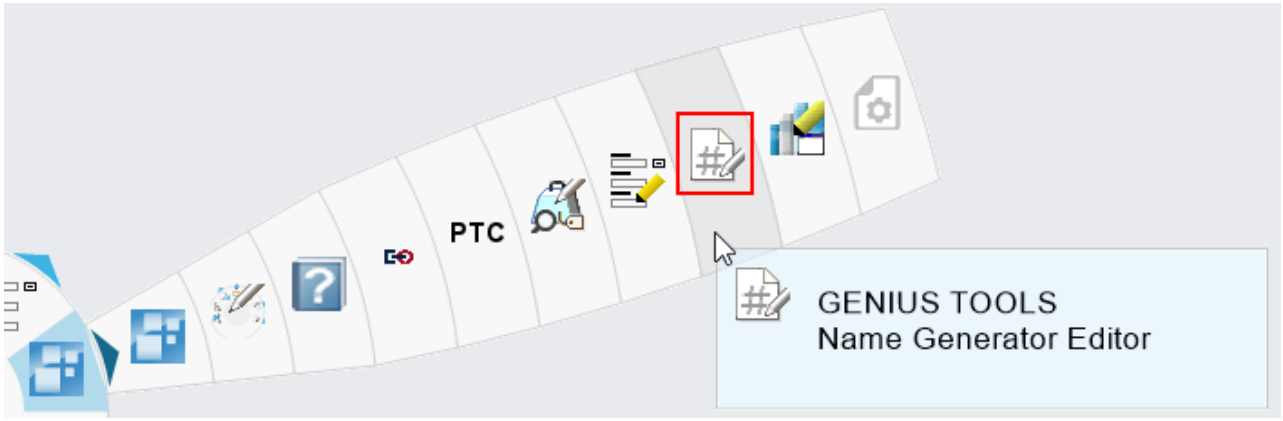
– Local name configurations are saved locally in the appdata directory of the current user.

– For users of *Startup TOOLS* global configurations are saved in the Main server *caddepot\serveronly*. (If you use *GENIUS TOOLS* for Creo without *Startup TOOLS*, you can find the global configuration in the resource folder.)

**Warning**: Global name configurations require read and write access to the files for all users.

### Starting the program

Start *Name Generator Editor* via *Quick Access* ([<] key).




**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].

### Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

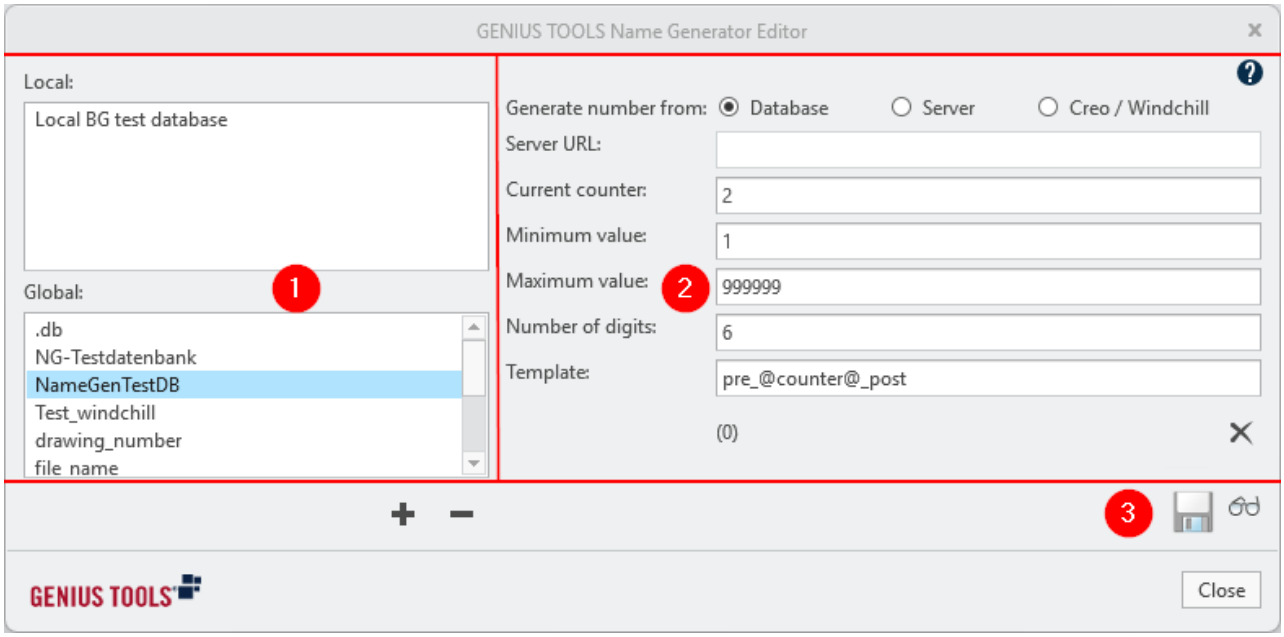
To ensure that the changes are applied, you should work directly in Caddepot.


© 2026 INNEO Solutions GmbH

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## 16.2.1.1 User interface

The user interface of *Name Generator Editor* consists of the following elements:




1. Available name configurations[447]
2. Configuration details[448]
3. Save and preview. Click on the glasses-symbol to see a preview of the generated file name.

## 16.2.1.2 Available name configurations

Existing name configurations are displayed on the left side of the dialog. Click an entry in the lists to display the associated options in the configuration details. Use the plus button to create new name configurations. Select a name configuration and click the minus button to delete it.

### Local and global name configurations

In the configurations chose either the option for local name configurations (`gtng_local_folder`) or the option for global name configurations (`gtng_folder`).

For different usage scenarios, there are local and global name configurations.

Local name configurations are suitable for constructions in which no other persons are involved or which are edited exclusively with one user account on one workstation.


© 2026 INNEO Solutions GmbH

Local name configurations, by default, are saved under *%appdata%\INNEO\GENIUS_TOOLS\for_Creo\name_generator* on a local computer.

**Tip**: Customize the storage location of local and global name configurations with the *GENIUS TOOLS* for Creo configuration options.

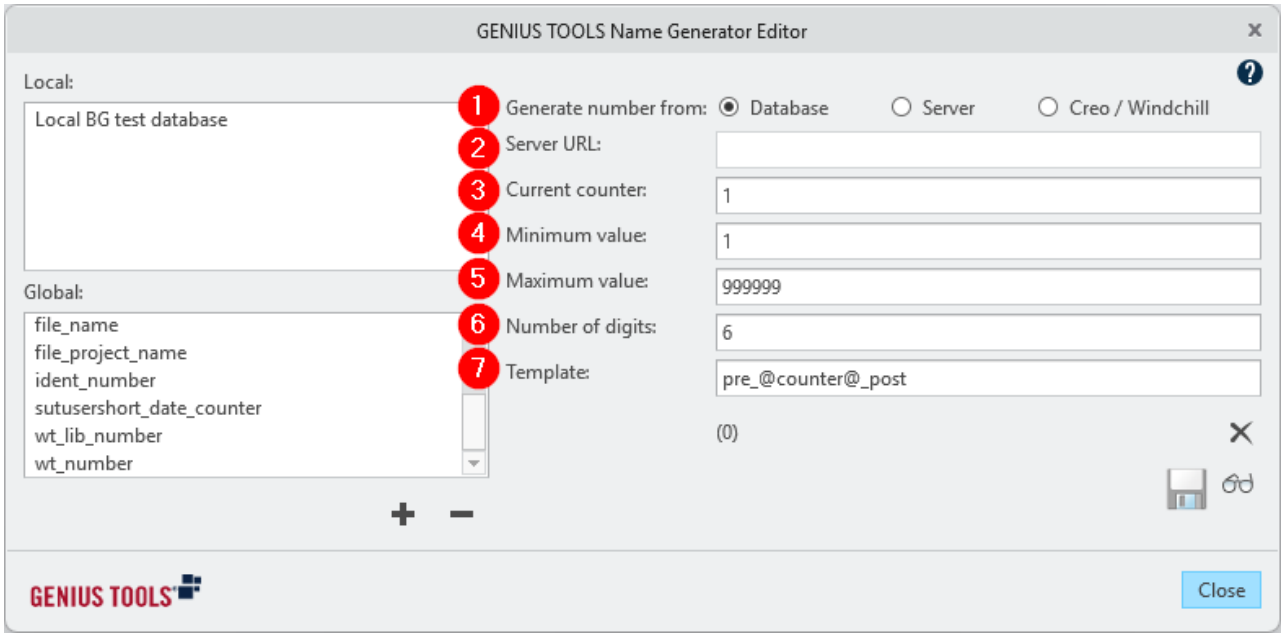
Global configurations are stored by default in the main server under *caddepot\serveronly\gt_name_generator* and are applicable for all users.

In order to create new names with the current counter, there always must be an active network connection.

**Warning**: Global name configurations require read and write access to the files for all users.

## 16.2.1.3 Configuration details

The current configuration of the selected element is displayed in the configuration details.



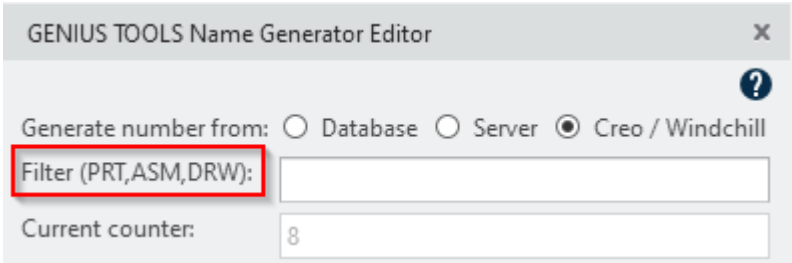

**1. Generate number from**

Specifies whether names are generated from the name generator databases or a Webserver.

If you are working with Windchill, use the option *Creo / Windchill*. This option lets you request numbers from Windchill without installing a customization on the Windchill server. The functionality uses the currently connected server and the current Creo user.


© 2026 INNEO Solutions GmbH

2. **Second line**
   - **If Database is selected: no specifications possible**
   - **If Server is selected: Server URL (2.)**
     The URL of the Webserver. The URL must return a number. This number is adopted into a name configuration instead of `@counter@`. The URL only has to be specified when you are using the Server option.
   - **Selection Creo / Windchill: filter**
     Specify whether you want Windchill to number parts (PRT), assemblies (ASM) or drawings (DRW).




The filter only has to be specified when you are using the *Creo / Windchill* option to address a specific Windchill-number generator. If nothing is selected, PRT (part) is used automatically. Ignore this setting if numbers are generated from databases.

3. **Current counter**
Displays the current counter of the name configuration. If necessary, the counter can be customized in this field.

**Warning**: The current counter must at least equal the minimum value.

4. **Minimum value**
The minimum value is a name configuration's start value. Enter the smallest number for name generation.

5. **Maximum value**
The maximum value is the last number that can be assigned in the name generation.

6. **Number of digits**
Specifies the number of digits for number output.

7. **Template**
Specifies the name configuration. Use the variable `@counter@` for upwards counting name configurations.

**Warning**: In older GENIUS TOOLS for Creo versions `@number@` was used for the counter. To ensure compatibility, the variable can still be used.

You can additionally use all variables and string operations of *GENIUS TOOLS* for Creo. See the variables list 913. All *GENIUS TOOLS* for Creo variables can be used in name


© 2026 INNEO Solutions GmbH

configurations as desired. However, filenames are limited to 31 characters. Where variables return longer values, the generated name is truncated.

## Sample for name generation

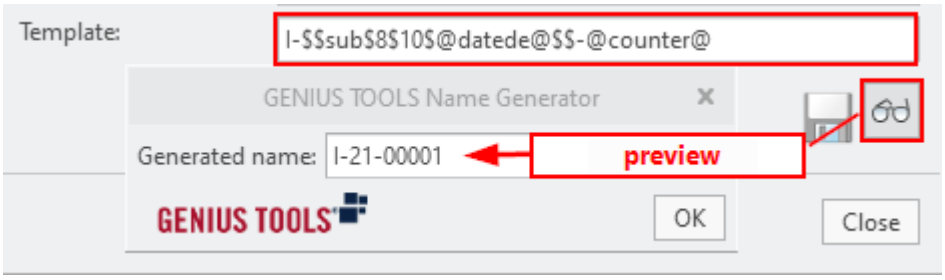
Enter the following variable: `I-$$sub$$8$10$@datede@$$-@counter@` under Template.

`$$sub$$8$10$:` shows the 8th to 10th digit from the date format, whereas the 0 from the 8th digit is not displayed.

`@datede@:` date format dd-mm-yyyy (de for german)

`@counter@: generates` a consecutive number

Result: The generated file name is: I-21-00001.




## 16.2.2 Use cases

In this section you will find use cases for the *Name Generator* module.

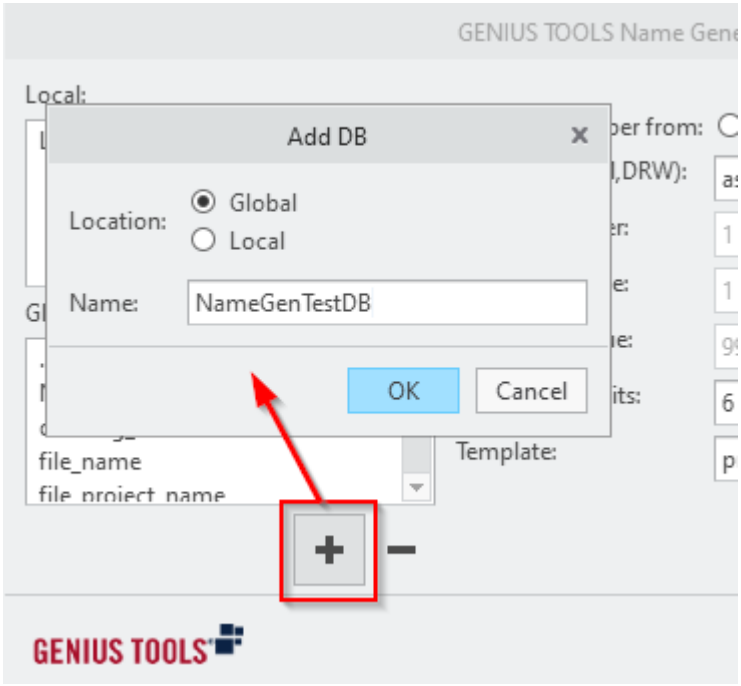
### 16.2.2.1 Global name configurations with fallback

A name configuration is created in this example. With this name configuration, a parameter is read from a model that is truncated and returned together with the counter. If the parameter does not exist in a model, a static fallback text is returned instead.

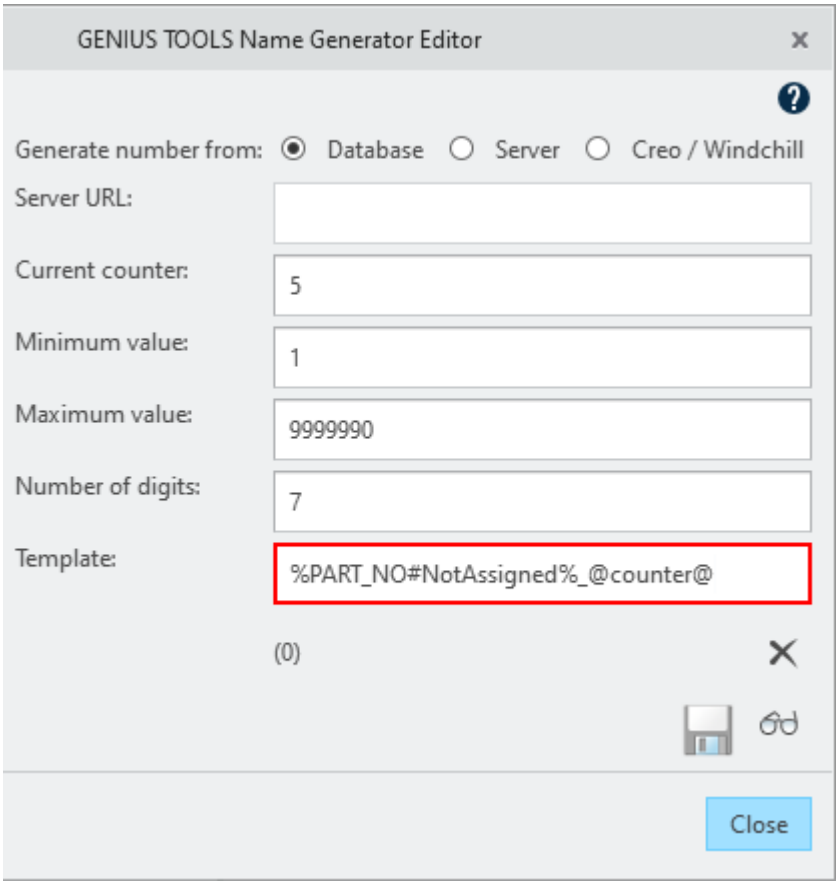
Proceed as follows to create a new name configuration with fallback:

1. Open *Name Generator Editor*.
2. Click the plus button in the command bar.
3. Set the storage location to *Global*. Enter a name for the new name configuration. Make sure to use only special characters that can be used for filenames in windows.


© 2026 INNEO Solutions GmbH




4. Confirm the dialog with *OK*.
5. Select the newly created name configuration at Global.
6. Configure the details of the name configuration.





*© 2026 INNEO Solutions GmbH*

7. Save the name configuration. The global name configuration with fallback is ready to use for all users.

**Explanation of the created name rule**

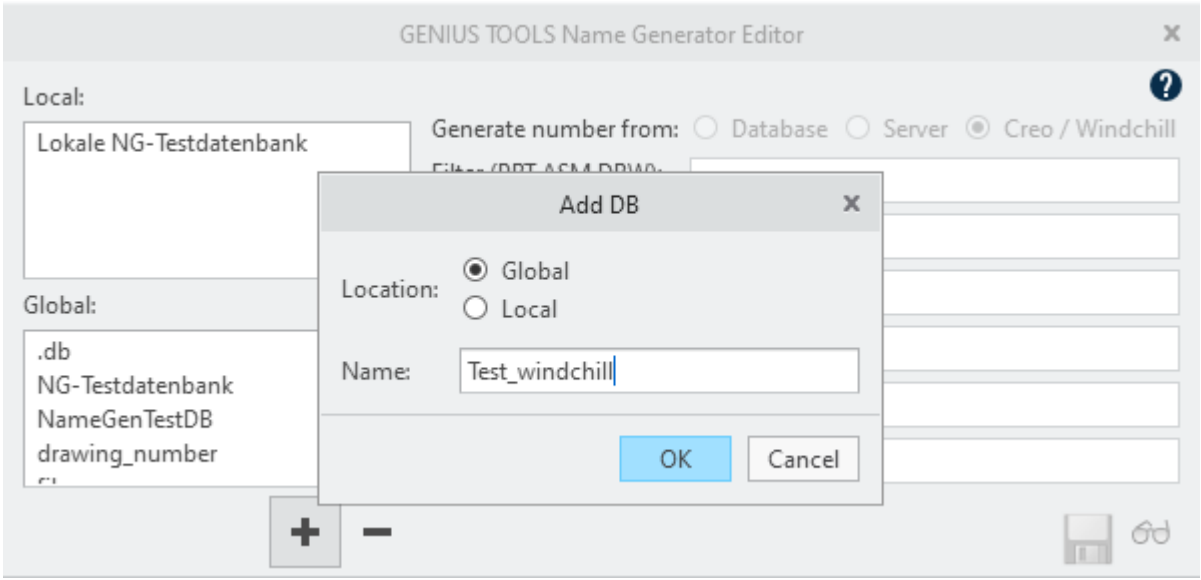
`%PART_NO#NotAssigned%_@counter@`

A parameter is read from the models and after the hyphen _ the counter is incremented. If no parameter is found, the static text `NotAssigned` is written in front of the counter instead.

## 16.2.2.2 Global name configurations with Windchill

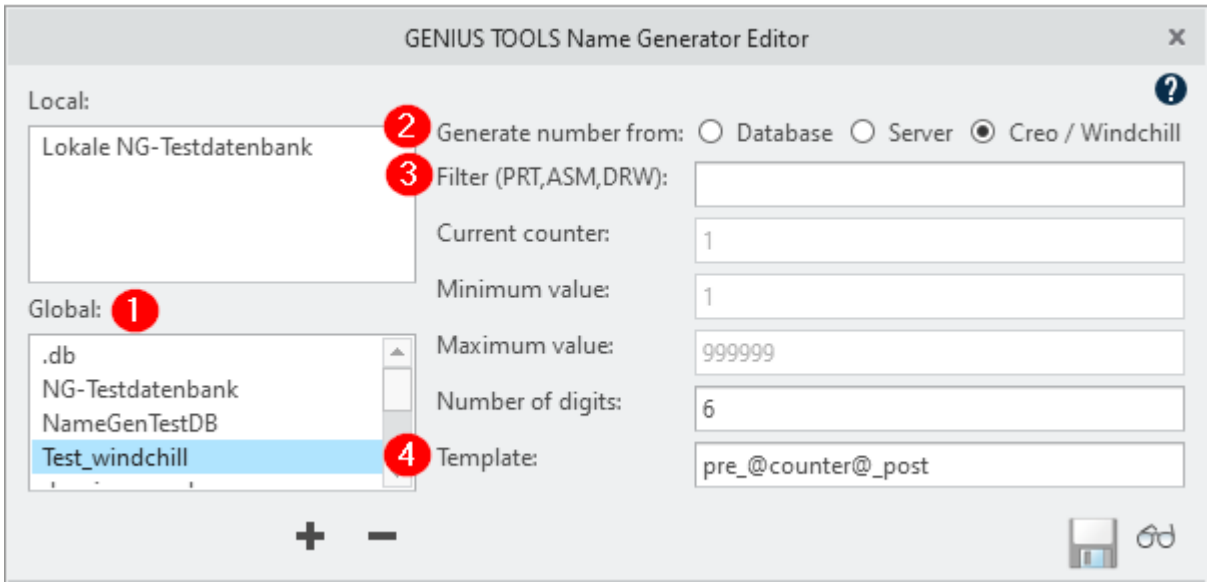
A global name configuration is created in this example. In the name configuration a number from a Windchill server is used.

1. Open *Name Generator Editor*.
2. Select the Plus-button in the command line. A dialog opens.
3. Set the storage location to *Global*. Enter a name for the new name configuration and confirm the dialog with *OK*. Make sure to use only special characters that can be used for filenames in windows

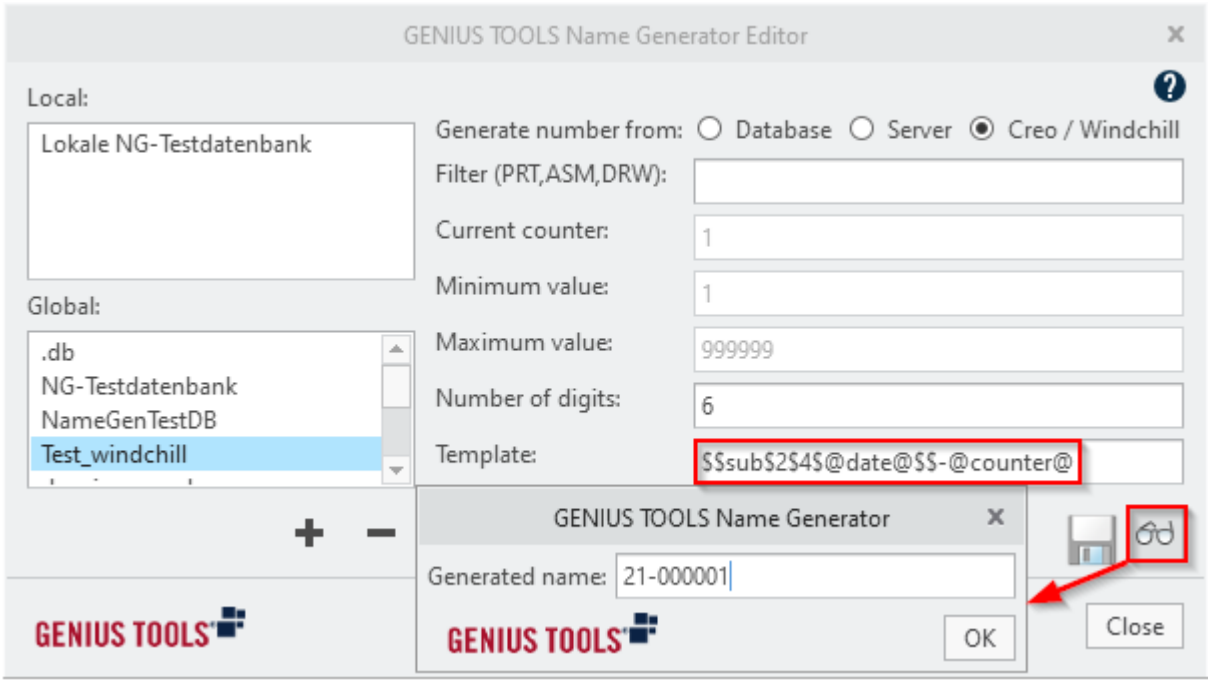



4. Select the newly created name configuration under *Global* (1). Under *Generate Number from* (2) select the option *Creo / Windchill*.
5. Chose the *Filter* (3), select the specific Windchill-number generator. If nothing is selected, the Windchill-number generator for PRT (parts) is used automatically.
6. Under *Template* (4) an example is already entered (here: *pre_@counter@_post)*, which can be customized.


© 2026 INNEO Solutions GmbH




7. Configure under *Template* (4) the name configuration to your requirements. Enter for example: `$$sub$2$4$@date@$$-@counter@`. In this operation (`$$sub$2$4$`) the 2nd to 4th digit is cut out from the date format yyyy-mm-dd (@date@). The 0 from digit 2 is not displayed. With `@counter@` a consecutive number is generated and attached.




8. With the preview-symbol (glasses) a window opens and shows the generated filename. In this example the filename is: 21-000001.
9. Save the name configuration (disc-symbol).

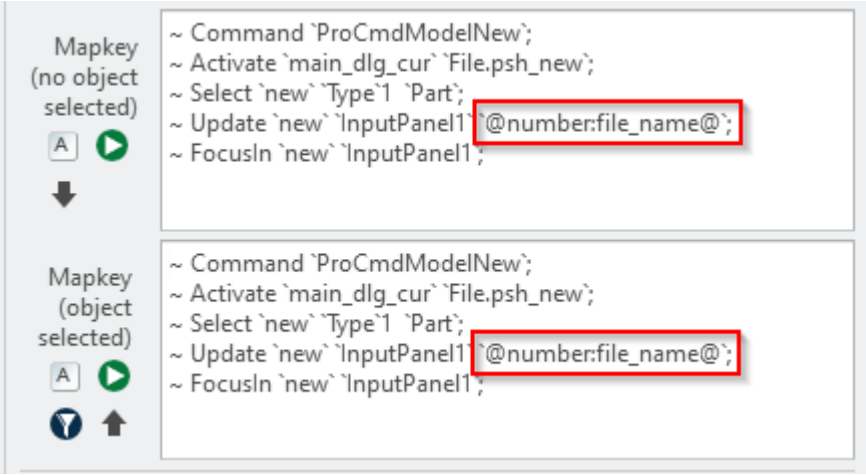
The counter `@counter@` in the name configuration now uses Windchill's number generator.

**Tip**: Windchill name generators do not have to be on the global Mainserver.


© 2026 INNEO Solutions GmbH

## 16.2.2.3 Integrating name configurations in Quick Access

*Quick Access* uses intelligent Mapkeys extended with variables and operations that can only be used with *GENIUS TOOLS* for Creo.




Customize the Mapkeys in *Quick Access* (*Quick Access Editor*).

For being able to use any name configurations, the Mapkeys have to be customized in *Quick Access*.

The variable *@number:...@* needs to be customized in the predefined Mapkeys.

The Mapkey for *Create new part with number*:

```
~ Command `ProCmdModelNew`;
~ Activate `main_dlg_cur` `File.psh_new`;
~ Select `new` `Type`1 `Part`;
~ Update `new` `InputPanel1` `@number:file_name@`;
~ FocusIn `new` `InputPanel1`;
```

Now customize the variable *@number:...@* in the Mapkey.

## Variant 1: With name configuration selection

*@number@* opens *Name Generator* without pre-selected configuration. After selecting the configuration, the Creo dialog for creating parts is opened with a pre-entered name.

## Variant 2: With defined name configuration

*@number:nameconfiguration@* directly opens the dialog for part creation. A filename – created with the entered name configuration – is predefined and can be customized.

**Please note**: If the name configuration is not unique, a name configuration will still have to be selected via the selection dialog.


© 2026 INNEO Solutions GmbH

## 16.2.2.4 Using project numbers

In this example, a project number with a consecutive number is to generate new names using the *Name Generator*. This new and unique name is used, for example, for new parts and assemblies.

**Variant A:** There is only one consecutive number for all project numbers.

**Variant B:** Each project number has its own consecutive number (number range).

## Query: Where can the project number be obtained from automatically?

### Method 1: Project number comes from a parameter of the current model

**Set project number - initial assignment**
The initial assignment is done manually, in which a defined model parameter receives the project number as a value. By changing the active model (there can only ever exist one in the Creo session), the value of the defined model parameter will also change, and thus with it the project number.

**Advantage**
If models from different projects are edited in one session, the "correct" new name is automatically created if the "correct" model is active.

**Disadvantage**
If there is no model in the Creo graphics window (e.g. in the start window), no project number can be determined.

### Method 2: The project number is defined in the session environment (operating system variable)

**Set project number - initial assignment**
If a defined operating system variable is assigned a value before Creo is started, Creo Parametric must be restarted every time the project changes.

If the defined operating system variable is to be changed in the current session, you can define a Javascript-function, that changes the value (see below) or use *Model Processor*. By default, it is not possible to change an environment variable in a Creo Parametric session.

**Advantage**
The project number can be determined in any Creo Parametric situation.

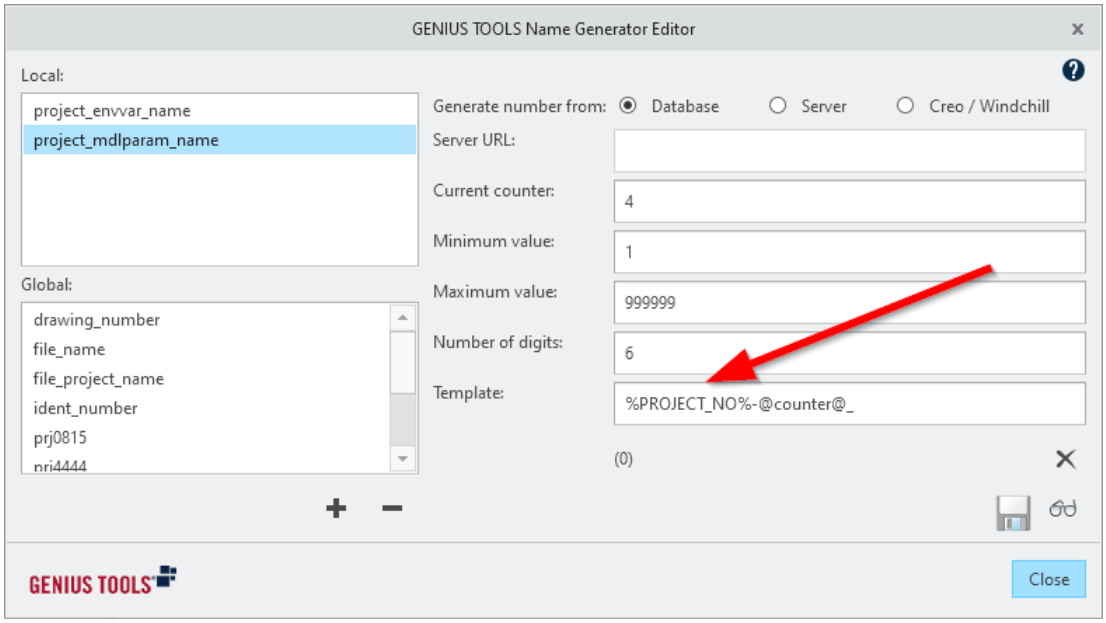

© 2026 INNEO Solutions GmbH

**Disadvantage**
Additional software functionality is required to change the project number. When changing models from different projects, it is necessary to manually set the project number each time you change.

# Solution for variant A: One counter for all project numbers

## Method 1: Model parameter value as project number

The model parameter is specified in the *Name Generator Editor*.



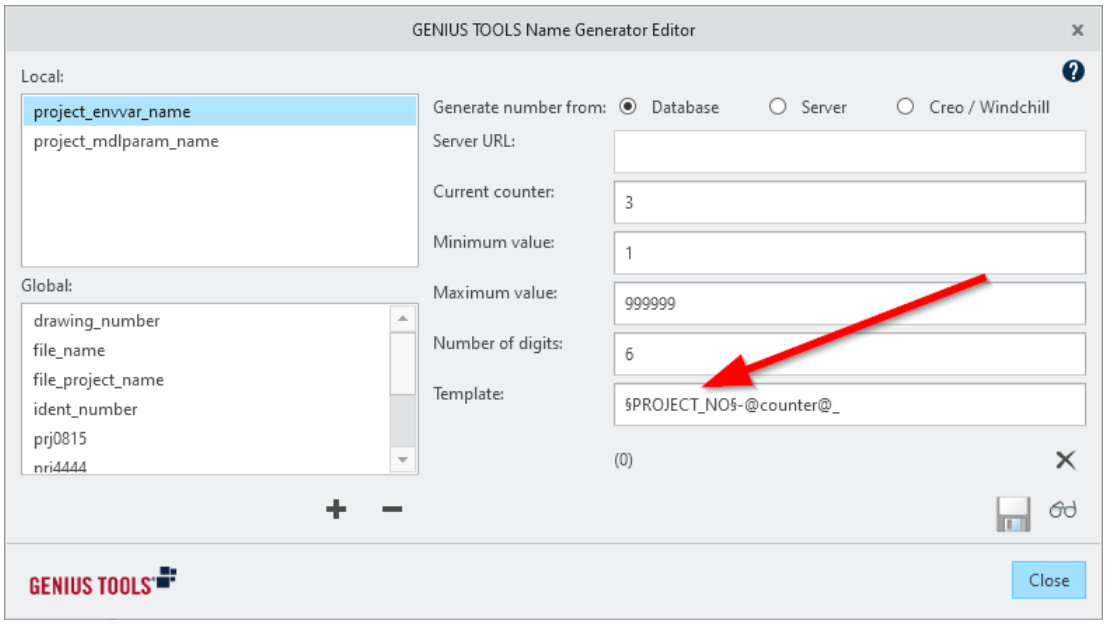

**Please note:** The *Name Generator* can only generate a valid name if a part or assembly is active that also has the corresponding model parameters.

**Create or change model parameter value**
The first assignment or change of the model parameter value takes place via the parameter dialog of Creo Parametric or *Parameters*.

## Method 2: Operating system variable value as project number

The operating system variable is specified in the *Name Generator Editor*.


© 2026 INNEO Solutions GmbH




**Please note:** The *Name Generator* can only generate a valid name if an operating system variable with corresponding model parameters exists.

### Generate, change or view operating system variable value

An operating system variable in a Creo Parametric session cannot be created, modified, or viewed using standard Creo Parametric functionality. You can, however, use the JavaScript function *SetVar()*, which creates or changes the value of an operating system variable, e.g. by using *JavaScript Editor*[784].

```
function SetVar () {
  setEnvVar ("PROJECT_NO", "0000");
}
```

If you combine this with an „intelligent" mapkey in *Quick Access*, users can enter a project number in a separate dialog window (1).

The display of an operating system variable value can also be implemented with an intelligent mapkey (2).

### (1) Change operating system variable value by user input

You can create a mapkey or button in *Quick Access* that allows users to enter a project number which will be transferred to the operating system variable. (See also Creating mapkeys[557].)

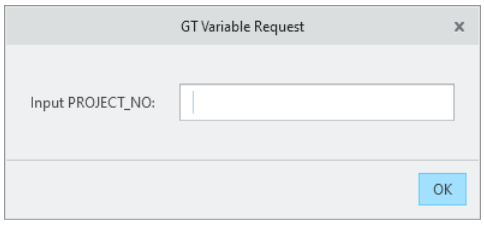
A *Quick Access* command might look like this: (See also Use of variables for user input[913] as well as Javascript function[788] *replaceVars ()*.)

```
js:var pn = replaceVars("§PROJECT_NO§");
var regexp = new RegExp("^\\d{4}$");
do {
var pn = getStringUI("Input PROJECT_NO:");
```


© 2026 INNEO Solutions GmbH

```
if (!regexp.test(pn)) {
alert("The project number must consist of 4 numbers.");
}
} while (pn != "" && !regexp.test(pn));
if (pn != "") setEnvVar("PROJECT_NO", pn);
```

After clicking on the defined mapkey or button in *Quick Access*, this dialog appears.

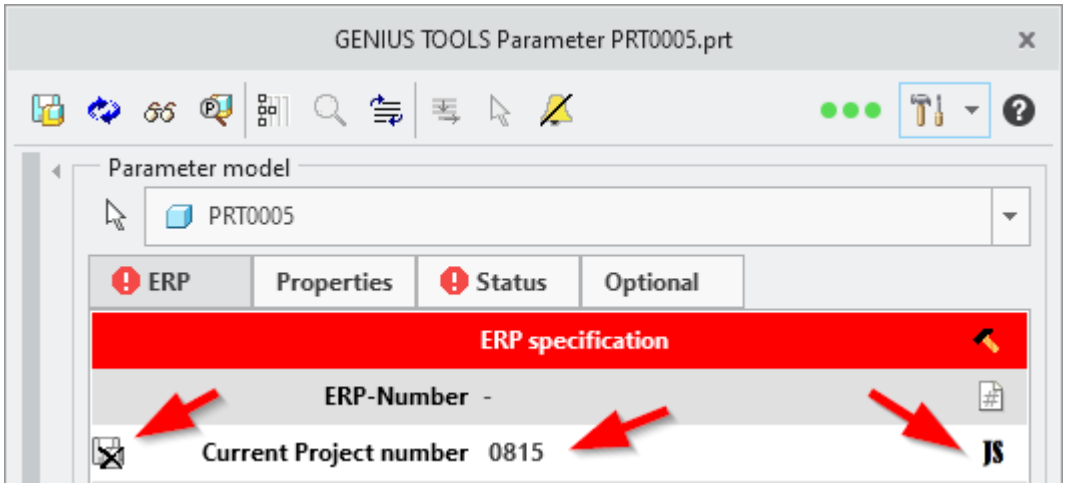



**Alternative**

A Javascript function can also be triggered in *Parameter*, e.g. when opening *Parameters* or after changing a model parameter value or as a click function on a parameter.

```
function PostLoad () {
   setInputValue ("PROJECT_NO_ENV", getEnvVar ("PROJECT_NO"));
return true;
}
```

```
function SetVar() {
  setEnvVar("PROJECT_NO", getInputValue("PROJECT_NO_ENV"));
  alert("Project number "+getEnvVar("PROJECT_NO")+" set!");
}
```




However, a model must always be available for this route. We recommend offering of offering both, *Quick Access* as well as *Parameters* (click function) so that querying and setting project numbers can be carried out conveniently in different ways.

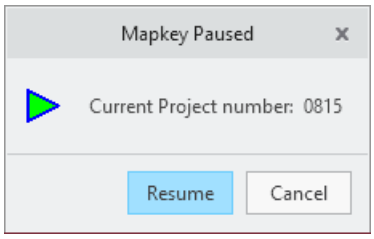
**(2) Display operating system variable value**

The following intelligent mapkey can be stored in *Quick Access* to display the current project number. Mapkeys can include variables by default.

```
@MANUAL_PAUSECurrent project number: $PROJECT_NO$;
```


© 2026 INNEO Solutions GmbH

The output then looks like this:




## Solution for variant B: One counter for each project number

If each project number shall receive a counter, a *Name Generator Definition* (database) must be created for each project number (e.g. prj0815.db, prj4711.db, prj1234.db). It is irrelevant whether model parameter values or operating system variable values are used.

The manual creation and maintenance of these databases is very time-consuming. *Name Generator* offers the possibility to create these databases automatically. The principle works like this:

– Search for a project number counter in data base.

– If it does not exist, a new data base is created using a configured template file.

A template database can be stored with the configuration option `gtng_copy_template_if_filter_db_not_found` (e.g. project number). The definition can look like the one shown in variant A.

When using *Name Generator* within *Quick Access* or in the copying rules of *Library*, you only need to specify the required generator:

```
@number:prj§PROJECT_NO§@
```

If this is not found, it is generated automatically.

**Please note:** If this technology is used, every user needs write rights in the directory of the configuration option *gtng_folder*.

**Example: Count number for Javascript**

If the count number of a project-specific number generator is required in Javascript, the function (e.g. for *Parameters*) could look like this:

```
function GetProjectSubNo () {
var numgenname = getInputValue("PROJECT_NO");
setEnvVar("PROJECT_NO", numgenname);
var new_num = replaceVars("number:" + numgenname + "");
setInputValue("PROJECT_SUB_NO", new_num);
}
```


© 2026 INNEO Solutions GmbH

# 17 Orientation Manager

Available in the ribbon menu GT MBD

The module *Orientation Manager* allows you to create and update orientations. It is also possible to assign combined views to the orientations.

The following defined orientations can be created:

- Rectangle projection
- Isometric projection (from above/below)
- Trimetric projection (from above/below)

Associated annotation planes are created for the defined projections, which are filtered in the *Annotate* tab when switching between combined views. This makes it easier for the user to select the annotation plane when creating new annotations.

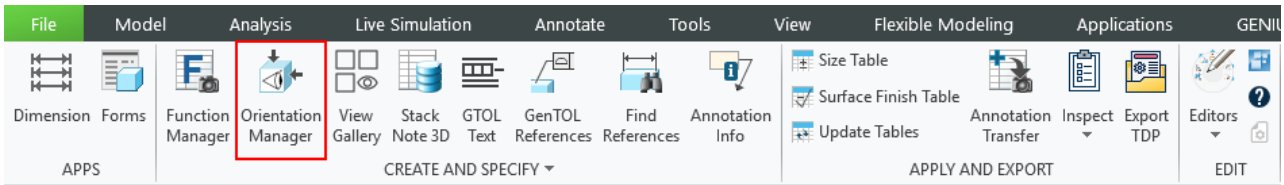
Furthermore, the Orientation Manager displays the number of times each orientation is used in the combined views.

## 17.1 Usage

In this section you will find information on the use of *Orientation Manager*.

### Starting the program: in part mode and in assembly mode

To start *Orientation Manager*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




*Starting via the GT MBD in the ribbon menu*

### Configuration

**gtom_projection_menu_type**
Defines whether the menu is displayed as a drop-down bar (0) or with radio buttons (1).
Default setting: 1

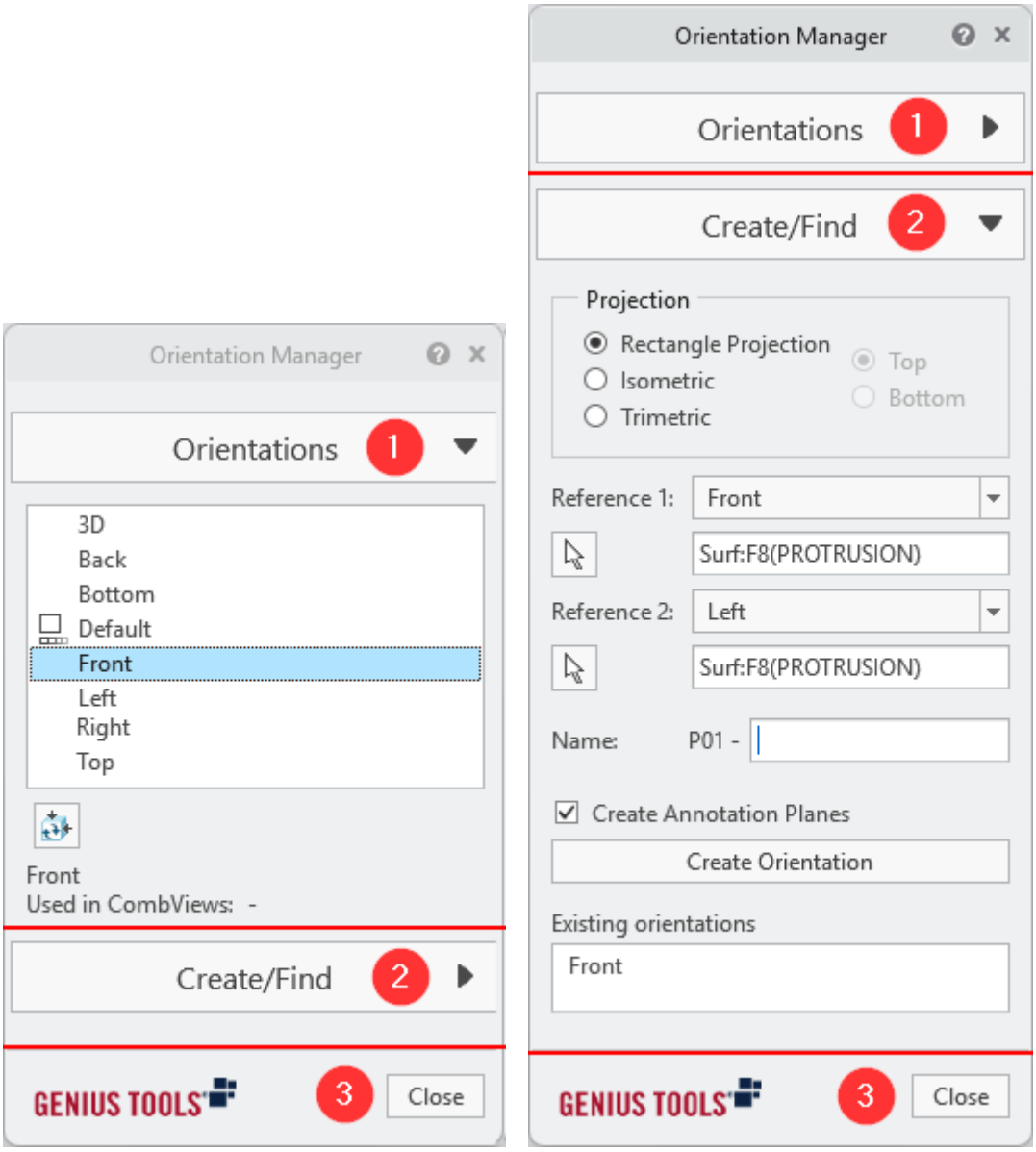

© 2026 INNEO Solutions GmbH

**gtom_projection_type_init**
Defines which option is selected by default: (0) Rectangle Projection, (1) Isometric Top, (2) Trimetric Top. Default setting: 1

**gtu_ui_change_filter_annotation_planes**
Defines whether GENIUS TOOLS should filter the annotation planes based on the orientation set in the active combined view (1) or not (0). Default setting: 1

## 17.1.1 User interface

The user interface of *Orientation Manager* consists of the following elements:




1. List of orientations with help button
2. Creating and finding orientations[463]


© 2026 INNEO Solutions GmbH

3. Close

## 17.1.2 Command bar

The following button is included in the command bar:

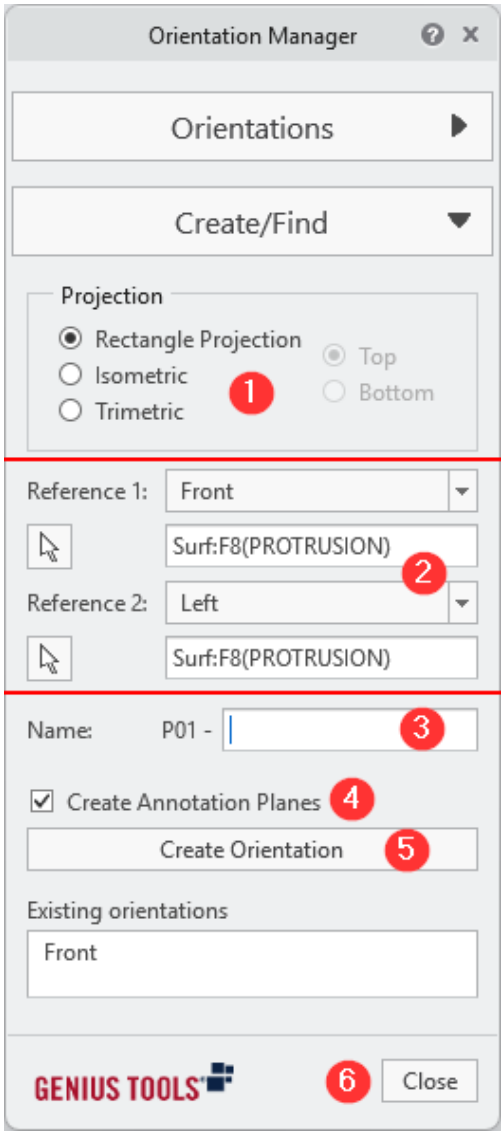
| Icon | Name | Description |
|---|---|---|
| | Help | Opens the help. |

This function includes the following icons:

| Icons | Name | Description |
|---|---|---|
| | Update the assigned orientation of the active CombView | The orientation of the combined view is overwritten after the change. |
| | Select Reference | Enables selection of the reference in the model to be assigned to the entry from the drop-down menu. |


© 2026 INNEO Solutions GmbH

## 17.1.3 Create Orientation




1. Select the projection
2. Click *Select Reference* and select a reference in the model
3. Name the orientation
   - PXX - refers to rectangle projection
   - IXX - refers to isometric projection
   - TXX - refers to trimetric projection
   - XX - refers to the numbering
4. Check the box if you want to create annotation planes for the respective orientation.
5. Click *Create Orientation*
6. If the same orientation already exists, they will be displayed in the list.


© 2026 INNEO Solutions GmbH

**Tip:** By right-clicking on an existing orientation in the list you can set it as the active combined view.


© 2026 INNEO Solutions GmbH

# 18 Parameter

Available in the ribbon menu GENIUS TOOLS

*Parameter* is a program to easily edit meta data (parameters) of parts, bodies, assemblies and drawings in Creo Parametric.

In assembly mode, the parameters of any sub-model can be edited. In drawing mode, you can switch between the drawing parameters and the parameters of the drawing models. Parameters to be edited (name, type, displayed text, etc.) are defined in the parameter definitions. They can be created and edited with the included *Parameter Editor*.

It is also possible to work with multiple parameter definitions. The parameter definitions that will be used, can be specified manually or automatically, depending on the parameters (selection parameters) of the active model.

*Parameter* is available in assembly mode, part mode and drawing mode providing the following features:

1. A classifiable meta data definition including
   - an auto-selection mechanism for parameter definitions
   - freely configurable groupings
2. Easy parameter creation
   - with default properties for all types
   - specification for effective ranges (parts, bodies, assemblies and drawings)
3. Easy editing of parameters by means of
   - free input
   - lists and tables (from files and databases)
   - auto-suggestion function
   - format checks
   - input dependencies
   - adoption of parameters from other Creo models
4. Individual model tree configuration


© 2026 INNEO Solutions GmbH

# 18.1 Fundamentals

This introductory section gives you an insight into the operating principles of *Parameter*. In addition, you will find a glossary of important terms for a better understanding.

## 18.1.1 Glossary

**Selection parameter**
The selection parameter is a Creo model parameter. This parameter and the *Parameter* parameter definition list are used to determine which parameter definition will be applied to a model. The parameter used as the selection parameter by default is `MC_CHECKTYPE`.

**Field function**
Field functions are predefined functions to determine parameter values in *Parameter*. They are displayed as buttons after the entry fields in the parameter form.

Field functions can set parameter values in different ways, e.g., set the current date or a model parameter value from Creo.

**Parameters**
Parameters in terms of *Parameter* define the display and behavior when assigning values in *Parameter*. Creo model parameters are driven depending on the properties of a parameter.

**Parameter definition and sub parameter definition**
A parameter definition contains parameters, separators and references to sub parameter definitions. A sub parameter definition contains only parameters and separators. The information is stored in an XML data structure.

**Parameter model**
The parameter model is the Creo model, whose parameter values are currently displayed in *Parameter*.

**PTC Common Name**
The parameter *PTC_COMMON_NAME* is treated like any other parameter in *Parameter*. Please note that this parameter is used by Windchill and must not be changed after a model has first been checked in to Windchill. It is recommended to make this parameter read-only.

The possibility to rename Windchill objects in Creo is also controlled by the configuration option `let_proe_rename_pdm_objects`.

**Rolling**
Rolling allows to pass incremental values stepwise in a Creo parameter chain. The value in the last Creo parameter will be discarded. The purpose of rolling is to obtain history of parameter values.


© 2026 INNEO Solutions GmbH

**Separator**
Separators are elements used to subdivide the parameters in the element list of *Parameter Editor* and the parameter sets in the parameter form in *Parameter*. Separators are also needed for the two functions: *reset* and *execute field function*.

**Table connection**
A table connection is the connection between the values of two parameters. Parameter values can either be controlled by another parameter or filtered.

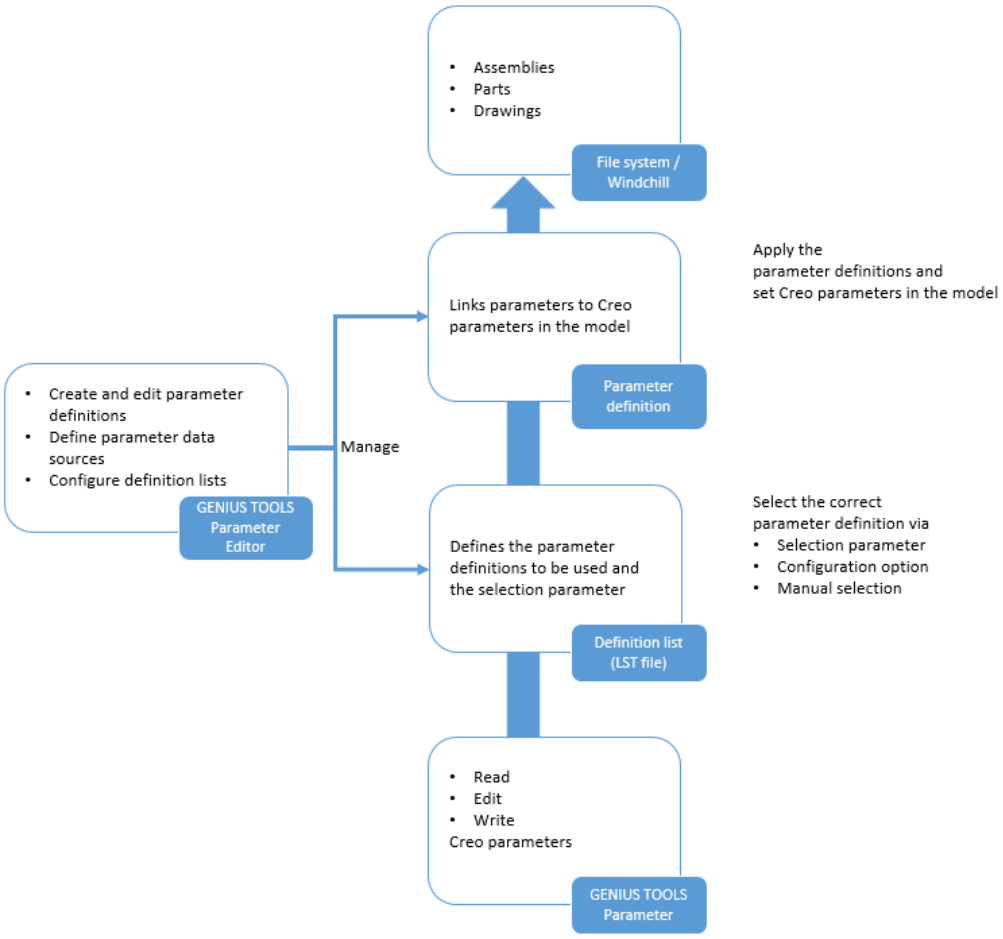
## 18.1.2 Parameter management concept

One of the major advantages of working with a uniform database in Creo Parametric is that both geometric information and administrative information is pre-defined. The benefit of this is that the information for drawing header, bills of material, and position flags in assembly drawings only needs to be created and maintained once. Administrative information in terms of Creo Parametric are parameters that are saved along with a part, a body, an assembly or a drawing. Examples of such parameters could be *Name*, *Description*, *Material*, *Weight*, and *Drawing number*.

When creating a drawing of a part or an assembly, the drawing header can be filled in automatically with these parameter values. When building assemblies, a current bill of materials can be generated at any time. The content of the BOMs (number and content of the individual columns) is freely configurable. The position flags on the assembly components can be automatically filled in with any information from these bills of material.

*Parameter* was developed to guarantee this uniform database. The program uses parameter definitions, which contain all usable parameters, to display them in a configurable view. For this purpose, a definition exists for each parameter. In addition to the parameter name, these definitions also contain localized descriptions and determine from which data sources (free input, text files, databases, etc.) the parameters can be filled. The parameters are displayed in a form and can be easily filled in by user input.


© 2026 INNEO Solutions GmbH




## Steps for preliminary considerations on the parameter concept

– Analyze the existing title boxes

– Analyze the BOMs

– Analyze design details in the ERP

## 18.1.3 Mechanisms on starting and saving

The parameters that are shown and changed in *Parameter* are specified in parameter definitions. It depends on the configuration which parameter definition will be applied to a Creo model and which parameters will be saved to the models.

For a better understanding, this section outlines the general mechanisms on starting *Parameter* and when saving parameters.


© 2026 INNEO Solutions GmbH

## Start mechanism of Parameter

1. Determines the parameter definition to be used (see below *Selecting the parameter definition*)
2. Reads parameters from the current model or assembly component and checks PDM status
3. Assigns parameters automatically as specified by the configuration of *Parameter*
4. Displays the parameters in the form section of *Parameter*
5. Output of status information (status indicator)

## Selecting the parameter definition

**Automatic selection based on a model parameter**
Automatic selection based on a model parameter (selection parameter) is the default behavior of *Parameter*. A parameter in the Creo model is compared with a list of parameter definitions.

Important criteria for the selection based on a model parameter are the configuration options `gtp_lst` and `gtp_file_param`.

The configuration option `gtp_lst` specifies the location of the file containing all parameter definitions in list form. By default, this file (*gtp.lst*) is located at *<GTfCInstallation>\gt_resource_folder\parameter*.

The list contains all available parameter definitions in following format:

```
Location|Description|ParameterValue
```

**Example**: `%GT_RESOURCE_FOLDER%\parameter\gtp_sut_int_de\pmm_int_de.xml|Production|PRODUCTION`

The configuration option `gtp_file_param` specifies the selection parameter. If the value in the model parameter matches with a parameter value in the list, the corresponding parameter definition will be applied to the model.

**Please note**: If a list file contains duplicate parameter values, only the last entry will be applied.

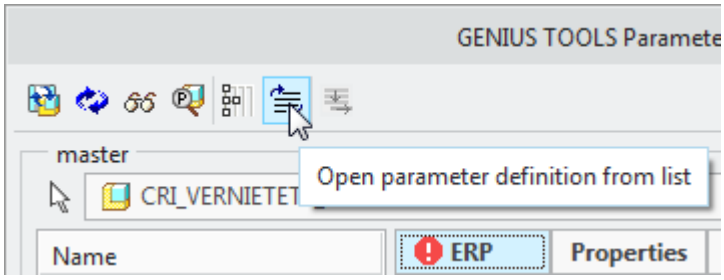
The chapter Auto-select parameter definitions 533 describes the procedure step by step.

**Automatic selection based on a configuration option**
If the selection based on a model parameter returns no result, it will be checked whether a parameter definition has been specified via the configuration option `gtp_file`. If a valid parameter definition is returned, it will be applied to the model.


© 2026 INNEO Solutions GmbH

**Manual selection**
If the automatic selection does not return a valid parameter definition, or if another parameter definition should be used, parameter definitions from the parameter definitions list can also be applied manually to a model.

Make sure to not select a sub parameter definition when selecting manually!




*You can also assign parameter definitions to a model manually with this button*

## Storage mechanism of Parameter

After changes in the parameter form in *Parameter*, the following mechanism will be applied when saving the parameter values:

1. Verification of the parameter values according to the rules defined in the parameter definition
2. Output of warning messages (depending on the configuration)
3. The values are adopted by the model parameters:
   a. designations are removed (depending on the configuration)
   b. revised parameter values are adopted into the model
   c. designations are set
4. The model is being regenerated (depending on the configuration)
5. *Parameter* is completely reloaded with the new model parameters
6. If the option is selected, the model can also be saved.

**Warning**: Changes to the selection parameter or manual selection of a parameter definition may lead to loading a different *Parameter* definition!

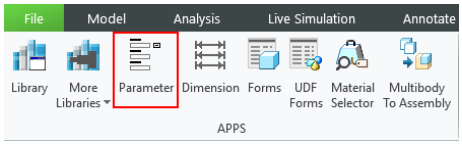
# 18.2 Usage

This section contains information on using *Parameter*. It describes the general structure of the program.


© 2026 INNEO Solutions GmbH

# Starting the program

To start *Parameter*, click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

The program opens the parameter form for a specific parameter definition. Selection can be carried out manually or automated; see the section Selection of parameter definition[469] .

# Automatic start when creating new models

To automatically open *Parameter* for specific file types, set the configuration option `gtl_gtp_start_gtp_after_model_creation` for a part, assembly, drawing.

To prevent *Parameter* from starting automatically for specific file types, specify the desired types in the configuration option `gtp_ignore_start_gtp_after_model_creation`.

# Modality of the dialogs

The modality of the Parameter dialogs (locking the rest of the application or individual dialogs) is determined by the configuration option `gtp_dock_dialog_to_mdl`. You can edit multiple models using the same Parameter dialog and also open and operate the editor at the same time. If you change modality to set `gtp_dock_dialog_to_mdl=0`, *Parameter* will remain open and available when you change the model.

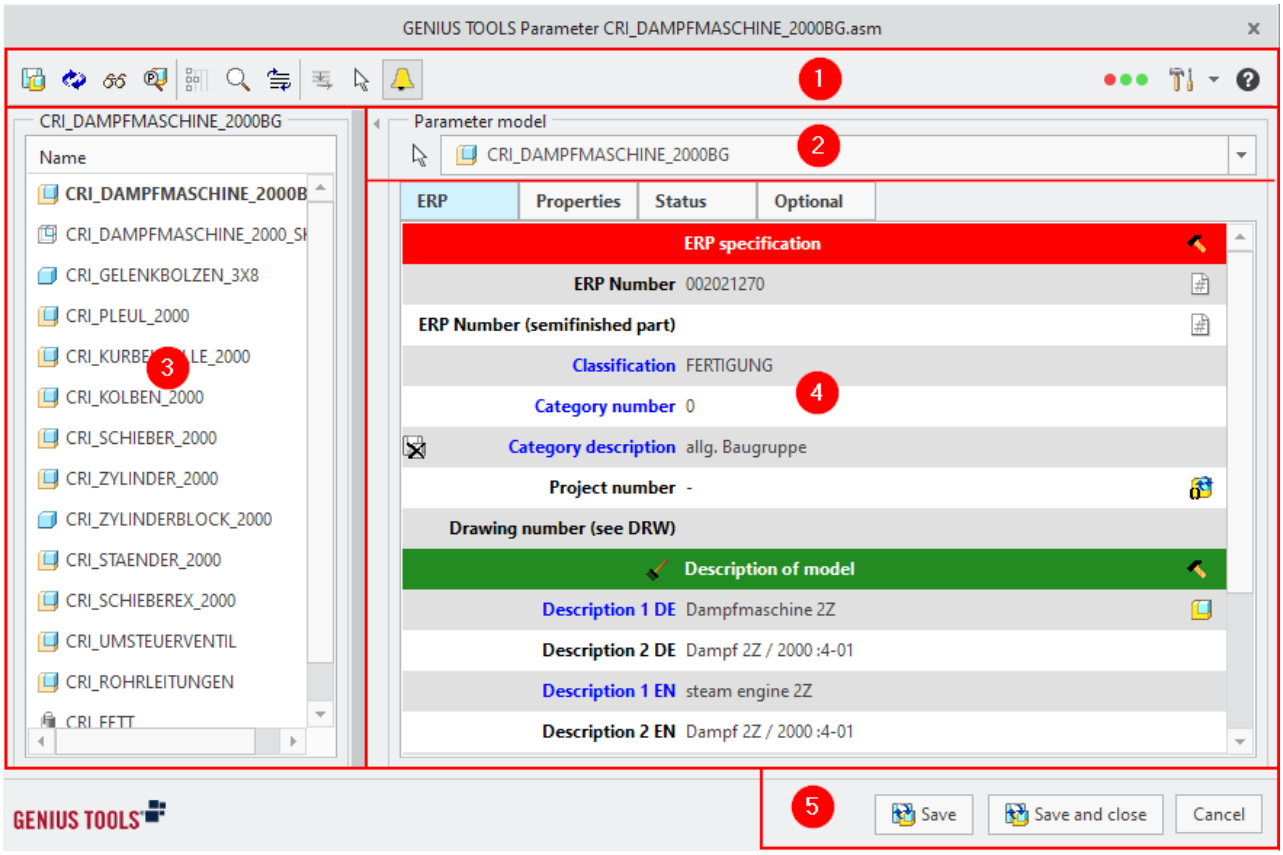
Click *Discard changes and reload* to make the current Creo model the parameter model and load the corresponding parameter definition.


© 2026 INNEO Solutions GmbH

**Warning**: You can only change parameter values if the active model is the parameter model! Save any changes to the parameter form before you change the active model.

## 18.2.2 User interface

The user interface of *Parameter* consists of the following elements:




1. Command bar[473] with status indicator[476] and tools menu[475]
2. Model selection[477]
3. Model list[478]
4. Parameter form[480]
5. Save / Cancel buttons[472].

## Save and cancel

Provides the option to save the model if the option is set.

The *Save* button turns green when changes are made to the parameter form. Click on the button to apply changes in the parameter form to a Creo model. The status display is updated. *Save and close* skips updating the status display. If errors occur, the parameters will not be saved.


© 2026 INNEO Solutions GmbH

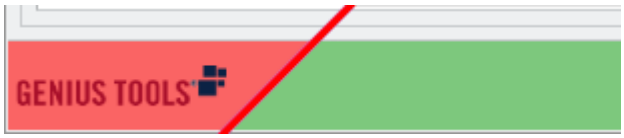
The footer of the parameter window is colored depending on the save status:

**Green:** Parameters were transmitted, no warnings or errors occurred during transmission. The PreSave method returned `true`.

**Yellow:** Parameters were transferred, but warnings or errors occurred during transfer. The PreSave method returned `true`.

**Red:** Parameters were not transferred because conflicts exist and the configuration options `gtp_do_not_save*` are set to 1.

Alternatively, the PreSave method returned `false` and therefore no parameters were transferred.




The footer of the Parameter window shows the save status.

**Warning:** Note the modality of the Parameter window! By default, *Parameter* is closed when other windows in Creo Parametric (Creo standard window and *GENIUS TOOLS*) are opened.

## 18.2.3 Command bar

The command bar contains the following buttons:

| Icon | Name | Description |
|---|---|---|
| | Save | Saves all parameters and their corresponding values in the model.<br>**Note**: Depending on the configuration, saving may only be possible if there are no conflicts. |
| | Discard changes and reload | The current parameter definition and the parameters of the model are reloaded.<br>**Note**: Unsaved changes will be discarded! |
| | Adopt model parameters | Model parameters are read again, the parameter form is refilled.<br>**Note**: Unsaved changes will be discarded! |


© 2026 INNEO Solutions GmbH

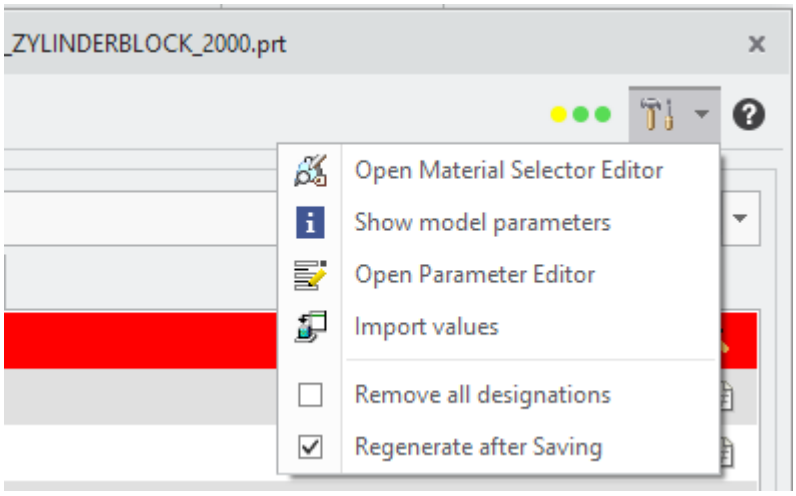
| Icon | Name | Description |
|---|---|---|
|  | Select model and adopt parameter values | After selecting a model in the Creo window its model parameters are read and displayed in the current parameter form.<br>**Note**: Unsaved changes will be discarded! |
|  | Update model tree | Reloads the model tree and displays it in the Creo Parametric sidebar.<br>Parameters, flagged for the model tree in the editor will be additionally shown in the model tree.<br>The model tree can only be adjusted if the *Parameter* is not bound to the Creo Parametric window (`gtp_dock_dialog_to_mdl=0`). |
|  | Highlight current model | Highlights the model selected in the model list in the Creo window. |
|  | Open parameter definition[494] from list | Opens an existing parameter definition and applies it to the model parameters of the current model. |
|  | Roll values | Executes rolling when configured in the current parameter definition.<br>**Note**: Button is only activated in drawings. |
|  | Select start model | Displays a model selected in the Creo window or in the model tree in the parameter form. |
|  | Activate or deactivate Save message | Activates/ deactivates in assemblies a warning message for unsaved changes in parts. |
|  | Status indicator[476] | Shows the current status of *loading*, *working* and *saving* using traffic light colors and opens the status dialog. |
|  | Tools | The Tools menu contains various supporting functions. |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
| | Help | Opens the Help. |

# Tool menu

Supporting functions are available in the tool menu :




**Open Material Selector Editor**
Opens the *Material Selector Editor* to modify the material properties.

**Show model parameters**
Shows model parameters of each opened part (prt, asm, drw) in an extra window. The function only opens a view. No changes can be made, but parameter values needed can be copied.

**Open Parameter Editor**
Opens the *Parameter Editor*[490] to edit the parameter definitions.

**Import values**
Imports a text file or a CSV file whose content is written into the corresponding parameters.

**Remove all designations**
Specifies whether all designations are removed in Windchill mode. The display of this checkbox can be hidden by setting the configuration option `gtp_designate=0`.

**Regenerate after saving**
Specifies whether models will be automatically regenerated after clicking *Save*. The display and standard setting of this checkbox can be changed in the configuration option `gtp_regen` as follow:

0: display unchecked box
1: display checked box
2: automatic regeneration and no display of checkbox


© 2026 INNEO Solutions GmbH

-1: no automatic regeneration and no display of checkbox

## 18.2.4 Reading parameter values from text files

You can import parameter values from text files into the parameter form. To do so, use the function *Import values* in the tool menu.

The file *GTP_values.txt* is read from the work directory and the values are written to the corresponding parameters. After importing, table connections and JavaScript OnChange functions are executed.

The path to the directory can be defined using the *gtp_import_folder* configuration option. The default setting *@curworkdir@* is the working directory.

**Warning**: The function *Import values* also writes read-only fields!

Values can be read from files with the following character encodings:

– UTF16LE/BE

– UTF8 BOM

– ANSI

The text file has to follow the format given below with a parameter name and a value separated by a colon in each line.

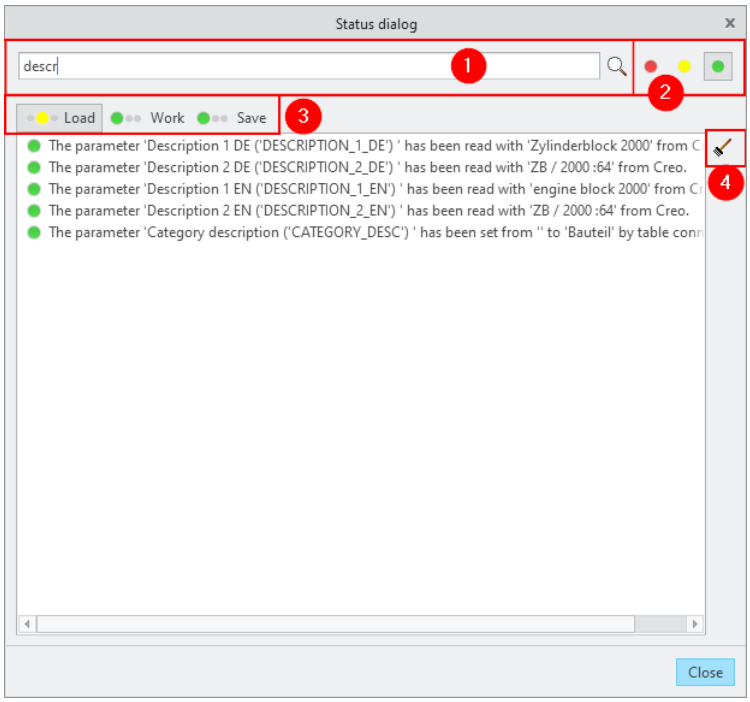
**Example**

```
DESCRIPTION_1_DE:Abdeckkappe
DESCRIPTION_1_EN:Cover cap
PART_NO:004526-12
ARCHIVE:YES
```

## 18.2.5 Status dialog

The command bar contains the status dialog, which is used to track successful and faulty work steps. This dialog's functionality is consistent across all modules.

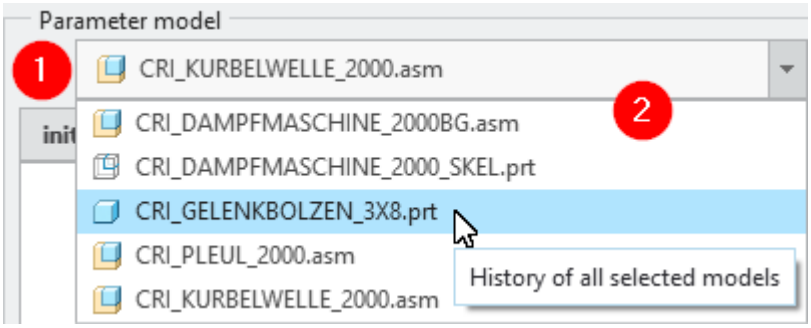

© 2026 INNEO Solutions GmbH




1. **Text search:** Searches for message content in all tabs. Press ENTER or click the magnifying glass icon to begin.
2. **Traffic light buttons:** Use the three traffic light buttons (2) to turn the displayed messages (errors, warnings and/or successes) on or off.
3. **Load, Work and Save:** The messages are divided into these three tabs. For each tab, a separate status traffic light provides an overview of the status of that work step. Whether there is a warning or an error is indicated by the traffic light color in the tab.
4. **Reset:** Deletes the displayed status messages. These messages will only reappear after the module is restarted.

## 18.2.6 Model selection

The model selection consists of two elements:




*The model selection with object selection (1) and drop-down list (2)*


© 2026 INNEO Solutions GmbH

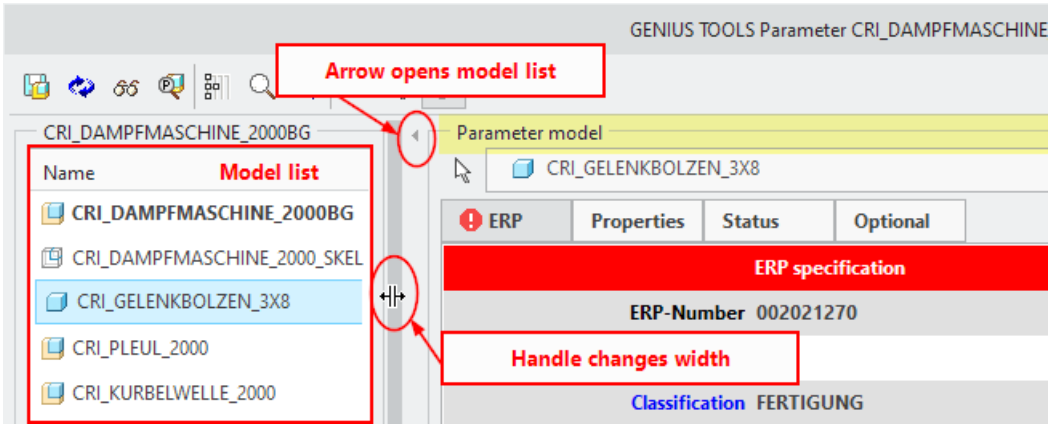
Select models using the Object selection (1). Their parameters will be read into the parameter form.

The drop-down list (2) shows the current selection. Open the list and select from the recently selected models.

The name of the current parameter definition is displayed above the model selection. Hover over the name of the definition to see its storage location as a tooltip.

## 18.2.7 Model list

The model list opens on the left below the command bar when clicking the arrow next to *Parameter model*.




*Model list and parameter model CRI_GELENKBOLZEN_3X8*

Click on a sub-model to display its parameters in the parameter form. The model, for which the parameter values are displayed, is referred to as *parameter model*.

### Mode dependency of the model list

The model list depends on the configuration option `gtp_show_mdl_list`.

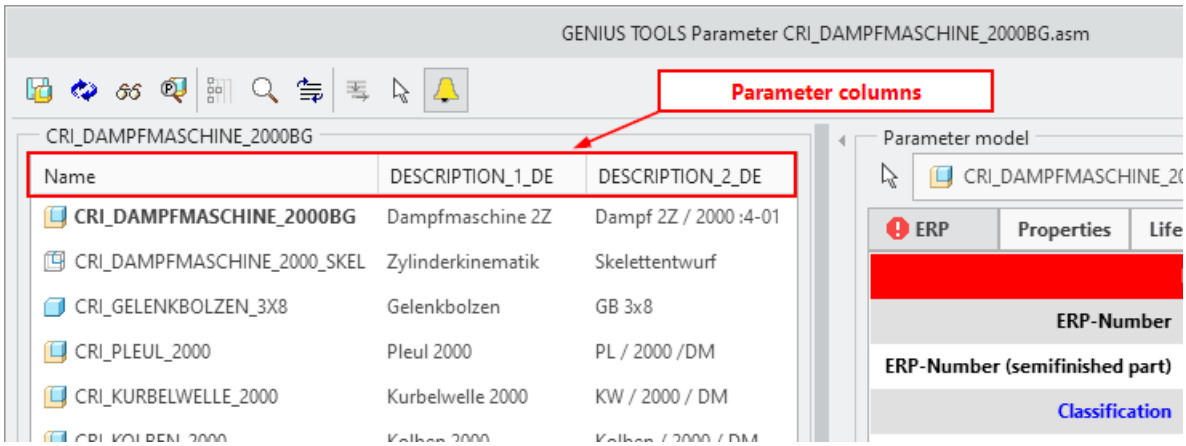
If the configuration option is set to 0, the model list is not displayed. If the configuration option is set to 1, the model list is displayed. If the configuration option is set to 2, the model list is displayed or not, depending on the model type (for example, show list for assemblies and for generic parts with instances).

If the configuration option `gt_window_size_position_save` is set to 1, the size of the model list is saved.

### Adding extra columns for parameter

You can add additional columns for parameters in the model list by entering the parameters needed, separated by comma, into the configuration option `gtp_model_tree_columns`. The option `gtp_model_tree_column_width` allows you to set the width of these additional columns, measured in the Windows standard width of a character. The default value is 10.

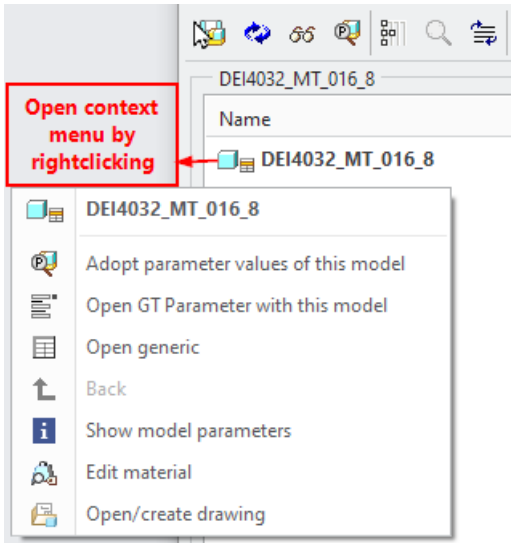

© 2026 INNEO Solutions GmbH




*Additional columns for the parameters DESCRIPTION_1_DE and DESCRIPTION_2_DE*

## Context menu

There is a context menu for the model list, which appears when you right-click on an element.



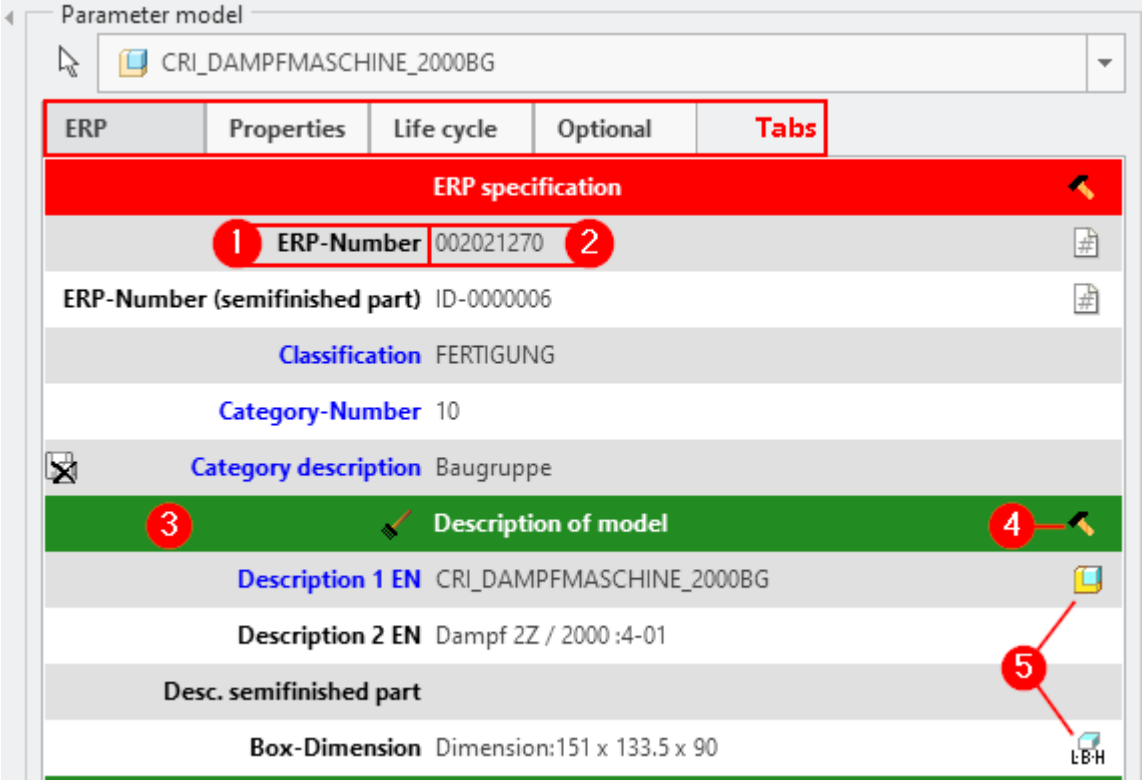

- **Adopt parameter values of this model**: The parameters of the selected model are transferred to the fields of the parameter model in the parameter form. Ensure you save after executing this action to apply the values to the parameter model.
- **Open GT Parameter with this model**: Opens the selected model in *Parameter*.
- **Open generic**: Opens the instance as a Generic.
- **Back**: Switches the view to the previous model.
- **Show model parameters**: Opens a new window displaying an overview of all the selected model's parameters. It is not possible to make changes via this window. If you need parameter values from this overview, copy them.
- **Edit material**: Opens *Material* for the selected model.
- **Open/Create Drawing**: Opens the model's drawing or generates a new drawing.


© 2026 INNEO Solutions GmbH

## 18.2.8 Parameter form

The parameter form displays parameter sets of the currently selected model – depending on the parameter definition.[494] Only those parameters specified in the editor are displayed in the parameter form.

Each **parameter set** consists of the localized parameter title (1) and the model's parameter value in the input field (2).




Click on an input field to activate it.

Use the Tab key (TAB) to switch to the next input field and Shift+Tab to switch to the previous field.

Click on the localized parameter name to display additional information about this parameter.

The background color of the heading bar (3) depends on the configuration in the parameter definition.

## Input fields

Enter the parameter values either manually or via predefined automatic field functions (5). The *Auto-fill* and *Reset functions* affect input fields up to the next separator.


© 2026 INNEO Solutions GmbH

Click the *broom symbol* to reset the input fields to their initial value. Whether an input field can be reset depends on its configuration. The setting is made in the parameter definition under *Resettable*. See Adding parameter > Field: Value[500].

*Execute field functions* automatically executes the field functions that have been defined for fields. If more than one function has been configured for a field, the first one will be executed. Only functions that do not require user interaction will be executed.

To create the hammer symbol, refer to the chapter Adding separators[514].




The different field types of the input fields are listed in the section Field types[484].

# Tabs

Tabs will show up above the parameter form if a parameter definition contains parameter subdefinitions[494]. The tabs subdivide parameters for better accessibility.

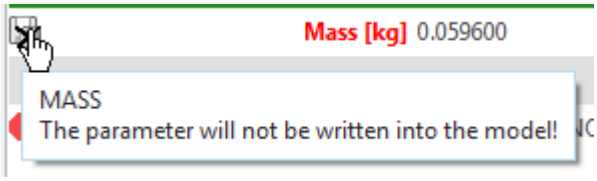



If there are any unfinished tasks for a sub-parameter definition (e.g., an unfinished mandatory field), a warning symbol (exclamation mark) will appear before the name in the tab.

If you have previously opened *Parameter* in a Creo model, the program will remember which tab you were working on when you stopped.

# Info symbols in parameter sets

Info symbols appear before the localized name to provide important information about a parameter. Hover over the icon for more information.




*Info symbols before the localized name*

The following info symbols are displayed:

| Icon | Description |
|---|---|
| | The parameter will not be included in the model when it is saved. |


© 2026 INNEO Solutions GmbH

| Icon | Description |
|---|---|
|  | The parameter is not contained in the model.<br>Saving creates the parameter and transfers its value into the model. |
|  | The input field is a mandatory field and must be filled in. |
|  | The input does not correspond with the specifications for the input field. For example, letters in a number field. |
|  | The parameter is locked by a relation. |
|  | The model is an instance, the parameter is not in the generic's family table. |

Localized parameter titles are displayed in different colors when one of the following conditions is met:

– Dark blue: The parameter is filled by the database connection.

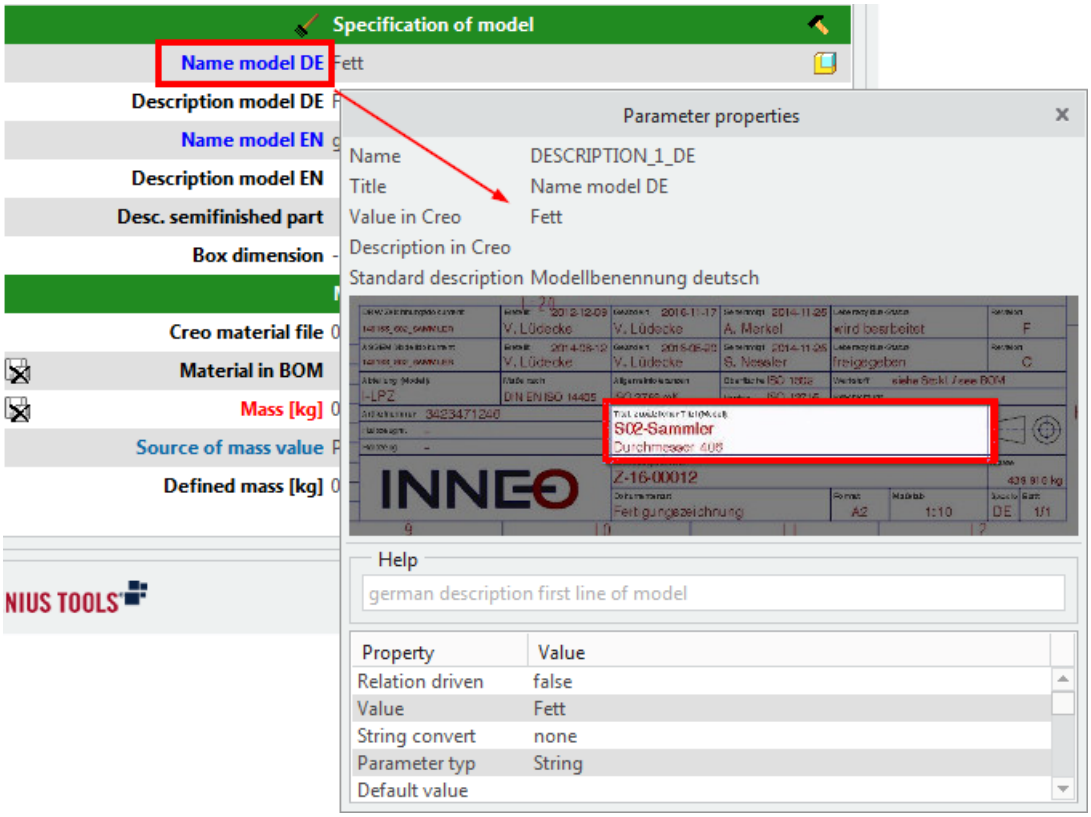
– Light blue: The parameter is filled by a CSV file.

– Orange: The parameter is controlled or locked by a relationship.

## Parameter properties

Additionally, you can provide more information on parameter sets. Click on the localized parameter title to open the *Parameter properties* dialog. The dialog shows properties and information about the selected parameter.


© 2026 INNEO Solutions GmbH




*Properties dialog of the parameter*

## 18.2.8.1 Calculating / assigning masses

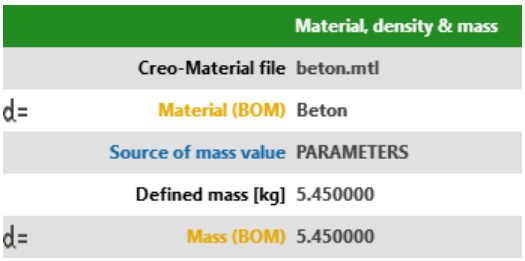
There are several ways to calculate mass. The parameter `mp_source` is displayed as the `Source of mass value`.

– **Mass can be calculated using density and volume**
By default, the mass of the part is calculated based on density and volume of a part (`Source of mass value: GEOMETRY`). If a Creo-Material file is selected, the density from this file is used automatically, and the mass is calculated.

| Material, density & mass | |
|---|---|
| Creo-Material file | beton.mtl |
| Material (BOM) | Beton |
| Source of mass value | GEOMETRY |
| Density [g/cm³] | 2.300000 |
| Mass (BOM) | 7.938500 |

– **Determining the mass**
Alternatively, you can specify as `Source of mass value: PARAMETERS`. In this case, the density is ignored and removed from the parameter form, allowing you to determine the value of the mass. This procedure is recommended for simplified models, such as purchased parts and standard parts.


© 2026 INNEO Solutions GmbH




## 18.2.9 Field types

A parameter value is displayed in an input field that can contain predefined values and automated field functions[509].

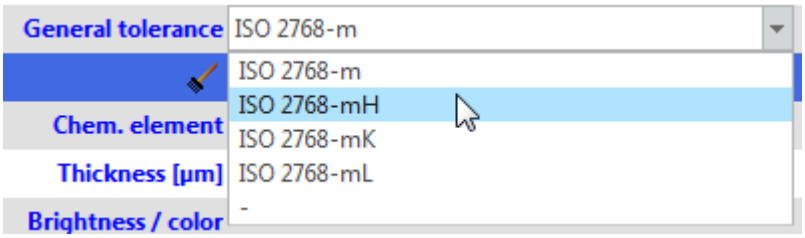
Values for parameters can be entered and reset automatically, see input fields[480].

Depending on the configuration in the editor, the input fields can have a different function:

**Input field**
Regular input fields accept any string for input. Restrictions (e.g. numbers only) are possible through configuration in the editor.




**Drop-down list**
Drop-down lists contain a selection of possible inputs. These lists are specified through text/CSV files or databases.




**Combined input field**
Combined input fields allow you to enter text freely or select from a drop-down list. When assigned to a database, the entries in the drop-down list will be filtered as you type.

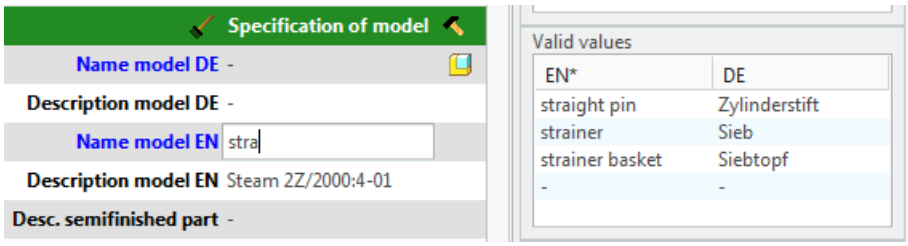



**List selection**
A list selection automatically displays a list with appropriate entries for the current input. This requires a table connection. The columns displayed are specified in the configuration and are retrieved from a CSV file or a database. When entering text to a list selection with database connection, the list is filtered for matching entries. You can also enter regular


© 2026 INNEO Solutions GmbH

expressions[487] (RegEx) to filter the list. Clicking on an entry applies its matching value to the field.

**Please note**: A selection list provided by a webserver as a JSON file cannot be filtered further in the parameter form.




### Material input field

Material input fields behave similarly to combined drop-down lists. The drop-down list displays all materials and subfolders of the material folder. You can select subfolders to navigate the tree. Additionally, you can open the material selection using the magnifier icon.




### Read-only input field

Read-only input fields cannot be edited.




**Warning**: The function *Import values* or JavaScript functions can edit read-only fields.

### Checkbox

Checkboxes allow to choose from two options. These can be yes or no decisions, for example.



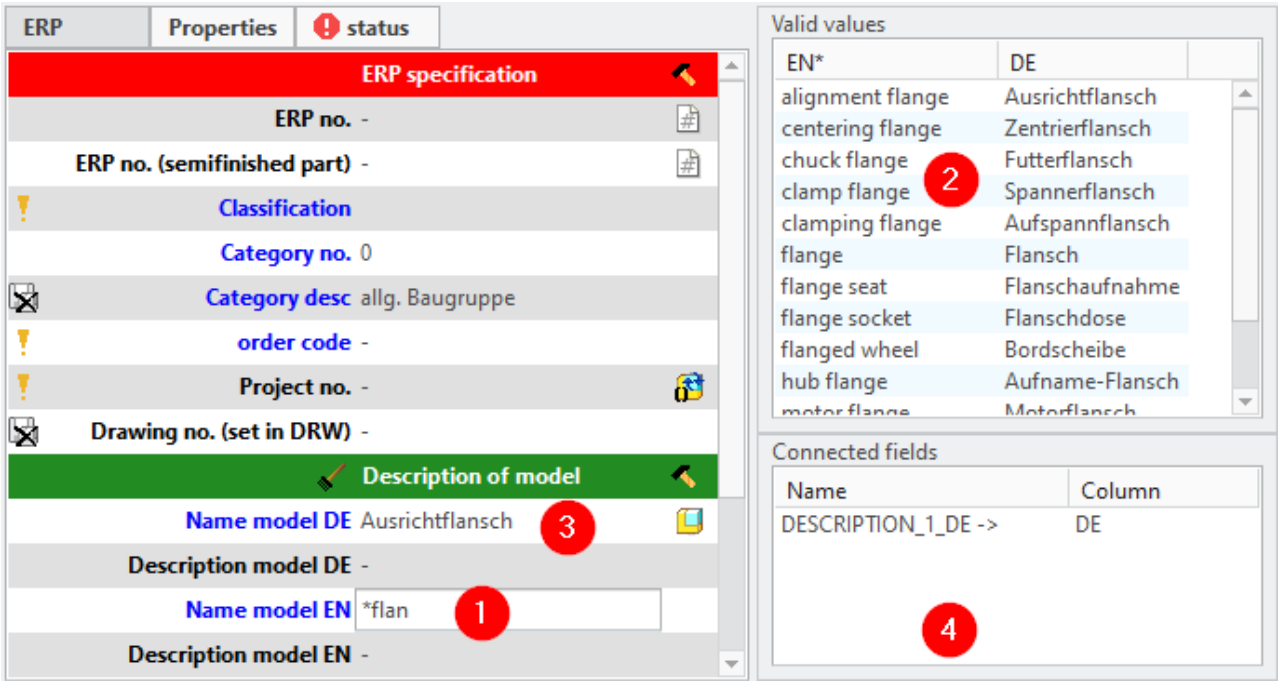

## Field functions

Field functions are predefined functions for determining a parameter value in *Parameter*. Field functions can set parameter values in different ways, e.g. for setting the current date or transfer a model parameter value from Creo.

Click on a field function icon to execute the function. Find a list of all functions in chapter Field functions[509].

## Connected fields

Input fields can be interlinked. This makes them dependent on one another. When one field's value is modified, the appropriate values are automatically entered into the other fields. Information from databases or CSV files is evaluated in the background.




Following the input (1) appropriate values (2) are displayed. After selecting a value, the corresponding value is entered into each connected field (3). The names of the connected fields (4) are displayed below the valid values.

Click on a connected field and enter a search term. You can also use the asterisk (*) placeholder to search for subterms. Then, select the value to be entered from the list. The appropriate values will be adopted by each connected field.

# 18.3 Configuration

This section contains further information on

– available regular expressions for the Parameter Editor, and

– the structure of the Parameter Editor

In addition, you will find information on configuring *Parameter* and the parameter definitions in the section Use cases[516].


© 2026 INNEO Solutions GmbH

## 18.3.1 Regular Expressions

Use Regular Expressions in *GENIUS TOOLS* for Creo to check value inputs or allow only rule-compliant inputs to be saved.

| Character | Description |
|---|---|
| \ | Indicates the following character as a special or verbatim character. For example, “n” corresponds to the character “n”. “\n” corresponds to a line-break character. The sequence “\\” corresponds to “\”, “\(” corresponds to “(”. |
| ^ | Corresponds to the beginning of the input. |
| $ | Corresponds to the end of the input. |
| * | Corresponds to the preceding character zero or multiple times. “zo*” matches e.g. “z” or “zoo”. |
| + | Corresponds to the proceding character one or multiple times. “zo+” for example matches “zoo”, but does not match “z”. |
| ? | Corresponds to the proceeding character zero or one time. For example “a?ve?“ matches the “ve” in “never”. |
| . | Corresponds to all single characters except for a line-break character. |
| (Pattern) | Matches Pattern and saves the equivalent. The compared substring can be retrieved from the resulting matches listing using the elements [0]... [n]. To compare characters in parentheses ( ) use “\(” or “\)”. |
| x\|y | Corresponds to either x or y. For example matches “l\|red” either “l” or “red”. “(l\|r)ed” matches “led” or “red”. |
| {n} | n is a positive integer. Corresponds to exactly n times. “o{2}” for example does not match the “o” in “Robert” but the first two “o”s in “Booooat”. |


© 2026 INNEO Solutions GmbH

| Character | Description |
|---|---|
| {n,} | n is a positive integer. Corresponds to at least n times. “o{2}” for example does not match the “o” in “Robert” but all “o”s in “Booooat”. “o{1,}” is equivalent to “o+”. “o{0,}” is equivalent to “o*”. |
| {n,m} | m and n are positive integers. Corresponds to at least n and at most m times. For example “o{1,3}” matches the first three “o”s in “Booooooat”. “o{0,1,}” is equivalent to “o?”. |
| [xyz] | A group of characters. Corresponds to any of the included characters. “[abc]” for example matches the “a” in “falling”. |
| [^xyz] | A group of excluded characters. Corresponds to any character not included. “[^abc]” for example matches the “f” in “falling”. |
| [a-z] | A character range. Corresponds to any character in the specified range. For example, “[a-z]” matches any lowercase alphabetic character in the range from “a” to “z”. |
| [^m-z] | An excluded range of characters. Corresponds to any character not included in the specified range. “[m-z]” for example matches all characters not included in the range from “m” to “z”. |

## Examples

| Regular expression | Description | Example |
|---|---|---|
| [a-zA-Z]*_[a-zA-Z]*-+, | Any alphabetic string with an underscore | user_tbx |
| [0-9]{5} | Five random numbers | 12345 |
| ^.{7}$ | 7 random characters | t_p.prt |
| ^[A-Z]{1}[a-z]{2,10} | A capital letter at the beginning followed by two to ten lower case letters | Tuser |


© 2026 INNEO Solutions GmbH

| Regular expression | Description | Example |
|---|---|---|
| [0-9]{2}\.[0-9]{2}\.[0-9]{4} | German date format | 01.07.1975 |
| [0-9]{4}-[0-9]{2}-[0-9]{2} | Date format ISO 8601 | 1975-07-01 |
| [0-9]{2}/[0-9]{2}/[0-9]{4} | Date format US | 01/07/1975 |
| [0-9]{2}-[a-zA-Z]{3}-[0-9]{4} | English date format | 01-Jul-1975 |
| [a-zA-Z]{3,9}\s[0-9]{1,2},\s[0-9]{4} | English date format (long form) | July 01, 1975 |
| ^(?!ST\|B00\|DEF).* | combstatefilter="^(?!ST\|B00\|DEF).*"<br>All combined states that do not begin with ST, B00, or DEF are used. This filters out the Creo Parametric standard state “Default all”. | Combined states are filtered with:<br>ST*<br>B00*<br>DEF* |
| ^ABC | Starts with ABC | ABCDEF |
| ABC$ | Ends with ABC | UVWXYZ |
| ^. {0,3}$ | String with a maximum of 3 characters | AB |
| ^\d+\.\d+\.\d+$ | Version number | 1.0.3 |
| ^[A-Z] {3}-\d{4}$ | Format | ABC-1234 |
| \b[A-Z] {2}\d{6}\b | Serial number | AB123456 |
| (?<![a-zA-Z0-9])CAD(?![a-zA-Z0-9]) | Exact term (not part of a word) | CAD |
| (?<=\s)Rev\.\s?[A-Z] | “Rev A” or “Rev B” after a space | Version Rev A |


© 2026 INNEO Solutions GmbH

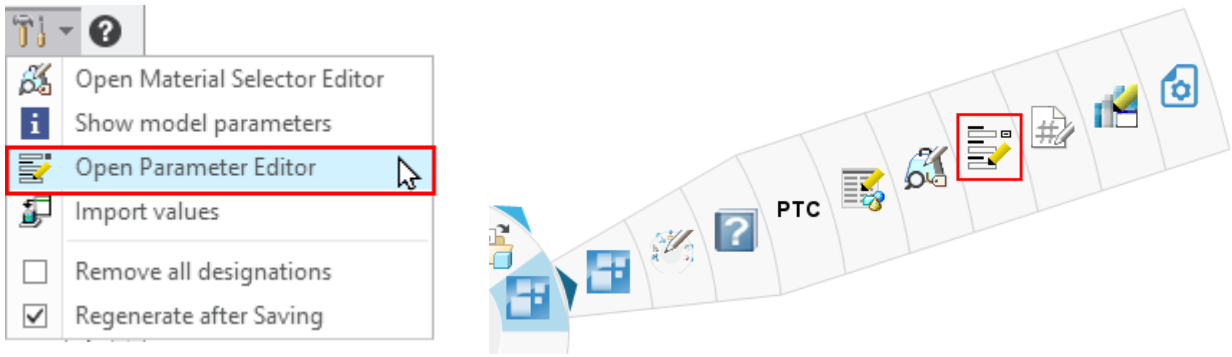
| Regular expression | Description | Example |
|---|---|---|
| \s{2,} | More than one space (for format check) | Scale: 1:100 |
| ^DWG\d{6}$ | Drawing number | DWG123456 |
| ^\d+(.\d+)?\s?(mm | Unit of measurement | cm |
| ^±\d+(\.\d+)?$ | Tolerance specification | ±0.2 |
| ^\d{1,4}x\d{1,4}x\d{1,4}$ | Dimension chain: width x height x depth | 120x50x80 |
| ^PART_[0-9]{3}$ | part | PART_001 |

## 18.3.2 Parameter Editor

The *Parameter Editor* is used to specify parameter definitions for use in *Parameter*. Parameter forms consist of one or more parameter definitions.

### Starting the program

Start *Parameter Editor* from the tools menu in the *Parameter* window or via *Quick Access* ([<] key).




**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].


© 2026 INNEO Solutions GmbH

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

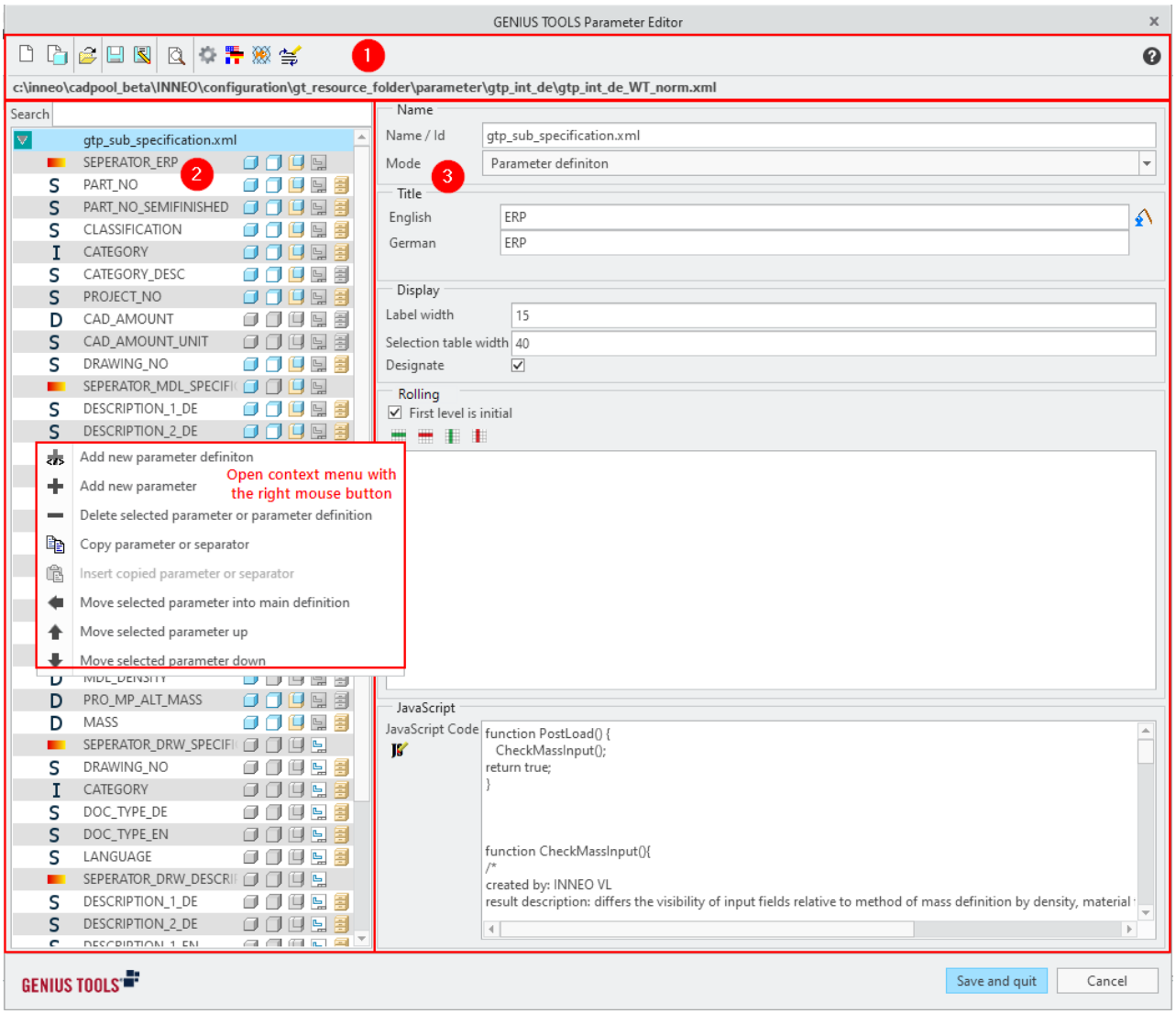
SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## 18.3.2.1 User interface

The user interface of the *Parameter Editor* consists of the following elements:





© 2026 INNEO Solutions GmbH

1. Command bar[492]
2. Element list[493] with context menu[493]
3. Detail view for parameter definitions[494], parameter[498] and separators[514]

## 18.3.2.2 Command bar

The following buttons are contained in the command bar:

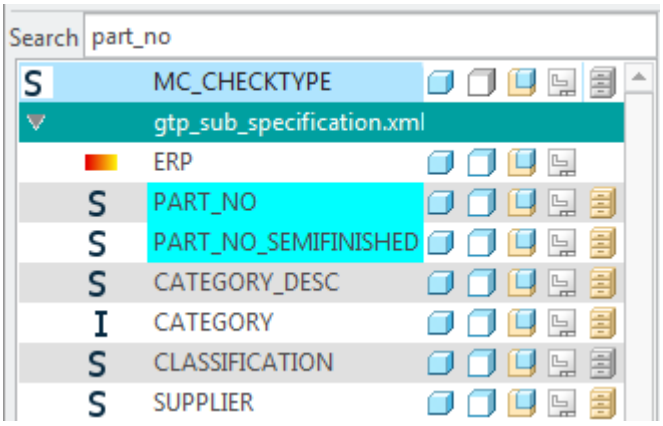
| Icon | Name | Description |
|---|---|---|
|  | New | Creates a new parameter definition in XML format. **Note**: The file is only created after saving. |
|  | New parameter definition from model | Creates a new parameter definition based on **all** parameters of the currently opened model. |
|  | Open parameter definition | Opens an existing parameter definition for editing. |
|  | Save | Saves the current parameter definition. |
|  | Save as | Saves the current parameter definition with a new name. |
|  | Preview | Displays the result of the current configuration in *Parameter* without the need to save. |
|  | Properties | Opens the dialog to edit the parameter definition properties. |
|  | Manage languages | Opens the dialog to manage the languages of a parameter definition. |
|  | Automatic standard texts | Opens the dialog for automatically setting default texts for all parameters and separators. The names of the parameter definitions are used as separators in the table.<br>The parameter names are used as the translation key.<br>See Set standard texts for multiple elements[678] |
|  | Edit list of parameter definitions | Opens the dialog to edit the list of existing parameter definitions[515]. |


© 2026 INNEO Solutions GmbH

## 18.3.2.3 Element list

The element list displays all parameters and sub-parameter definitions contained in the current parameter definition. Click on an element in the list to view its details. Click on the arrow symbol before a sub-parameter definition to expand it and display its contained parameters.

A search field is located above the element list. Use the search field to search for elements in the list by part of their name. Search results will be highlighted in color.




*Color highlight when searching "part_no"*

You can rearrange parameters within the same parameter definition using drag-and-drop. Press the CTRL key to select multiple parameters.

The availability settings are displayed next to the element on the right. They indicate if an element (parameter or separator) will be displayed in the parameter form in parts, bodies, assemblies or drawings. Click on the symbols to show or hide an element in the corresponding mode.

Parameters have a fourth icon. It specifies whether designation is allowed for a single parameter. This function is only relevant when using a PDM system, such as Windchill.

## Context menu

The element list has its own context menu. Right-click on an element to open the context menu.

You can select multiple elements using CTRL.

The following options are available:

**Add new parameter definition**: Adds a new sub-parameter definition to the element list.

**Add new parameter**: Creates either a new parameter definition or a separator.

**Delete selected parameter or parameter definition**: Deletes the currently selected element.

**Copy parameter or separator**: Copies parameters or separators to the buffer memory.


© 2026 INNEO Solutions GmbH

**Insert copied parameter or separator**: Pastes the previously copied element into the element list at the current position.

**Move selected parameter into main definition**: Moves a parameter from a sub parameter definition into the main parameter definition file.

**Moving the selected parameter up/down**: Moves a parameter or separator one position in the list.

## 18.3.2.4 Parameter definitions

Various parameters can be defined in a parameter definition. Since such a list can become confusing, parameters are organized in subparameter definitions. The subparameter definitions are displayed as a tab in the parameter form[481].

**Master parameter definition**
A parameter definition that contains subparameter definitions is called a master parameter definition.

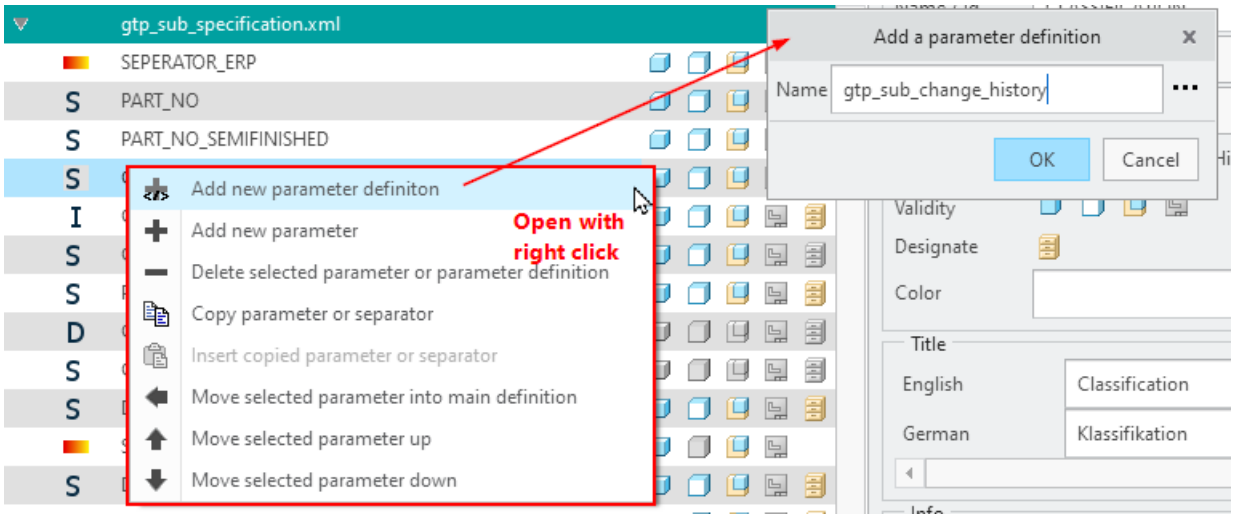
All definitions are stored in an XML file.

## Create a parameter definition

Open the list of all parameter definitions[515] via the button. Create a new entry with the + Symbol and select this new definition in the parameter dialog with the function *Open parameter definition from list*. Open the editor for configuration.

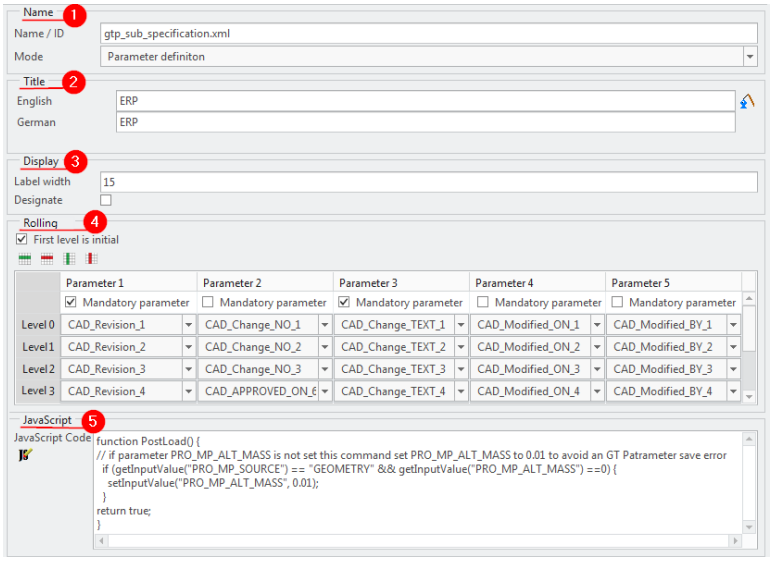
## Create a parameter subdefinition

In the element list open the context menu and select *Add new parameter function*.




In the Detail view select the mode *Parameter definition*. The dialog is divided into the following sections.


© 2026 INNEO Solutions GmbH




*Detail view of a sub-parameter definition*

# 1. Name

**Name/ID**: The name represents the filename of the parameter definition file. If the filename value is changed and then saved, this will be similar to a *Save as* command. The parameter definition will be saved with a new name and the file with the old name remains unchanged.

**Warning**: The mode of sub parameter definitions cannot be changed at a later date.

# 2. Title

Specifies the language-dependent (localized) names that will be displayed as tabs in the parameter form.

The number of available input fields depends on the language configuration.

Standard texts can be added via the button (Description of the standard text selection dialog678).

# 3. Display

**Label width**: Specifies the width (in characters) of the localized parameter title in the parameter form.

**Selection table width**: Specifies the width of the tables (for example, for database connections) that are displayed next to the parameter form. The width is given as number of characters.

**Designate**: Specifies whether designations in PDM mode should be priorly removed for all parameters contained in the definition.

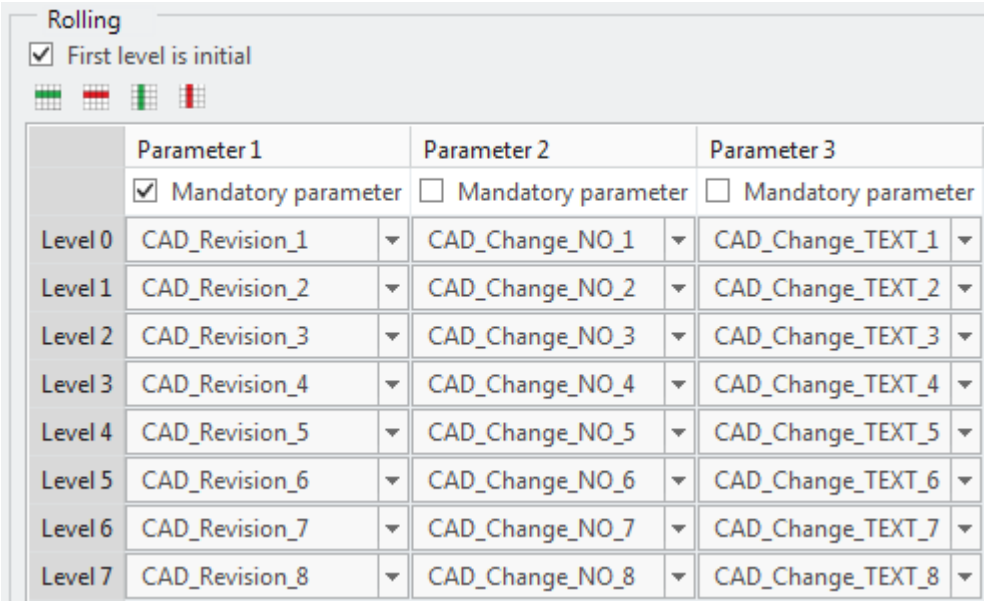

© 2026 INNEO Solutions GmbH

# 4. Rolling

In the *Rolling* section, parameters are specified whose values are transferred to other parameters. Rolling allows to pass incremental values stepwise (per level) in a parameter chain.

**Please note:** The function for rolling parameter values is only available in drawing mode.

The parameter values are transferred FIFO (first in, first out). This allows to create a change history, for instance. For each editing process, the parameter value is moved one step. When the end of the configured chain is reached, the parameter values are deleted.



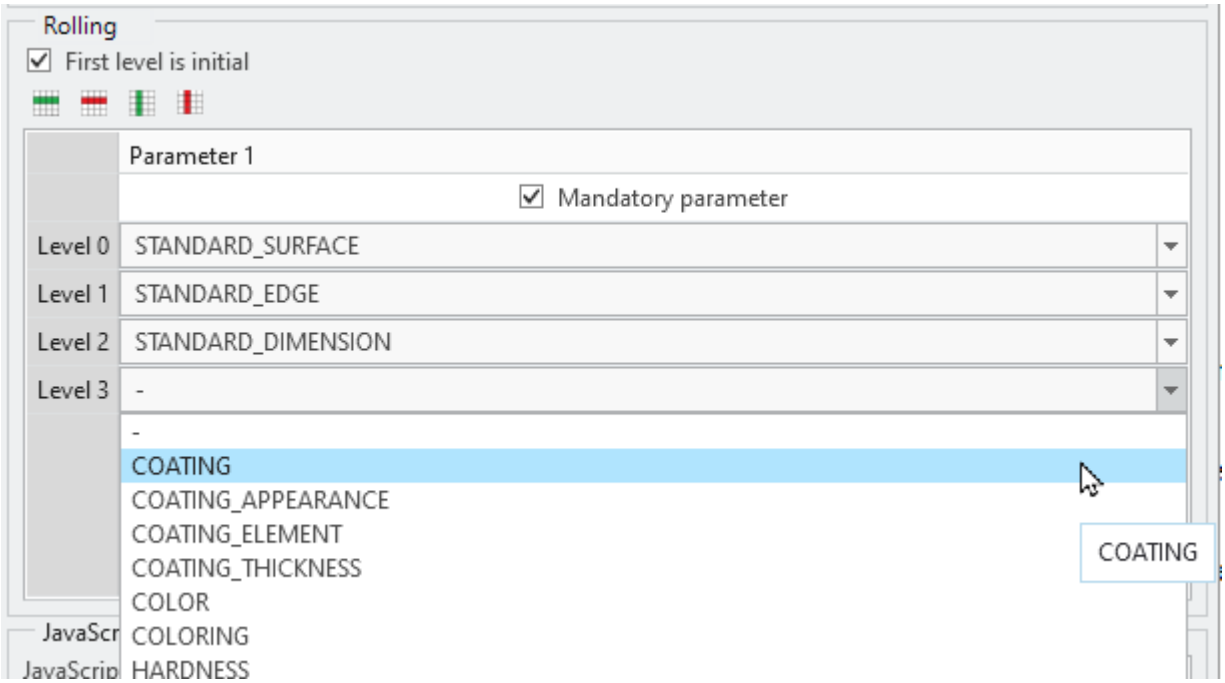

*The parameter values are transferred step by step*

**First level is initial:** If this checkbox is set, the parameter values of the first step will not be overwritten after the parameter has been filled once. This means that the configured chain starts at level 1.

**Add or remove rows:** / Adds rows to the table. You can add rows to determine after how many editing steps a parameter value is deleted.

**Add or remove columns:** / Adds columns to the table, defining more parameters for which values should be transferred.


© 2026 INNEO Solutions GmbH




*Determine parameters for Rolling*

# 5. JavaScript in parameter definitions

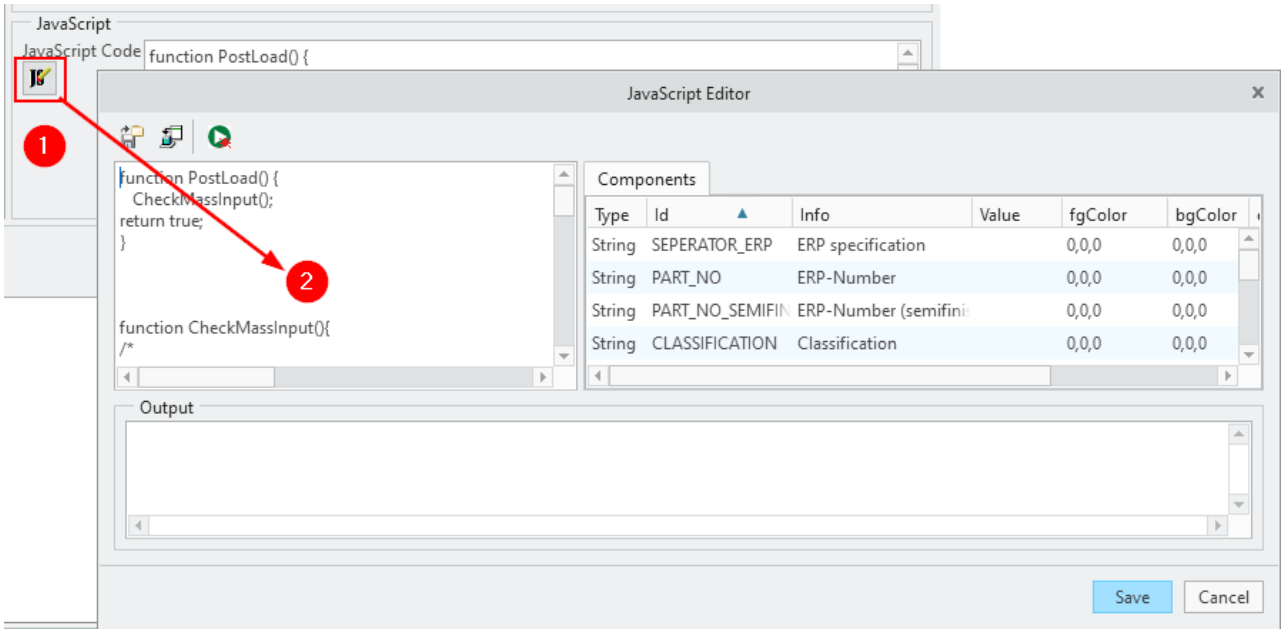
You can add executable JavaScript code to each parameter definition. It is only available in the respective parameter definition.

**Parameter definition:** Open the dialog *Properties* and navigate to the section *JavaScript*.

**Sub parameter definition:** Select the sub parameter definition in the element list and navigate to the JavaScript section in the detail area.

Enter the JavaScript code directly into the input field or click on the JavaScript icon to open the JavaScript Editor[784]. Go here[788] for an explanation of JavaScript functions and short examples. Under Use cases[535] you will find a complex example of using JavaScript with *Parameter*.


© 2026 INNEO Solutions GmbH




**Execution time of functions:** JavaScript can be executed at different times.

| Time of execution | Function |
|---|---|
| After loading a form | PostLoad |
| Before saving values in a form | PreSave |
| After saving values in a form | PostSave |
| After editing a value or after clicking enter | OnChange |

Several of these functions can occur in the JavaScript code. The names of the PostLoad and PreSave functions are fixed. Functions of type OnChange can be named freely.

Table connections are also covered by OnChange functions. If a table connection changes parameter values when GENIUS TOOLS is started, no OnChange function is executed. To correct these parameter values, the PostLoad function must be used.

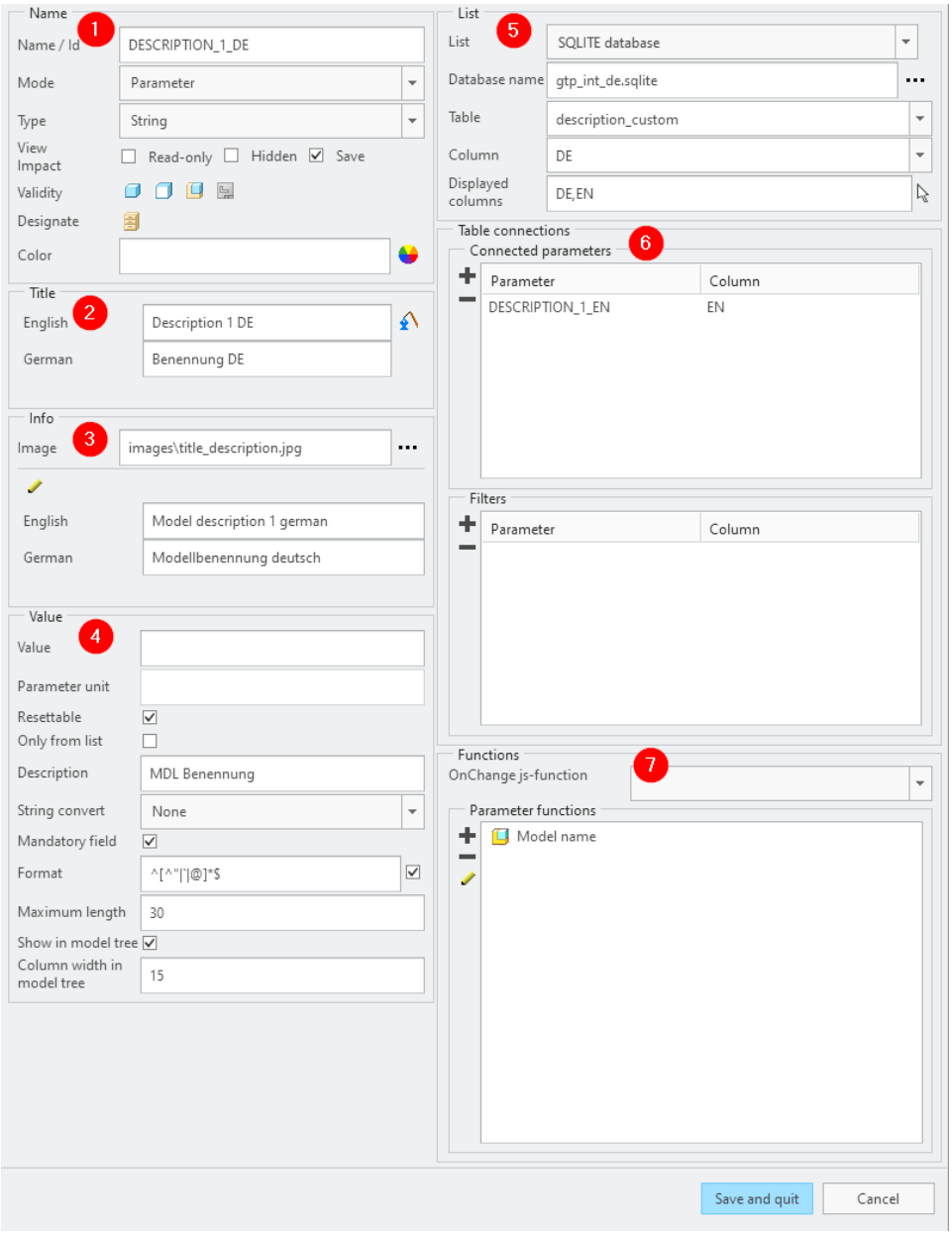
**Please note:** Use the JavaScript function *creoMapkeyAddToStack* only as a PostSave function, because it executes mapkeys and mapkeys usually close windows.

## 18.3.2.5 Adding Parameters

Parameters in *Parameter* contain all settings to specify whether and how a parameter is saved in a Creo model and which input field type is to be used for input.

Field functions such as *Name Generator* or *Material search* are also stored in parameters.


© 2026 INNEO Solutions GmbH




*Example for a detail view of a parameter*

The detailed view for parameters is divided into different sections:

# 1. Name

**Name:** Specifies the Creo model parameter that is addressed by the parameter.

**Mode:** Specifies whether it is a separator or a parameter.

**Type:** Specifies the parameter type to be saved in Creo models. Possible types include: String, Integer, Real number, Yes/No and Material.

---

**Warning:** Parameters can contain values in exponential notation. Check the Creo configuration option `param_dec_places`.

---

**View Impact:** Specifies general visibility settings of the parameter:


© 2026 INNEO Solutions GmbH

– Read-only: The input field cannot be modified.

– Hidden: Specifies whether the parameter is displayed. Hidden parameters are displayed in the *Hidden* tab if the configuration has been adjusted accordingly (`gtp_show_hidden_params`). Also saving these parameters depends on the configuration (`gtp_save_hidden`).

– Save: Specifies whether a parameter is adopted into the model when saving.

**Tip**: Use the *Save* option to specify temporary parameters to allow a conditional selection for another parameter without saving the temporary parameter in the Creo model.

**Validity:** Specifies the use of a parameter in parts, bodies, assemblies and drawings. Parameters that are invalid in an object type can neither be saved nor used.

**Designate:** Specifies whether to allow or prevent designations in PDM mode. When saving in GENIUS TOOLS Parameter, the parameters are flagged *Designate* in the Creo model.

**Color:** Specifies an individual color for the input field. Type the color name directly into the entry box or choose an appropriate color via the color circle  to select a color (Description of the color dialog 796).

## 2. Title

Specifies the language-dependent (localized) names that will be displayed in the GENIUS TOOLS Parameter Name column.

Standard texts can be added via the button  (Description of the standard text selection dialog 678).

## 3. Info

Language-dependent (localized) is stored under *Info*. In addition, an image can be stored for each parameter. The image and the localized information are displayed when the parameter name is clicked. URLs are automatically converted into clickable links in *Parameter*.

Use images and localized information to add support information about parameters.

Always specify URLs including the network protocol (http://, https://).

If images are changed later on, the change will only be displayed after restarting Creo.

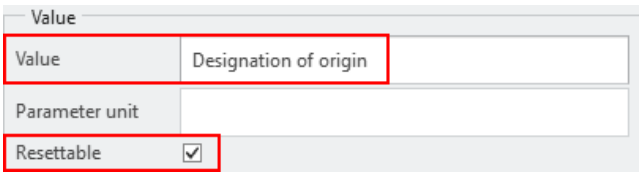
## 4. Value

**Value:** Defines a default value. It is used when the parameter is not in the model and the field in the parameter form is not filled. This can be for example: "-", "not defined" or "n. d.".

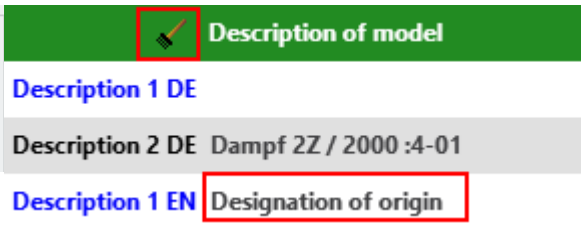
**Parameter unit:** When the parameter type is *real number*, a unit can be specified here. The unit will be adopted by the model parameter when saving.


© 2026 INNEO Solutions GmbH

**Resettable:** Allows resetting the input value in the parameter form to the default value with the *broom symbol* in the parent separator. The broom symbol is displayed if the Resettable checkbox is activated in at least one parameter. The function takes effect until the next separator.




*Activated function in the Parameter Editor*




*Applied function in the Parameterformular*

**Warning**: The *Reset* function can only be used if there is a parent separator. The reset icon can only be displayed there.

**Only from list:** Specifies whether users can only select from predefined list items. It is not possible to directly type into the input field. This option requires a database or a list connection.

With database connections typing into the input field can be used for searching, but only list values will be accepted. When this option is disabled, a field will also accept free input.

**Description:** The description entered here will be adopted by the model as parameter description on saving.

**String convert:** Allows automatic conversion of the input to upper or lower case. This option is only available for string type parameters.

**Required field:** Specifies whether an input field must be filled in by the user. Mandatory field are marked with an asterisk in the parameter form *.

**Format:** Specifies a regular expression[487] to verify the input. If a format has been specified for an input field, the users will be prompted when the input does not comply with the format.

The format can only be checked if the parameter is configured for writing into the model (*Save* is set) and is not hidden.


© 2026 INNEO Solutions GmbH

**Tip:** Activate the checkbox behind the *Format* input field to automatically match values that do not match the format (if possible).

Example
Format: ^\w-\d{2}-\d{5}$
Meaning: "1 Character","Minus","2 Digits","Minus", "5 Digits"

The following entry is incorrect: ABZ-23-00005 (there are three characters at the beginning and not one).

If the checkbox is not set, the correction must be made by the user.
If the checkbox is set, the entry is automatically corrected to Z-23-00005 (i. e. characters are removed at the beginning).

**Maximum length:** Specifies the maximum length for the input in characters.

**Show in model tree:** Specifies whether the parameter will be shown in the model tree when the model tree is displayed.

**Column width in model tree:** Specifies the column width for display of the parameter values in the model tree.

## 5. List

The list configuration is located on the right side of the detailed view.

The list configuration allows to deposit a list of values for a parameter from the following sources:

– text and CSV files: see File-based lists for parameters[503],

– SQLite or Access databases (until version 2003): see Database file lists for parameters[504],

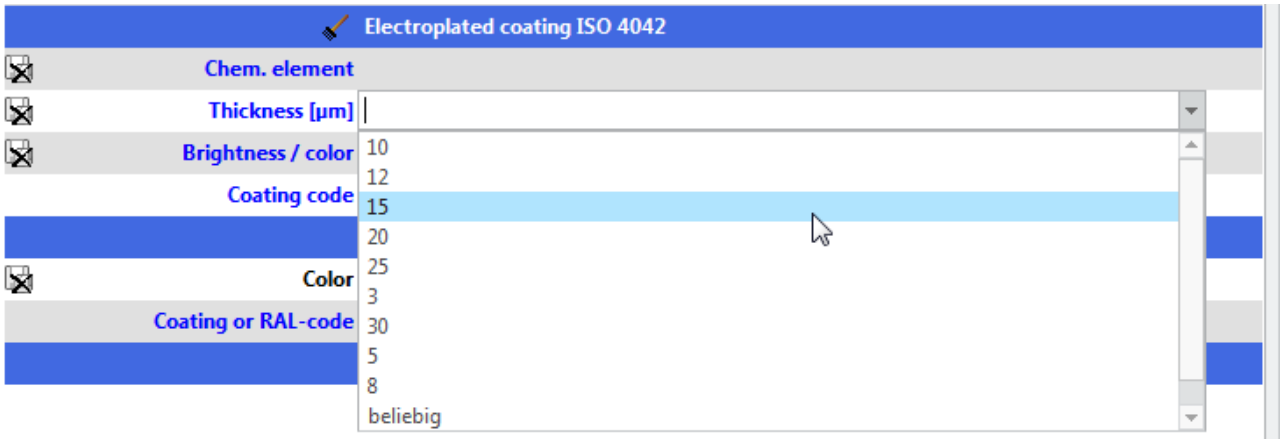
– webserver database systems: see Webserver lists for parameters[505].

– REST API: see REST API lists for parameters with Windchill[507]

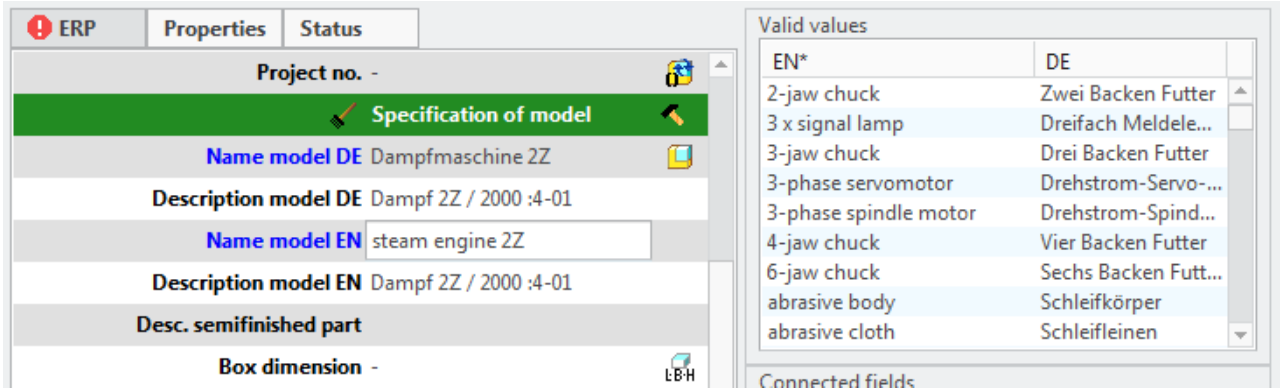
Please pay attention that file-based lists have to be under *gt_resource_folder*. Also UNC-paths (*servername**sharename**path*) are possible, thereby the accessibility has to be ensured.

The configuration depends on the list type.


© 2026 INNEO Solutions GmbH




*A list from a text file was deposited in an input field*




*A database was deposited in an input field*

## 6. Table connections

In a table connection, two or more input fields are connected. Specify the connected parameters[507] and, if necessary, limit the values with a filter list.[508]

## 7. Functions

For each input field you can assign an OnChange functions (Javascript)[509] as well as various parameter functions.[510]

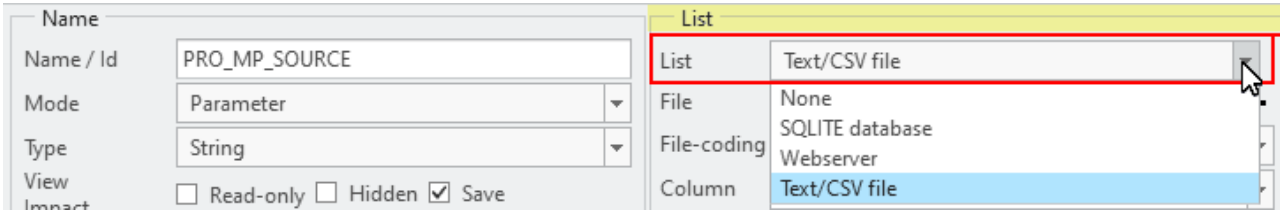
## File-based lists for parameters

File-based lists can be created using plain text files (TXT) or CSV files (Comma-Separated Values).

Text files allow the easy selection of entries from the file (one entry corresponds to one line). CSV files allow a more complex configuration with table connections similar to databases.

To configure a list from a text file you have to specify the file and the file encoding it uses. The list can be used after saving the parameter definition and reloading it in *Parameter*.


© 2026 INNEO Solutions GmbH




*List configuration with CSV file*

**Warning**: Pay attention to the file encoding when creating list files! Only Unicode and ANSI are supported.

**File**: Specifies the list file.

Select the list file using the three dots button or create a new file using the Edit button (pen icon). If a file has already been specified, Edit opens the file using the editor specified in the configuration.

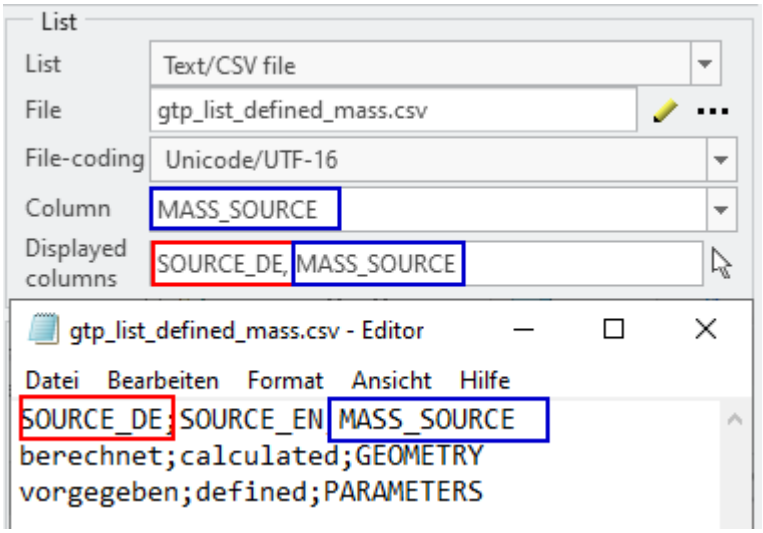
Only the file name is displayed, not the path.

**File encoding**: Specifies the file encoding of the file.

If the selected file is a text file, no further configuration is required.

**Column**: Specifies the CSV file column to retrieve the values for the input field.

**Displayed columns**: Specifies the CSV file columns displayed in the parameter form to allow the user to make a selection.




*Column (blue) and displayed columns (red)*

**Tip**: Activate the *Read-only* option for a list field. This way the list entries can be selected but cannot be modified.

## Database file lists for parameters

Database files can be used for lists: SQLite or Access databases (until version 2003).

The following information needs to be entered.


© 2026 INNEO Solutions GmbH

**Database name**: Specifies the name of the database file. Via the button behind the field the file can be selected via the file browser. Only the file name is always displayed, not the entire path to the database.

**Table**: Specifies the name of the table in the database.

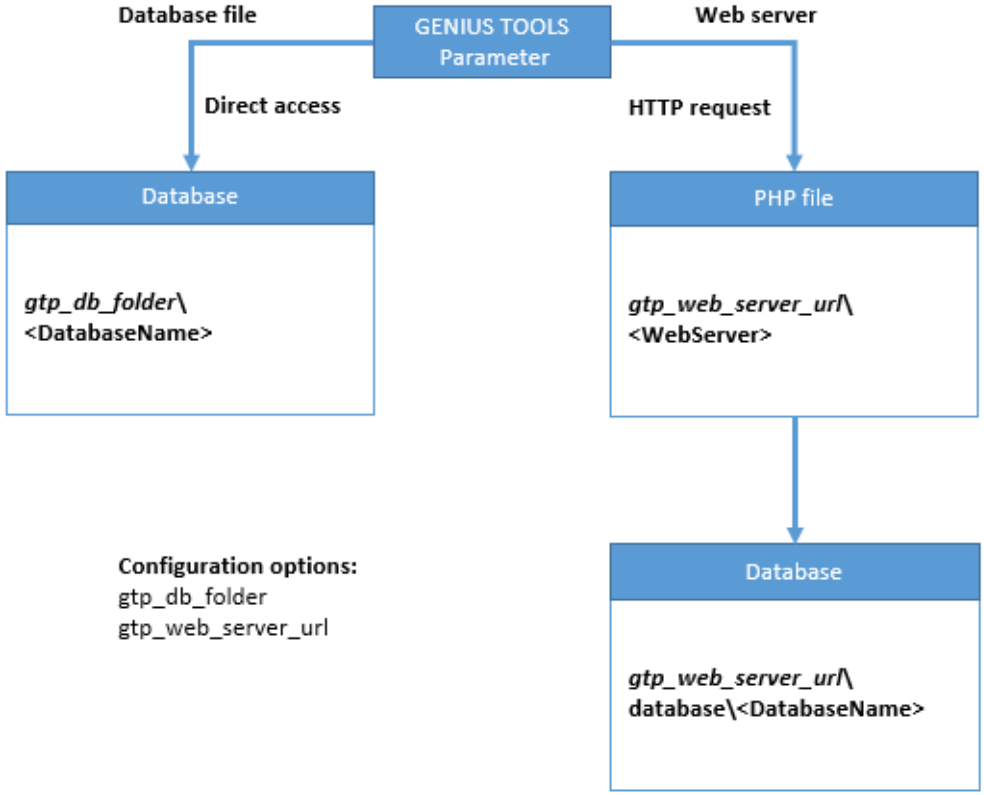
**Column**: Specifies the table column to retrieve the values for the input field.

**Displayed columns**: Specifies the table column displayed in the parameter form to allow the user to make a selection. If multiple columns should be displayed, specify them comma-separated.

The configuration of connected parameters is done in the same way as the configuration of CSV files with connected parameters.

**Please note:** Do not use mathematical symbols +-~/* for describing field names. This can otherwise lead to the problem that field names cannot be recognized unambiguously.

## Overview of database access paths for lists




*Access to database files or web server*

## Webserver lists for parameters

In addition to database files, *Parameter* can also use databases on a webserver.

An HTTP-POST request is sent to a webserver awaiting a JSON (JavaScript Object Notation) response.


© 2026 INNEO Solutions GmbH

**Please note**: Any filtering of the data has to be implemented on the webserver. A selection list provided by a webserver as a JSON file cannot be filtered further in the parameter form.

**Webserver**: Specifies a URL that returns valid JSON data. You can enter a part path. This will be appended to the value of the configuration option `gtp_web_server_url`. If your input contains a colon (as in *http://localhost/testdata.json*), it is assumed to be a complete URL, and `gtp_web_server_url` will not be prefixed. You can find an example PHP file in the installation directory under *Tools/gtp_server*.

**Database name**: Specifies the database to be used. Is passed as POST parameter *DB*.

**Table**: Specifies the name of the table in the database. Is passed as POST parameter *TABLE*.

Database name and table may be left blank if they are referred to automatically by the webserver PHP page

**Column**: Specifies the table column to retrieve the values for the input field. Is passed as POST parameter *SFIELD*. The column names under **Column** and **Displayed columns** are also used to label the columns in the user interface.

**Displayed columns**: Specifies the table column displayed in the parameter form to allow the user to make a selection. Is passed as POST parameter *FIELDS*. If the column with the values for the input field (**Column**) is not listed under **Displayed columns**, it will be added to the POST parameter automatically.

## Example of an HTTP-POST request

```
DB=gt_parameter.db&TABLE=namen&SFIELD=german&FIELDS=english,german&SEARCH=**%
```

| Argument | Description |
|---|---|
| DB | Database name |
| TABLE | Name of the table in the database |
| FIELDS | Name of the columns to be returned |
| SFIELD | Name of the search field |
| SEARCH | Search string |

## Example of the expected JSON response

```
{"timedb":10, "debug":"DB=gt_parameter.db TABLE=namen FIELDS=english,german
SFIELDS=german SEARCH=**%","varr":["Abdeckkappe|cover cap",
"Abdeckscheibe|cover disc","Abdeckung|cover"],"time":Verbrauchte Zeit}
```

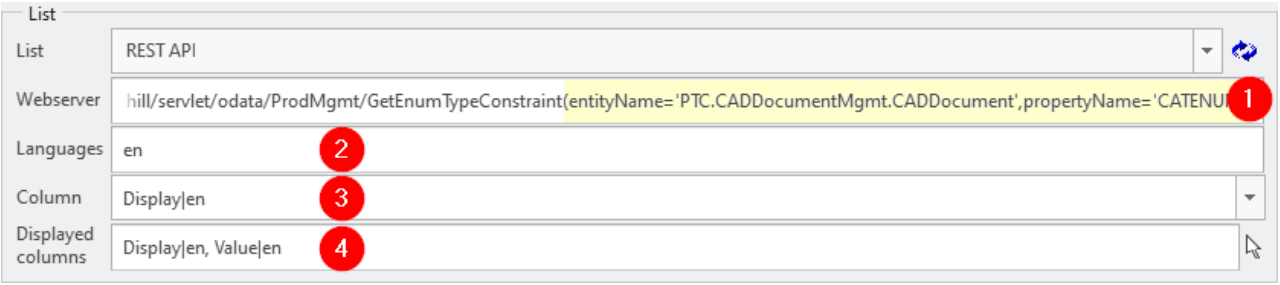

© 2026 INNEO Solutions GmbH

The output *varr* contains the data required for the parameter selection. The other parts of the JSON response could also be passed empty.

| Argument | Description |
|---|---|
| timedb | Duration of the database query |
| debug | Arbitrary text output |
| varr | Output of the data from the defined fields. Each record is delimited by straight double quotation marks and a comma, the fields are separated by pipe symbols (\|). |
| time | Duration of the entire query |

## REST API lists for parameters with Windchill

The REST API list function is available for PTC Windchill with 2.5 REST Services installed. `BasicAuth` authentication protocol is supported.




*Example of creating a REST API list*

1. **Webserver:** Defines the web server. After entering the web server, you will be prompted to authenticate to Windchill.
   In the example shown above, `entityName` and `propertyName` have to be adjusted according to the Windchill configuration.
2. **Languages:** Defines the displayed languages. The languages entered must match the browser language codes: e. g. en, de or fr-CH, en-US.
3. **Column:** Defines the column from which the values for the input field are taken.
4. **Displayed columns:** Defines the columns of the table that are displayed in the parameter form.

## Connected parameters

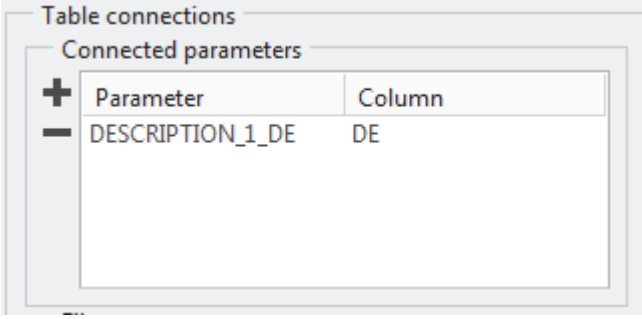
Lists with table connections can be specified for parameters. A table connection interlinks two or more input fields.


© 2026 INNEO Solutions GmbH

For instance, if a field lets the user make a language-dependent selection, a connected field can be automatically filled with the appropriate entry for another language.

In order to be able to use connected parameters, there has to be a list of values that contains a column for each affected parameter and a row for each combination of values.

All parameters that you want to be connected need to be added to the connected parameters list. You have to specify the corresponding column name for each parameter.




*Connected parameter and column in the database*

After saving and reloading it in *Parameter* the updated list is available with the connected input fields.

**Please note**: If values for connected parameters are set automatically, the *first* row of the value list is used that has the defined value for the leading parameter.
Parameter values can be set automatically, e. g., when a JavaScript function is executed, when values are read from the model, or when the configuration option `gtp_check_connections` is set to add missing values after loading the model.

## Filters for parameters

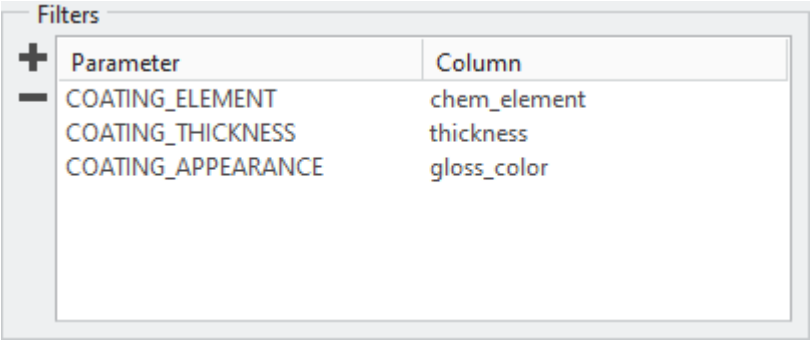
Filters are used for 1:n-relations in tables. Specifying one parameter value controls the selection options of other parameters.

In order to be able to use filters, there has to be a list of values that contains a column for each affected parameter and a row for each combination of values.

**Example**
The code for a galvanic coating per ISO 4042 is filtered by the coating material, coating thickness and coating gloss and color parameters. The users can now enter an ISO code to fill in the coating material, coating thickness and coating gloss and color automatically. Alternatively, they can enter values for coating material and coating thickness to filter the selection options for coating gloss and color and for the ISO code.

The parameters that are required to filter another parameter need to be added to the list under *Filter*. You have to specify the corresponding column name for each parameter.


© 2026 INNEO Solutions GmbH

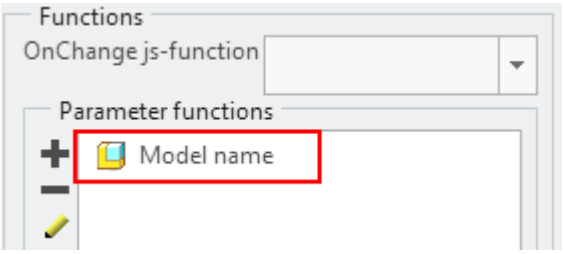



# Field functions

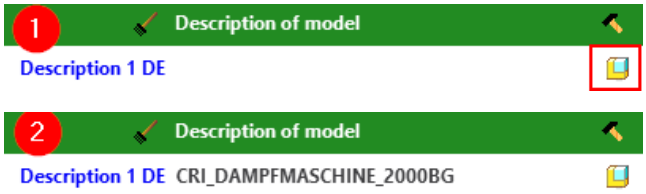
Field functions fill an input field with the value of a parameter form after one click on the symbol at the end of the input field.

With the *fill hammer* in the header you can fill ["hammer"] all values of a section that have field functions with one click.

Example: Field function *Model name* at parameter *Description_DE*:

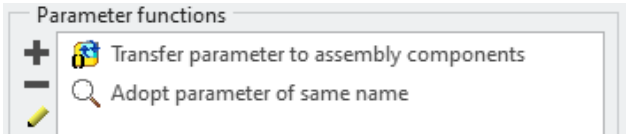


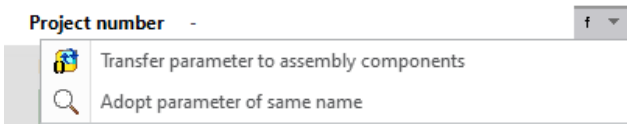

*Parameter Editor*




*Filling the input field in the parameter formular*

If you configure multiple functions for an input field, these functions will be displayed in a drop-down list for the input field in the parameter form. When field functions are executed using the hammer button, only the first field function will be used.



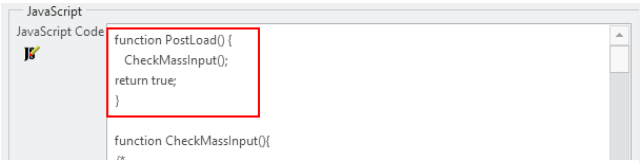




# OnChange functions (Javascript)

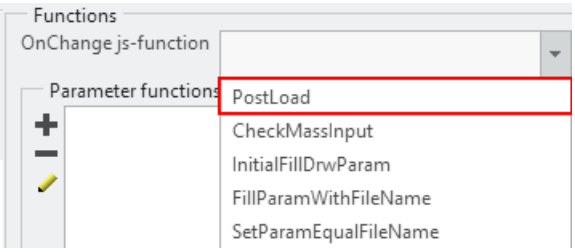
Each input field can have OnChange functions attached to it. OnChange is executed when the value of a form element is changed automatically (e. g. by value tables) or manually by leaving the input field or pressing Enter.

OnChange functions have to be entered in the Javascript section of a parameter definition[497] before they are available for selection as a field function.


© 2026 INNEO Solutions GmbH




*Entering Javascript function in the parameter definition*




*Selection for a specific parameter*

# Adding field functions

Use the ✚ button to add one or more parameter function for an input field. These are available:

| Icon | Parameter function | Description |
|---|---|---|
| | Date | Allows to apply the current date to the input field. Several prebuilt date formats are available. Alternatively, the required format can be manually entered. |
| | Login name | Adopts the current Windows user name or STOOLS User (Log/SHORT) into the input field. |
| | Adopt parameter of same name | Adopts the parameter value of a Creo parameter of the same name from another part or assembly into the input field. |
| | Free parameter search | Adopts the value of a parameter from the Creo *Parameter selection* dialog into the input field.<br>**Note:** The function supports the transfer of only one parameter. If multiple parameters are selected, only the first parameter value will be adopted. |
| | Model name | Adopts the model name from the currently active model in *Parameter* into the input field. In a drawing the active model is adopted as parameter value. |


© 2026 INNEO Solutions GmbH

| Icon | Parameter function | Description |
|---|---|---|
|  | Drawing name | Adopts the drawing name from the currently active model in *Parameter* into the input field.<br>**Note:** Can only be used in Drawing mode. |
|  | PTC Common Name | Adopts the PTC Common Name from the currently active model in *Parameter* into the input field. |
|  | Material selection | Opens *Material* and adopts the selected material into the input field.<br>**Make sure** you have a properly maintained material database. Missing or double materials in the database can cause models that cannot be regenerated. Make sure to only use MTL-files when material selection is used as a field function. GENIUS TOOLS for Creo does not support MAT-files. |
|  | Bounding box | Adopts the bounding box dimensions into the input field.<br>Requires further input, see extra dialog[512] below. |
|  | Name Generator | Adopts a generated name from *Name Generator* into the input field.<br>Parameter looks for the name generator database in the directory specified by the configuration option `gtng_folder`. |
|  | Use predefined parameter value | Adopts the parameter value of a Creo parameter that must be specified during configuration from a model or part that has to be selected at runtime. |


© 2026 INNEO Solutions GmbH

| Icon | Parameter function | Description |
|---|---|---|
|  | Copy parameter value into active drawing model | Copies the parameter value into the parameter of the active model.<br>**Note:** Can only be used in Drawing mode. |
|  | Copy parameter value from active drawing model | Copies the parameter value from the parameter of the active model.<br>**Note:** Can only be used in Drawing mode. |
|  | Transfer parameter to assembly components | Opens the *Value Transfer*[595] dialog window or predefined values from a configured XML file, see below[513].<br>**Note:** Can only be used in assembly mode.<br>**Note:** Parameters of the active sub-assembly are passed on, not of the main assembly! |
|  | Replace variable | Replaces a variable specified in the editor with its value and adopts it into the input field. |
|  | Run JavaScript | Executes a JavaScript function. |

## Bounding box

The function asks for information on how to specify the dimensions of the bounding box. The table illustrates this for a bounding box with the dimensions x=151, y=133.5, z=90.0.

| | Input example 1 | Input example 2 |
|---|---|---|
| Complete prefix | Dimensions: | Size: |
| Prefix X | x= | |
| Prefix Y | ,y= | x |
| Prefix Z | ,z= | x |
| Complete postfix | | |
| Position after decimal point | | 1 |


© 2026 INNEO Solutions GmbH

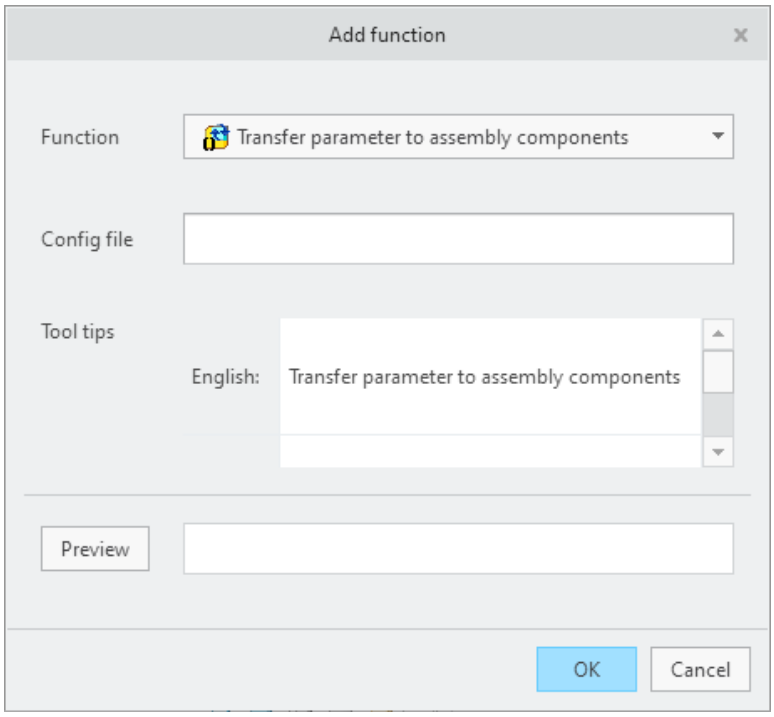
| | Input example 1 | Input example 2 |
|---|---|---|
| Shown values | | xyz |
| Delete followed zeros | ja | nein |
| Result | Dimensions:x=151,y=133.5,z=90 | Size: 151.0x133.5x90.0 |

## Transfer parameter to assembly component: in assembly mode

This function allows you to transfer parameter values from an assembly to its individual components. The field function can be used

– without predefined values: the *Value Transfer* dialog box opens, or

– with predefined values: by clicking on the icon the values from a configuration file (XML file) are entered. The entire section can thus be filled in with the fill hammer.[480]




**Procedure for using the function with filtered values**

Creating a configuration file

1. Open the *Value Transfer* component.
2. In the Filter section[598], add a parameter and define its values, e.g. `@mdln@` contains `CRI_DE`.
3. Using the button in the command bar, save the specifications as an XML file in a folder, e. g. as in the screenshot above: *cri_d*.

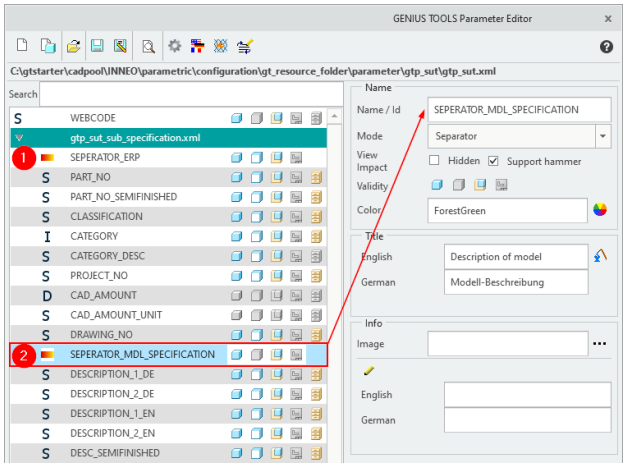
Specifying the configuration file


© 2026 INNEO Solutions GmbH

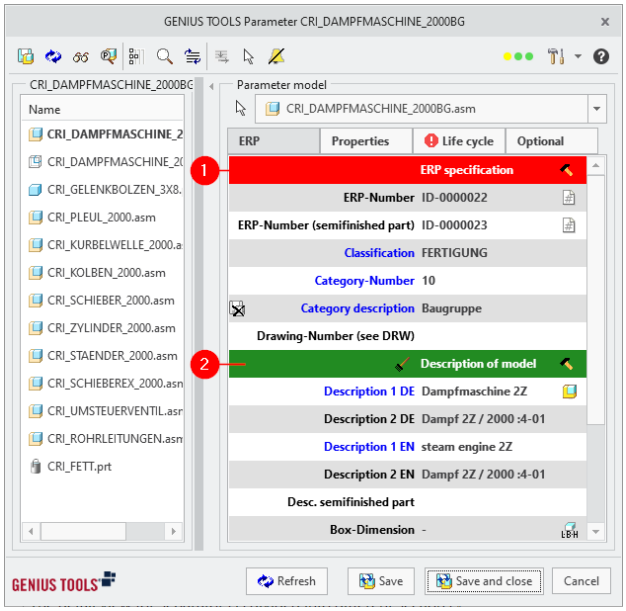
4. In the Parameter Editor in the dialog *Add function* specify the complete path to the XML file or only the file name with file extension (in the screenshot above *cri_d.xml*).
   Variables can be used in the name specification.
   If only the file name is specified, the folder specified in the configuration option `gtp_gtr_rules` will be read. (Default is *%gt_resource_folder%parameter*).

## 18.3.2.6 Adding separators

Use separators to logically structure the parameters, e. g. as section headers. Separators are also needed to reset input fields and to have them filled in by field functions automatically.




*Setting of a separator in the Parameter Editor*




*Result in the Parameter dialog*

The detail view for separators is divided into different sections:

### Name

**Name:** Specifies the name of the separator.

**Mode:** Specifies whether it is a separator or a parameter.

**View/Impact:** If a separator is marked *Hidden*, it is only used to structure the view in the Editor. It is not displayed in *Parameter*.
If the setting *Support hammer* is active, the button for automatically filling in values is displayed in the separator as soon as field functions are defined for parameters.

**Validity:** Specifies the visibility of the separator in Part, Assembly and Drawing mode.


© 2026 INNEO Solutions GmbH

**Color:** Specifies a custom color for a separator. Type the color name directly into the entry box or choose an appropriate color via the color selection (Description of the color dialog[796]).

## Title

Specifies language-dependent (localized) names that are displayed in the *Parameter* forms section. Leave the title empty to create an empty separator.

Standard texts can be added via the button (Description of the standard text selection dialog[678]).

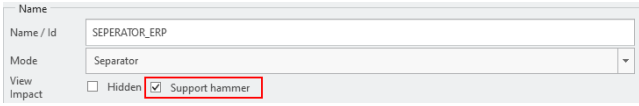
## Info

Language-dependent (localized information) and a preview image can be deposited at Info. They are currently not displayed in the forms section.

## Automatic filling of parameter values: Create hammer symbol

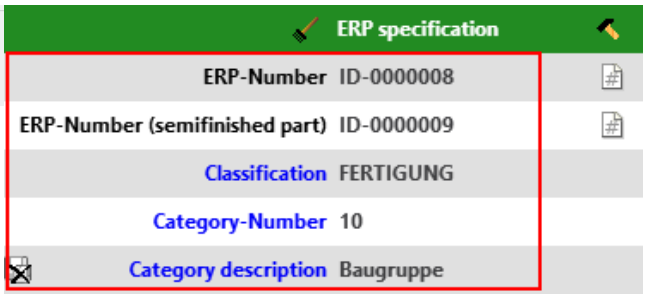
In the parameter form, users can use the *hammer symbol* to have parameter values filled in automatically. The *hammer symbol* has an effect up to the next Separator. The symbol is displayed if a field function is stored in at least one parameter and if the *Support hammer* setting is set in the Separator. The functions are defined in each parameter. To display the symbol, it is mandatory that a Separator is located above the parameters in the form.

**Setting**




*Activated function in the Parameter Editor*

**Result**




*Applied function in the Parameter form*

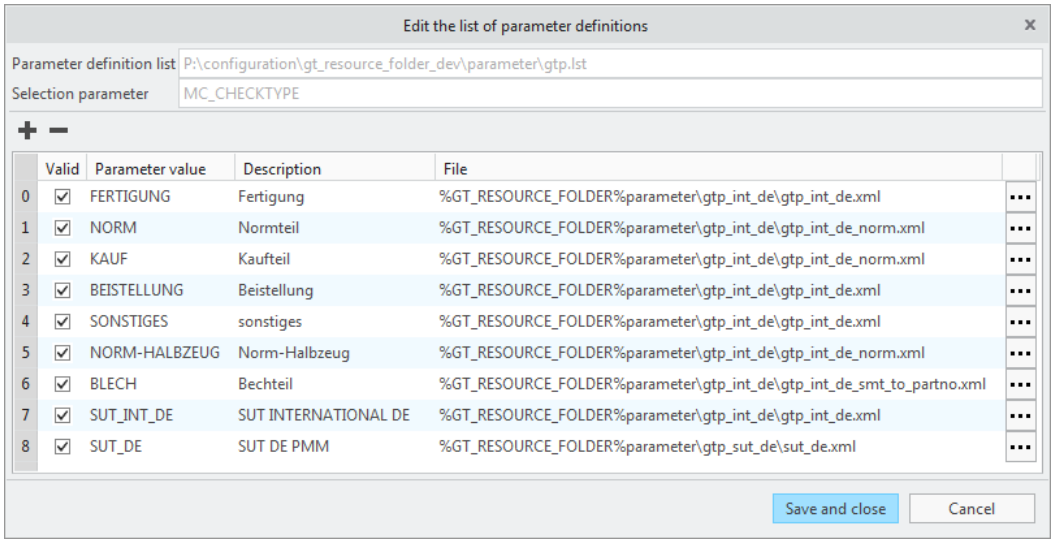
The broom symbol for resetting the values is also displayed in the separator, but set individually for each value. As soon as a value can be reset, the broom symbol appears. See Adding parameters > Range: Values[500].

## 18.3.2.7 Editing parameter definition list

Open the list of parameter definitions in the command bar with the button . It shows the content of the file *gtp.lst*. Entries can be edited directly in the dialog.


© 2026 INNEO Solutions GmbH

The fields above the list show the defined location of the file (configuration option: `gtp_lst`) and the current selection parameter (configuration option: `gtp_file_param`).




Add new entries by clicking the ✚ button. Click on a line number and use the minus button ━ to remove an entry.

Click on a cell in the table to edit it.

The individual line columns have the following meaning:

**Valid:** Activates or deactivates the parameter definition. Deactivated parameter definitions are ignored on selection.

**Parameter value:** Defines the value of the selection parameter for a parameter definition (default: `MC_CHECKTYPE`).

**Warning**: Only assign parameter values once.

**Tip:** You can use fixed parameter names or wildcards, e. g. `*_CRI*`.

**Description:** A meaningful description of the parameter definition. Descriptions can also be used as comments or for separation, even if they don't have a parameter value or definition.

**File:** Specifies the location of the parameter definition.

Click the button ... and select a parameter definition to add via the file browser.

Save the list of parameter definitions to apply the changes.

## 18.3.3 Use cases

In this section you will find use cases of the *Parameter Editor*. The tasks in the following sections mainly build upon one another.


© 2026 INNEO Solutions GmbH

*Parameter* can be customized for use in companies in different ways. You can customize individual parameters of the parameter definition included with *GENIUS TOOLS* for Creo and *Startup TOOLS* or develop individual parameter definitions.

The following sections describe the second approach and also explain how to import parameter definitions from the Web.Link *TOOLBOX Parameter* manager.

## Preliminary considerations on creating new parameter definition files

Before creating new parameter definitions think about your parameter concept:

– Which parameters should be added in the model, and in which mode (prt, asm, drw)?

– How should the parameters be structured (e. g. in sub-parameter definitions)?

– Which separators should be used for structuring the parameters?

## Parameter definition backup

To avoid loss of data and restrictions in productive operation, we recommend to save a copy of your existing parameter definitions. In case of misconfiguration you can quickly restore a working configuration.

## Customizing the configuration settings

Customize the *Parameter* configuration settings to your requirements. Closely inspect the following configuration settings:

**gtp_db_folder**
Inspect this configuration option if you want to use information from databases in your parameters.

**gtp_designate**
Adjust this configuration setting if you are using a PDM system such as Windchill.

**gtp_file and gtp_file_param**
Adjust the configuration options `gtp_file` (path to a parameter definition when no model parameter is found for assignment) and `gtp_file_param` (selection parameter specifying which parameter definition is used) to have your new settings being used.

## Creating the parameter definition files

The following section describes the creation of new parameter definitions in various examples.

**Warning**: Note that sub-parameter definitions can also refer to other parameter definitions, but these references are not evaluated.


© 2026 INNEO Solutions GmbH

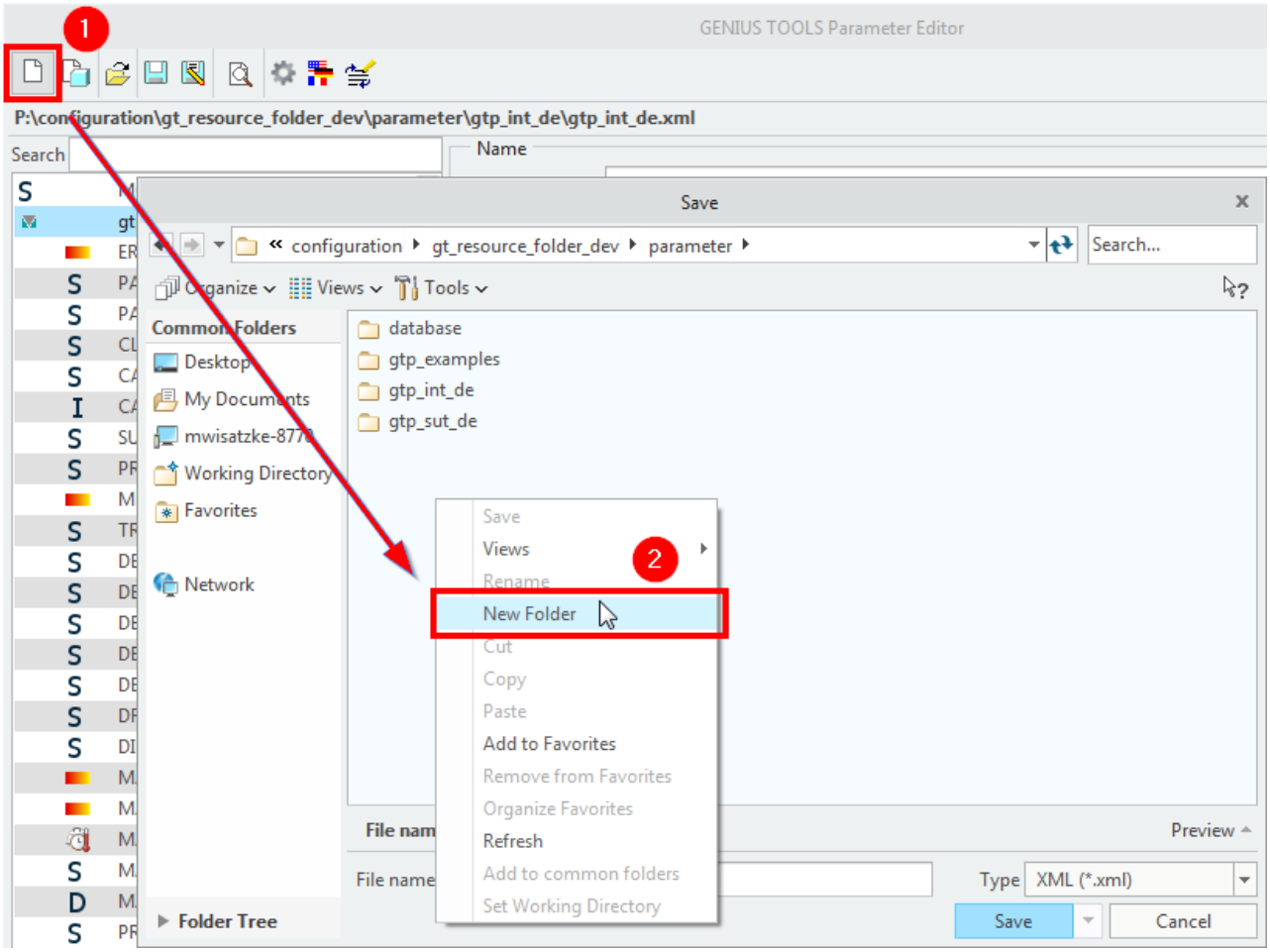
Another example shows the import of an existing parameter definition from the *TOOLBOX Parameter* manager.

The examples partly build on one another; so depending on the operational scenario the procedure of creating a parameter definition may be combined using the examples.

## 18.3.3.1 Creating a simple parameter definition

A parameter definition without references to other definitions consists of a single file. All parameters and separators are included in it and are displayed as a list in the parameter form.

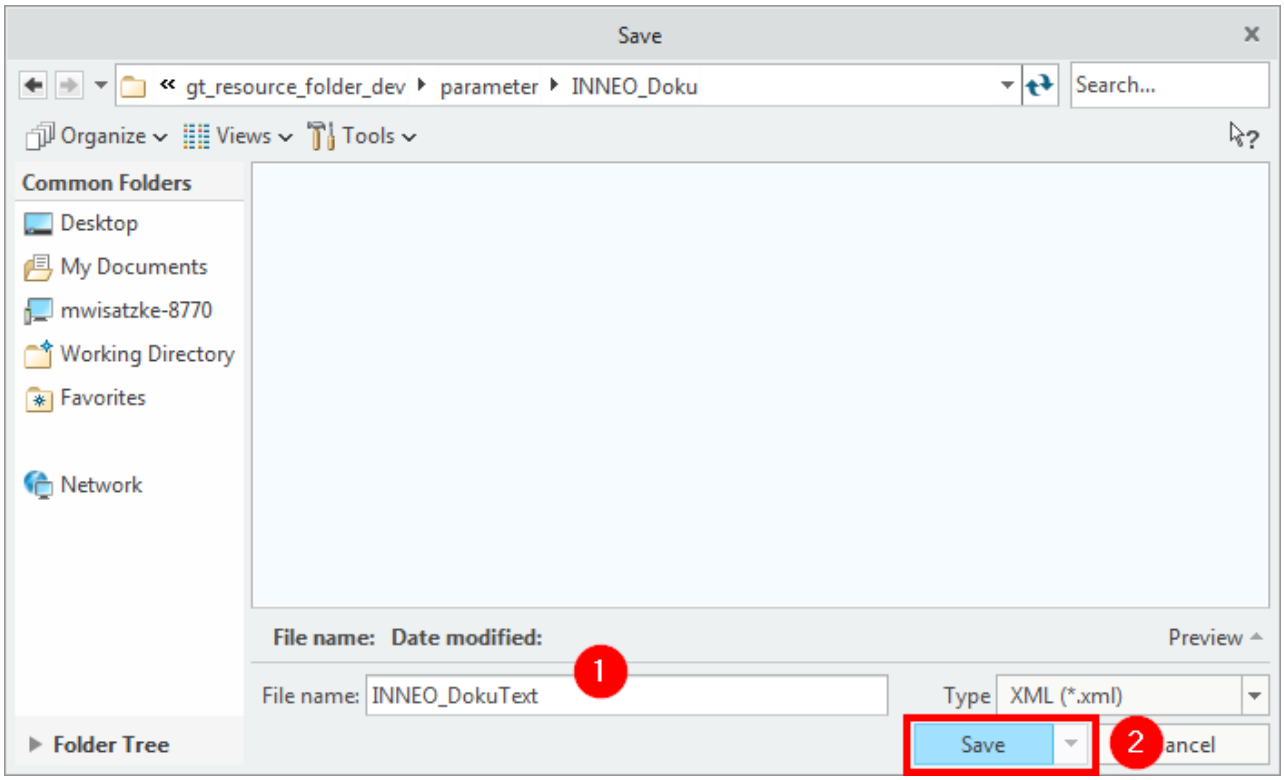
1. Open *Parameter Editor*.
2. Create a subfolder in the *<GTfCInstallationDirectory>\configuration\gt_resourcefolder\parameter* following the name scheme *companyname_parameterdefinitionname*.




*In the editor, click on "Create new parameter definition" (1) and create a new folder (2) via the context menu*

3. Create a new parameter definition inside the folder. Click *Save*.

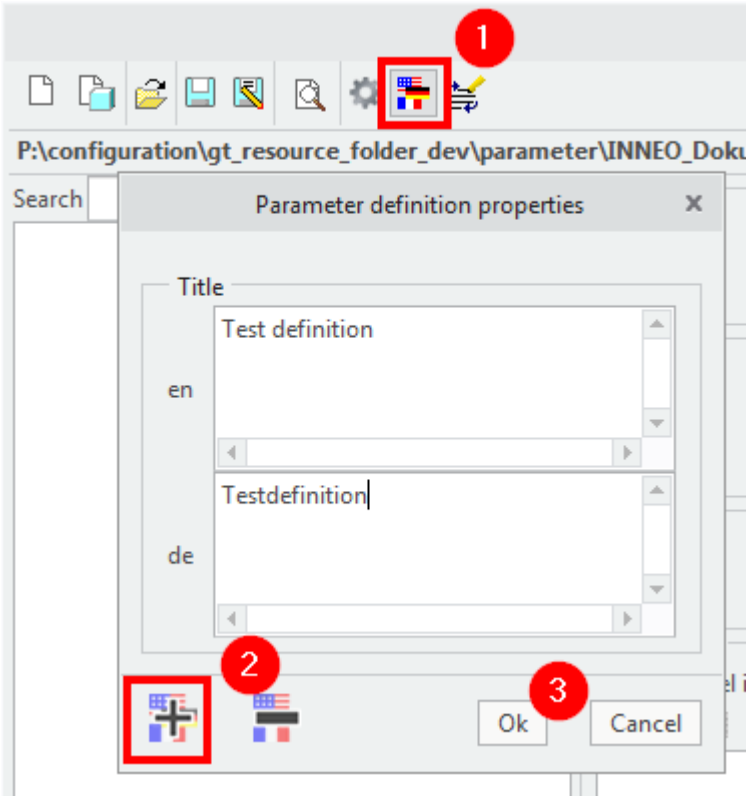

© 2026 INNEO Solutions GmbH




*Enter a title (1) and confirm by clicking "Save" (2)*

4. Click *Edit definition title*. Add additional languages to the parameter definition and enter the parameter definition titles for these languages. As soon as a new language is added, a name can be assigned to every parameter in this language.

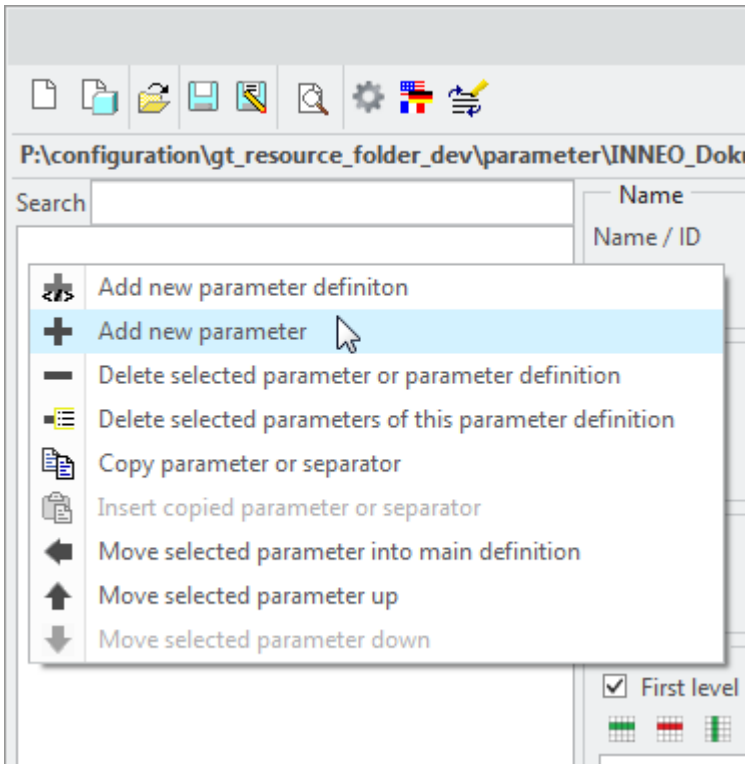

*© 2026 INNEO Solutions GmbH*




*Open the definition titles dialog (1). Add another language (2). Confirm the dialog after entering the definition titles (3)*

5. Click *Save* in *Parameter Editor*. The file is created.
6. Create your parameters via the element list in the editor.


© 2026 INNEO Solutions GmbH




*New parameters are added via the context menu in the element list of the editor*

In the following sections, parameters with the following properties are created as an example:

– Parameter with simple input with list selection

– Parameter with database connection

– Parameter as a mandatory field

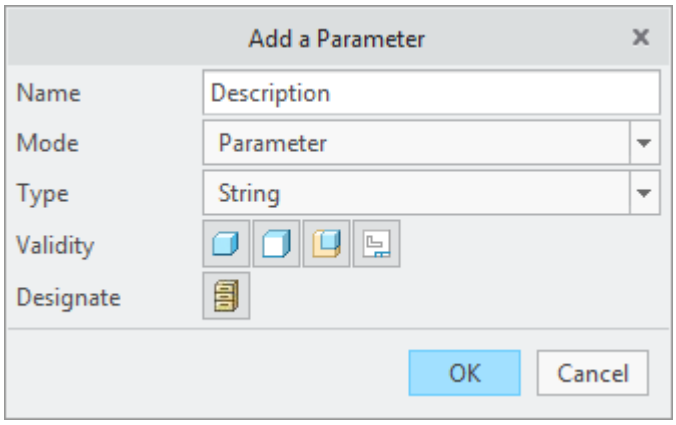
In addition, a colored separator is created that allows to reset input fields.

## Parameter with input field - Description

For non-standardized descriptions free input may be suitable. Note that *String* type parameters are restricted to a maximum of 80 characters.

Open the context menu in the element list and create a *String* type parameter. If you need descriptions in multiple languages, create an individual parameter for each language.


© 2026 INNEO Solutions GmbH




*Create a new parameter*

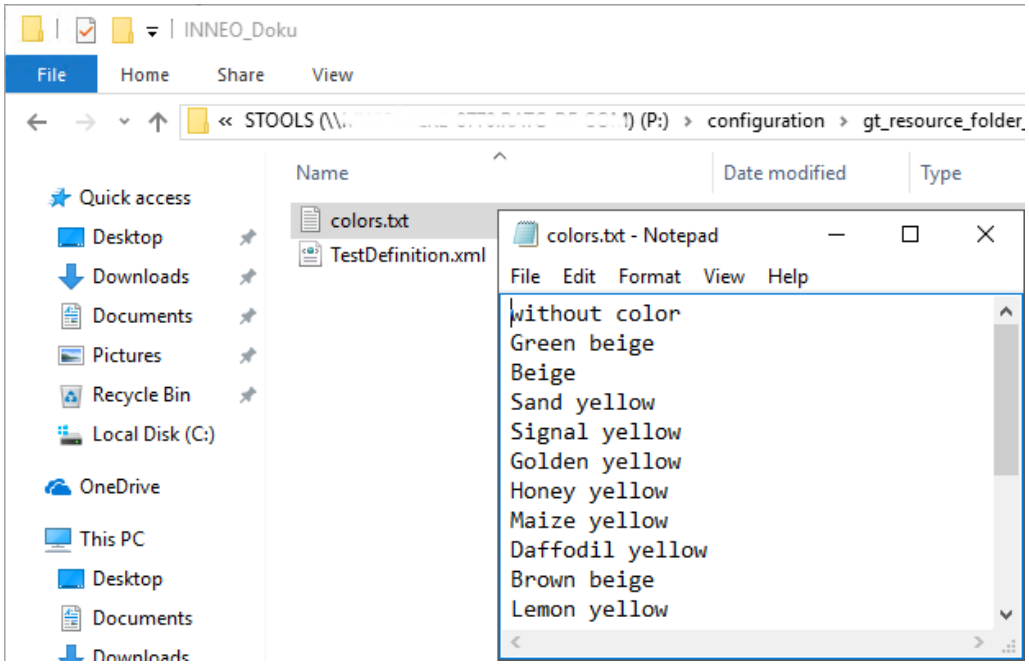
1. Enter a title for the field in *German* and *English* language.
2. Save the parameter definition.

For free input into the field no further input is required.

## Parameter with list selection - Color

In order to use the same color names in all parts, a list can be deposited for colors.

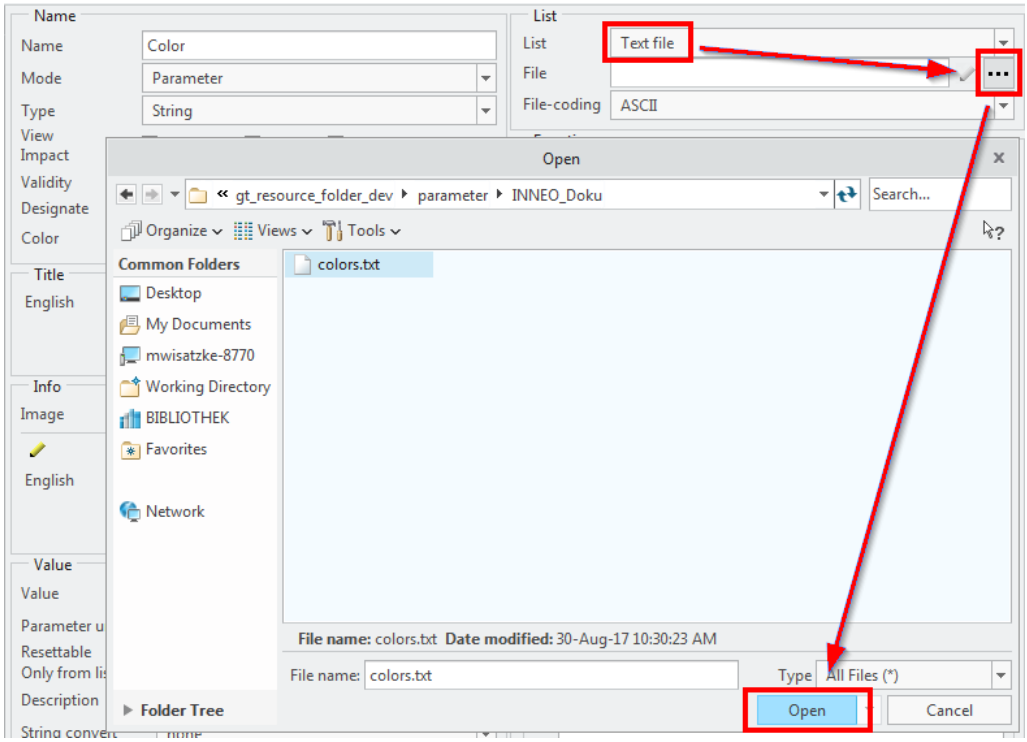
1. Create a *String* type parameter.
2. Create a new file *colors.txt* in the folder containing the Parameter definitions using the Windows Text Editor.
3. Enter the desired color names and save the file.




*Make sure to use the proper file encoding when saving!*

4. In the Parameter Editor list section, select *Text file* and select the file you just have saved via the button ....


© 2026 INNEO Solutions GmbH




*For List select "File" , then select the TXT file and confirm your input*

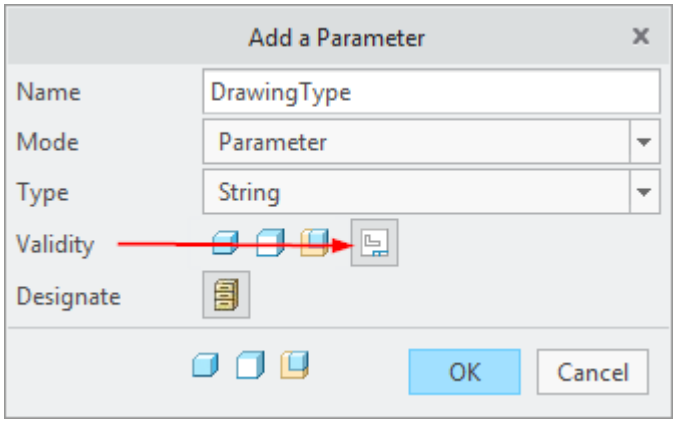
5. In the *Value* section, select the option *Only from list*.
6. Enter a title for the field in *German* and *English* language.
7. Save the parameter definition. The parameter with list selection is fully configured.

## Parameter with list from database file - Drawing type

In the *gt_resource_folder\parameter\database* folder there is a file named *gtp_int_de.sqlite*. This file contains a selection of several standardized records related to different topics.

For drawing types there is a prebuilt table in the database that will be used in this example.

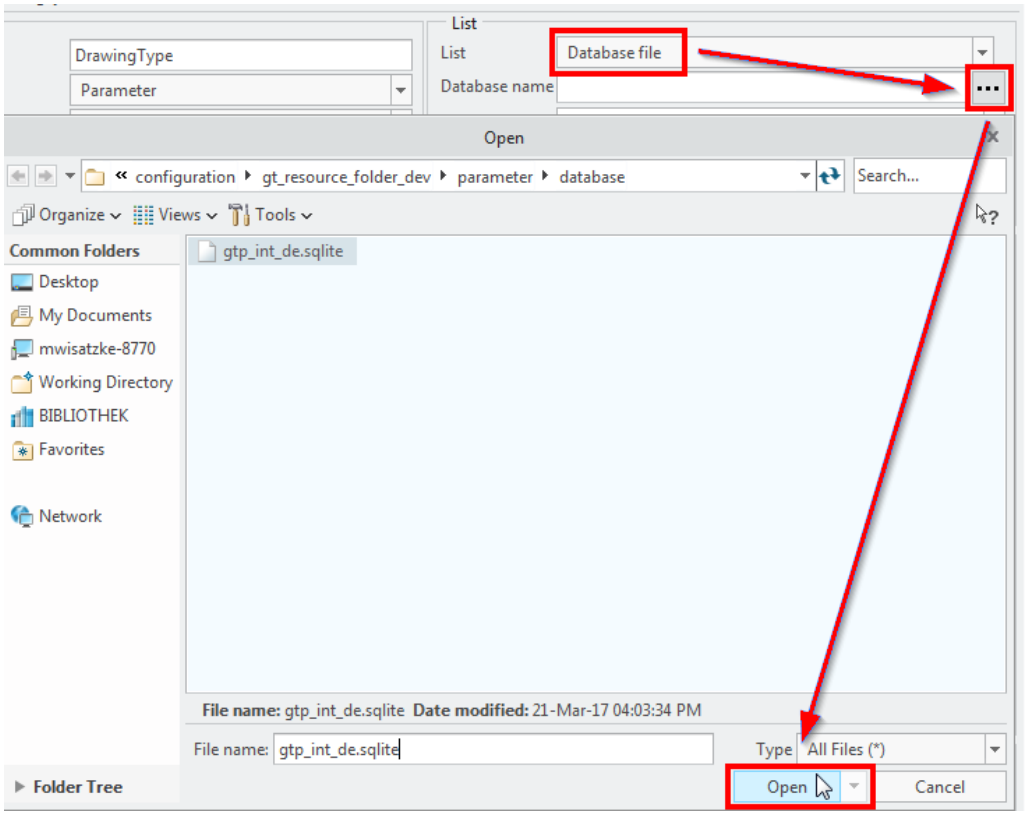
1. Create a *String* type parameter. For Validity select *Drawing*.




*Make sure to use the proper validity setting*


© 2026 INNEO Solutions GmbH

2. In the list section, select *Database file* and select the SQLite file in the ... *\parameter\database* folder via the button ...




*For List, select "Database" , then select the SQLite file and confirm your input*

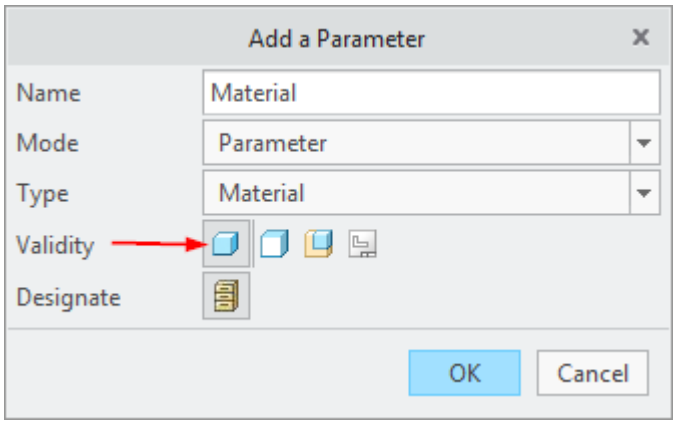
3. For Table, select *document_type*.
4. In the *Column* field below, select *DE* for German.
5. For *Displayed columns*, select *DE* and *EN* with the Ctrl key pressed.
6. In the *Value* section, select the option *Only from list*.
7. Enter a title for the field in *German* and *English* language.
8. Save the parameter definition. The parameter with a list from a database is fully configured.

## Parameter with mandatory field - Material

Materials are an important property of a model and are usually only added in parts (prt).

1. Create a new *Material* type parameter. Restrict the *Validity* to parts.

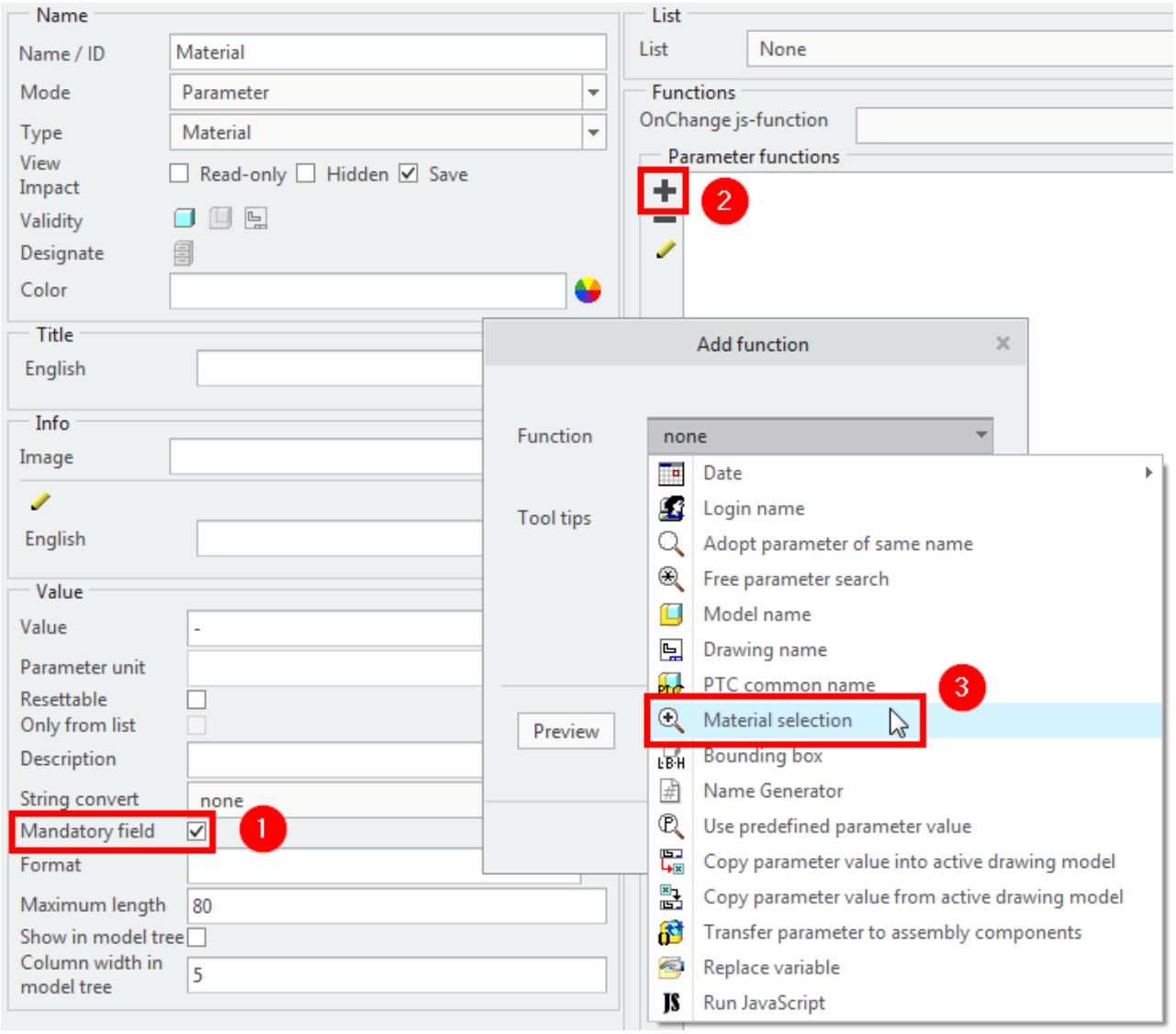

© 2026 INNEO Solutions GmbH




*Make sure to use the proper parameter type at creation*

2. Activate the *Mandatory field* option in the value section.
3. Go to the *Parameter functions* area and click *Add function*
4. and select *Material selection*.


© 2026 INNEO Solutions GmbH




*Activate the Mandatory field option (1), click on Add function (2) and select Material selection (3)*

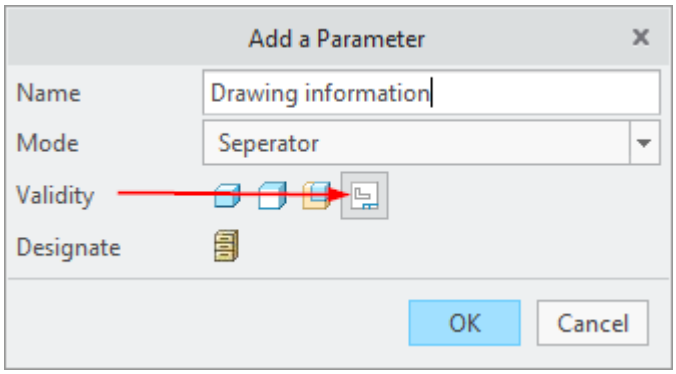
4. Enter a *Title* for the field in German and English language.
5. Save the parameter definition. The material parameter as a mandatory field is fully configured.

## Colored separator with reset-function

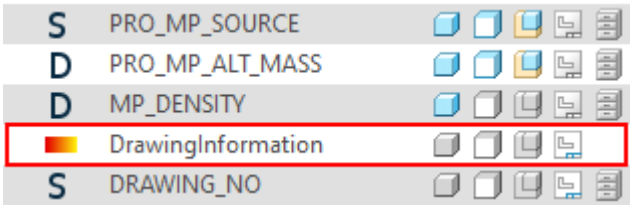
Separators are used for logically structuring the input fields for parameters. If *Reset* and *Fill automatically* are required in a parameter form, at least one separator for each parameter definition has to be created.

1. Open the element list context menu and click on *Add parameter*.
2. Select the *Separator* mode and enter a name.


© 2026 INNEO Solutions GmbH

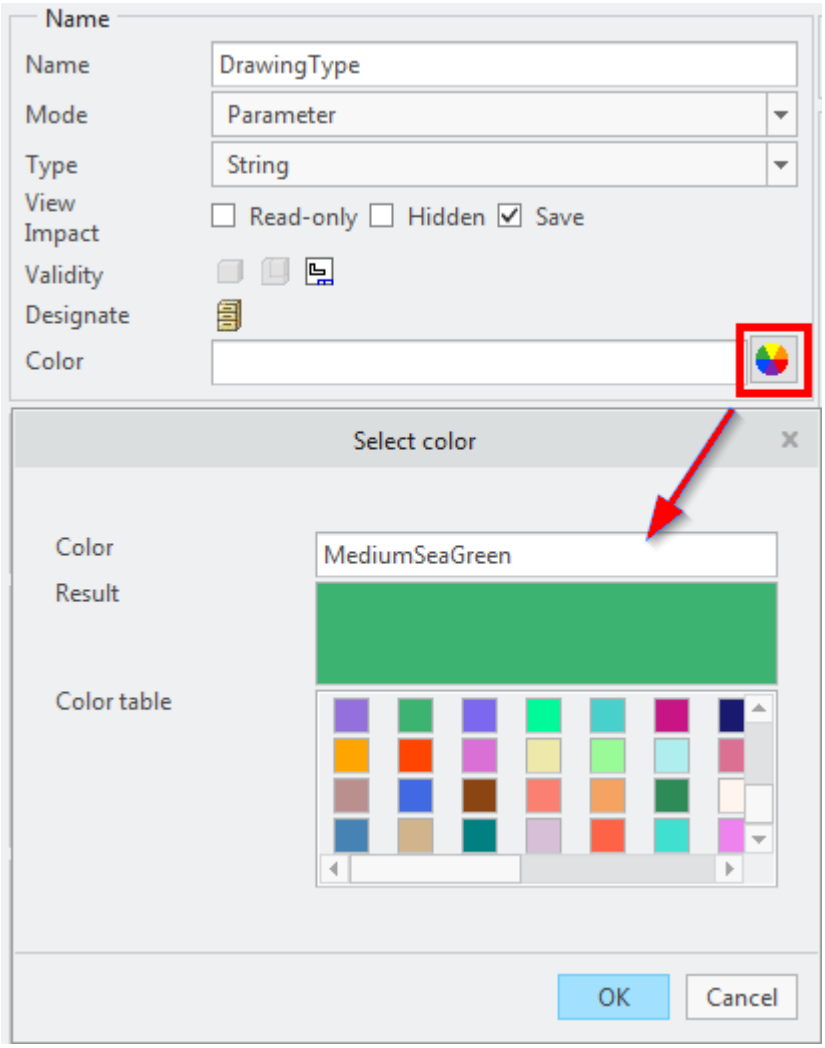



3. Drag the separator in the list and drop it before the parameter that should be reset. Move separators by Drag-and-Drop in the parameter definition list.




4. Click on the color selection icon to the right of the *Color* field and select an appropriate color. Confirm the selection.

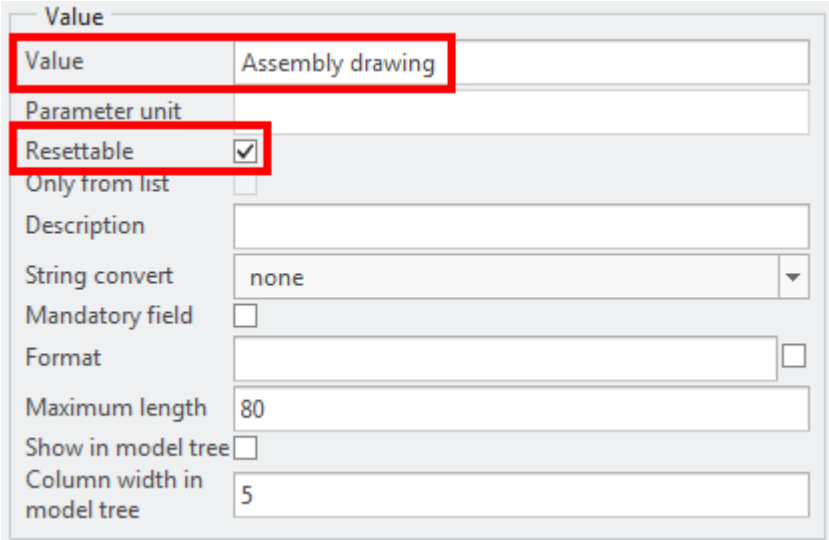

© 2026 INNEO Solutions GmbH




*A color can be directly selected via the pen icon*

5. Enter a title for the field in *German* and *English* language. The *Reset* function is not specified in the separator itself. The separator is only needed to display the reset button.
6. Click the parameter that should be reset. In this example the *DrawingType* parameter.
7. In the *Value* section, activate the *Reset* checkbox.
8. Enter a default value at *Value*. This will automatically be entered into the field when resetting.


© 2026 INNEO Solutions GmbH




*Enter a default value and activate the "Reset" function*

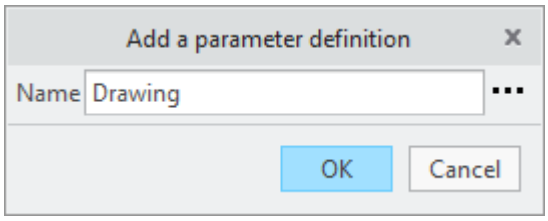
9. Save the parameter definition.

## 18.3.3.2 Creating a Parameter definition with tabs

*Parameter* supports to splitting a parameter definition into several sub-parameter definitions. These are displayed as additional tabs in the parameter form and increase clarity.

In the following example an existing parameter definition (Creating a simple parameter definition 518) is extended with a sub-parameter definition.

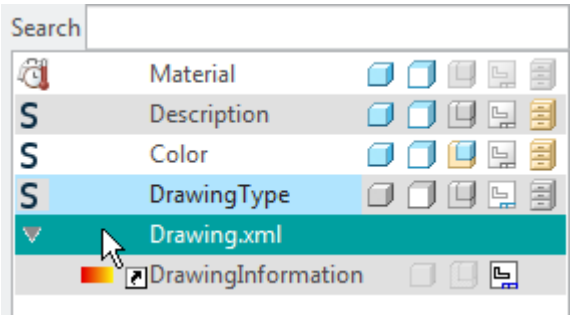
1. Open *Parameter Editor* and load the parameter definition from the example under Creating a simple parameter definition 518.
2. Open the context menu in the element list. Click on *Add parameter definition*.
3. Enter a name for the parameter definition.




*Create a new Parameter definition via the context menu*

4. Drag-and-Drop the parameter into the new parameter definition.


© 2026 INNEO Solutions GmbH




*Drag and Drop the desired parameters into the new sub-parameter definition*

5. Save the parameter definition.

## 18.3.3.3 Importing old parameter definitions

*Parameter* can import parameter definitions that were created with the Web.Link *TOOLBOX Parameter* manager.

Before importing, the following two preliminary considerations should take place:

– Does the visibility/validity of parameters in the individual modes (prt, asm, drw) need to be customized?

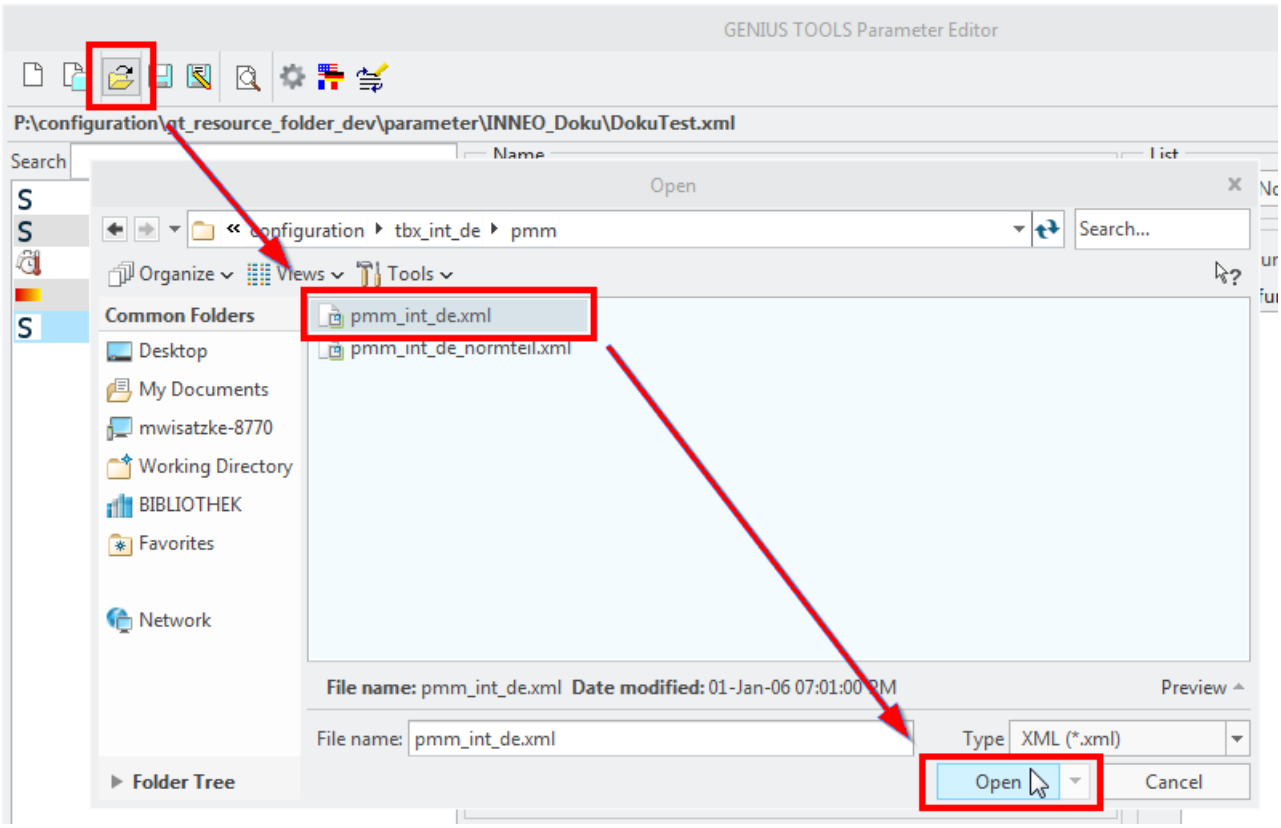
– Should the existing parameters be split into sub-parameter definitions for logical structuring?

Proceed as follows to import a parameter definition:

### Adopting the parameters

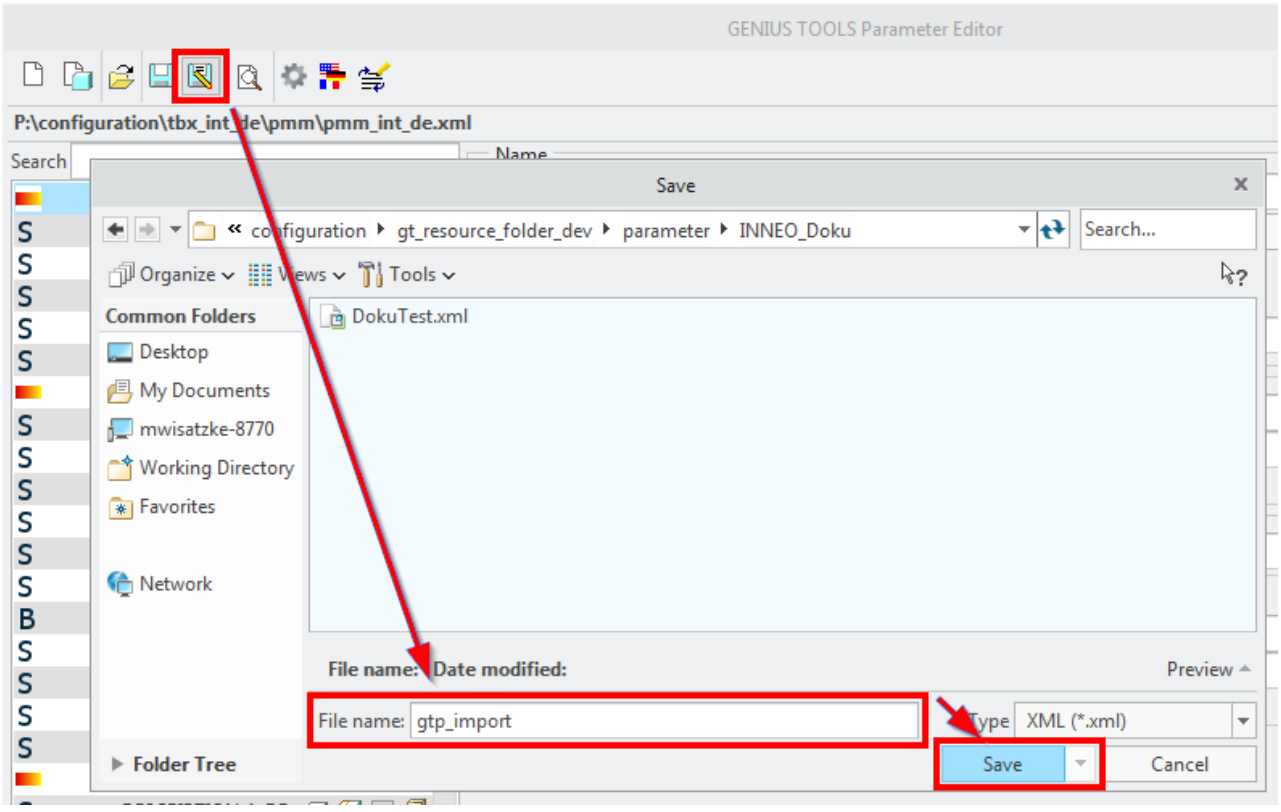
1. In Windows Explorer, create a subfolder under *<GTfCInstallationDirectory>\configuration\gt_resourcefolder\parameter* following the name scheme *companyname_parameterdefinitionname*.
2. Open the old parameter definition in *Parameter Editor*.


© 2026 INNEO Solutions GmbH




*Open the old Parameter definition in the editor*

3. Click on *Save as* and save the parameter definition in the newly created folder. The old parameter definition is adopted.



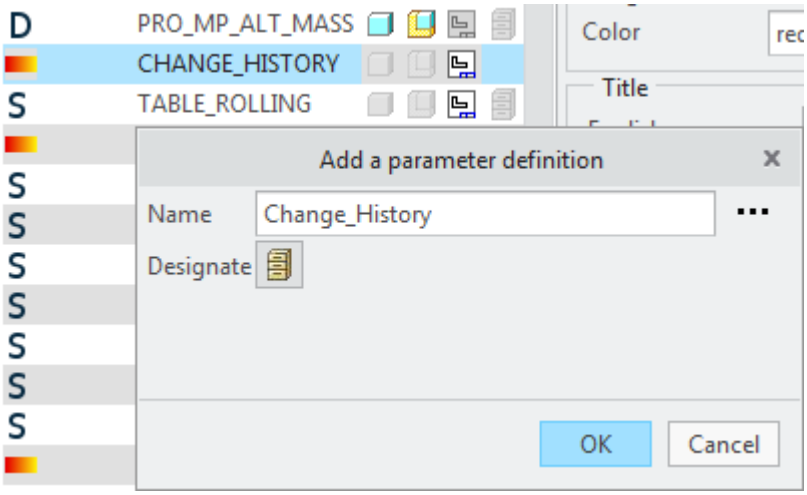

*Save the parameter definition with a new name*


© 2026 INNEO Solutions GmbH

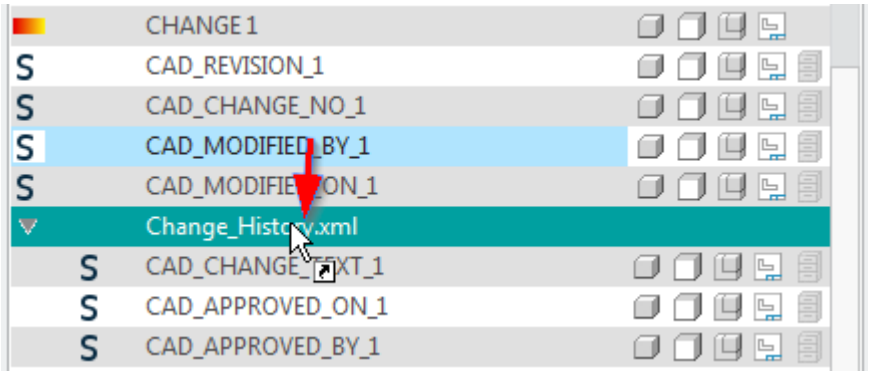
## Splitting into sub-definitions

If several tabs should be displayed in the parameter form rather than a long list of parameters, the parameter definition has to be split up into sub-definitions. If separators were used in the *TOOLBOX Parameter* manager to logically separate different parameters, they can now be grouped into topics via sub-definitions.

Open the context menu in the element list and add the required sub-definitions.




Select parameters or separators that should be moved and drag-and-drop them into a parameter definition.




When the creation of the sub-definitions and moving of the parameters and separators are done, the splitting process is complete.

## Customizing the file and database connections

Verify whether the file and database connections in the parameters are up to date and functioning.

Open each parameter by clicking it in the element list and check the settings in the details view. Then, save the parameter definition.

Your new parameter definition for *Parameter* is now ready for use.


© 2026 INNEO Solutions GmbH

## 18.3.3.4 Auto-select parameter definitions

Which parameter definition is applied to a model is specified via the list of parameter definitions (lst file) and a parameter in the model. The list of parameter definitions is specified via the `gtp_lst` configuration option and the selection parameter via the `gtp_file_param` configuration option.

Default for the list of parameter definitions is the *gtp.lst* file; the selection parameter to determine the parameter definition is `MC_CHECKTYPE`.

Additional information can be found in the Configuration[755] section.

# List of parameter definitions structure

Each line in the list must contain three values separated by a pipe (|) character:

– File path of the parameter definition

– Description of the parameter definition

– Parameter value of the selection parameter in models

**Example**

```
Parameter definition|Description|Parameter value
%GT_RESOURCE_FOLDER%\parameter\gtp_sut_int_de\pmm_int_de_fe.xml|manufacturing|MANUFACTURING
%GT_RESOURCE_FOLDER%\parameter\gtp_sut_int_de\pmm_int_de_be.xml|provision|PROVISION
%GT_RESOURCE_FOLDER%\parameter\gtp_sut_int_de\pmm_int_de_so.xml|other|OTHER
```

When a model is opened with GENIUS TOOLS Parameter and the parameter `MC_CHECKTYPE` has the value *PROVISION*, the second parameter definition for the model will be used automatically.

**Please note**: If the list contains duplicate parameter values, always the last entry will be used.

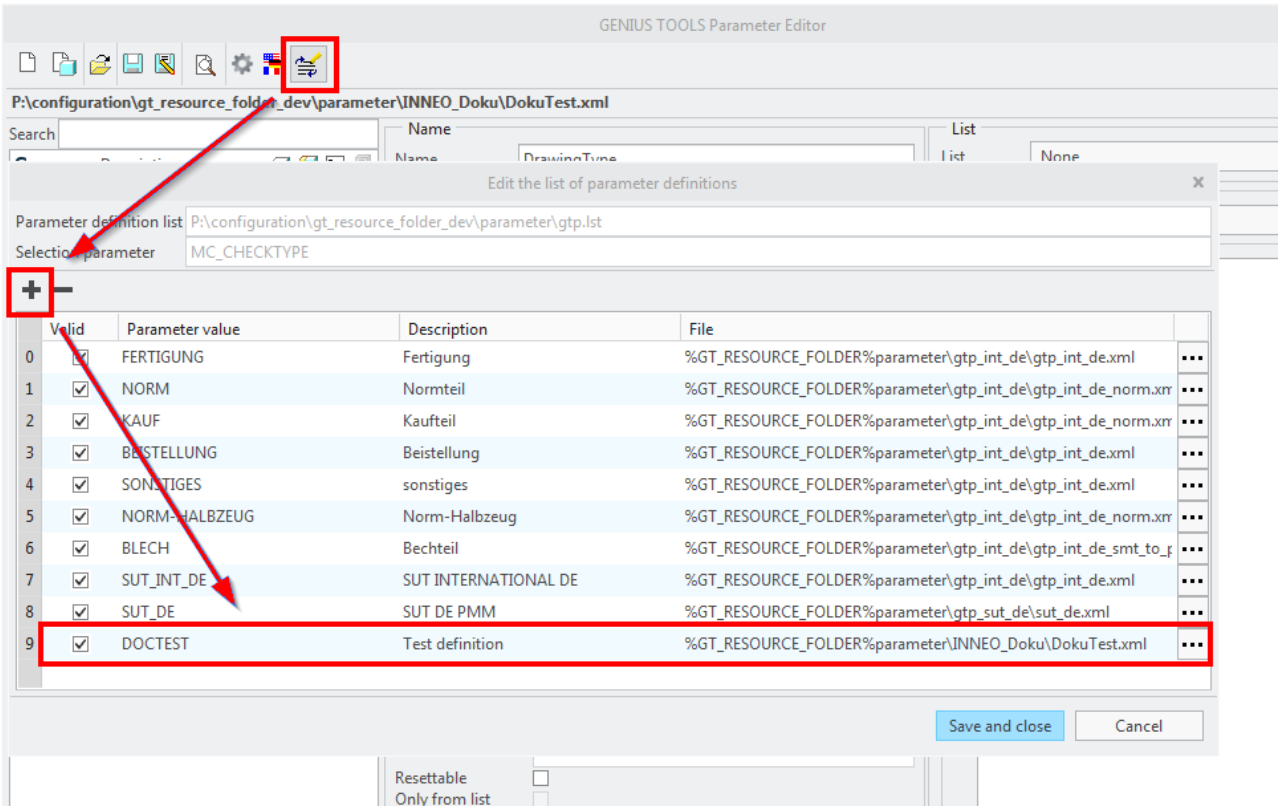
# Expanding the list of parameter definitions and start models

The list of parameter definitions can be edited manually as well as via the Parameter Editor user interface.

Proceed as follows to expand the parameter list in the editor with an additional entry:

1. Open the Parameter Editor.
2. Click on the *Edit list of parameter definitions* button.
3. Add a new entry to the list.
4. Enter the value for the selection parameter and a short description of the parameter definition.

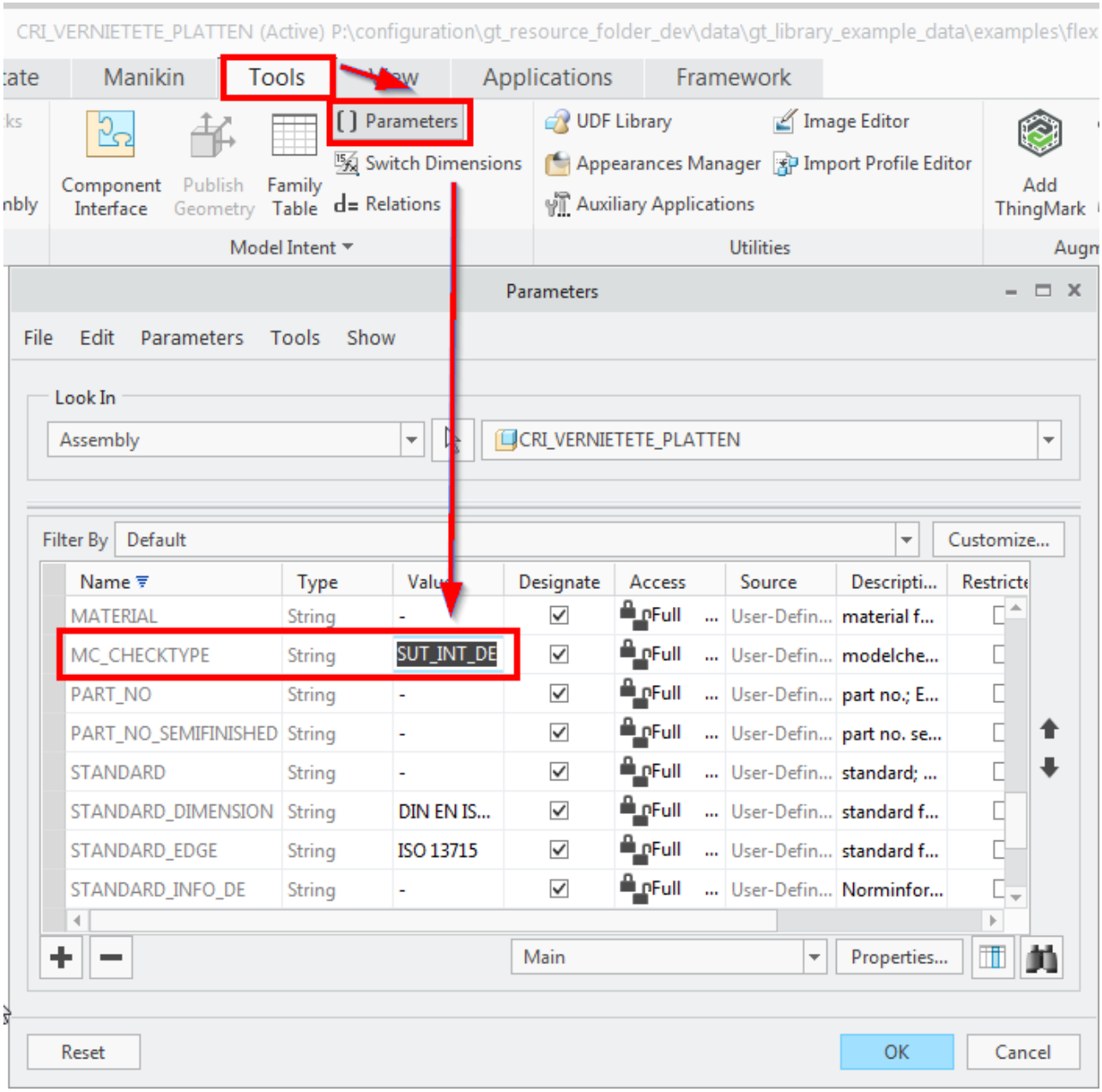

© 2026 INNEO Solutions GmbH




*Open the parameter definition list and add a new entry*

**Please note**: If you have modified the `gtp_file_param` configuration setting, the parameter value must be deposited in the parameter specified there.

5. Navigate to your start models folder in Creo Parametric. You will find the start models (when using Startup TOOLS ) at *Library/start_model_dir* (<*ProjectName*>*library_dir**start_model_dir*).
6. Open a start model. In the *Tools* ribbon click *Parameter* and search for the `MC_CHECKTYPE` parameter. If you are using an individual selection parameter, create it in the start model.
7. As the parameter value enter the value you have specified in the list file.




*Switch to the "Tools" ribbon in Creo, open the Parameter dialog and edit the selection parameter*

8. Save the start model.
9. Repeat this procedure for each start model used.

Your parameter definition will now be applied to each model that has been created with the modified start models.

## 18.3.3.5 Using JavaScript with Parameter

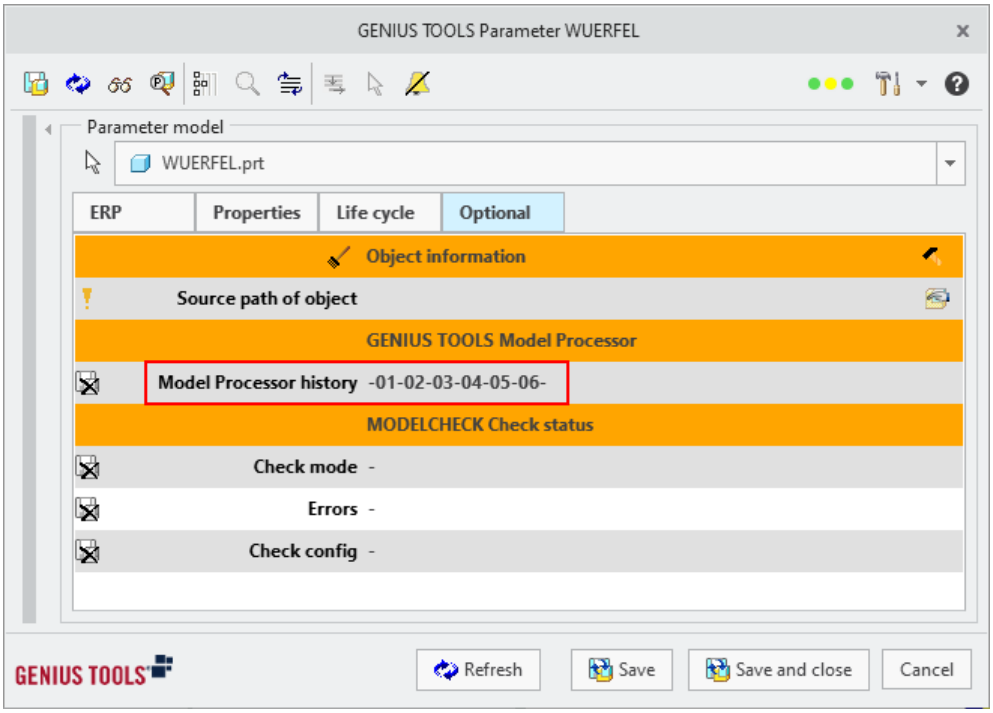
This example demonstrates the practical use of JavaScript functions for managing parameters with the module *Parameter*[465].


© 2026 INNEO Solutions GmbH

# Requirements

First, consider what goals are to be achieved with the JavaScript functions. To record this, the requirements are specified here. The required JavaScript functions are then derived from the requirements.

(a) When opening the module *Parameter*, check whether the opened model meets the current data requirements. For the data requirement to be recognized as current, a model must have reached revision level 6 through revision with the *Model Processor*.




(b) In addition, the unit for specifying the mass in drawings is to be checked: For mass specifications < *1 kg*, the mass has to be indicated in *g* instead.

**Tip:** This requirement can also be solved with relations. However, the implementation with JavaScript provides the advantage that the function is more flexible and does not require any regeneration time. A JavaScript function is executed only once, while relations are executed continuously.

# Necessary JavaScript functions

First, functions are needed that define the content and second, functions that define the time of execution. In this example there are two functions that define the time and two functions that define the content. For clarity of the functions, the execution times are described here first and below the functions for the actions. Each function is separated from the next by a blank line. So the scope of a function can be seen directly.


© 2026 INNEO Solutions GmbH

## Writing JavaScript functions

1. Open the model in which you want to apply the JavaScript code.
2. Open *Parameter*.
3. Open *Parameter Editor*.
4. Select the subparameter definition *gtp_sut_sub_optional.xml* to see the properties stored there.
5. Go into the Details view to the JavaScript section. Click the icon  to open the JavaScript Editor.

**Tip:** It is recommended to work with the *JavaScript Editor* as it provides access to predefined functions that can be used as templates. These functions help you maintain the formatting required for the JavaScript code to be executable, such as the position and number of braces.

## Writing JavaScript functions: Defining points in time

6. Insert via *right click* > *Insert function* > *Function Bodies* the function `PostLoad` to define the execution time.
7. According to the defined requirements, two checks should take place ((a) and (b)) when opening *Parameter*. Therefore, the `PostLoad` function is needed to execute the function immediately after loading when the model is opened.

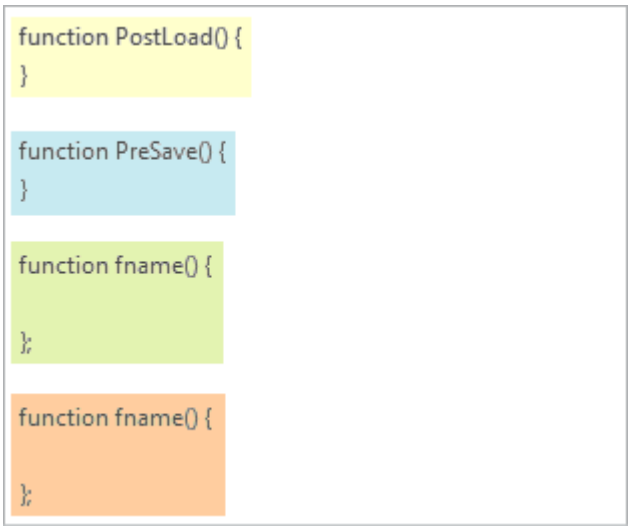
```
function PostLoad() {
}

function PreSave() {
}
```

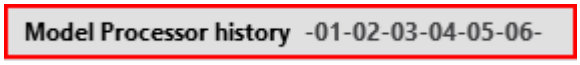
## Writing JavaScript code: Defining contents

8. For the contentual execution of the requirements, both self-written and predefined functions are necessary. This is implemented via *right click* > *Insert Function* > *Empty Function*. Insert an *Empty Function* twice. You will receive templates into which the requirements will now be inserted.


© 2026 INNEO Solutions GmbH




9. Define requirement (a):
   a. Name the function. `fname` becomes `MP_HISTORY_ALERT`.
   b. Describe the information to be searched for as a variable. Base this on how the value to be checked is created in *Parameter*:
      `var CURR_MP_STATUS="-01-02-03-04-05-06-";`

   c. If the revision level of a model corresponds to this format, there is no effect. However, in a model that has not yet reached this revision level, a warning message should be issued to indicate the need for the user to revise the model. To achieve this distinction, insert an IF loop: `if(){}`
   d. Fill this IF loop:
      i. Insert the query for the `MP_HISTORY` parameter: *right-click* > *Insert Function* > *getInputValue* > *getInputValue("MP_HISTORY");*
      `getInputValue` queries the valid revision level with the parameter `MP_History`.
      `!=` compares the result with the actual revision state with the variable `CURR_MP_STATUS` defined above.
      ii. To make it clear that the model needs to be revised, the corresponding field in *Parameter* can be colored red. Insert the color change of the field in question: *Right click* > *Insert Function* > *setBgColorValue* > *setBgColorValue("MP_HISTORY", 10,100,200);* and set as color *red (255,0,0)*.
      iii. Add the output of a warning message: *Right click* > *Insert Function* > *Print messages* > *alert*
      Here the message is created bilingual: `alert('Datenqualität veraltet ... Bitte GT MPU starten! \n Data quality outdated ... Please run GT MPU!');`
      The function checks if the opened model corresponds to the current status of the revision with the *Model Processor*: "-01-02-03-04-05-06-". The Model Processor status must correspond to this scheme so that it is clear that all necessary updates


© 2026 INNEO Solutions GmbH

have passed through this model. If this is not the case, this error message will be issued.

```
function MP_HISTORY_Alert(){
 var CURR_MP_STATUS = "-01-02-03-04-05-06-";
 if (getInputValue("MP_HISTORY") != CURR_MP_STATUS) {
    setBgColorValue("MP_HISTORY", 255,0,0);
    alert('Datenqualität veraltet ... Bitte GT MPU starten! \n Data quality outdated ... Please run GT MPU!');
  }
}
```

*JavaScript function for requirement (a)*

10. Define requirement (b):
    a. Name the function. `fname` becomes `MassDRW`.
    b. Create a new variable: `var MASS_Creo`
       Describe the information to be searched for as a variable.
    c. Insert the query for the parameter and set it to `false` to not round the value according to the rules of the Creo configuration option `PARAM_DEC_PLACES`. Since the parameter refers to the active model of a drawing, insert the function `creoDrwActiveMdlNameGet()` and enter the name of the drawing:
       `creoParameterValueGet( creoDrwActiveMdlNameGet() ,"PRO_MP_MASS", false);`
    d. Include an `if-else` loop so that if one rounding rule does not apply, the other is executed:
       ```
       if(){
       }
       else{
       }
       ```
    e. Specify `if`:
       Instead of using the Creo configuration option's rounding rules, now define your own rule to set *kg* as the default unit.
       `var MASS_round = Math.round (MASS_Creo * 100) / 100;`
    f. Specify `if`:
       If the value can be rounded according to this rule, the unit is set to *kg*:
       `var MASS_unit = " kg";`
    g. Specify `else`:
       Define a second rule to move the decimal place backwards. The indication of *0.001 kg* becomes *1 g*.
       `var MASS_round = Math.round (1000 * MASS_Creo * 100) / 100;`
    h. Specify `else`:
       If the value can be rounded according to this rule, the unit is set to *g*:
       `var MASS_unit = " g";`
    i. Define a parameter that controls the display of the defined rounding rules:
       `var MASS_display = MASS_round.toString()+MASS_unit;`


© 2026 INNEO Solutions GmbH

j. Set the parameter to a value with the corresponding unit:
Use *right click* > *Insert Function* > *setInputValue* to add the function `setInputValue("MASS", val);`.

k. Assign the variable `MASS_display` to the function.

```
function MassDRW(){
  var MASS_Creo = creoParameterValueGet( creoDrwActiveMdlNameGet() ,"PRO_MP_MASS", false);
  if (MASS_Creo >= 1) {
  //alert(MASS_Creo);
    var MASS_round = Math.round (MASS_Creo * 100) / 100;
    var MASS_unit = " kg";
  }
  else
  {
    var MASS_round = Math.round (1000 * MASS_Creo * 100) / 100;
    var MASS_unit = " g";
  }
  var MASS_display = MASS_round.toString()+MASS_unit;
  setInputValue("MASS_DRW", MASS_display);
}
```

11. Enter the functions `MP_HISTORY_Alert` and `MassDRW` in the functions for executing the time points.
`MP_HISTORY_Alert();` is executed once when *Parameter* is opened.
`MassDRW();` is executed once when opening and once when saving, so that the correct mass information is loaded (`PostLoad`) and the correct mass information is saved if the material has been modified with *Parameter* (`PreSave`).

```
function PostLoad() {
  MassDRW();
  MP_HISTORY_Alert();
}

function PreSave() {
  MassDRW();
}
```

**Please note:** Function names are case sensitive. Make sure that the function name and the function call are written the same way.

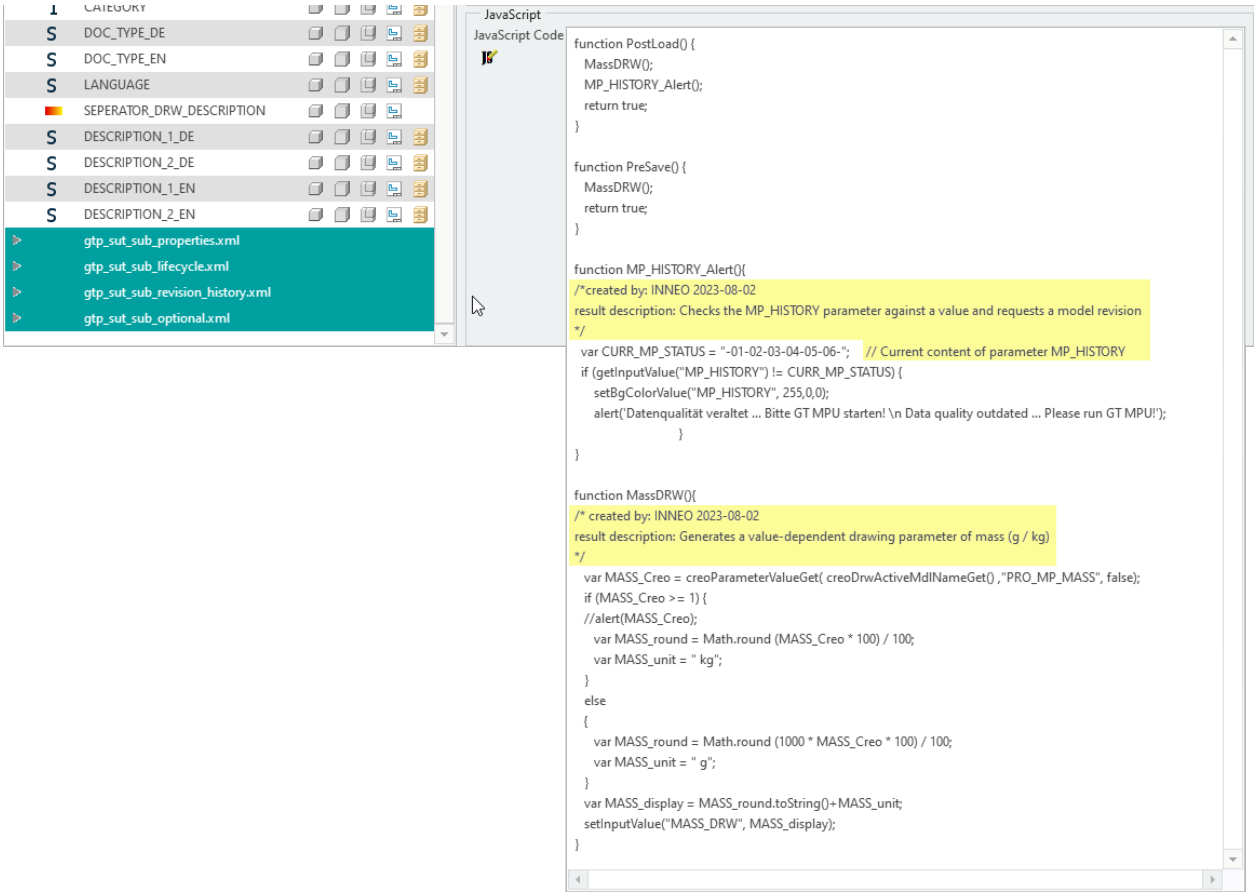
12. At the end of these functions select via *right-click* > *Insert Function* > *return...* > *return true;*. This function confirms that saving *Parameter* is allowed in any case. With `return false;` you would prevent that a model that does not correspond to the current revision level can be saved.


© 2026 INNEO Solutions GmbH

```
function PostLoad() {
  MassDRW();
  MP_HISTORY_Alert();
  return true;
}

function PreSave() {
  MassDRW();
  return true;
}
```

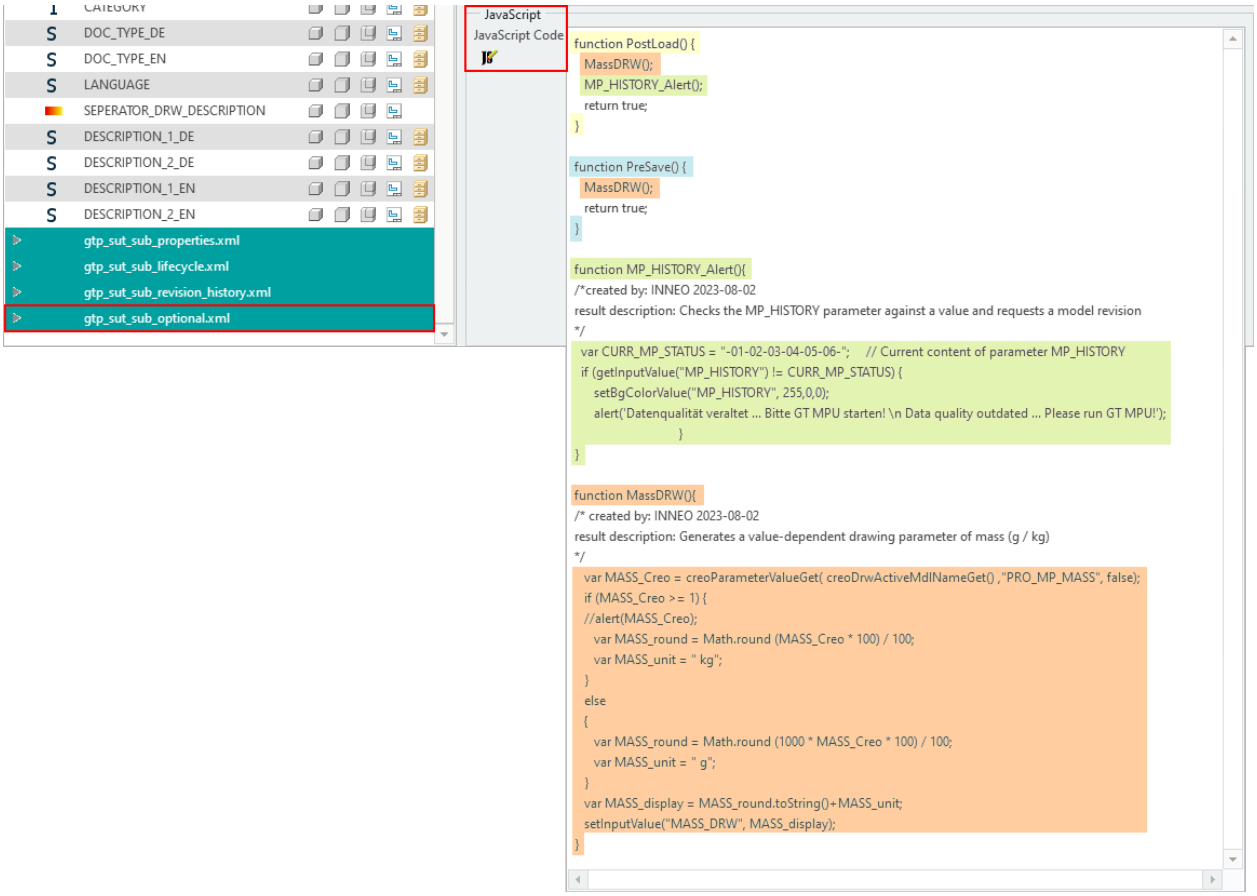
13. Add comments to document intent, creator, date of creation, etc.




# Complete JavaScript code

14. The final JavaScript code looks like this:


© 2026 INNEO Solutions GmbH




15. In the *JavaScript Editor*, click *Save*.
16. In the *Parameter Editor*, click *Save and quit*.
17. In the *Parameter*, click *Refresh* . Now the JavaScript code is applied.

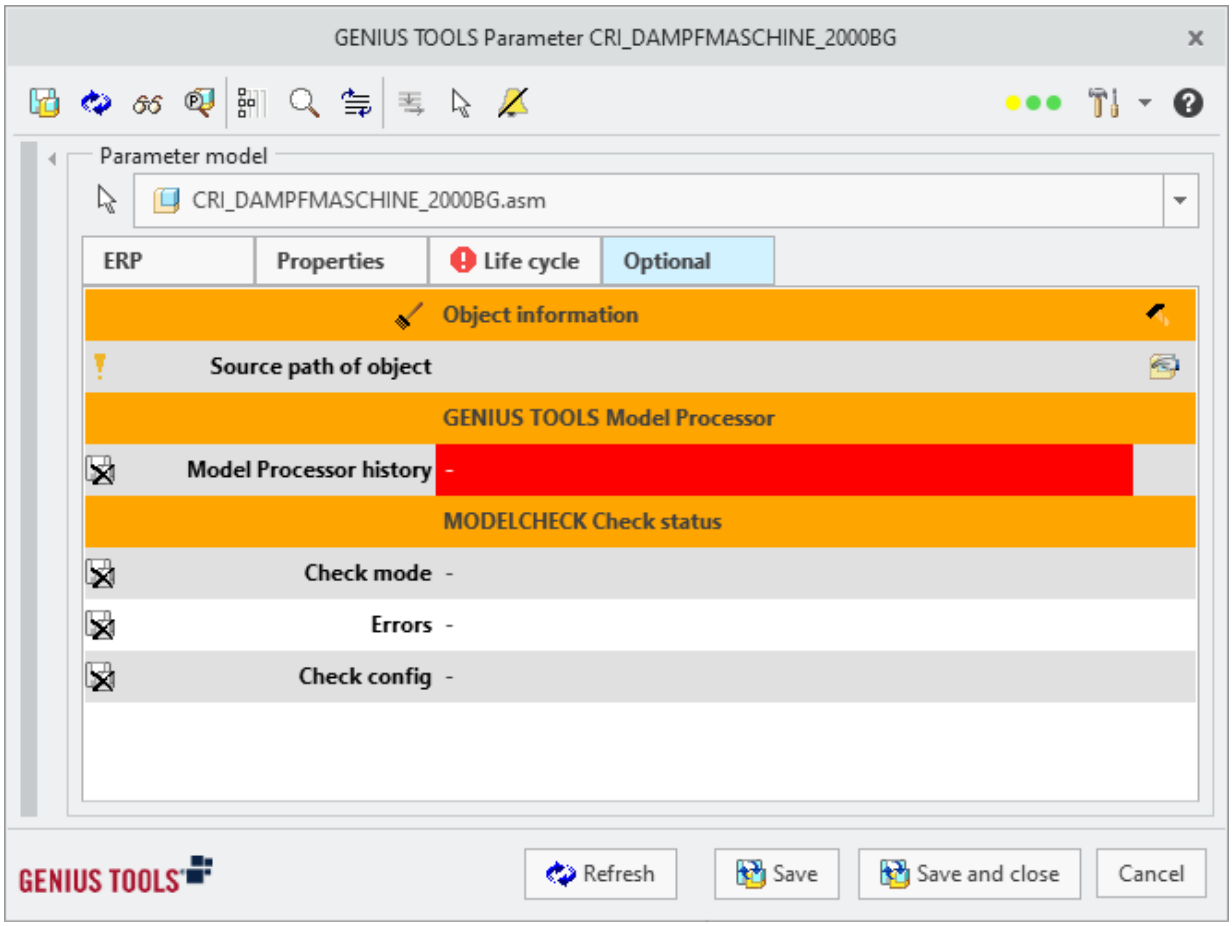
## Applying JavaScript functions

18. If *Parameter* is opened in a model and revision level 6 is not available according to requirement (a), this warning message is displayed. This message can be closed by clicking *OK*.




19. In *Parameter*, the line *Model Processor History* is then displayed with a red background.


© 2026 INNEO Solutions GmbH




20. You can fix this error by using the *Model Processor* to revise the model according to current company guidelines. Alternatively, you can ignore the error and save and close *Parameter*.

21. When you open *Parameter* from a drawing, the mass specification appears in the desired unit (Implementation of requirement (b)).




*Masseangabe bei ≥ 1 kg*




*Masseangabe bei < 1 kg*


© 2026 INNEO Solutions GmbH

# 19 Quick Access

Available in the ribbon menus GENIUS TOOLS | GT MBD

*Quick Access* is a database-dependent ring menu for Creo Parametric. It allows you to start normal and intelligent commands (mapkeys) with short mouse movements. This significantly speeds up the work with Creo Parametric.

Intelligent mapkeys have extended functionality and allow the use of variables, parameters and placeholders.

*Quick Access* is available in all Creo modes with the following functions:

1. Using regular and intelligent mapkeys
2. Launching *GENIUS TOOLS* for Creo and any Creo functions via short mouse movements
3. Different usage scenarios[548]:
   - central configuration (global)
   - user-specific configuration (local)
   - simultaneous central and user-specific configuration (mixed mode)
4. Easy-to-use editor for homogeneous working environment
5. Export and import of all created mapkeys with images and descriptions for easy data exchange

## 19.1 Usage

This section contains information on using *Quick Access*. It describes the general structure of the program.

### Starting the program

Start *Quick Access* via the key [<] in a Creo session. You can change the start key, see Changing the start key[561]. Assign the action to open to a button on your multi-button mouse. The Creo command is only active within the Creo environment.

*Quick Access* opens at the current mouse position. Depending on the current Creo mode, different commands are displayed in the ring menu.


© 2026 INNEO Solutions GmbH



To close *Quick Access*, press the ESC key, the designated Start key [<], or click outside the ring menu without executing a command.

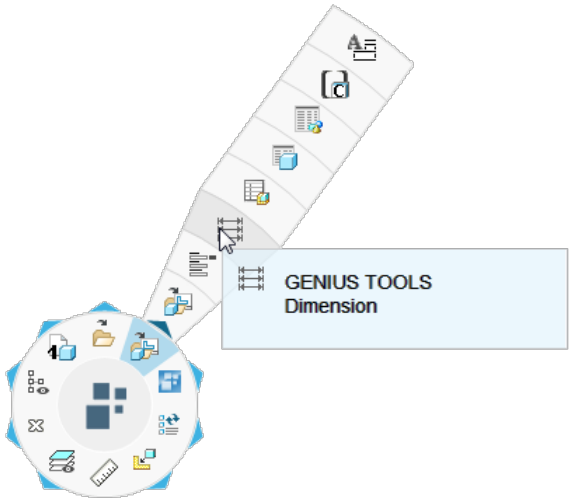
## 19.1.2 User interface

Quick Access displays command icons in two ways:

1. Single icons: Single icons are displayed directly in the menu ring. Clicking the icon starts the associated mapkey.
2. Group icons: Group icons are indicated by a blue arrow. The displayed icon is the first icon of a group. Move the mouse over the blue arrow to expand a group.

Greyed-out icons cannot be used for the current selection.

If you are not sure about what an icon will execute, leave the mouse pointer over the icon. Tooltips will be displayed shortly.

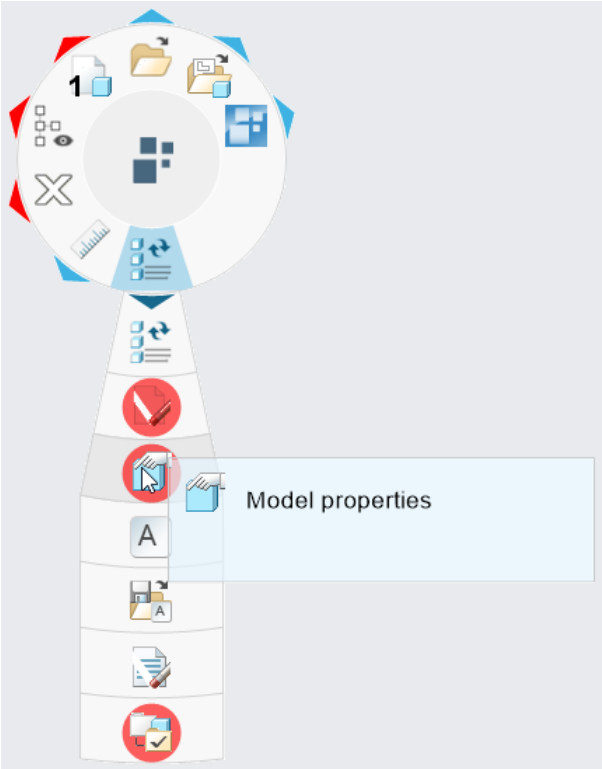



*Leave the mouse pointer over an icon to display its tooltip*

## Searching for elements

You can search for elements in *Quick Access*. Enter the letters you are looking for, e. g. "mo", "model". Categories containing search results will be highlighted in red. Click on an arrow to view the search results it contains, which are also highlighted in red. For clarity and performance reasons, only one category can be expanded at a time.


© 2026 INNEO Solutions GmbH




*Results for the sample search for "mo"*

You can reset the search by pressing the "Enter/Return" key or by closing and reopening *Quick Access*.

## Undocking individual groups

Individual command groups can be undocked and are permanently visible as a toolbar. Click a group using either the middle (default) or right mouse button to undock it.

Undocked groups are not redrawn mode-sensitively. In addition, they are not greyed-out selection-sensitively.


© 2026 INNEO Solutions GmbH




*Permanently visible toolbar*

## 19.1.3 Mode and selection dependence

Displayed command icons depend on the current Creo mode.

The following modes are available for commands in Creo Parametric and in Quick Access:

| Mode (Acronym) | Description |
|---|---|
| NO MDL | No part opened in Creo Parametric |
| PRT | Part |
| SMT | Sheetmetal part |
| ASM | Assembly |
| DRW | Drawing |
| SEC | Sketch |
| MFG | Sheetmetal assembly |


© 2026 INNEO Solutions GmbH

| Mode (Acronym) | Description |
|---|---|
| DGM | Diagram |
| FRM | Frame |
| LAY | Layout |

## Selection-dependent mapkeys

Apart from the Creo modes, there is another dependency in *Quick Access*: Selection.

Each displayed icon represents two Mapkeys executing different command sequences:

1. without selection in Creo Parametric
2. with selection in Creo Parametric

This ensures executing the correct command sequences as Mapkeys can have a slightly different sequence depending on the selection in Creo. If a Mapkey is not deposited, the icon is greyed-out in this selection state.

# 19.2 Configuration

In this section you will find information on how to configure *Quick Access* with the Editor[551] and for the display options of the *Quick Access* ring menu, i. e. different zoom levels[563] and the use of custom images.

## 19.2.1 Different types of use

*Quick Access* can be customized for use within companies in different ways. Depending on the preferred way of working, it is possible to work with a global, a local or a mixture of both databases. Thus, there are three different ways to work with *Quick Access*:

1. **Global configuration (as-delivered condition)**[549]
   In a global database, all users access the same database. If one user makes changes to the database, these changes are also valid for all other users.
2. **Local configuration**[549]
   A local database is a copy of the global database. Each user gets his own local database, which he can freely modify and extend. Each user can view and edit only his own database.


© 2026 INNEO Solutions GmbH

3. **Global and local configuration: Mixed mode (recommended mode of operation)**[550]
   Working with global and local databases can be combined, e. g. by predefining one part of the ring menu globally and releasing the other part to the user for individual configuration.

**Warning:** GENIUS TOOLS Quick Access uses a different database for each Creo version. The databases are addressed via the variable `%GT_CREO_VERSION%`. If you use multiple Creo versions in your company, all databases must be adapted for the corresponding versions. Alternatively, the backward compatibility of a database must be tested by an administrator.

The following sections describe the most important configuration options for these types of use. In addition, further configuration options can be set. Additional information on the configuration options can be found in the section Configuration[799].

The ring menu groups 8 and 9 are predefined by the default settings of *GENIUS TOOLS* and can only be adjusted with a change in the configuration option `gtqa_change_closed_groups` afterwards.

# Global configuration (as-delivered condition)

Global configuration means that all employees in the company work with a *Quick Access* configuration that is the same throughout the company. To allow all employees in the company to work with the global database managed by the administrator, set the following configuration options. No user other than the administrator is authorized to make changes.

– **gt_start_quick_access_editor**
  Must be activated for administrator. Value: 1. For all other users, this configuration option must be deactivated. Value: 0.

**Please note:** If you work with the *GENIUS TOOLS Startup TOOLS*, `%GTFC_ADMIN%` is preset.

– **gtqa_command_file**
  Refers to the file of the global *Quick Access* database.
  `(%GT_RESOURCE_FOLDER%\quick_access\quick_access_%GT_CREO_VERSION%.db`

– **gtqa_command_icon_folder**
  Refers to the global icon resource directory of Quick Access.
  `(%GT_RESOURCE_FOLDER%\quick_access\img_w20)`

# Local configuration

Local configuration means that each user works with his own local database (= copy of the global database). The administrator does not make any specifications in the ring menu and therefore there are no company-wide specifications for the use of *Quick Access*. Each user can and must freely configure his own *Quick Access* ring menu and customize it himself via the editor.


© 2026 INNEO Solutions GmbH

– **gt_start_quick_access_editor**
  The Quick Access Editor must be executable. Value: 1

– **gtqa_command_file**
  Refers to the user-specific file of the *Quick Access* database:
  `(%appdata\INNEO\GENIUS_TOOLS\for_Creo\quick_access\quick_access_%GT_CREO_VERSION%.db)`

– **gtqa_command_icon_folder**
  Refers to the global icon resource directory of *Quick Access*.
  `(%appdata\INNEO\GENIUS_TOOLS\for_Creo\quick_access\img_w20)`

– **gtqa_local_command_group_split**
  Must use the value "all" (default).

## Global and local configuration: Mixed mode (recommended mode of operation)

We recommend using a combination of global and local configuration. You can make part of the commands available to all users via the global database and at the same time prevent users from making unintended changes to this database. Users who want to configure further commands via *Quick Access* in addition to the predefined commands can do so via their own local database.

– **gtqa_command_file**
  Refers to the file of the global *Quick Access* database.
  `(%GT_RESOURCE_FOLDER%\quick_access\quick_access_%GT_CREO_VERSION%.db)`

– **gtqa_command_icon_folder**
  Refers to the global icon resource directory of *Quick Access*.
  `(%GT_RESOURCE_FOLDER%\quick_access\img_w20)`

– **gtqa_local_command_file**
  Refers to the user-specific file of the *Quick Access* database:
  `(%appdata\INNEO\GENIUS_TOOLS\for_Creo\quick_access\quick_access_%GT_CREO_VERSION%.db)`

– **gtqa_local_command_group_split**
  Defines how many *Quick Access* command groups may be customized by end users (0: only the first group can be edited to 7: all groups can be customized).

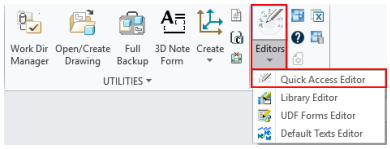
– **gtqa_local_icon_folder**
  Refers to the user-specific icon resource directory of *Quick Access*.
  `(%appdata\INNEO\GENIUS_TOOLS\for_Creo\quick_access\img_w20)`

## 19.2.2 Quick Access Editor

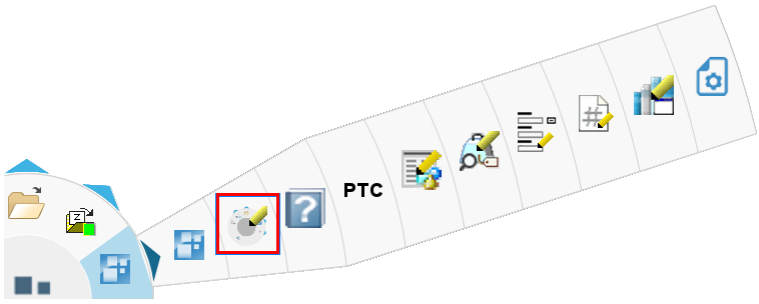
The Editor is used to manage the commands and functions in *Quick Access*.

### Starting the program

Start *Quick Access Editor* from the ribbon menu in the *GENIUS TOOLS* tab or via *Quick Access* ([<] key).




*Starting via the ribbon menu*




*Starting via Quick Access*

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors 781.

### Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

### Save database

Modifications made with *Quick Access Editor* are immediately saved to the *Quick Access* database.

A backup copy of the database is created once a day by default. The first change starts this process. You can disable the creation of the backup copy with the configuration option `gtqa_editor_create_db_security_copy_once_a_day`.


© 2026 INNEO Solutions GmbH

## 19.2.2.1 User interface

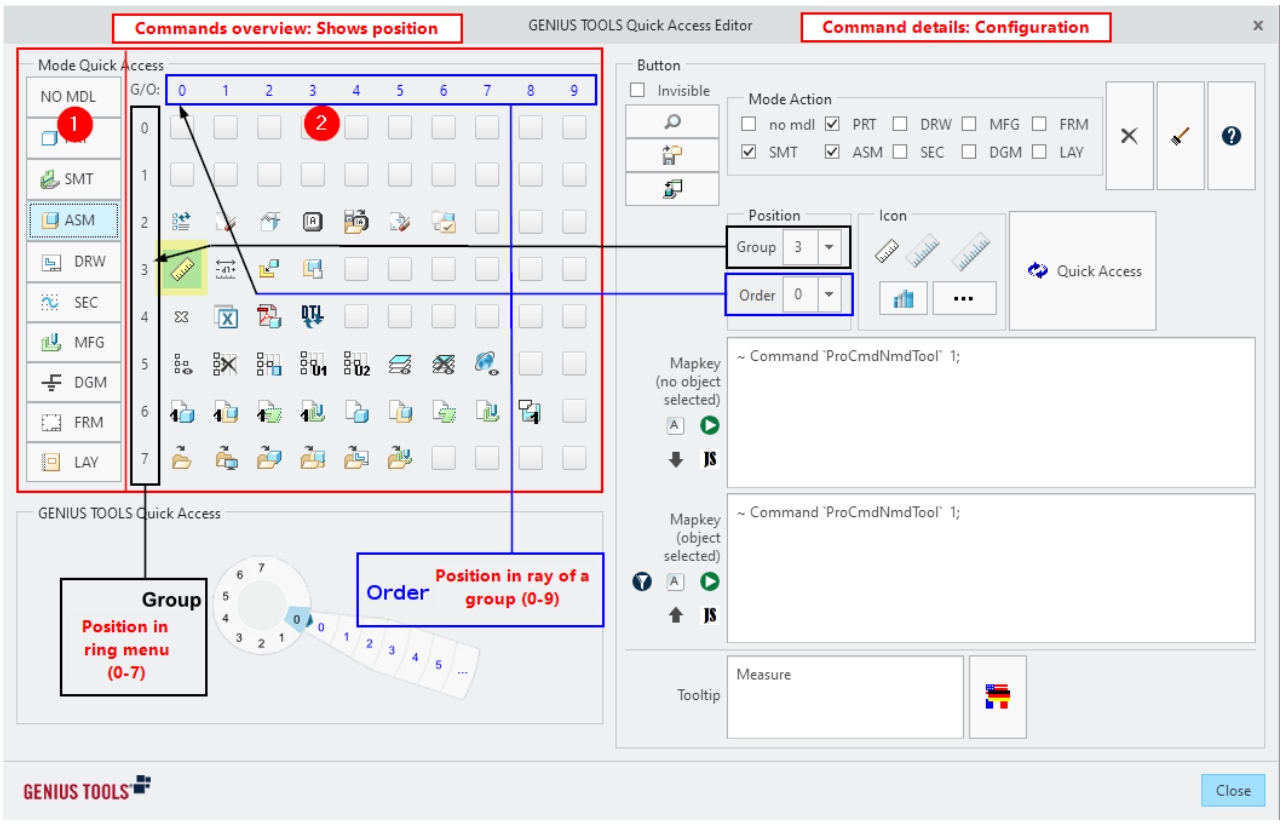
*Quick Access Editor* is divided into two areas:

1. Command overview[552] (left): Displays commands of all groups in a specific mode.
2. Command details[553] (right): Configuration of individual commands.

## 19.2.2.2 Commands overview

The commands overview displays commands already created in the available Creo modes.

Commands displayed in *Quick Access* always depend on the Creo mode. The commands overview shows all created commands in the available Creo modes. On the left side, the different Creo modes (1) are displayed. Click on a mode to see the command grid (2) for that mode.




*Creo mode (1) and command grid (2) in the commands overview*

Click on a command in the command grid (2) to display the configurable properties of the command in the Command details area on the right. Control the positioning of the commands there with *Group* and *Order*.

The different groups show the position of the command in the ring menu (0-7, vertical display). The position within a group is the order which is displayed horizontally in the command grid (0-9).


© 2026 INNEO Solutions GmbH

You can store 80 commands (8 groups x 10) with individual icons per mode. To increase the number of customized commands you can adjust the number of groups[548] with the configuration option `gtqa_change_closed_groups`.

**Warning**: Pay attention to sufficient space for displaying the commands! Additional commands are displayed in all other modes via the ALWAYS mode. This may cause that not all modes can be displayed in the ring menu. A warning message will be displayed when saving overloaded groups.

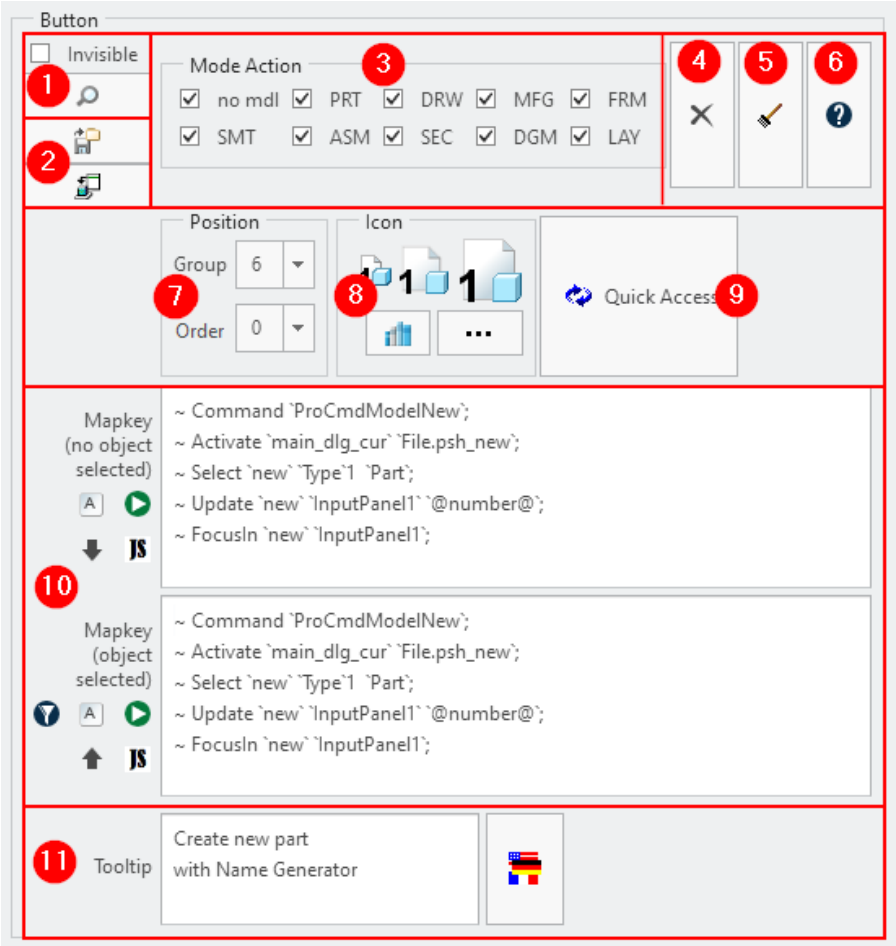
The following Creo modes (1) are available:

| Acronym | Description |
|---|---|
| ALWAYS | Always (This mode must be released via a config option `gtqa_always_option`) |
| NO MDL | No part opened in Creo Parametric |
| PRT | Part |
| SMT | Sheetmetal part |
| ASM | Assembly |
| DRW | Drawing |
| SEC | Sketch |
| MFG | Sheetmetal assembly |
| DGM | Diagram |
| FRM | Frame |
| LAY | Layout |

## 19.2.2.3 Command details

In the command details area you can configure a command.


© 2026 INNEO Solutions GmbH




1. **Visibility settings:** Deactivates the command without deleting it. The position is saved. To retrieve a command, click the magnifying glass icon. After a refresh of *Quick Access*, the command is available again.
2. Import / export[560]
3. **Mode action:** Select the Creo modes in which a command is available in the *Quick Access* ring menu.
4. **Delete command**
5. **Clean up database**: Deletes all images not in use in the *Quick Access* database.
6. **Open help**
7. **Position:** Determines the position of the command in the *Quick Access* ring menu. Specify a group for each command as well as the order in the ring menu beam for this group. Commands with the same group and order are displayed one after the other.
8. **Image:** Select an icon for the display. You can select different sizes to support zoom levels for different monitor sizes and resolutions, see Adapting zoom levels and icons[563].
9. **Refresh Quick Access:** Reloads all commands. Updated commands are available instantly.


© 2026 INNEO Solutions GmbH

10. **Mapkey:** Defines a command[555], e.g. a mapkey,[556] with and without having selected an object.
11. **Tooltip:** Enter copy to get a monolingual tooltip. To store a multilingual tooltip, click on the flag icon  . In the dialog that opens, a tooltip can be defined for each language required. Standard texts can be selected via the  button, see also the description of the standard text selection dialog.[678]

## 19.2.2.4 Creating commands

*Quick Access* command are created as follows:

1. Open the *Quick Access Editor*.
2. In the Commands overview[552] select the Creo mode in which to create a new command.
3. Select a free field in the commands overview. Define the group and sorting in the area Position (7).
4. Select the Creo modes in which to use the command at *Mode Action* (3).
5. Select an icon (8) from the library or from your own files.
6. Specify the sequence of commands[556] (10), e.g. for a mapkey, with and without selection.
7. Enter a tooltip (11).
8. Refresh *Quick Access*. (9)

**Tip:** Store templates in the *Quick Access* database by making ready configured commands invisible (1).

## 19.2.2.5 Defining commands

There are different ways to create a command:

– Mapkeys: see next chapter[556]

– Commands with starttags: Javascript functions, keyboard entries, opening web sites

## Commands with start tags

You can define commands with the following start tags:

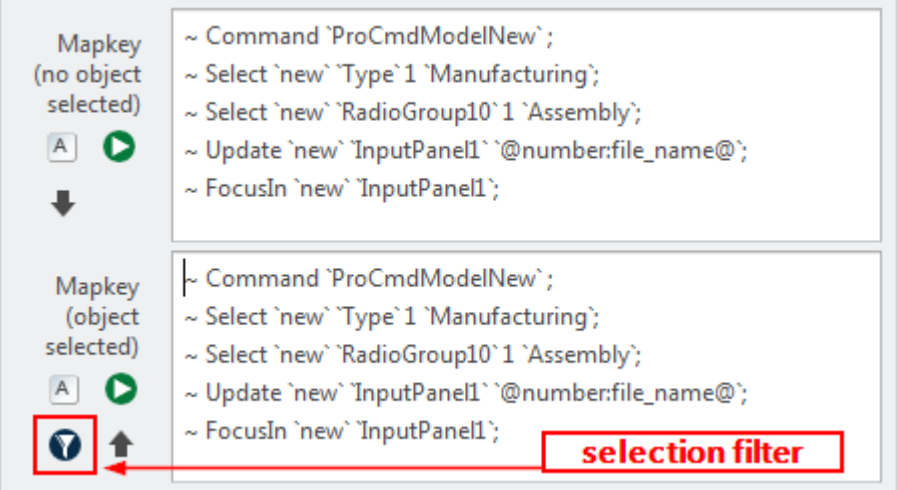
1. **Javascript function:** `js:`
   Executes a Javascript function directly. Use the *JavaScript Editor*[784], which opens with the key  , to configure it.
   Example: `js: alert("Hello engineer!")`

2. **Keyboard entry:** `key_input:`
   Passes a string as keyboard input. Only letters and numbers are supported.
   Example: `keyinput:abc`

3. **Open a web site:** `browser:`
   Passes a character string as URL to the Creo internal browser.
   Example: `browser:www.inneo.de`

## 19.2.2.6 Defining mapkeys

For each command two mapkeys can be specified. They will be executed depending on the situation: with and without selection of an object. Icons are automatically greyed out if the corresponding situation-dependent mapkey is empty.




*Mapkeys with and without selection in Creo Parametric*

Enter the a mapkey into the two fields by either:

– creating a new mapkey[557]

– copying a recorded mapkey[559]

– inserting an existing mapkey[559] with `%Acronym`

For an example of recording a mapkey, see Use cases[565].

We recommend entering a mapkey into each of the fields, even when they do not differ, as to have them always available.

Additional functions are available before the mapkeys:

| Icon | Description |
|---|---|
| A | Shortens a mapkey. |
| ▶ | Plays the mapkey in the current Creo context. |

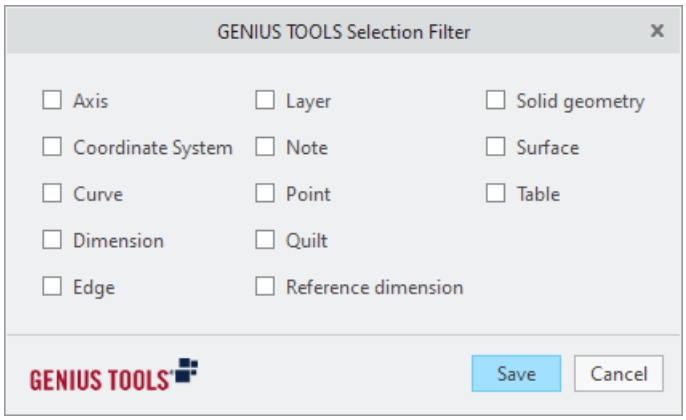

© 2026 INNEO Solutions GmbH

| Icon | Description |
|---|---|
| /  | Copy mapkey downward/upward. |
| JS | Opens *JavaScript Editor*[784] . |
|  | Specifies the Creo objects for which the mapkey is executed on selection. |

**Please note**: Starting a mapkey may close the editor.

## Selection filter

The selection filter defines the Creo objects for which the command should be applied when the object is selected.




Commands with a selection filter are only available in Creo Parametric with the proper selection (axes, layers, edges etc.). If the selection filter is empty, the mapkey is valid for any selected object.

## Creating new mapkeys

Mapkeys are created in the Creo-dialog *Record Mapkeys* and saved in a file. You can either copy the mapkey from that file into the *Quick Access Editor* manually or you can use the *Record Mapkey* function in *Quick Access*.

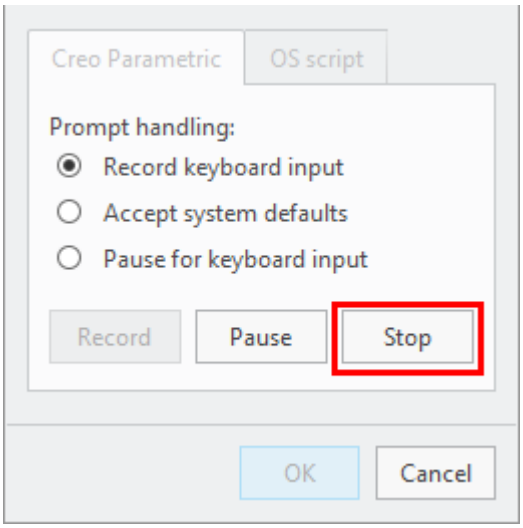
## Creating new mapkeys with Quick Access

1. In *Quick Access* ring menu select the *Record Mapkey* command . This opens the Creo dialog *Record Mapkeys*.


© 2026 INNEO Solutions GmbH



2. Click *Record* in the record dialog.
3. Record the sequence of commands by clicking on all the necessary buttons.
4. Click *Stop* in the record dialog, then click *OK* to end the recording.




5. In *Quick Access* ring menu select *Save modified Mapkeys and open with Notepad*.

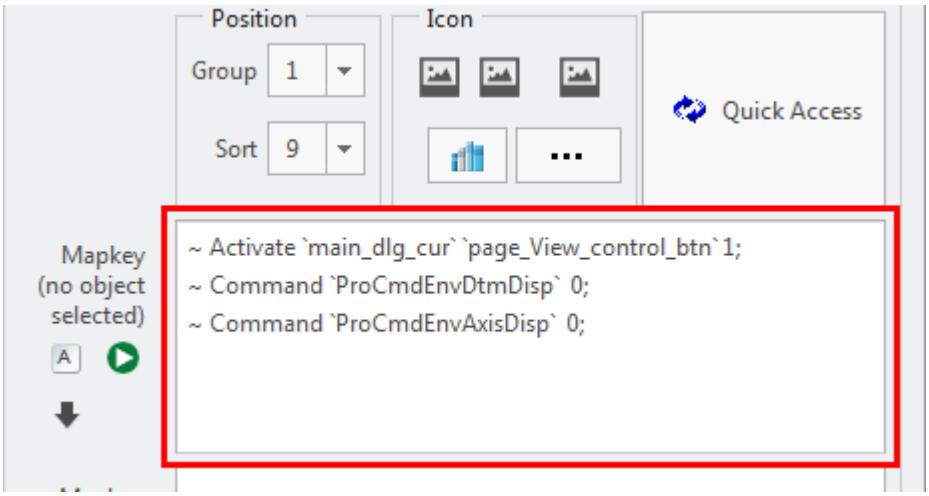


Result: The sequence of commands for this mapkey is saved in the file *config_quick_access_temp.pro* in the working directory.


© 2026 INNEO Solutions GmbH

## Inserting recorded mapkeys from file

1. Open the file to which the mapkey has been saved by
   – clicking on the button *Save modified Mapkeys and open with Notepad* in the *Quick Access* ring menu or
   – by opening the file with a text editor, e.g. Notepad (Mapkeys can be created directly in Creo: *File* > *options* > *Environment* > *Mapkey Settings* > *Mapkeys* > *New*)
2. Copy the entire content of the file (Ctrl+C).
3. Open *Quick Access Editor*.
4. Insert the copied mapkey into the upper right field in the command details section (Ctrl+V).




*Copied content from file config_quick_access_temp.pro (above) and shortened content (below)*

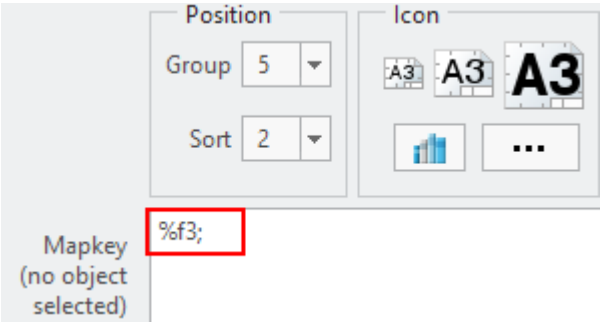
5. Shorten the mapkey by clicking A.
6. Test it by clicking the play symbol.
7. Copy the mapkey into the lower input field to use when an object is selected in Creo.
8. Close the editor. This refreshes Quick Access automatically.

## Inserting existing mapkeys as acroynym

Existing mapkeys are mapkeys that are already available in a Creo session, i. e. that are already in inserted in a Creo configuration file (config.pro). Mapkey acronyms that are saved in a file in the working directory cannot be read into *Quick Access*.

You can insert existing mapkeys by entering the acronym for the keys as follow: `%Acronym`


© 2026 INNEO Solutions GmbH




*Existing mapkey "Load A3 drawing frame" with acrynom f3*

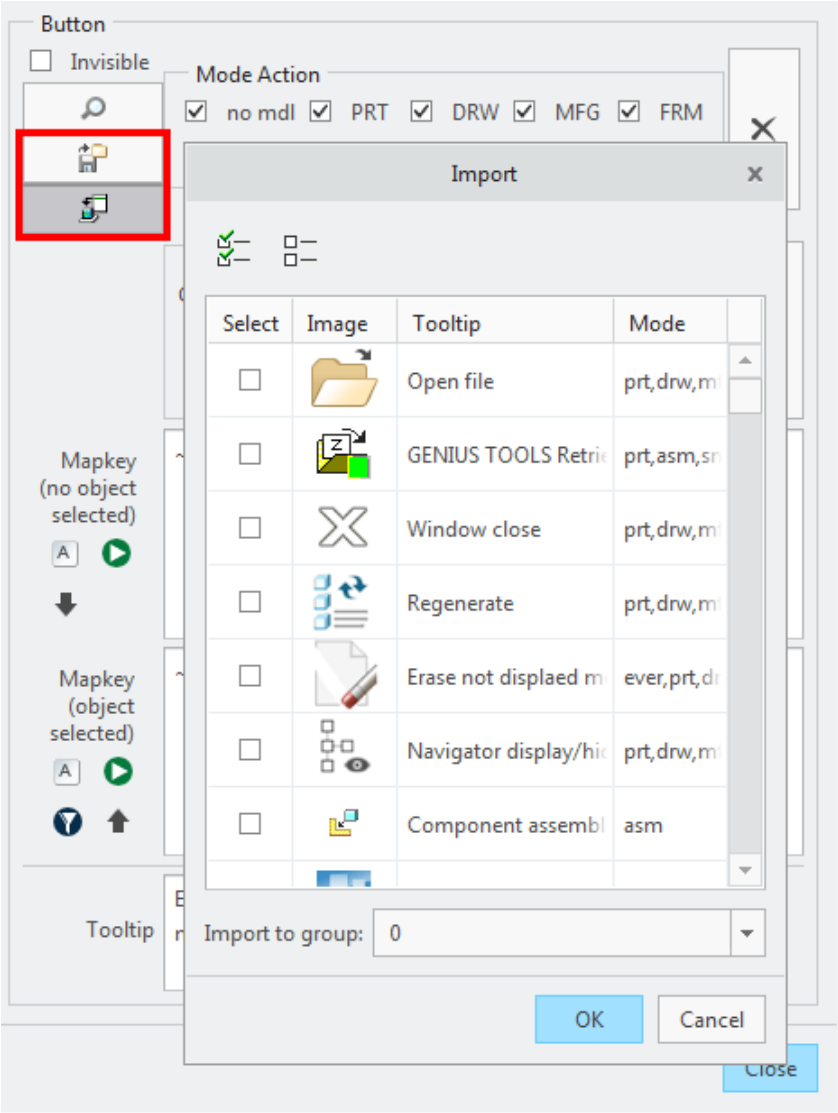
You may insert several existing Mapkeys with `%Acronym1; %Acronym2;` one after the other. These Mapkeys are executed one after the other.

Use `key_input:` to emulate keyboard entries.

## 19.2.2.7 Export and import

Created commands can be exported into and re-imported from a QAEX file (*Quick Access* export file). Images, tooltips, the modes in which a command can be executed and the position are also transported. The position of the commands can be controlled with an import. Import only single commands into a group for this.


© 2026 INNEO Solutions GmbH




*Commands can be re-imported into Quick Access*

You can export and import QAEX files only to the current database. You can implement a cross-database exchange of commands via a separate folder, e.g. the Button Store folder in the *GENIUS TOOLS* resource folder <*GT_Resource_Folder*>.

## 19.2.3 Changing the start key

*Quick Access* opens with the [<] key ("less-than" key) by default. This start key can be changed with the configuration option gtqa_start_mapkey.

**Tip:** We recommend placing the start key on the side of the keyboard where your hand is free, i.e. left-handed users should place the start key on keys on the right side of the keyboard, e.g. [,] or [.].

*Quick Access* closes by default with the [<] or ESC key.


© 2026 INNEO Solutions GmbH

## 19.2.4 Inserting project images

You have the option to place user-specific images in the center of the *Quick Access* ring menu, e.g. for your company logo or to indicate which project a user is currently in.



*Ring menu with the standard GENIUS TOOLS image*



*Ring menu with user-defined image*

**Inserting a project image**

1. Create your image in PNG file format.
2. If you work with different zoom levels,[563] place two additional image files with these additional abbreviations:
   - For medium-sized files: <*nameprojectimage*>_30.png
   - For large files: <*nameprojectimage*>_40.png




*Image files in three different sizes for zoom levels*

3. For users of GENIUS TOOLS Startup TOOLS: Create a directory with the name *img_background* in the *GENIUS TOOLS* resource folder of caddepot of the installation computer under <*nameofworking environment*>*parametric**configuration*\<*GT_RESOURCE_FOLDER*>*quick_access*.

**Please note:** Files outside the *GENIUS TOOLS* resource folder *(GT_RESOURCE_FOLDER)* are renewed at each update, e. g. files under <*operatingenvironment*>*apps**gtfc**text**resource*.


© 2026 INNEO Solutions GmbH

4. Specify the path to the image file in the configuration option `gtqa_background_picture` (for example, by using *Configuration Utility*[765]).
   You can use variables for the file or specify the following paths:
   *<GT_RESOURCE_FOLDER>\quick_access\img_background\<nameprojectpicture>*
   *<GTfC_install>\text\resource\<nameprojectimage>*.
   When using these paths, the project image is searched for a size-dependent version:
   - At zoom level 1.4 - 1.8, the medium sized file (*<image name>_30.png*) is used.
   - For zoom level greater > 1.9, the large file (*<image name>_40.png*) is used.

## 19.2.5 Adapting zoom levels and icons

*Quick Access* supports multiple zoom levels to improve usability and readability on high-resolution monitors. For icons to be clearly visible you can upload images of different sizes (resolution) in the subfolders of the *Quick Access* resources directory.

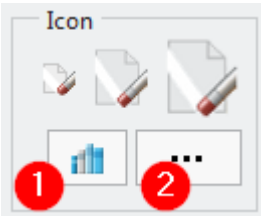
| Zoom level | Size of images | Subdirectory |
|---|---|---|
| 1 - 1.3 | 20x20 Pixel | *img_w20* |
| 1.4 - 1.8 | 30x30 Pixel | *img_w30* |
| From 1.9 | 40x40 Pixel | *img_w40* |

For best user experience the zoom levels 1.0, 1.6 and 2.1 are recommended.

# Icons

Icons are used from the directories *img_w20*, *img_w30* and *img_w40* depending on the defined zoom level.

**Assigning an icon to a command**
Load icon images in the Icon area of the command details[553] using the *Load icon from Quick Access library* button (1) for images stored in the resource directory. Use the right button *Import icon from Quick Access Library* (2) to insert external icons.




*Icons should at least have 20x20 pixels*


© 2026 INNEO Solutions GmbH

**External icons**
Once an external source has been selected with *Quick Access Editor*, the image file will be automatically added to the *Quick Access* library.

Make sure that the size of external icons is 40x40 pixels. The other sizes are downsized for the proper dimensions and copied into the images directories (*img_w20-40)*. Small icons are not imagestyleclass="Default" scaled up.

**Creating icons**
If you wish to copy icons into the images directories manually, make sure that identical icons with different resolutions always have the same name.

When creating your own icons pay attention to some specific characteristics:

– Export icons as PNG files. Check the export settings and avoid saving color values of transparent pixels.

– Take care of good recognition of the individual icons in different sizes. Modify the image composition if necessary.

– Pay attention to a strong foreground-background contrast.

## Define zoom levels

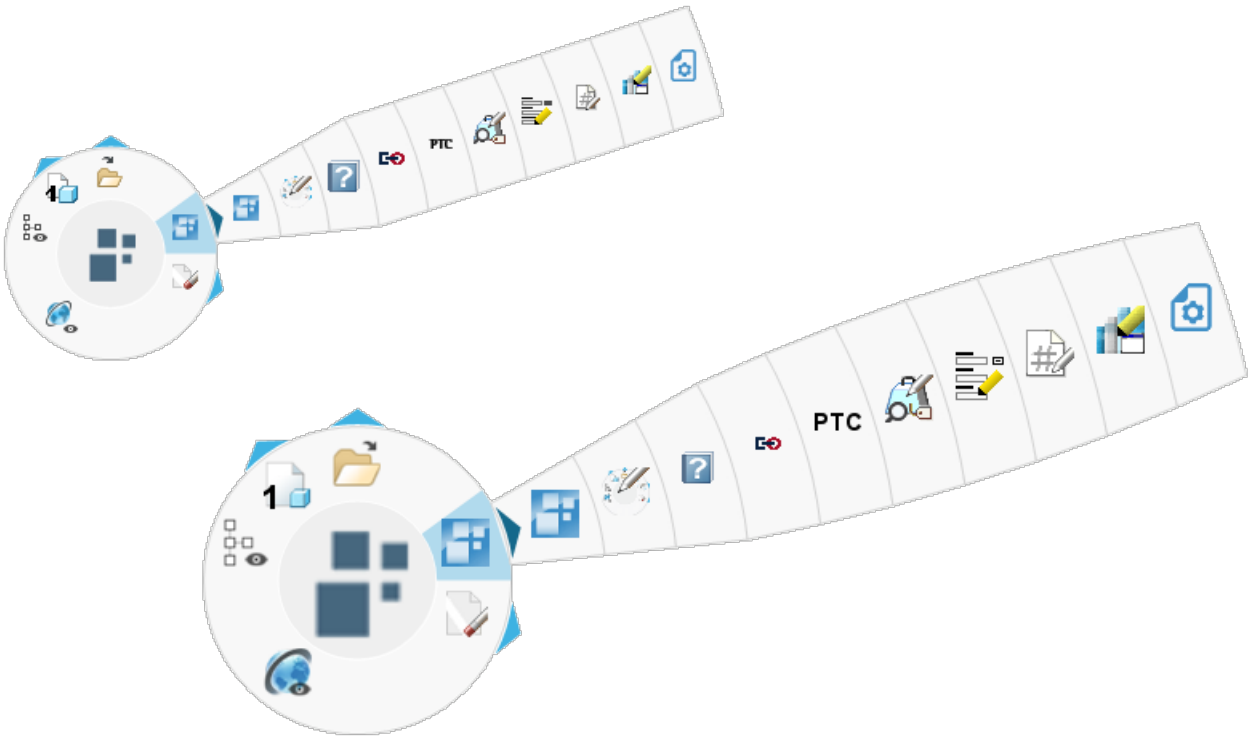
Use the following configuration options.

**gtqa_zoom**
Set the zoom level in steps of 0.1.

**gtqa_command_icon_folder / gtqa_local_command_icon_folder**
Make sure the *img_w20* directories are specified.

**gtqa_group_bow**
Use this option to display an arc in groups. This enlarges the display area for icons.


© 2026 INNEO Solutions GmbH




*Enlarged display area for icons (below)*

## 19.2.6 Use cases

### Recording a mapkey

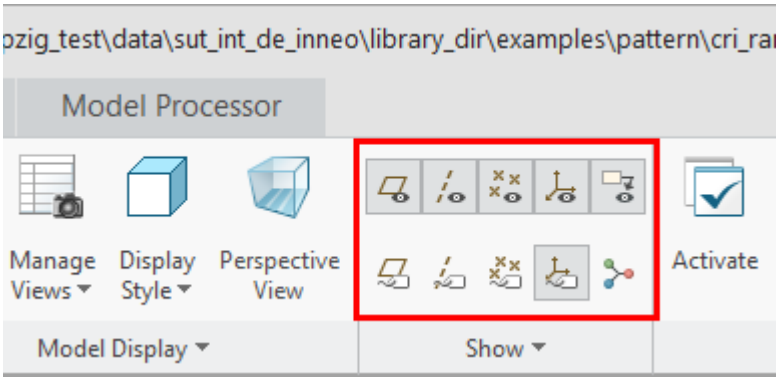
This example shows how to create a mapkey and set it up in *Quick Access*. The mapkey to be created shows and hides reference planes.

1. Open *Quick Access* by pressing the [<] key.
2. Select the *Record Mapkey* command, then click *Record* in the record dialog.

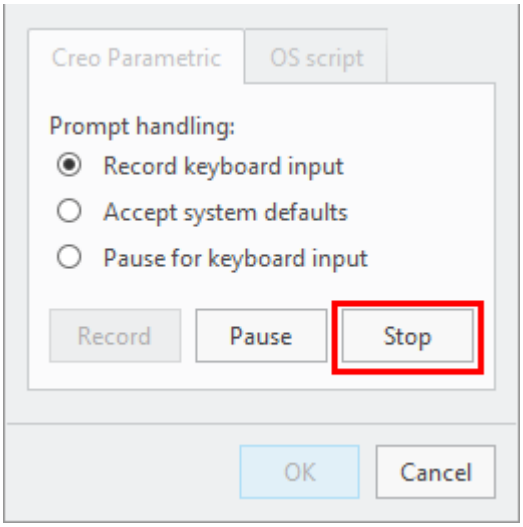



3. Activate or deactivate all datum display filters in the *Show* tab.


© 2026 INNEO Solutions GmbH




4. Click *Stop* (Creo 3) or *Pause* (Creo 4 and later) in the record dialog, then click *OK* to end the recording.

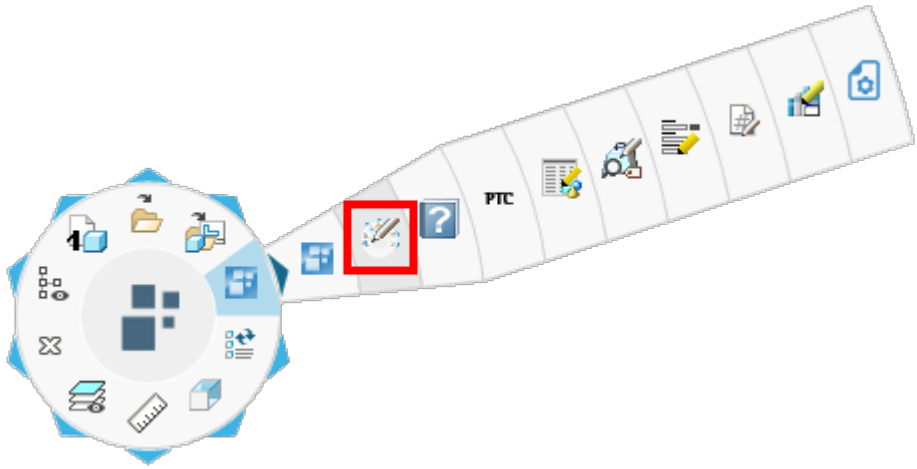



5. Open *Quick Access* again by clicking the [<] key.
6. Select the *Save changed* command to save the mapkey.
7. Open this file with an editor, e.g. Notepad.

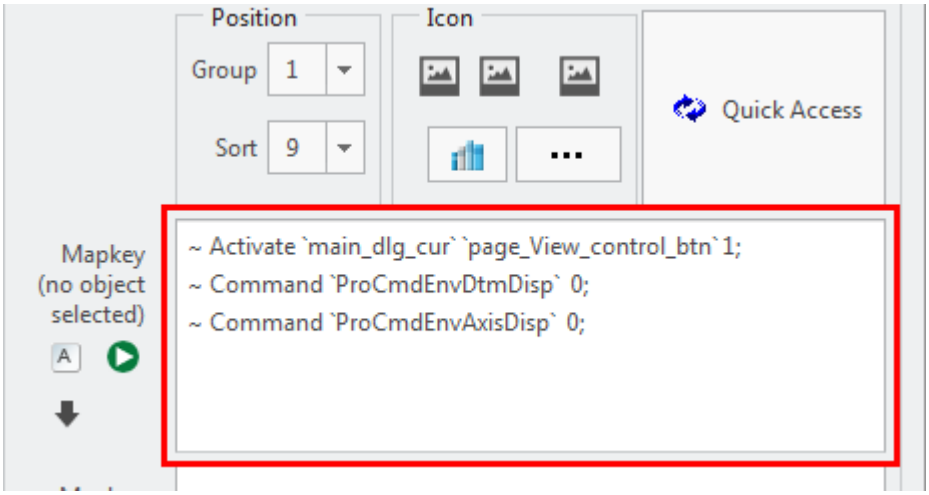


8. Copy the entire text from the editor (Ctrl+C).
9. Open *Quick Access Editor*.


© 2026 INNEO Solutions GmbH



10. Select an empty position in the grid or an existing command you wish to overwrite and insert the copied mapkey into the upper right field (Ctrl+V).




11. Shorten the mapkey and test it, then copy it into the lower input field to use when an object is selected in Creo.

12. Close the editor. *Quick Access* is automatically updated when closed.

# 20 Stack Note

Available in the ribbon menu GENIUS TOOLS

With *Stack Note*, you can create, manage and place note tables. These notes can be filled from a multilingual database. For instance, these notes refer to standards or regulations. Notes can be exported as XML files.

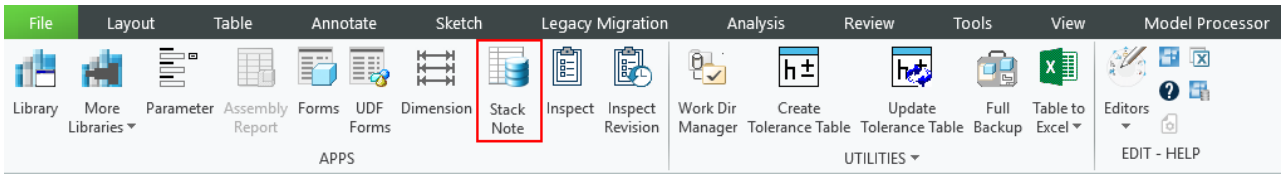
**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## 20.1 Usage

In this section you will find information on the use of *Stack Note*.

### Starting the program: in drawing mode

To start *Stack Note*, click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section.

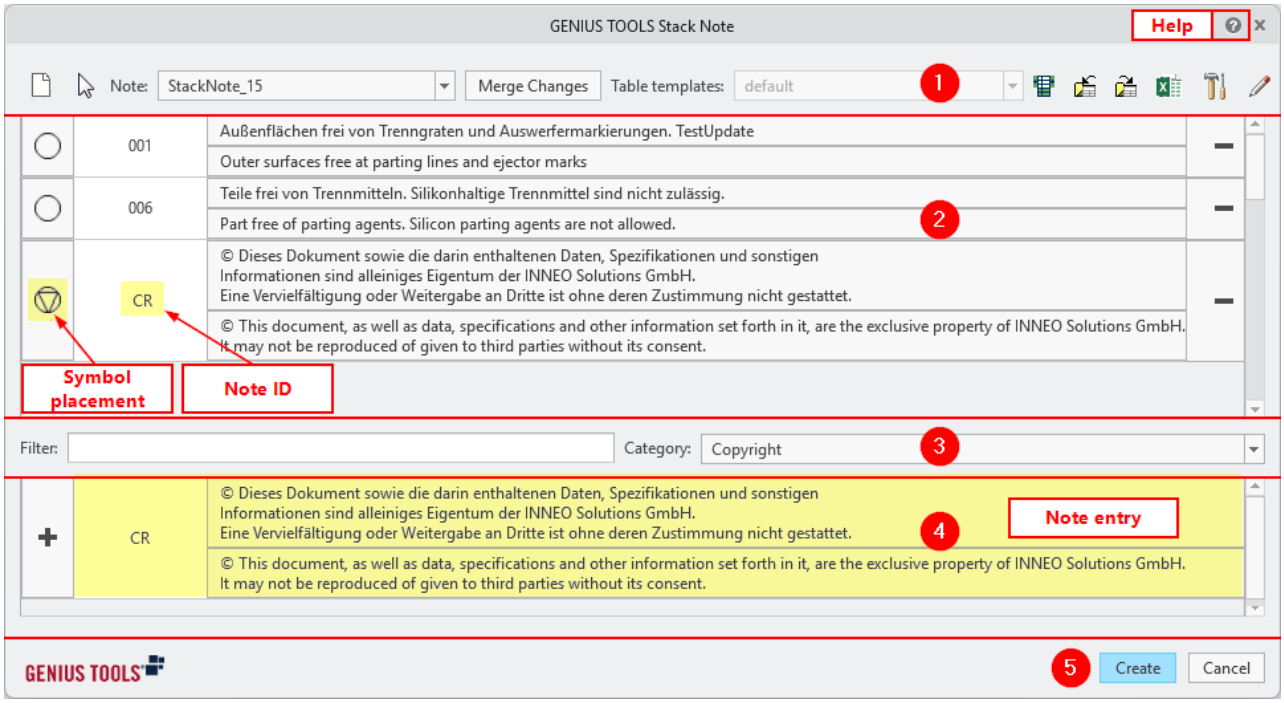



*Starting via the GENIUS TOOLS in the ribbon menu*

### 20.1.1 User Interface

The *Stack Note* user interface consists of the following elements:


© 2026 INNEO Solutions GmbH




1. Command bar[569] and help button
2. List of all note entries[570]
   A note entry consists of the button for symbol placement[578], a note ID, and the content in one or more languages[572].
3. Search input and category selection[571]
4. Selection of note entries[578]
5. Buttons for placing and updating the note and closing the window.

## 20.1.2 Command bar

| Symbol | Name | Description |
|---|---|---|
|  | Create new note | Create new, empty note |
|  | Select Note | Select placed note for editing |
|   | Note | Drop-down menu with placed notes |
|   | Merge database changes | Manage changes to note entries[575] by copying existing or modified note entries into the note. |

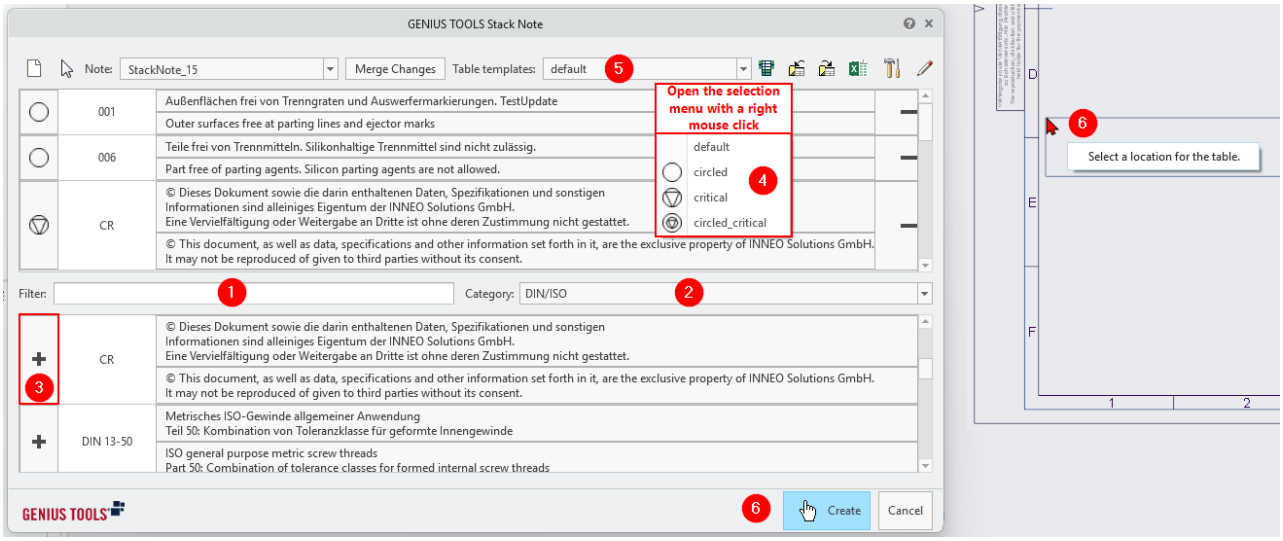

© 2026 INNEO Solutions GmbH

| Symbol | Name | Description |
|---|---|---|
|  | Table templates | Drop-down menu with available table templates (with a predefined header and a value row that specifies the formatting). Table templates must consist of 3 rows and 2 columns. |
|  | Split Table | Split an existing note table into two, see split note table[576]. |
|  | Import note from XML | Import an existing note |
|  | Export note to XML | Export current note as XML file |
|  | Export note to Excel table | Export current note as Excel file |
|  | Setting languages[572] | Set languages for a note |
|  | Editor[572] | Opens the editor to automatically insert a note. |

## 20.1.3 Creating notes

A note consists of one or more note entries that are managed by the administrator in the note database[578] and selected by the user. The selected note entries are placed on a drawing as a note in table form. The table is filled based on a table template in a TBL file stored in the resource folder.


© 2026 INNEO Solutions GmbH




## Searching for a note entry

1. Using the search input *Filter*, you can search either by note ID or by the content of the note entry. RegEx[487] can be used for this purpose. If two RegEx expressions are separated by a space, the filter interprets this as an 'OR' operator.
2. Using the drop-down menu *Category*, you can filter note entries by category.

## Adding a note entry

3. Click the ✚ icon to add a note entry to the note.

## Assigning a symbol

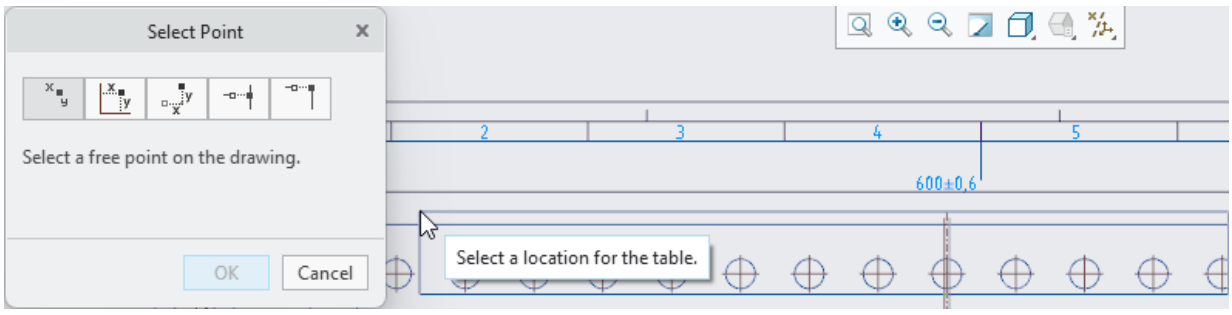
4. Right-click on the note entry to select the symbol from the drop-down list.

**Please note:** The colors of the symbols can be stored in the symbol database[580], e. g. red for critical symbols.

5. Select a suitable table template under *Table templates*.

## Placing a note

6. Confirm with *Create* and place the note in the drawing.

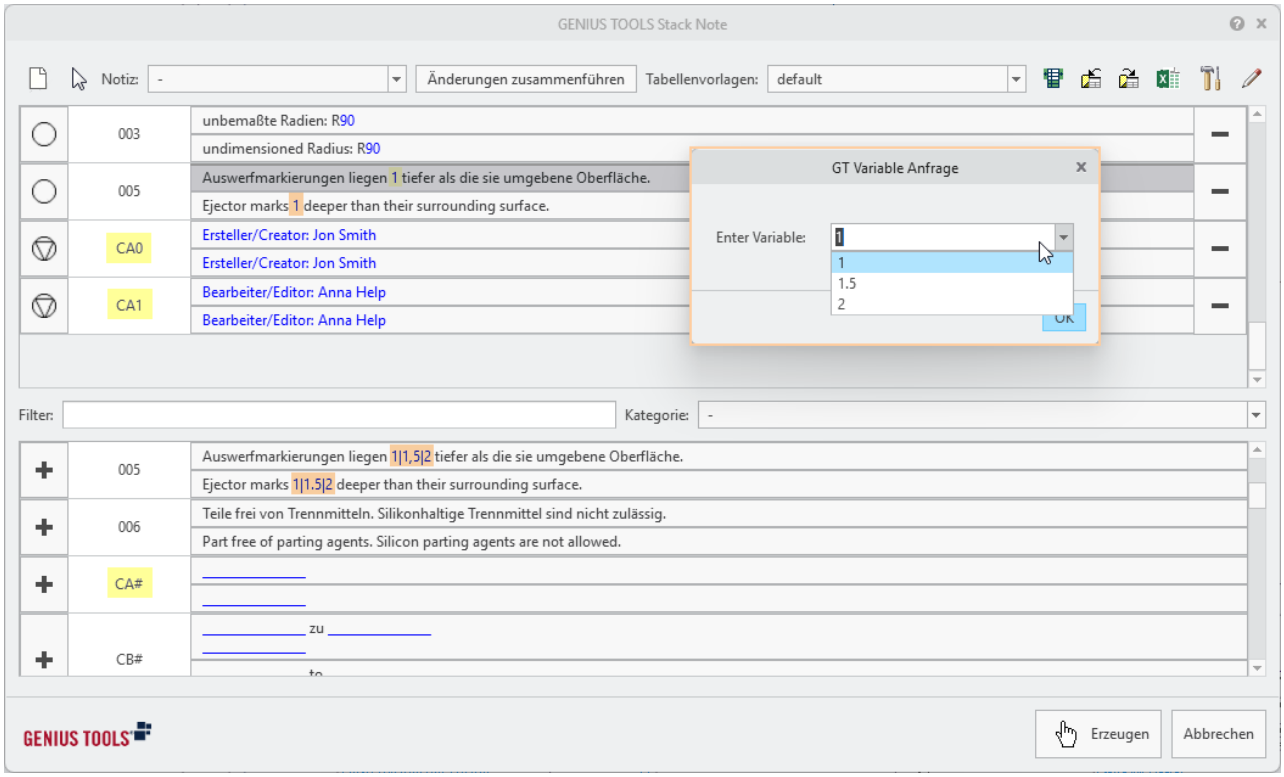



**Please note:** In the window *Select Point*, select the option *Select a free point on the drawing*. Otherwise, placement is not possible.


© 2026 INNEO Solutions GmbH

# Note entries with user input

Individual note entries can contain variables and placeholders to write user-defined texts.




*Note entries with example variables and placeholders*

### Variables

Once you have added a note entry, you can populate a variable. To do this, click on the blue formatting. The entered variable is displayed in blue. If there are multiple variables for a note in the database, a drop-down menu will appear, see Managing note entries 580.

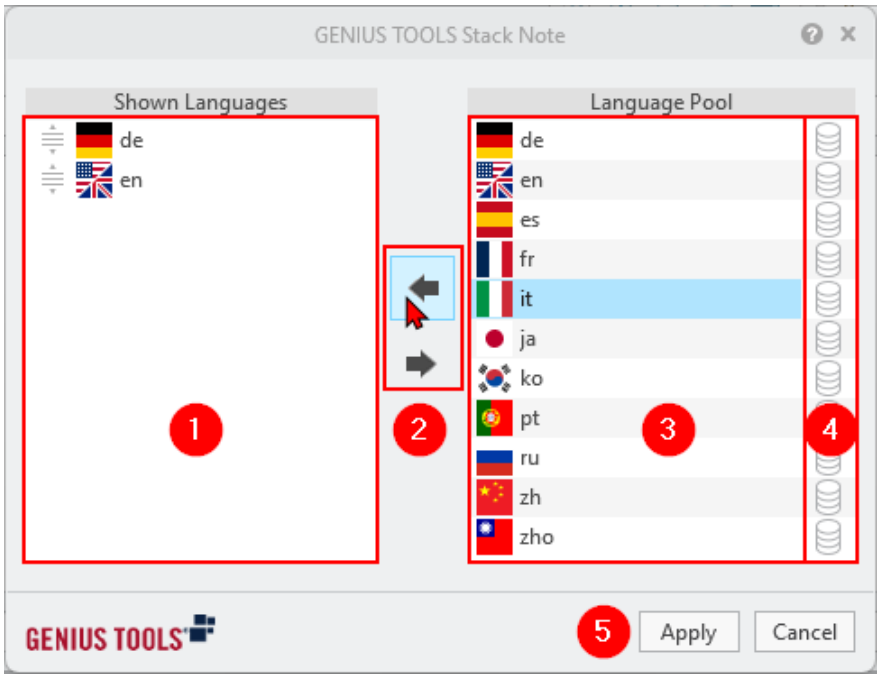
### Placeholders

A hashtag # is a placeholder for a number in the note ID and is numbered consecutively in the note table. For example, CA#, CA#, CA# changes to CA0, CA1, CA2.

# Setting languages

The comments can be displayed in multiple languages in the table. Open the tools menu .


© 2026 INNEO Solutions GmbH




1. Languages displayed in the note, which can be rearranged using drag and drop.
2. Icon for adding or removing a language

**Please note:** You can also drag and drop languages onto the *Shown Languages* (1) field and remove them by dragging the language to the *Language Pool* (3) field.

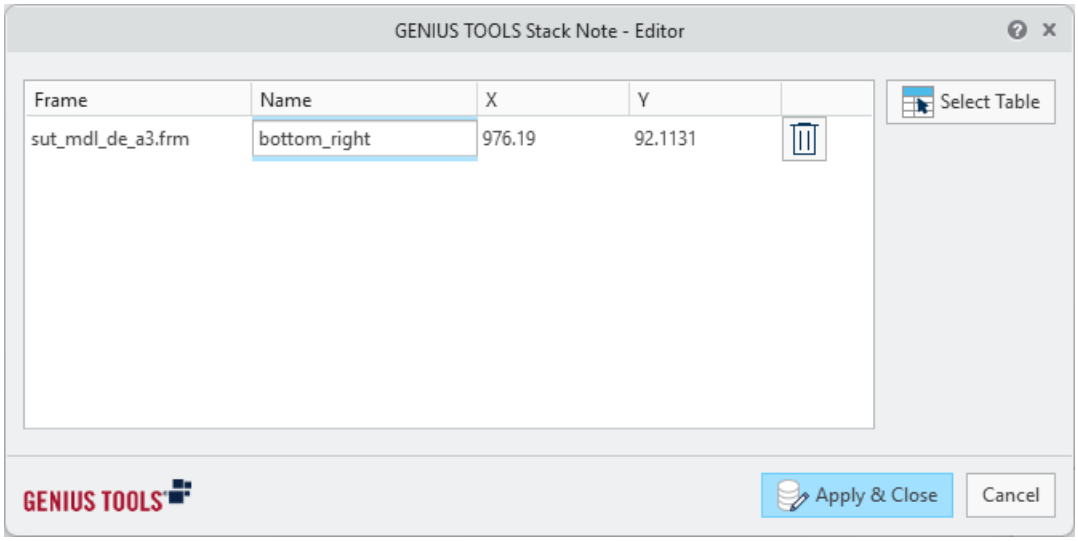
3. Language selection
4. Icon indicates that the language is defined in the database.
5. Icon indicates that the language is defined in the embedded data.
6. Click *Apply* to confirm the selected languages, or click *Cancel* to close the window without saving.

## 20.1.4 Placing notes automatically

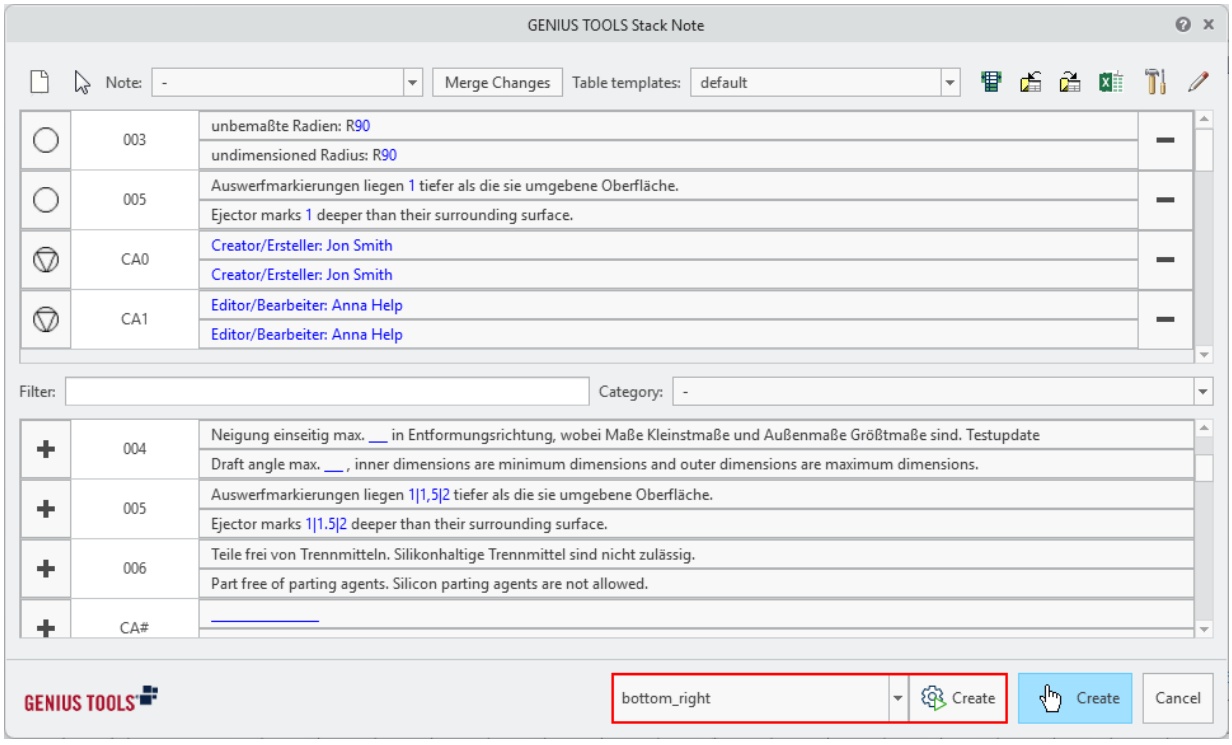
Use the Editor to select sample tables that have been placed in order to read their positions and reuse them.

1. Click on the pencil icon in the toolbar to open the Editor.
2. Click *Select Table*.
3. Click on a table with the left mouse button to read the value.
4. Under the category *Name*, enter a placement name, e.g. bottom_right.


© 2026 INNEO Solutions GmbH



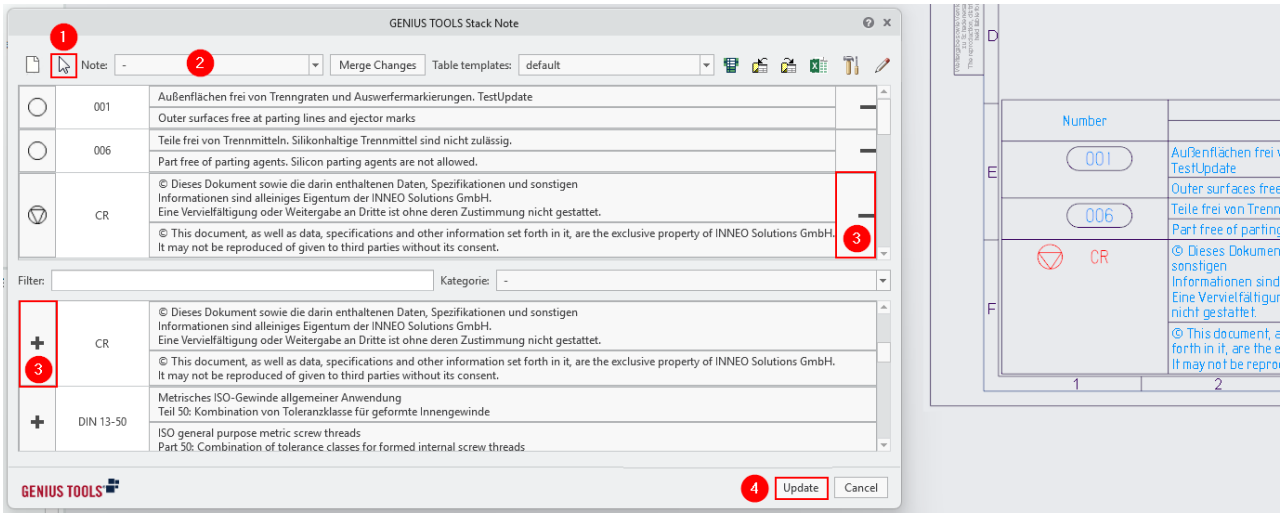

5. Click *Apply & Close* to save the placements to the corresponding database.
6. Select the note from the drop-down menu at the bottom.
7. Click *Create* to place the selected note automatically.




## 20.1.5 Updating notes

A note that has already been placed can be updated. You can add or remove note entries. If you have made changes to the database and want to change them in the note, you can find more information in the following chapter Applying changes from the database 575.


© 2026 INNEO Solutions GmbH



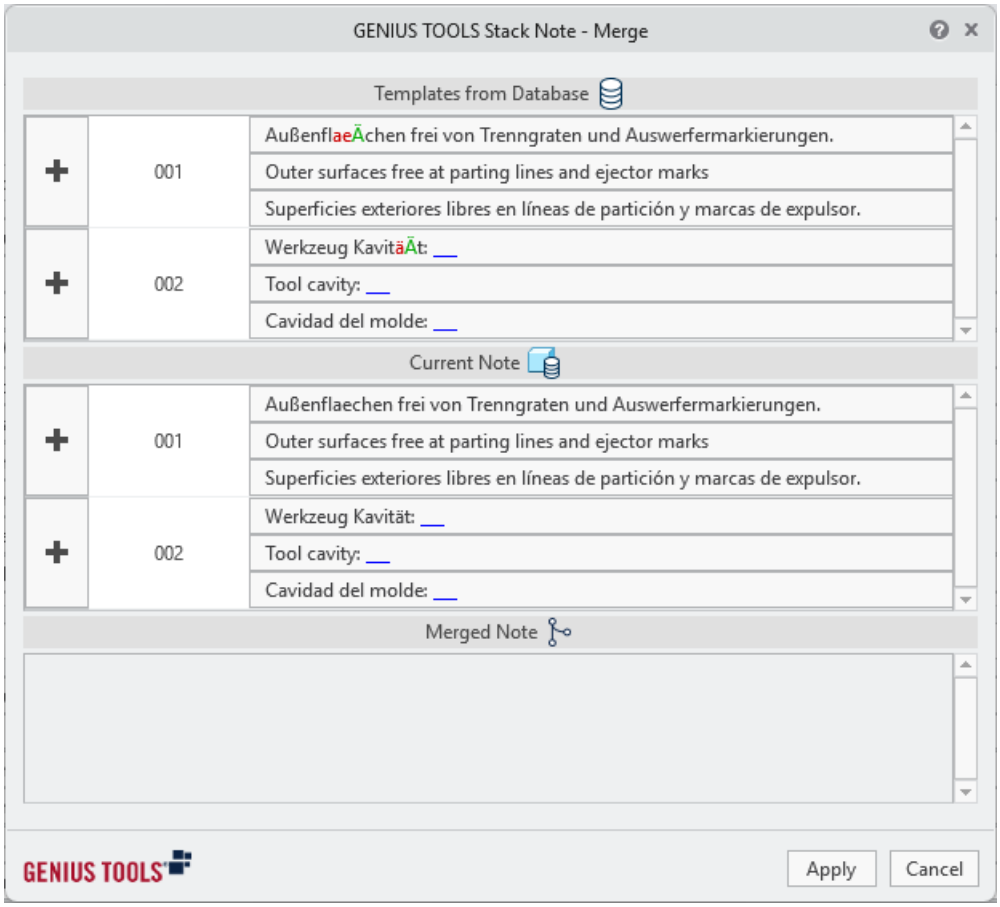

1. Click *Select Note* in the command bar and click on a note that has already been placed or
2. select a note from the drop-down menu *Note*.
3. Note entries can be removed using the adjacent ➖ icon or add a note entry using the ➕ icon.
4. Update the existing note with *Update*.

## 20.1.5.1 Applying changes from the database

Changes to note entries are made in the *Stack Note* database. Users can apply the changes from the database.

1. Open a note that has already been placed in *Stack Note*.
2. Edit the database as described in the following chapter Managing note entries 578.
3. Go back to *Stack Note* and click *Merge Changes*. You see a comparison of the different entries in the database and the current note. Changes from the database are highlighted in color.


© 2026 INNEO Solutions GmbH



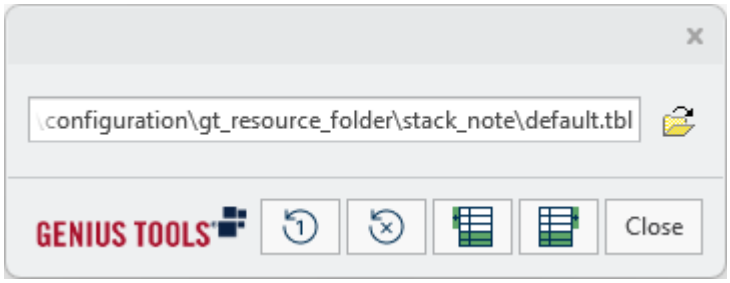

4. For each note entry, select whether to apply changes from the database or to keep the current note.
   - Database: Under *Templates from Database*, click the ✚ icon next to the note entry if you want to apply the change from the database to the current note.
   - Current note: Under *Current note*, click the ✚ icon next to the note entry if you want to transfer the note entry from the current note.

   Under *Merge Changes*, you see a preview of the note entries to be imported.
5. Confirm the selection with *Apply*.
6. Click *Update* to update the placed note.

## 20.1.6 Splitting a note table

The *Split Table* button can be used to split one note into two note tables, see also Split Table[747].


© 2026 INNEO Solutions GmbH




1. Select the table template by clicking on the folder icon .
2. Click either *Split table to left side* or *Split table to right side* .
3. Click in the note entry where you want to split the note table. Based on the click coordinates, all table cells belonging to the note entry are selected as well as all note entries below (framed in red) and written in a new table.

| Number | Zusätzliche Informationen<br>Additional Information |
|---|---|
| 001 | Außenflächen frei von Trenngraten und Auswerfermarkierungen.<br>TestUpdate |
| | Outer surfaces free at parting lines and ejector marks |
| 006 | Teile frei von Trennmitteln. Silikonhaltige Trennmittel sind nicht zulässig. |
| | Part free of parting agents. Silicon parting agents are not allowed. |
| CR | © Dieses Dokument sowie die darin enthaltenen Daten, Spezifikationen und sonstigen<br>Informationen sind alleiniges Eigentum der INNEO Solutions GmbH.<br>Eine Vervielfältigung oder Weitergabe an Dritte ist ohne deren Zustimmung nicht gestattet. |
| | © This document, as well as data, specifications and other information set forth in it, are the exclusive property of INNEO Solutions GmbH.<br>It may not be reproduced of given to third parties without its consent. |

*Selected cell as a divider for the table*

4. The table splits immediately after you click a cell:

| Number | Zusätzliche Informationen<br>Additional Information |
|---|---|
| 001 | Außenflächen frei von Trenngraten und Auswerfermarkierungen.<br>TestUpdate |
| | Outer surfaces free at parting lines and ejector marks |

| Number | Zusätzliche Informationen<br>Additional Information |
|---|---|
| 006 | Teile frei von Trennmitteln. Silikonhaltige Trennmittel sind nicht zulässig. |
| | Part free of parting agents. Silicon parting agents are not allowed. |
| CR | © Dieses Dokument sowie die darin enthaltenen Daten, Spezifikationen und sonstigen<br>Informationen sind alleiniges Eigentum der INNEO Solutions GmbH.<br>Eine Vervielfältigung oder Weitergabe an Dritte ist ohne deren Zustimmung nicht gestattet. |
| | © This document, as well as data, specifications and other information set forth in it, are the exclusive property of INNEO Solutions GmbH.<br>It may not be reproduced of given to third parties without its consent. |

*Results of the table split*

5. When you add new note entries and click *Update*, they will be added to the most recently created note table at the bottom.

| Number | Zusätzliche Informationen<br>Additional Information |
|---|---|
| 001 | Außenflächen frei von Trenngraten und Auswerfermarkierungen.<br>TestUpdate |
| | Outer surfaces free at parting lines and ejector marks |

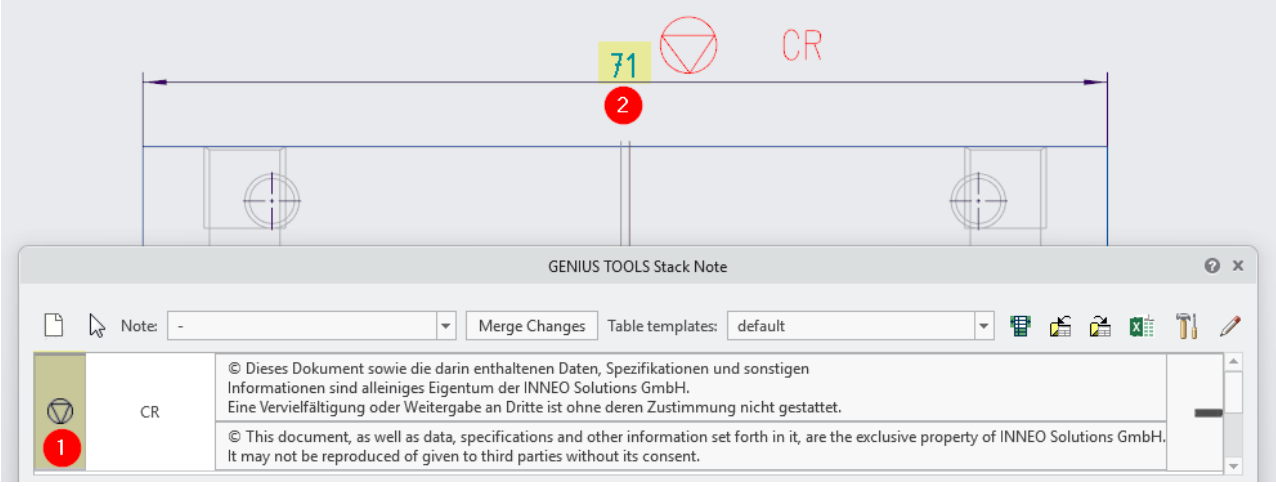
| Number | Zusätzliche Informationen<br>Additional Information |
|---|---|
| 006 | Teile frei von Trennmitteln. Silikonhaltige Trennmittel sind nicht zulässig. |
| | Part free of parting agents. Silicon parting agents are not allowed. |
| CR | © Dieses Dokument sowie die darin enthaltenen Daten, Spezifikationen und sonstigen<br>Informationen sind alleiniges Eigentum der INNEO Solutions GmbH.<br>Eine Vervielfältigung oder Weitergabe an Dritte ist ohne deren Zustimmung nicht gestattet. |
| | © This document, as well as data, specifications and other information set forth in it, are the exclusive property of INNEO Solutions GmbH.<br>It may not be reproduced of given to third parties without its consent. |
| T9 | Zylinderblock 2000 - Parameter |
| | Zylinderblock 2000 - Parameter |


© 2026 INNEO Solutions GmbH

6. Splits can be undone one at a time or all at once .

### 20.1.7 Placing symbols

Use the button for symbol placement (1) to link a symbol to a note, dimension, or symbol in the drawing.




1. Click the button for placing symbols next to a note entry. When the button is activated, it is highlighted in gray. Creo prompts you to select a link destination.
2. Click on a note, dimension or symbol.
3. You can place the symbol using the middle mouse button.

The symbol with the note ID is placed next to the selected element (2).

**Please note:** Symbols can only be placed once a note table has been placed.

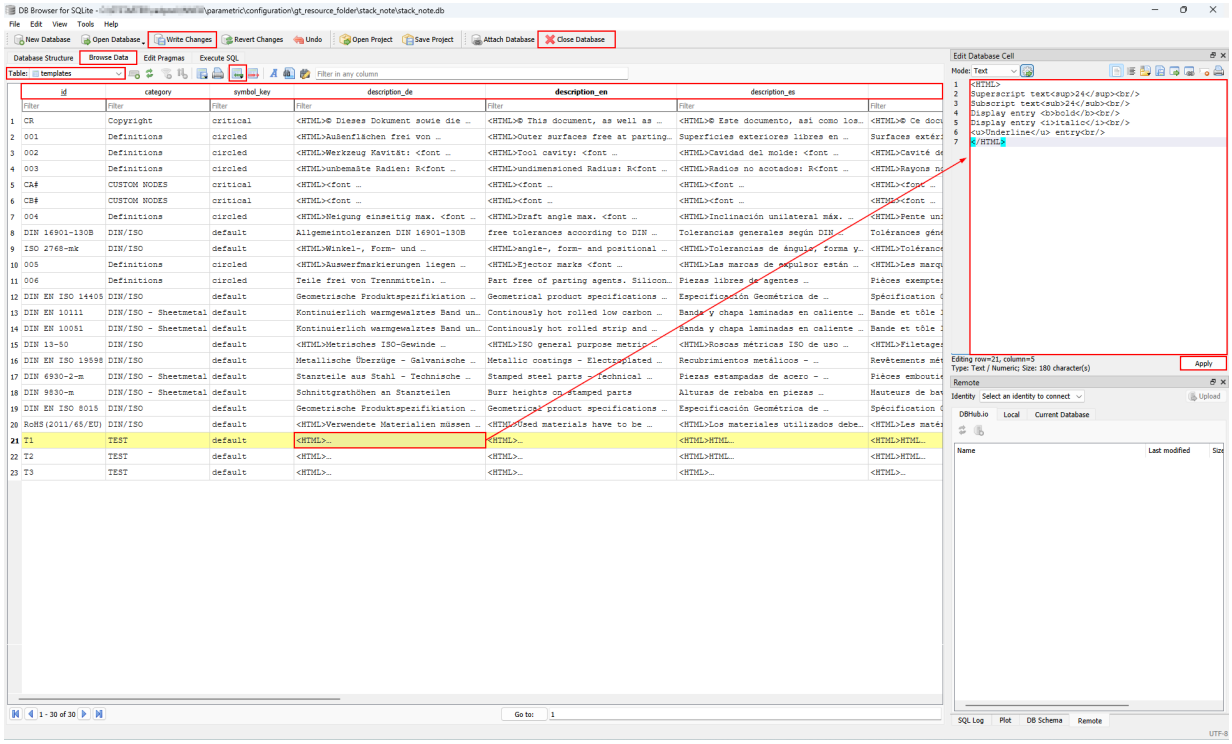
## 20.2 Configuration

This section provides information on configuring the databases.

### 20.2.1 Managing note entries

Available note entries are stored in the database *stack_note.db*. To edit the database, close GENIUS TOOLS Stack Note.

1. Open the SQLite database in the resource folder using a database editor, e. g. SQLite Browser:
   *<Operating environment>\parametric\configuration\gt_resource_folder\stack_note\stack_note.db*
2. Click the *Browse Data* tab in the upper left corner. Under *Table*, select *templates*:



*Database with an example of a formatted cell*

Each row contains a note entry consisting of the following columns:

- *id*: Note IDs. A note ID may contain numbers, letters, and other characters. They are assigned once and sorted alphanumerically.
- *category*: Category to which the notes are assigned.
- *symbol_key*: ID of the symbol defined in the symbol database.
- *description_<language abbreviation>*: Notes in the respective language

3. To edit a note, proceed as follows:

- To add a new note, click *Add New Row*  . A new, empty row will be added at the bottom of the table, which you can fill in.
- Click on a cell whose content you want to add or change. In the section *Edit Database Cell* on the right side, you can edit the cell's content.
- HTML tags are supported for editing note entries under *description_<language abbreviation>*. If you want to format the text, it must be embedded in HTML tags.
  `<HTML>`*Enter text here.*`</HTML>`
  Insert additional formatting within these HTML tags:
  - Superscript text: `<sup>`*Enter text*`</sup>`
  - Subscript text: `<sub>`*Enter text*`</sub>`
  - Bold: `<b>`*Enter text*`</b>`
  - Underline: `<u>`*Enter text*`</u>`


© 2026 INNEO Solutions GmbH

- Italic: *Enter text*
- Line break: <br/>
- Defining variables: <font color=#0303FF>___</font>

**Please note:** To ensure that the entries are read correctly, you must fill in all columns—including all available languages—for each note entry that uses variables.

**Tip:** To define a variable, you need the opening tag <font color=#0303FF> and the closing tag </font>. The variable format is defined here using underscores: ___. This is just one example of how you can format the variables. You can format the variables however you like, e. g., you can also use dots or the word 'Variable'.

- Defining variables with a drop-down menu: <font color=#0303FF>A|B|C</font>
  You can enter as many options as you like.
  After adding a note entry to a table, a drop-down menu containing all the specified options appears in the user interface. You can also adjust the variable manually. Manual entries are added to the menu.

4. After editing the contents of a cell, click *Apply* below the edit field. Repeat this step for each cell you edit.
5. When you are done editing the database, click *Write Changes* at the top.
6. Close the database.
7. Open Stack Note. The changes in the database are now available.

| | | |
|---|---|---|
| | T11 | Angabe hochstellen$^{24}$<br>Angabe tiefstellen$_{42}$<br>Angabe **fett** darstellen<br>Angabe *kursiv* darstellen<br>Angabe unterstreichen darstellen |
| | | Superscript entry$^{24}$<br>Subscript entry$_{42}$<br>Display entry **bold**<br>Display entry *italic*<br>Display entry *underlined* |
| | T12 | Eine Variable definieren___ |
| | | Defining a variable___ |
| | T13 | Auswahlmenü 0,5 |
| | | Selection menu 0,5 |

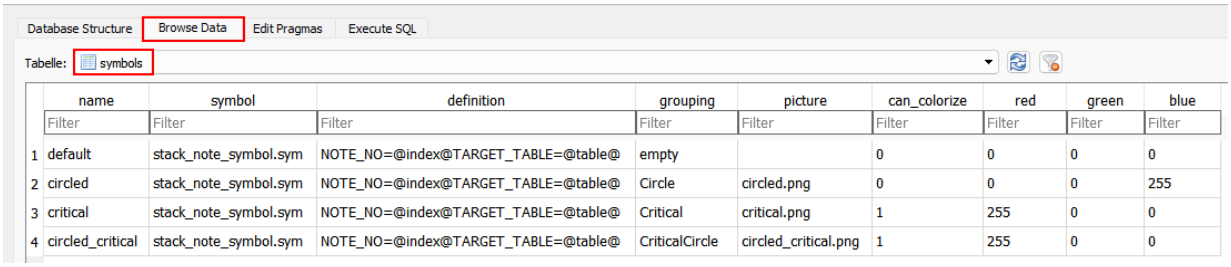
*Result in the user interface of GENIUS TOOLS Stack Note*

## 20.2.2 Managing symbols

The available symbols are managed in the database *stack_note_symbol.db*. To edit the database, close GENIUS TOOLS Stack Note.


© 2026 INNEO Solutions GmbH

1. Open the SQLite database in the resource folder:
   *<Operating environment>\parametric\configuration\gt_resource_folder\stack_note\stack_note_symbol.db*
2. Click the *Browse Data* tab in the upper left corner. Under *Table*, select *symbols*:




| | name | symbol | definition | grouping | picture | can_colorize | red | green | blue |
|---|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | default | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | empty | | 0 | 0 | 0 | 0 |
| 2 | circled | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | Circle | circled.png | 0 | 0 | 0 | 255 |
| 3 | critical | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | Critical | critical.png | 1 | 255 | 0 | 0 |
| 4 | circled_critical | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | CriticalCircle | circled_critical.png | 1 | 255 | 0 | 0 |

Each row contains a symbol definition consisting of the following columns:

- *name:* Name of the symbol specified as *symbol_key* in the table database.
- *symbol*: Symbol file in the resource folder.
- *definition:* The variable text can be found in Creo under *Variable Text*.

  **Please note:** *NOTE_NO=@index@* and *TARGET_TABLE=@table@* must be included in this column. These entries must be separated by a line break.

- *grouping:* Selection for grouping.
- *picture:* File for preview image of the icon.
- *can_colorize:* Option for the coloring of the symbol (0 - No, 1 - Yes). If you set this option to 1, use the following columns to specify the color with which to fill the image:
  - *red:* RGB value for red
  - *green:* RGB value for green
  - *blue:* RGB value for blue

For detailed instructions on how to edit a database, see Managing note entries[578].


© 2026 INNEO Solutions GmbH

# 21 Stack Note 3D

Available in the ribbon menu GT MBD

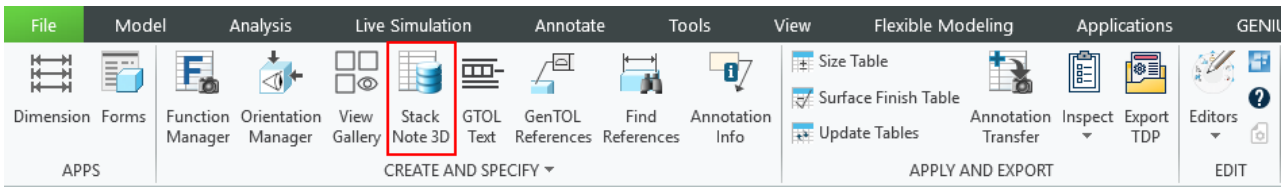
With *Stack Note 3D*, you can generate and manage note tables and place them in 3D. The notes can be filled from a multilingual database. These notes contain, for example, information related to standards or regulations. The note can also be exported as an XML file.

## 21.1 Usage

In this section you will find information on the use of *Stack Note 3D*.

### Starting the program: in part mode

To start *Stack Note 3D*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.

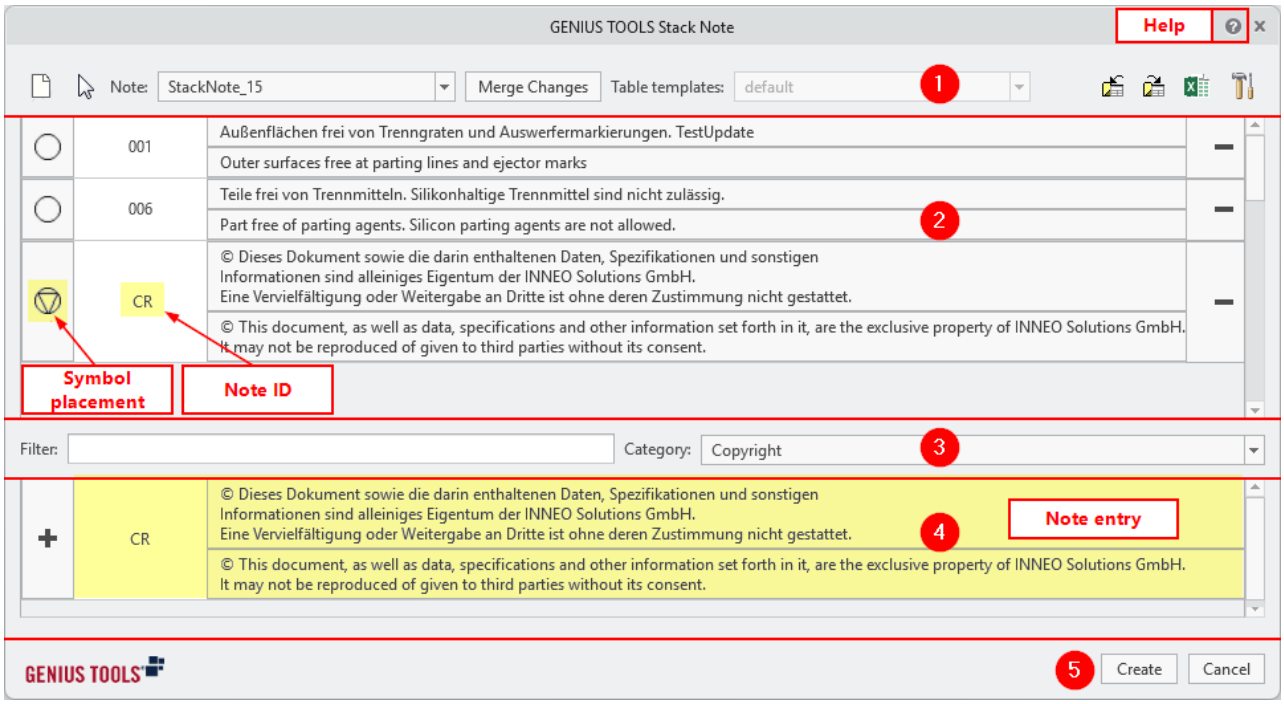



*Starting via the GT MBD in the ribbon menu*

### 21.1.1 User Interface

The *Stack Note 3D* user interface consists of the following elements:


© 2026 INNEO Solutions GmbH

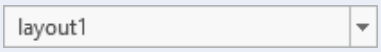



1. Command bar[583] and help button
2. Selection of note entries[590]
3. Search input and category selection[585]
4. List of all note entries[584]
   A note entry consists of the button for symbol placement[589], a note ID, and the content in one or more languages[586].
5. Buttons for placing and updating the note and for closing the window.

## 21.1.2 Command bar

| Symbol | Name | Description |
|---|---|---|
| | Help | Opens the help. |
| | Create new note | Create new note table |
| | Select Note | Select placed note table for editing |
| StackNote_1 | Selection of all placed notes | Drop-down menu with the created note tables |


© 2026 INNEO Solutions GmbH

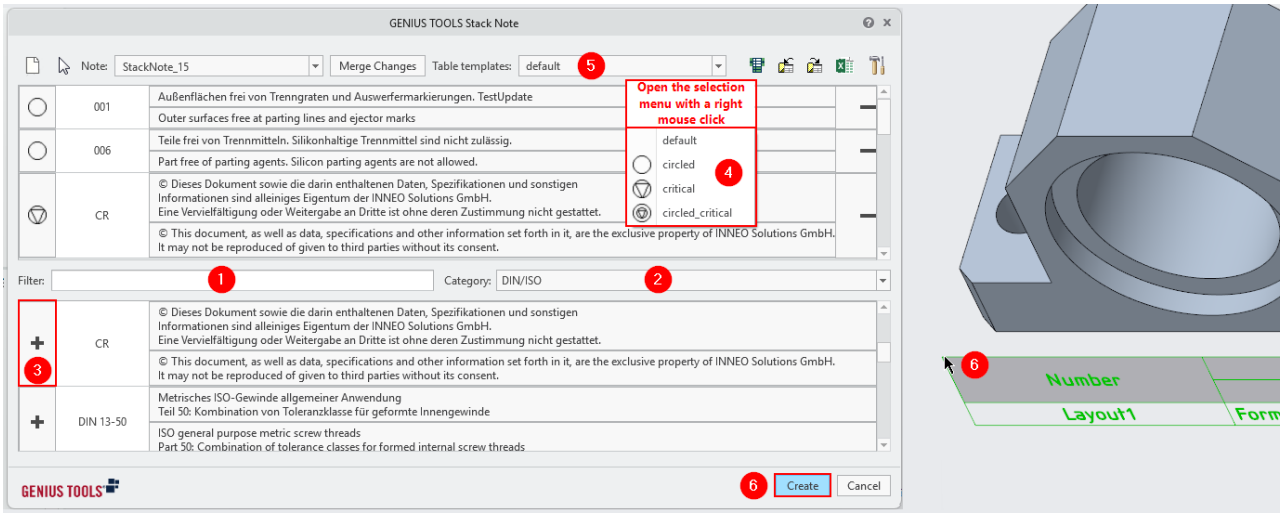
| Symbol | Name | Description |
|---|---|---|
|   | Merge database changes | Manage changes[588] to notes by transferring existing or modified notes to the notes table. |
|   | Table templates | Drop-down menu with templates for the note tables |
|  | Import note from xml | Import selected note table as an XML file |
|  | Exportnote to xml | Export selected note table as an XML file |
|  | Export note to Excel table | Export note table as an Excel file |
|  | Tools menu[586] | Set languages for note table |

## 21.1.3 Create note

A note consists of one or more note entries that are managed by the administrator in the note database[590] and selected by the user. The selected note entries are placed in 3D as a note in table form. The table is filled using a table template in a TBL file stored in the resource folder.


© 2026 INNEO Solutions GmbH

### Procedure




## Search for previous note entry[585]

1. Search for a note entry using the search field next to *Filter* or
2. in the category selection menu.

### Add note entry

3. Click the ✚ icon to add a note entry to the note.

### Assign symbol

4. Right-click on the note entry to select the symbol from the selection list.

**Please note:** The colors of the symbols can be stored in the symbol database[592], e.g., red for critical symbols.

5. Select the suitable table template under *Table templates*.

### Place note

6. Confirm with *Create* and place the note in the drawing.

# 21.1.3.1 Search note entry

You can search for note entries using the search input and category selection.

### Filter

You can use the search input to search for either the note ID or the content of the note entry. RegEx[487] can be used for this purpose. If two RegEx expressions are separated by a space, the filter interprets this as an OR link.

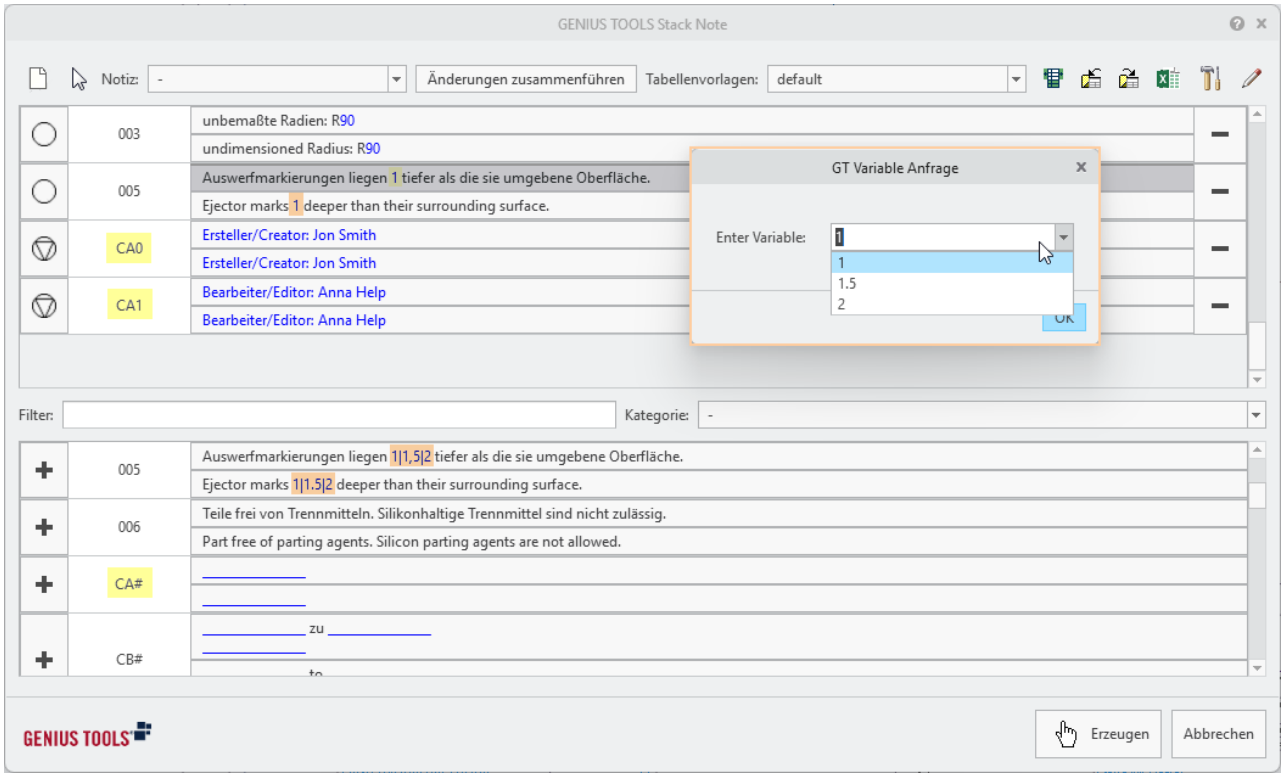
### Category

You can use this selection menu to filter note entries by category.


© 2026 INNEO Solutions GmbH

## 21.1.3.2 Note entries with user input

Individual note entries can contain variables and placeholders to write user-defined texts.



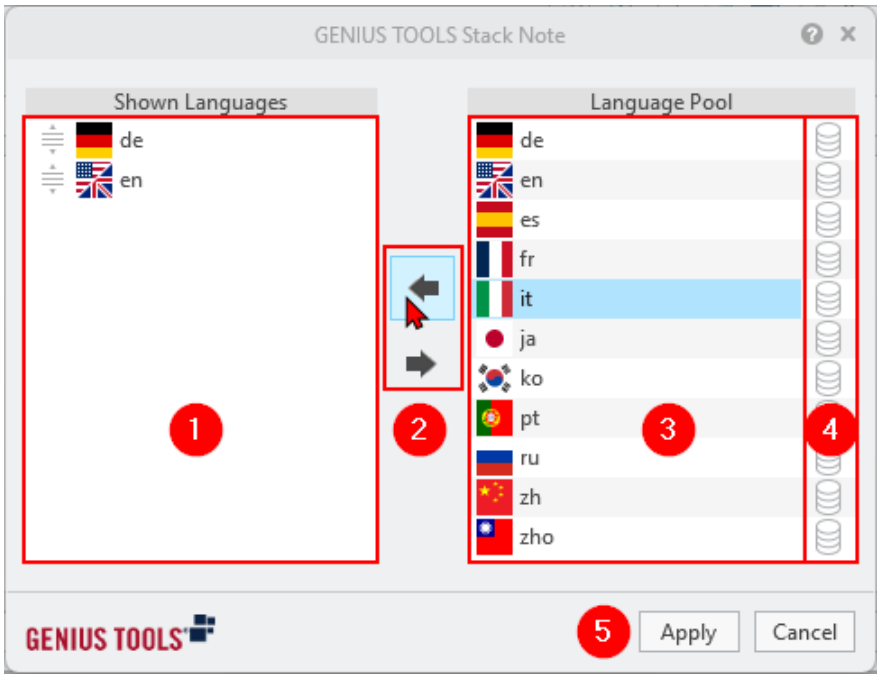

**Variables**
By clicking on a blue underscore, you can enter a variable in the upper section. The variable you enter will be displayed in blue font. If there are multiple variables for a note entry in the database, a drop-down menu will appear.

**Placeholders**
A # is a placeholder for a number in the note ID and is numbered consecutively in the note table. For example, CA#, CA#, CA# changes to CA0, CA1, CA2.

## 21.1.3.3 Set Languages

The comments can be displayed in multiple languages in the table.


© 2026 INNEO Solutions GmbH




1. Languages to be displayed in the note, which can be rearranged using drag and drop .
2. Icon for adding or removing a language

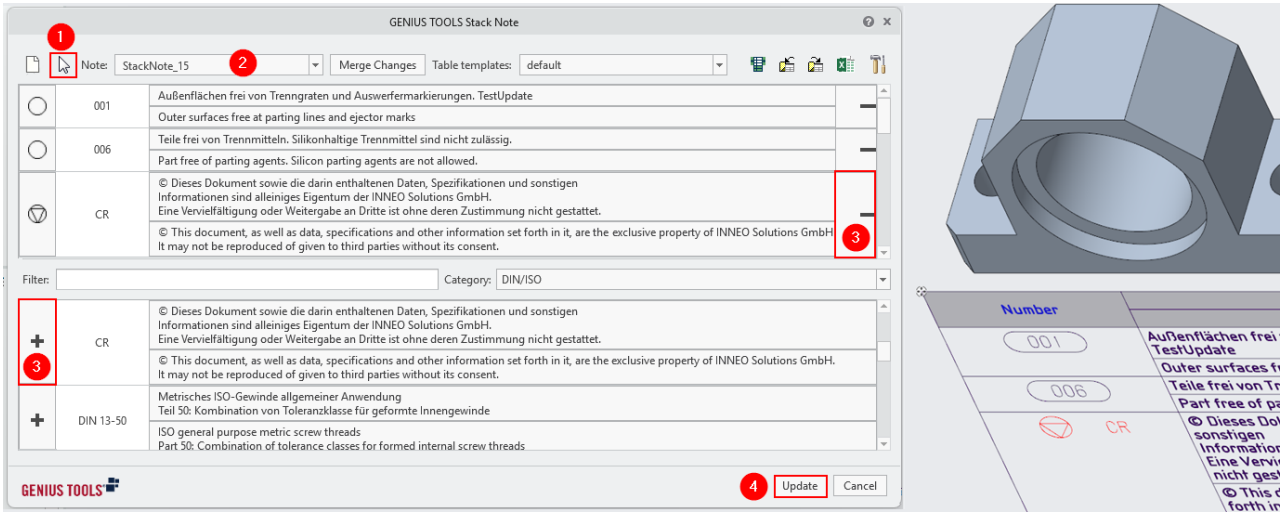
**Please note:** You can also drag and drop languages onto the *Shown Languages* (1) field and remove them by dragging the language to the *Language Pool* (3) field.

3. Language selection
4. Icon indicates that the language is defined in the database[578] .
5. Icon indicates that the language is defined in the embedded data .
6. Confirm the selected languages with *Apply* or click on *Cancel*.

## 21.1.4 Update note

A note that has already been placed can be updated. You can add or remove note entries. If you want to make changes to the database, you can find more information in the following chapter Apply changes from the database[588] .


© 2026 INNEO Solutions GmbH




1. Click *Select Note* in the command bar and click on a note that has already been placed or
2. select a note from the *Note* selection menu.
3. Note entries can be removed using the adjacent ➖ icon or add a note entry using the ➕ icon.
4. Update the existing note with *Update*.

## 21.1.4.1 Apply changes from the database

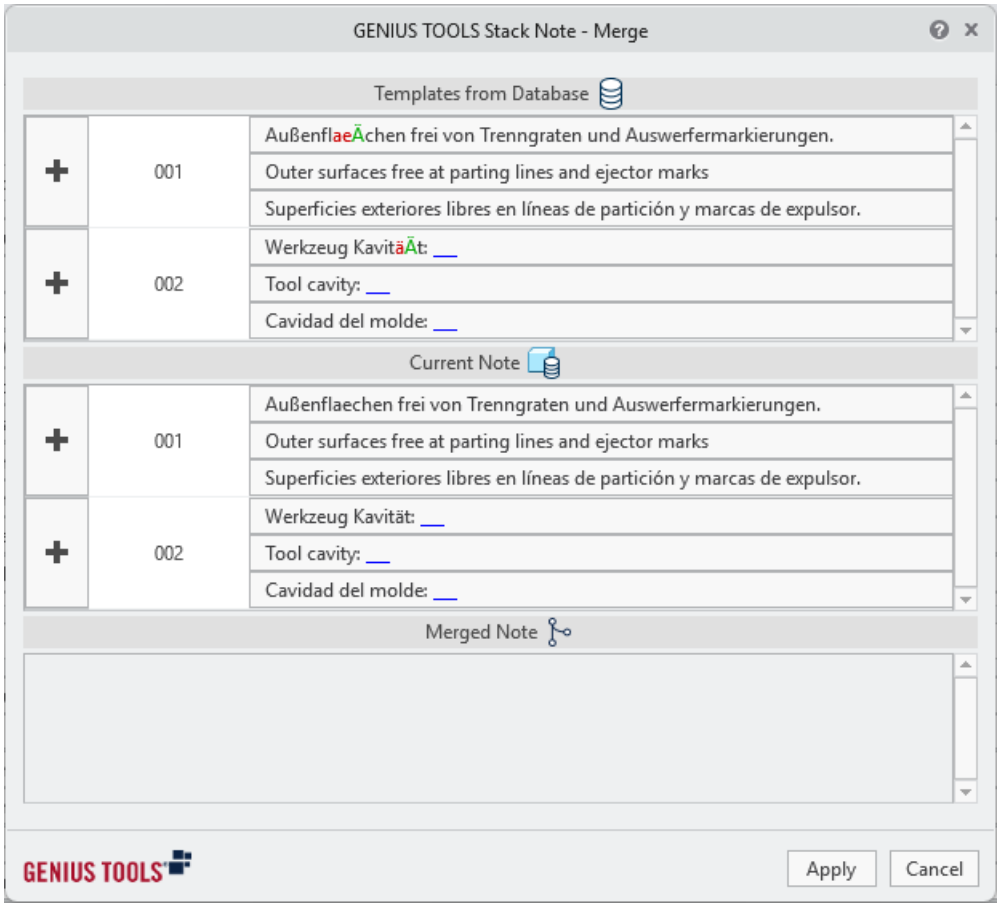
Changes to note entries are made in the *Stack Note 3D* database. Users can apply the changes either from the database or from the current note.

**Procedure**

1. Open a note that has already been placed in *Stack Note 3D*.
2. Edit the database as described in the following chapter Create and edit note entries 590.
3. Go back to *Stack Note 3D* and click *Merge Changes*.

There are three sections in *Merge Changes*:


© 2026 INNEO Solutions GmbH




*Templates from Database*: Changes from the database that are colored.

*Current Note*: The unchanged notes from the current note are displayed.

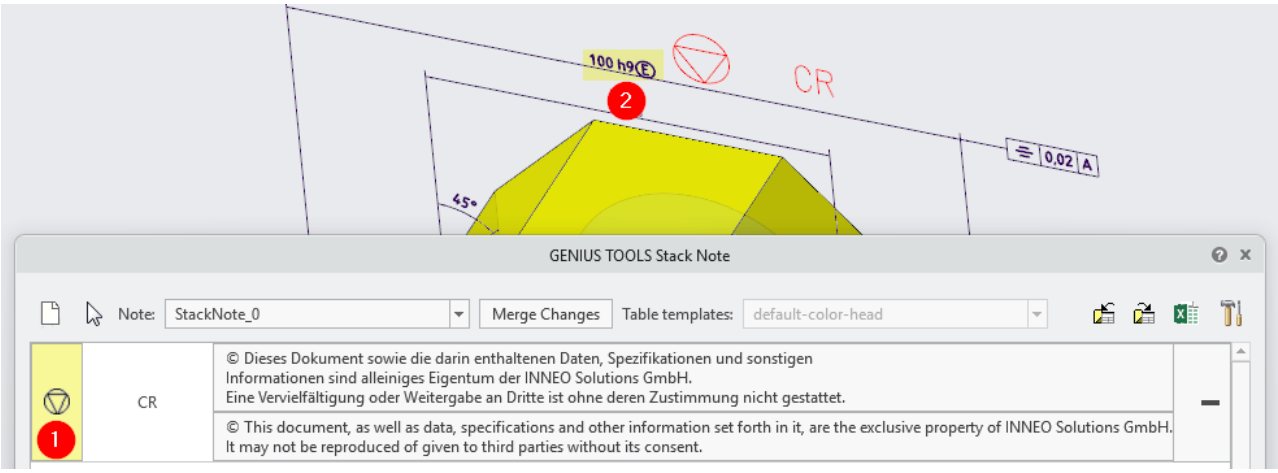
*Merged Note*: Notes which have been transferred and are to be included in the current note are displayed.

4. For each note entry, select whether you want to take it from the database or from the current note.
   - Database: Under *Templates from Database*, click the ✚ icon next to the note entry if you want to apply the change from the database to the current note.
   - Current note: Under *Current note*, click the ✚ icon next to the note entry if you want to transfer the note entry from the current note.
5. Confirm your selection with *Apply*.
6. If you agree with your selection, click *Update* to update the placed note.

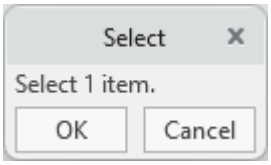
## 21.1.5 Placing symbols

Use the button for symbol placement (1) to link a symbol to a note, dimension, or symbol in 3D.


© 2026 INNEO Solutions GmbH




1. Click the button for placing symbols next to a note entry. When the button is activated, it is highlighted in gray. Creo prompts you to select a link target.




2. Click on a note, dimension, or symbol.
3. You can place the symbol using the middle mouse button.

The symbol with the note ID is placed next to the selected element.

**Please note:** Symbols can only be placed once a note table has been placed.

# 21.2 Configuration

In this section, you will find further information on example templates and how you can edit the database for the table and symbols.

## 21.2.1 Create and edit note entries

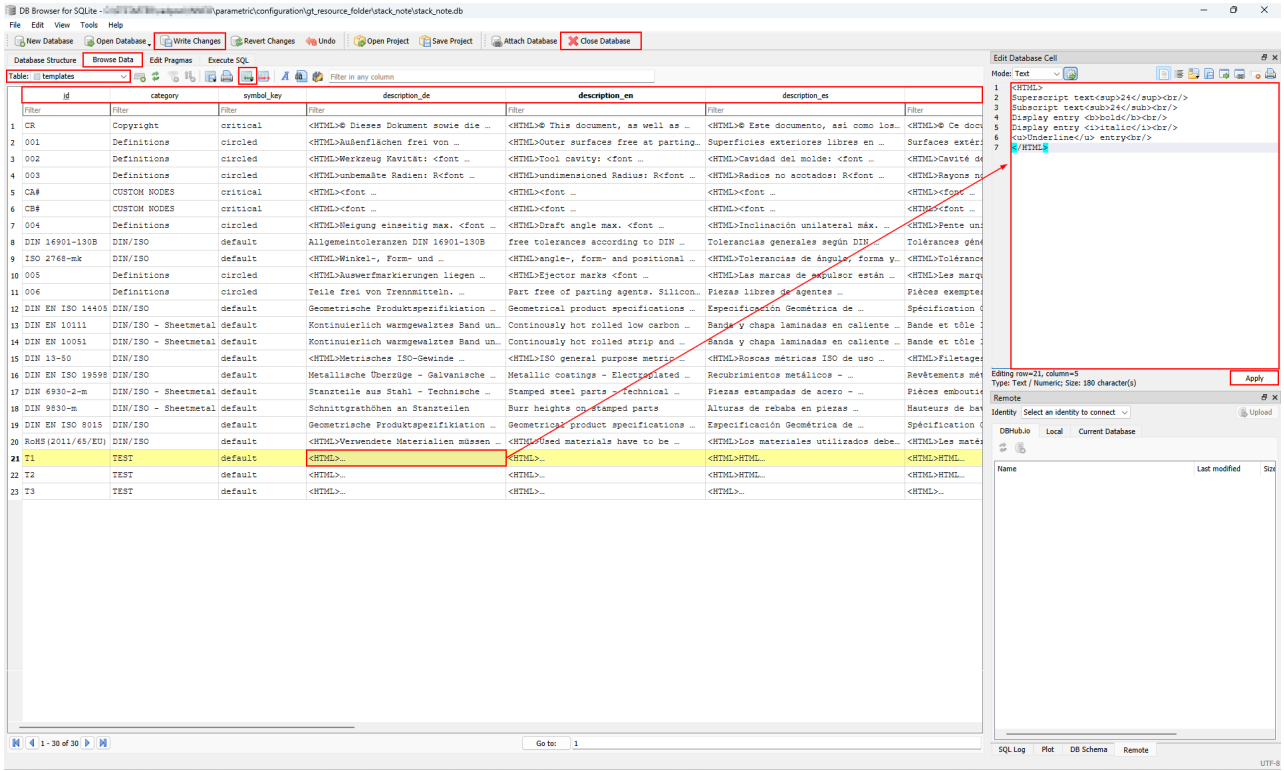
The available note entries are stored in the *stack_note.db* database.

1. Open the SQLite database in the resource folder:

   <Work environment>\parametric\configuration\gt_resource_folder\stack_note.db

2. Click on the *Browse Data* tab in the upper left corner.


© 2026 INNEO Solutions GmbH



Structure of the database in DB Browser for SQLite:

- *id*: Note IDs. A note ID can consist of numbers, letters, and characters. This will not be assigned twice and will be sorted alphanumerically.
- *categories*: Categories to which the notes are assigned
- *symbol_key*: ID of the symbol defined in the symbol database
- *description_<language abbreviation>*: Notes in the respective language

3. Change the desired notes under *description_<language abbreviation>*. The following HTML tags are supported:

– HTML: Use the start tag `<HTML>` and the end tag `</HTML>`

– superscript text: Use the start tag `<sup>` and the end tag `</sup>`

– subscript text: Use the start tag `<sub>` and the end tag `</sub>`

– bold: Use the start tag `<b>` and the end tag `</b>`

– underline: Use the start tag `<u>` and the end tag `</u>`

– italics: Use the start tag `<i>` and the end tag `</i>`

– line break: `<br/>`

4. Click *Apply* on the right.
5. Click *Write Changes* at the top.
6. Close the database.


© 2026 INNEO Solutions GmbH

## 21.2.2 Example Templates

**Excel templates**
A *Export Table to Excel* template

**Table templates**
Tables that are filled in by *Stack Note 3D* (with a predefined header and a value row that specifies the formatting). Table templates must consist of 3 rows and 2 columns.

**Symbol files**
Classification (example: critical, non-critical, undefined, etc.)

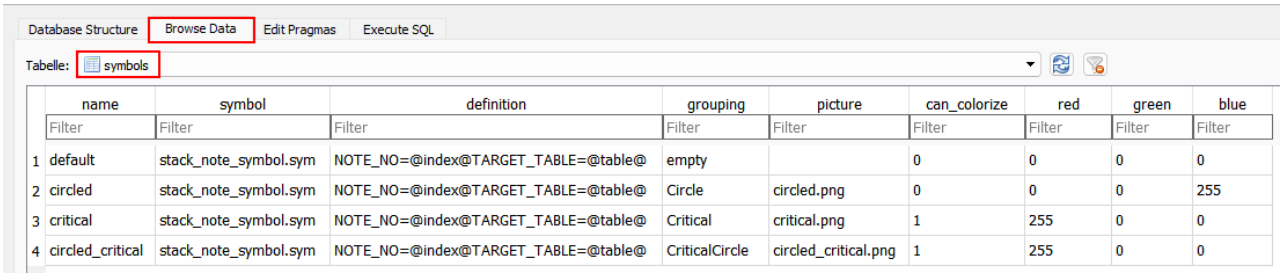
**Preview images**
Preview for symbols

**Database files**
Note and symbol configuration

## 21.2.3 Edit symbol database

The available symbols are managed in the stack_note_symbol.db database.

1. Open the SQLite database in the resource folder:
   *<Work environment>\parametric\configuration\gt_resource_folder\stack_note_symbol.db*
2. Open DB Browser for SQLite and click Open Database in the upper left corner to open a database (DB file).




| | name | symbol | definition | grouping | picture | can_colorize | red | green | blue |
|---|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | default | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | empty | | 0 | 0 | 0 | 0 |
| 2 | circled | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | Circle | circled.png | 0 | 0 | 0 | 255 |
| 3 | critical | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | Critical | critical.png | 1 | 255 | 0 | 0 |
| 4 | circled_critical | stack_note_symbol.sym | NOTE_NO=@index@TARGET_TABLE=@table@ | CriticalCircle | circled_critical.png | 1 | 255 | 0 | 0 |

**name**
Name of the symbol

**symbol**
This is the symbol file from *Stack Note 3D*, which is located in the resource folder.

**definition**
The variable text can be found in Creo under *Variable Text*. *NOTE_NO=@index@* and *TARGET_TABLE=@table@* must be included.


© 2026 INNEO Solutions GmbH

**grouping**
This is the selection for grouping.

**picture**
The file for the preview image of the icon.

**can_colorize**
Option for coloring the symbol

**red**
RGB values for red

**green**
RGB values for green

**blue**
RGB values for blue


© 2026 INNEO Solutions GmbH

# 22 Value Transfer

Available in the ribbon menu GENIUS TOOLS

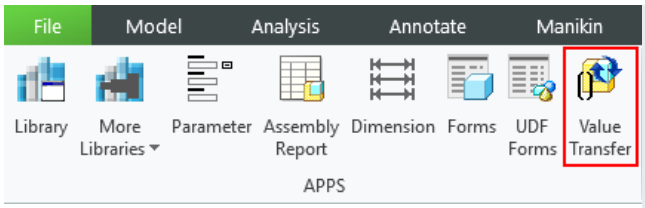
*Value Transfer* can be used in assembly mode to change multiple values in dimensions and parameters, as well as material definition files of subcomponents, in one step.

1. Search for subcomponents
   - with optional filters
   - display of search results in well-structured table format
2. Transfer parameter values
   - with control of the parameter values to be transferred by displaying the current parameter values for each component

By integrating *Value Transfer* into *Parameter*, it is possible to quickly transfer an assembly parameter (e.g. the project number) to submodels. In this case, the function for opening the Value Transfer dialog must be created for the project number parameter. The value transfer dialog will then be called from *Parameter* when the function of the parameter is clicked.

## Starting the program: in assembly mode

To start *Value Transfer*, click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

You can also start *Value Transfer*  in the following components:

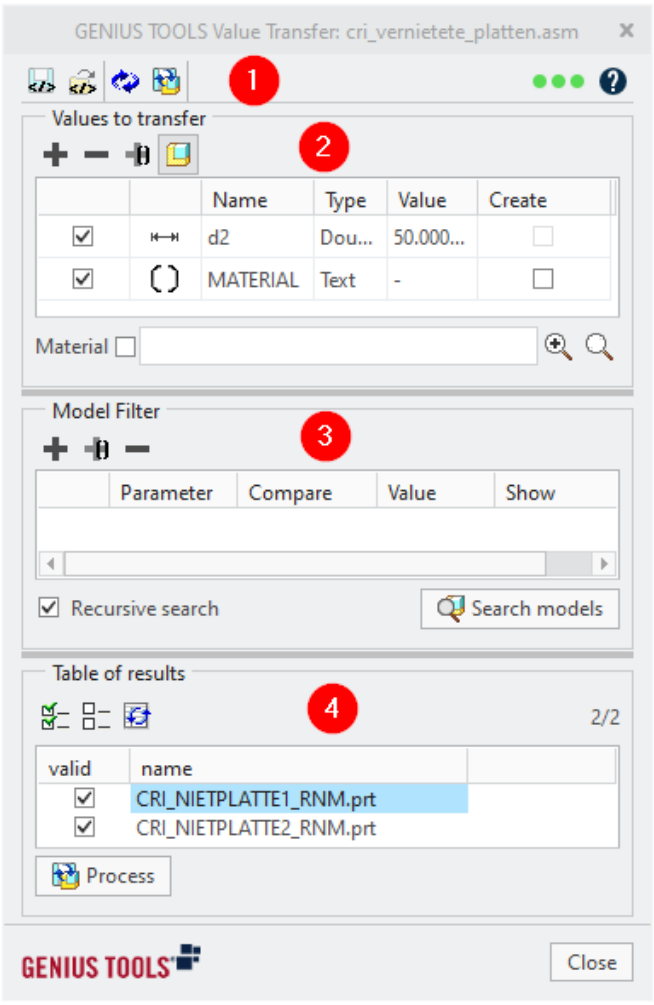

© 2026 INNEO Solutions GmbH

– In *Forms*[123] from the command bar[126].

– In *Parameter*[465] as a button in the input field of a parameter which has been equipped with the function Transfer parameter to assembly component[509] without filtered values.

## 22.2 User interface

The user interface of *Value Transfer* consists of the following elements:




1. Command bar[596]
2. Transfer list[596]
3. Filter list[596]
4. Results table[599]


© 2026 INNEO Solutions GmbH

## 22.3 Command bar

The command bar displays frequently used functions. The following buttons are included:

| Icon | Name | Description |
|---|---|---|
|  | Save parameter and model search as XML | Saves all settings for parameters, dimensions and filters in an XML file. |
|  | Load parameter and model search from XML | Loads settings for parameters, dimensions and filters from an XML file. |
|  | Refresh | Analyzes the current assembly, evaluates parameter and dimension values and reapplies filters. |
|  | Write parameter into model and regenerate | Transfers parameter and dimension values to sub-components and regenerates the entire assembly. |
|  | Status indicator[476] | Shows the current status for *loading*, *working* and *saving* with traffic light colors and opens the status dialog. |
|  | Help | Opens the Help. |

## 22.4 Transfer list

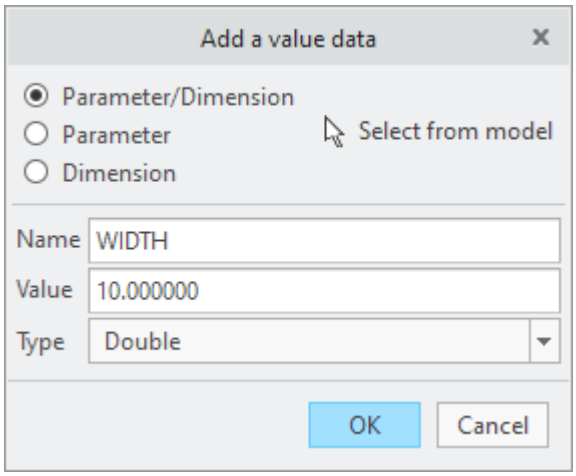
Parameters and dimensions, which should be transferred to sub-components, are specified in the transfer list. The entries in the list are displayed as column in the results table[599] .

### Creating parameters and dimensions

Add elements with the ✚ button and select:

- **Parameter/Dimension (combined):** allows you to enter free text. When transferring these values, the sub-components are searched for a dimension with that name. If such a dimension does not exist, an attempt is made to find a parameter with the same name. Otherwise, a parameter with that name is created and filled with the value.
- **Parameter:** allows to select an assembly parameter.
- **Dimension:** allows you to select dimensions from sub-components.


© 2026 INNEO Solutions GmbH

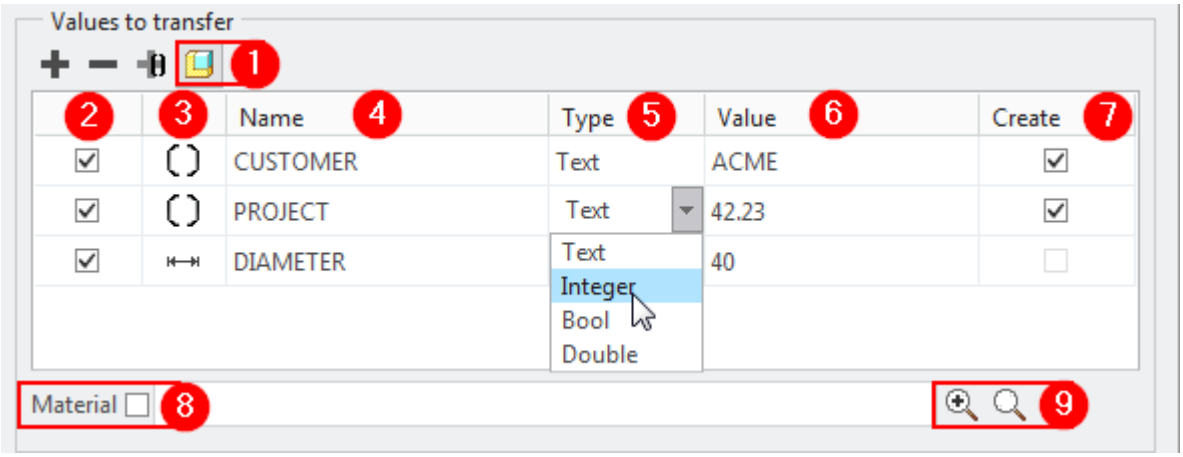



*Adding an element to the transfer list*

## Transferring values

To transfer values to an assembly, use the assembly symbol (1). If an assembly is locked by your PDM system, the button is grayed out. Non-locked subassemblies and parts can still be processed.

Use the checkbox (2) in the first column above to specify which parameters and dimensions to transfer to subcomponents. Specify the values to be transferred in the columns *Name* (4), *Type* (5) *Value* (6).




*Transfer list*

Use the second column (3) to change the type (Parameter, Dimension or Combined).

**Warning**: Always check the type of parameters and dimensions! A wrong type may cause existing parameters or dimensions to not be overwritable.

| Type of parameter | Description |
|---|---|
| Boolean parameter | *yes*, *true*, *1*, *ja* (engl. yes) and *wahr* (engl. true) are interpreted as *correct*, everything else is interpreted as *wrong*. |


© 2026 INNEO Solutions GmbH

| Type of parameter | Description |
|---|---|
| Double, Integer | Allows the input of numbers. |
| Text | Allows the input of text. |

If a parameter does not exist in a subcomponent, it can be created automatically by activating the checkbox *Create* (7) and the value will be transferred.

To delete an entry, click on the entry and then on the Minus button.

### Changing materials

You can also change materials of sub-components with *Value Transfer*. Click on the *Material* checkbox (8) and select the required material using the magnifying glass buttons (9) with the material selection or from a model.

## 22.5 Filter list

Criteria for filters to search for in sub-components are specified in the filters list. When the filter criteria are met, the specific sub-component is displayed in the results list.

Select the checkbox *Recursive search* to include subassemblies. Use the checkbox *Show* to change the filtering. Then refresh the results table. Add parameters using the Plus buttons with free input or by using the Creo parameter selection dialog. Click the parameter name, then click the Minus button to remove a parameter.

Specify a comparative operator in the *Compare* column. Available are: *equal* (=), *not equal* (!=), *greater-than* (>), *less-than* (<), *contains*, *contains not* and *regular expression*.

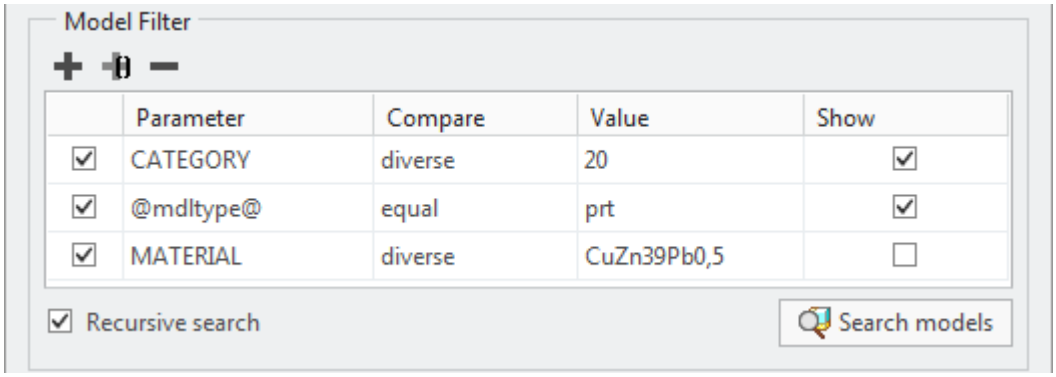
Enter the comparative value into the last column *Value*. Confirm the filters list by clicking the *Search models* button to refresh the results table.

**Warning**: Always refresh the results table, as only the subcomponents displayed in it are refreshed.

### Filtering with variables

Add an entry to the filters list and open the context menu for the entry. Select the desired variable, set the comparative operator and enter a comparative value.


© 2026 INNEO Solutions GmbH




*Use variables (here: @mdltype@) to filter by specific model properties*

The filters are case-sensitive. For variables, case is ignored for the *Equal* and *Not Equal* operators.

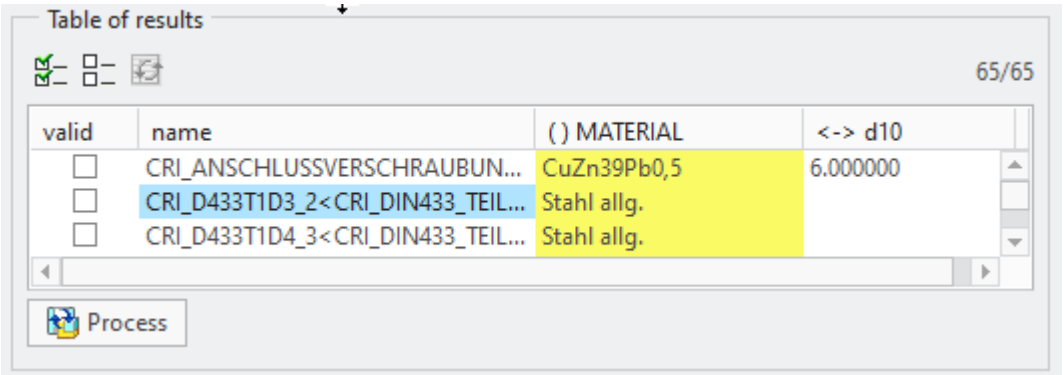
If the value of a filter parameter is empty and the comparison is set to *Equal*, all models are listed in which either this parameter does not exist or is empty. If only models with new parameter values in which the parameter exists are to be assigned, the check mark in *Create* must be removed.

**Tip**: Variables are always treated like text parameters.

You can find the variables available via the context menu in the Variables overview[913].

## 22.6 Table of results

The Table of results displays the filtered sub-components. The current values of the parameters from the parameters list are displayed in the individual columns.




Use the checkboxes in front of the components to select the subcomponents to which *Value Transfer* will be applied.

Parameters that are controlled by a relation or have access restrictions are marked with a yellow background. Dimensions that are controlled by a relation are also marked with a yellow background.


© 2026 INNEO Solutions GmbH

If a PDM system is used, the individual models can only be checked if they are unlocked. Models can be unlocked via the right mouse button menu. If several models are marked (via Shift and Control), several models can be unlocked.

By clicking the *Process* button, the parameter values specified in the parameter section are written into the sub-components.

**Please note**: Only active sub-models are considered when processing.

The parameter values in the results table can also be modified manually. Just click into a table cell and modify the value.


© 2026 INNEO Solutions GmbH

# 23 UDF Forms

Available in the ribbon menu GENIUS TOOLS

*UDF Forms* are graphical interfaces. Using *UDF Forms*, you can create groups of UDFs in models and then edit the individual properties later on.

*UDF Forms* is available in assembly, part, and drawing mode with the following features:

1. Creation of UDF groups in models
   - dependent or independent
   - optionally with variable dimensions from lists and tables
2. Use of UDF family tables - for form control
3. Use of variable parameters (only feature parameters located on the first feature of the UDF group)
4. Subsequent editing of already created UDF groups
5. Re-placing UDF groups with the same values of already created UDFs
6. UDFs with body references are supported
7. Rules between properties can be defined with JavaScript.
8. Loading of external data into the mask is possible (e. g. EXCEL, CSV)

The definition and construction of a UDF form is done with a graphical editor, see UDF Form Editor 614.

## 23.1 Fundamentals

With *UDF Forms* you can create UDF groups in models and manipulate individual properties later on.

### Glossary

In this glossary, the terms are sorted by relevance.

**UDF (generic term, Abbreviation for User-Defined Feature, built-in UDF group or saved UDF GPH file)**
User-defined feature with selected single features and their associated properties for placing the UDF in a part or assembly. UDFs can be dependent or independent. Beyond Creo Parametric functionality, *UDF Forms* allows you to create and subsequently edit UDF groups in models.


© 2026 INNEO Solutions GmbH

**UDF GPH File (*<udfname>.gph*)**
Creo internal file format in which all the details of a UDF are stored on disk. *UDF Forms* does not allow the creation of GPH files, but the assembly, subsequent editing and replacement.

**UDF-Group**
A UDF inserted into a model and previously saved. It appears in the model tree of the model.

**Target model**
The target model is the model opened in Creo into which a UDF group is placed.

**UDF Form**
Graphical mask for assembly and subsequent manipulation of properties. For the creation in the target model the UDF file and the UDF Forms definition are necessary.

**UDF Forms Definition**
XML structure that contains the description of the mask for a UDF. This XML structure can be stored either in the file directory or in the Creo model. If a UDF is assembled multiple times, the UDF Forms definition is included once in the target model.

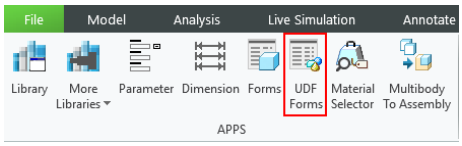
## 23.2 Usage

This section contains information on using *UDF Forms*. It describes the general structure of the program.

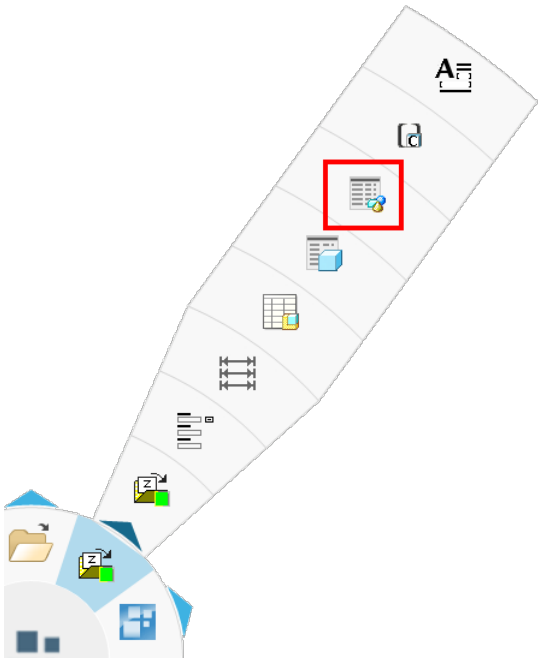
### Starting the program: in part mode

Start *UDF Forms* from the *GENIUS TOOLS* ribbon menu in the *APPS* section or by using the *Quick Access* (key [<]).


© 2026 INNEO Solutions GmbH




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

*UDF Forms* only starts after selecting a UDF in the Creo main window or a UDF group in the model tree.

## Forms and UDF Forms in the model

If a model contains a *Form* or *UDF Form*, a suitable Forms icon is displayed in the Creo Parametric main window. Click on the icon to open the form.

 Form in the model

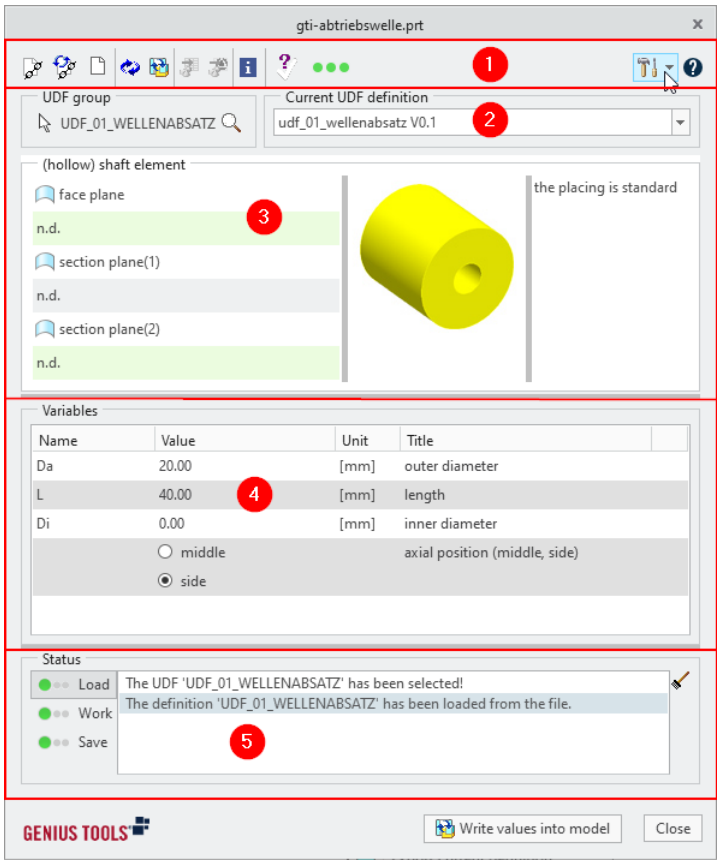
 UDF Form in the model

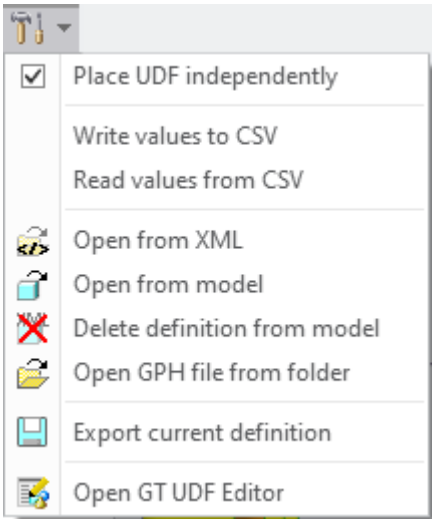
## 23.2.1 User interface

The *UDF Forms* user interface consists of the following elements:

1. Command bar[604] with tools menu[606]
2. UDF selection[607]
3. Placement references[607]
4. UDF Form[609]
5. Status indicator as short overview, a separate window for the status indicator with search function[476] can be opened via the tool menu


© 2026 INNEO Solutions GmbH




*Tools menu*

## 23.2.2 Command bar

The following buttons are included in the command bar:

| Icon | Name | Description |
|---|---|---|
|  | Create with references | Creates a UDF in a model. The placement references are used. |
|  | Repeated creation with references | Creates several UDFs in a row. The placement references are used. |
|  | Create | Creates a UDF in the model. References must be selected during placement.<br>The button can be hidden with `gtuf_show_create_without_references = 0`. |

| Icon | Name | Description |
|---|---|---|
|  | Create new UDF group by Creo std. Dialog (values would not be transferred) | Integration of the Creo function for creating a new UDF in *UDF Forms*. You can create a new UDF in an open model and then edit it and enter values with *UDF Forms* without changing the user interface.<br>This function is only available if the configuration option `gtuf_show_create_by_creo = 1` is set (Default: 0). |
|  | Refresh | Updates the view after editing the UDF Form with the editor. |
|  | Save values into model | Saves the entered values in the form into the current model and regenerates it. |
|  | Apply to table | Links the active UDF to an already placed drawing table.<br>(Only active in drawing mode) |
|  | Apply to symbol | Fills drawing symbols with the values of the UDF. The drawing is regenerated.<br>(Only active in drawing mode) |
|  | Show info | Opens a stored, language-dependent help document. |
|  | Run check function | Starts the JavaScript function CheckUI from the UDF definition to check the values entered in the form. (The CheckUI function must first be stored in the UDF object.) |
|  | Status indicator[476] | Shows the current status for *loading*, *working* and *saving* with traffic light colors and opens the status dialog. |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
|  | Tools | Opens the tool menu containing various supporting functions. |
|  | Help | Opens the user manual. |

## Repeated creation with references

Use the function *Repeated creation with references* if you want to place a UDF several times in a model:

1. Activate the part in which you want to place UDFs.
2. Define constant references beforehand and then click on the button *Repeated creation with references*.
3. Select the missing references in the part. Repeat this step until the required UDFs have been placed.
4. Click with the middle mouse button to cancel the placing.

## Tools menu

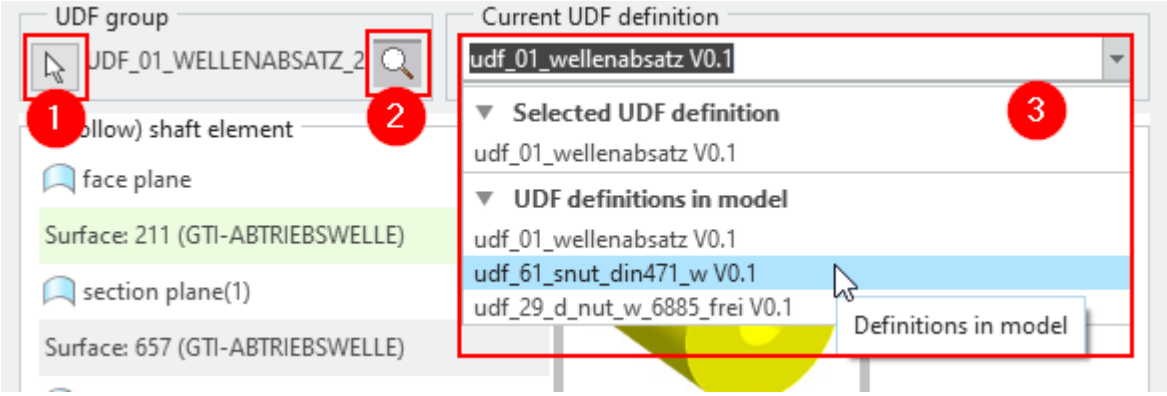
| Icon | Name | Description |
|---|---|---|
| | Place UDF independently | If a UDF is placed dependently, changes to the values of the UDF form (the GPH file) will be applied. Define the default setting for the checkbox in the editor.[618] |
| | Write values to CSV | Saves the current values in the form to a CSV file. |
| | Read values from CSV | Reads values from a CSV file and applies them to the current UDF form. |
|  | Open from XML | Opens a UDF group from an XML file (UDF definition). |
|  | Open from model | Opens a UDF definition from the current model. |
|  | Delete definition from model | Deletes a UDF definition from the current model. |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
| | Open GPH file from folder | Opens a UDF file from the file system. |
| | Export current definition | Saves the definition of the current UDF group as XML file. |
| | Open UDF Editor | Opens the *UDF Forms Editor* to edit the UDF definitions. |

## 23.2.3 UDF selection

The *UDF selection* consists of three elements:




*UDF selection*

Select UDF groups in a part, assembly or drawing directly in Creo or in the model tree with the arrow icon (1).

Use the magnifying glass icon (2) to highlight the currently selected UDF in models.

The drop-down list (3) shows the UDF definitions in the model. Open the list and choose between the UDF definitions.

Only one UDF definition is saved in the model if a UDF group is placed several times.

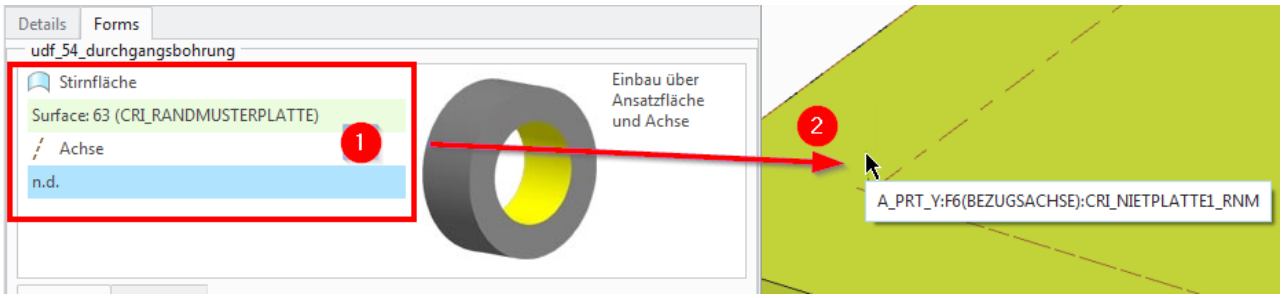
## 23.2.4 Placement

### Placement references

This section displays the placement references of the selected UDF group on the left side of the dialog. The right side contains a description and a preview image, which can be configured in the editor.


© 2026 INNEO Solutions GmbH

Click on a colored field (1) and select a placement reference in the model (2).

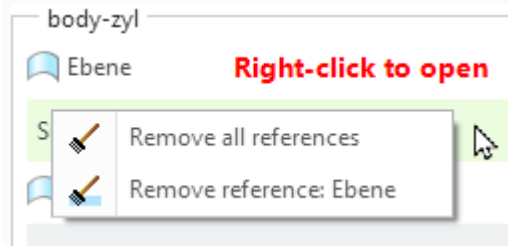



*Selecting the reference Axis in the model*

Use the functions *Create with references* and *Repeated creation with references* as soon as you have defined consistent placement references for multiple installations.

**Tip**: UDFs can be placed in assembly mode by enabling the required part.

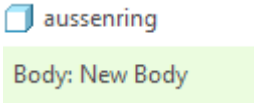
Use the context menu to delete placement references.




*Context menu*

## Placement references for bodies

The placement reference can be a body. If a UDF form requires only one body placement reference, this reference is filled in automatically with the active body. If the UDF shape has multiple body references, the bodies must be selected. If a body reference is to create a new body, this intent can be predefined in the *UDF Form Editor*.




To reach the "New Body" setting interactively, cancel the body reference selection with the middle mouse button.

## Place UDF dependently

A UDF can be placed dependent or independent. If a UDF is placed dependently, subsequent changes to the values in the UDF form (i. e. the GPH file) are applied to the UDF. This functionality can also be created in Creo.

Enter your choice in the checkbox *Place UDF independently* of the tools menu[606]. The default for the checkbox is activated. This can be changed in the editor[618].


© 2026 INNEO Solutions GmbH

## 23.2.5 Form section

The form displays variables of the currently selected UDF group. The form elements are arranged in a table with the following columns:

– Name of the variable

– Current values of the variables. Depending on the configuration, different input fields are displayed. Click on a displayed value to activate input fields.

– Units of the variables

– Title: the localized description

– Tools: Displays value tables or Javascript functions from appear in the Tools column, if available.

Both, the visibility as well as the order of the columns can be changed in the *Description* tab of *UDF Forms Editor*[628].

## Input field types

Depending on the UDF group used, different types of control elements are available in the form.

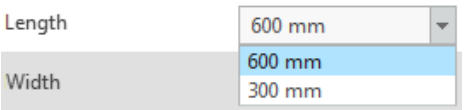
**Warning**: Check the Creo configuration option `show_dim_sign`, as *Forms* respects it.
`show_dim_sign=no`: Entering a negative value changes the direction vector of a dimension. The value becomes positive.
`show_dim_sign=yes`: When a negative value is entered, the value remains negative.

### Input field




Input fields accept any string for input. Restrictions (e.g. numbers only) can occur depending on the parameter type.

### Selection field




Selection fields (drop-down lists) contain a selection of possible inputs. They are predefined through the configuration.


© 2026 INNEO Solutions GmbH

**Radio button**




Radio buttons display predefined selectable values. Only one of the values can be selected.

**Binary checkbox**




Checkboxes allow to choose from two options. This can be Yes/No decisions, for example. Always two values are deposited in the configuration for checkboxes. With the checkmark set, the first value is used. Checkmark not set applies the second value.

**Value table**



Value tables control the presetting of form elements. Click on the icon to open the selection dialog.

## 23.2.6 Use cases

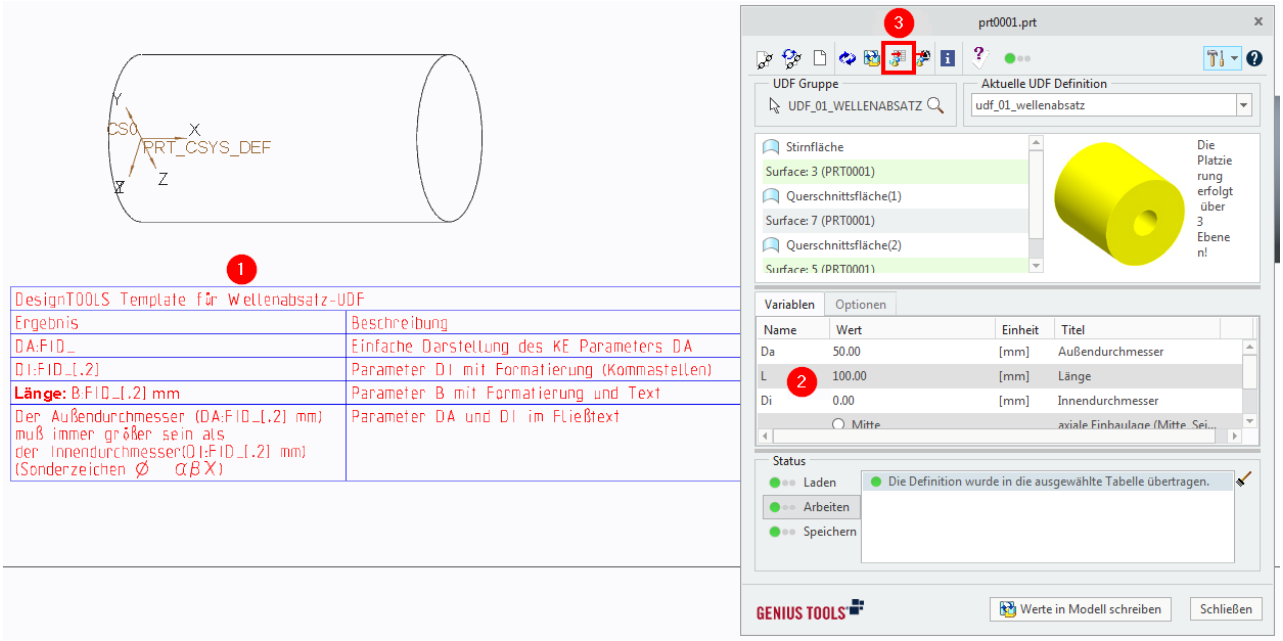
In this section you will find use cases related to *UDF Forms*.

# Linking UDFs to drawing tables

To link UDFs to tables, proceed as follows:

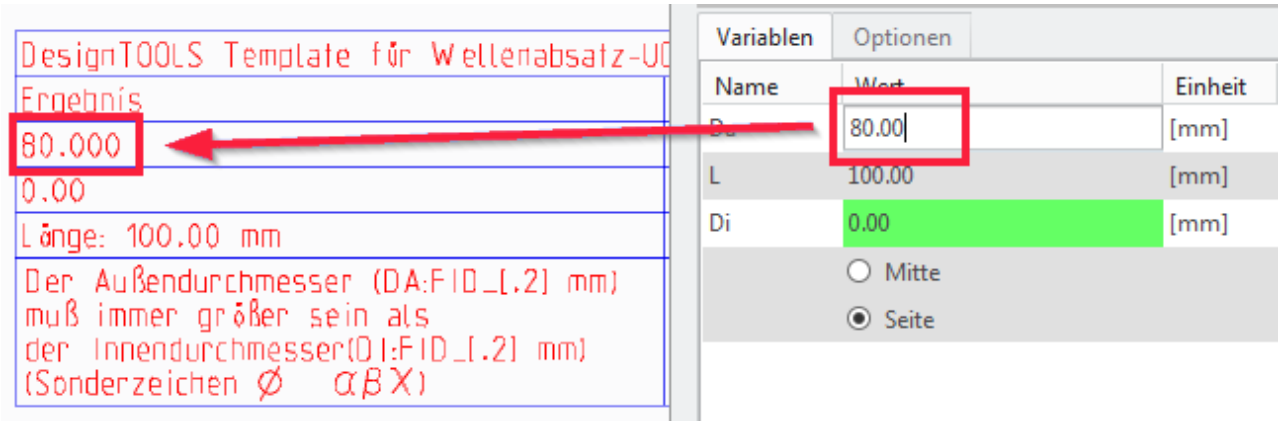
1. Place a UDF in a part.
2. Create a drawing.
3. Insert a table in the drawing.
4. Enter the required variables of the UDF in the table in the following notation:
   `VariableName:FID_[decimal places]`
5. In UDF Forms, click the button *Apply to Table*.


© 2026 INNEO Solutions GmbH




*Create a table (1)*
*Enter the variables from the UDF (2)*
*Click on "Apply to table" (3) and select the table*

Select the table created in the drawing. The UDF is now linked to the drawing table. The table is automatically updated each time the UDF is changed.




*If the UDF is changed, the table is updated automatically*

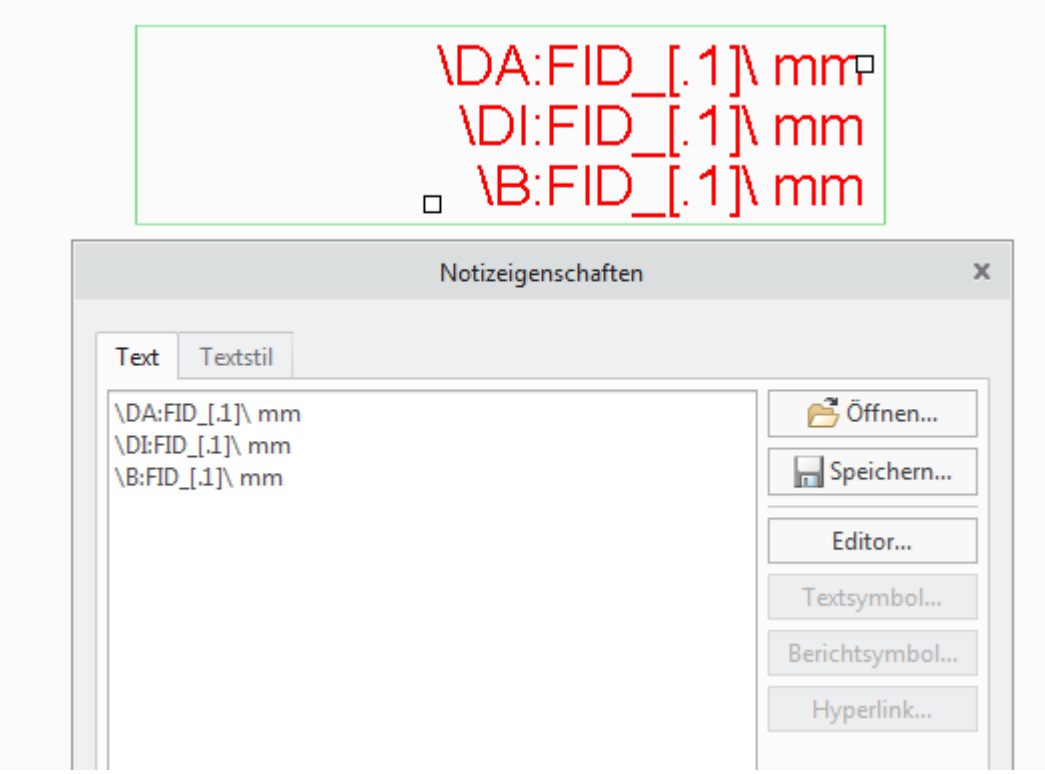
# Linking UDFs to drawing symbols

To link a UDF to an icon with variable text, proceed as follows:

**Preparing the symbols**
Create a separate symbol for each UDF!

Enter the variables of the UDF in the following notation as a note in the symbol:
`\VariableName:FID_[decimal places]\`


© 2026 INNEO Solutions GmbH

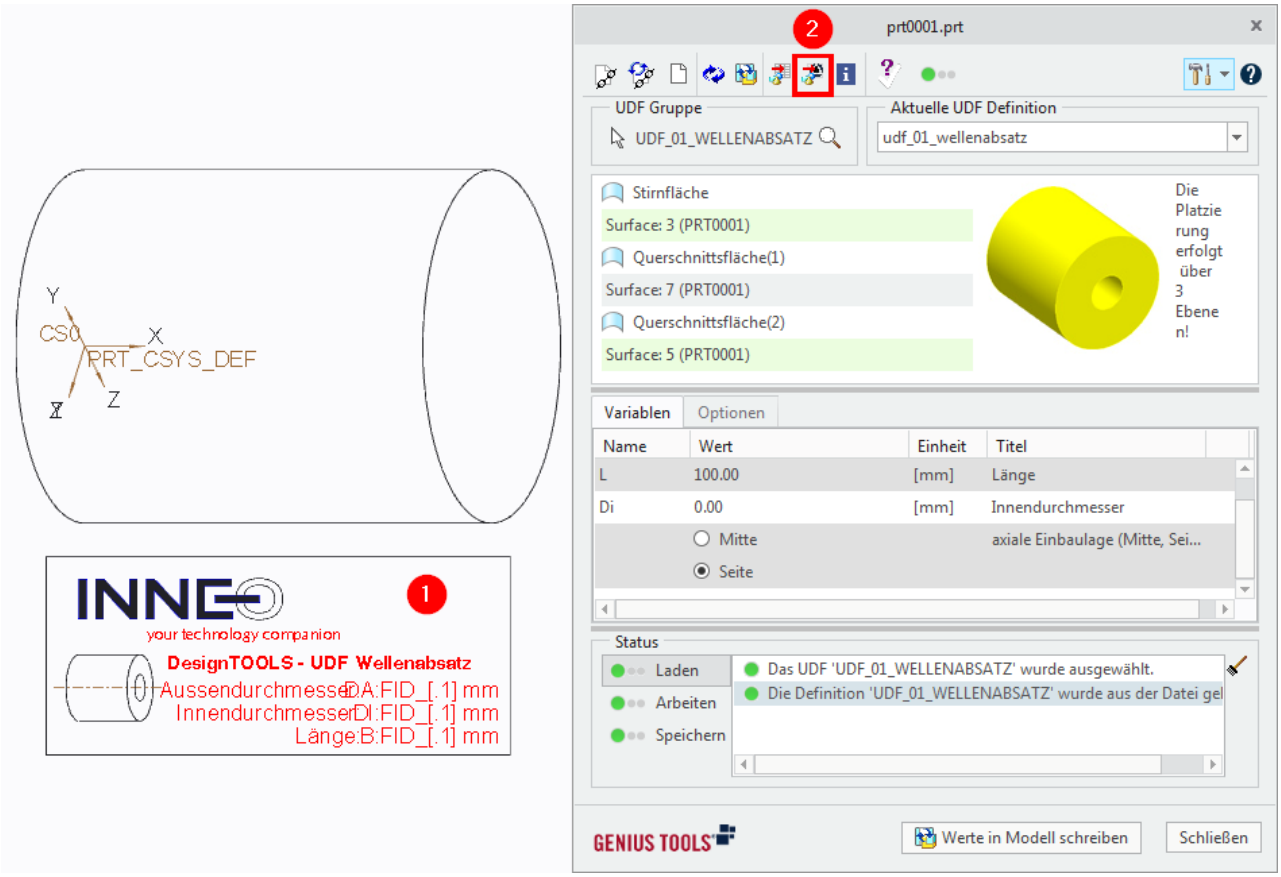



*Create a note with the desired variables in the symbol*

**Linking the UDF with the symbol**

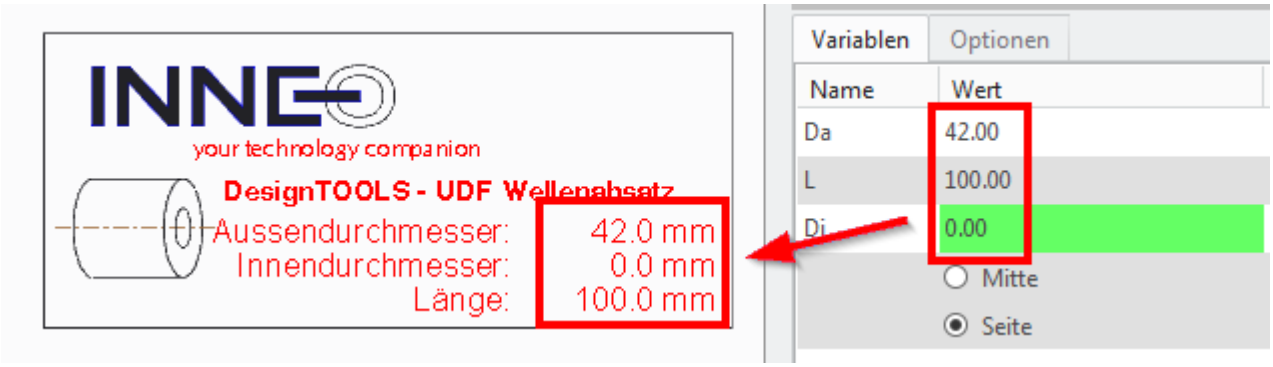
1. Place a UDF in a part.
2. Create a drawing.
3. Insert the appropriate symbol for the UDF in the drawing.
4. In UDF Forms, click the button *Apply to symbol*.


© 2026 INNEO Solutions GmbH




*Place the prepared symbol (1)*
*Click on "Apply to symbol" (2) and select the placed symbol*

Select the placed symbol in the drawing. The UDF is now linked to the drawing symbol. The symbol is updated automatically, each time the UDF is changed.




*If the UDF is changed, the symbol updates automatically*

## 23.3 Configuration

This section contains further information on configuration and structure of *UDF Forms*.

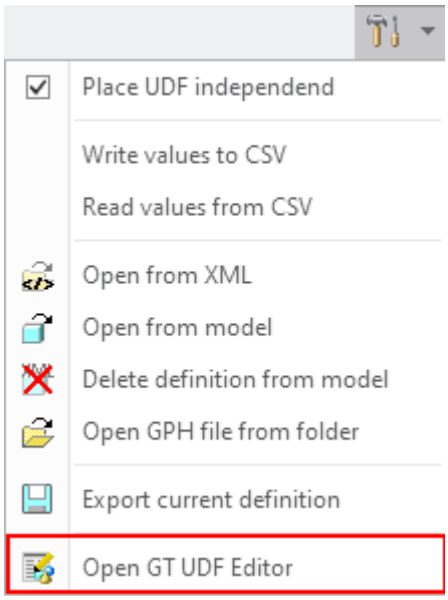

© 2026 INNEO Solutions GmbH

## 23.3.1 UDF Forms Editor

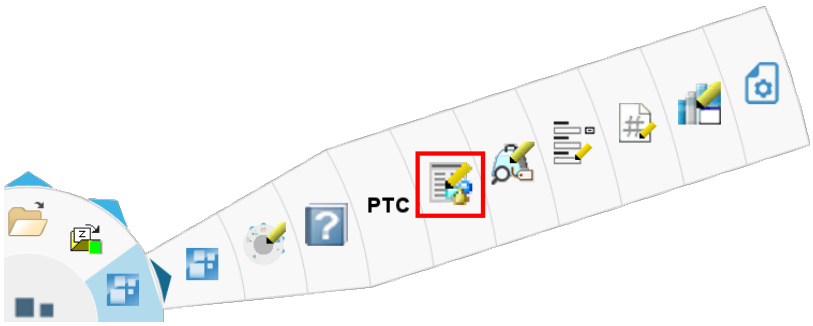
With *UDF Forms* UDF groups are created and edited. No changes are made to the UDF. All settings are stored in the XML structure of a UDF definition.

### Starting the program

Start the editor from the tools menu of *UDF Forms* or via *Quick Access* (key[<]).




*Starting from the tools menu*




*Starting in Quick Access*

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors[781].

### Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

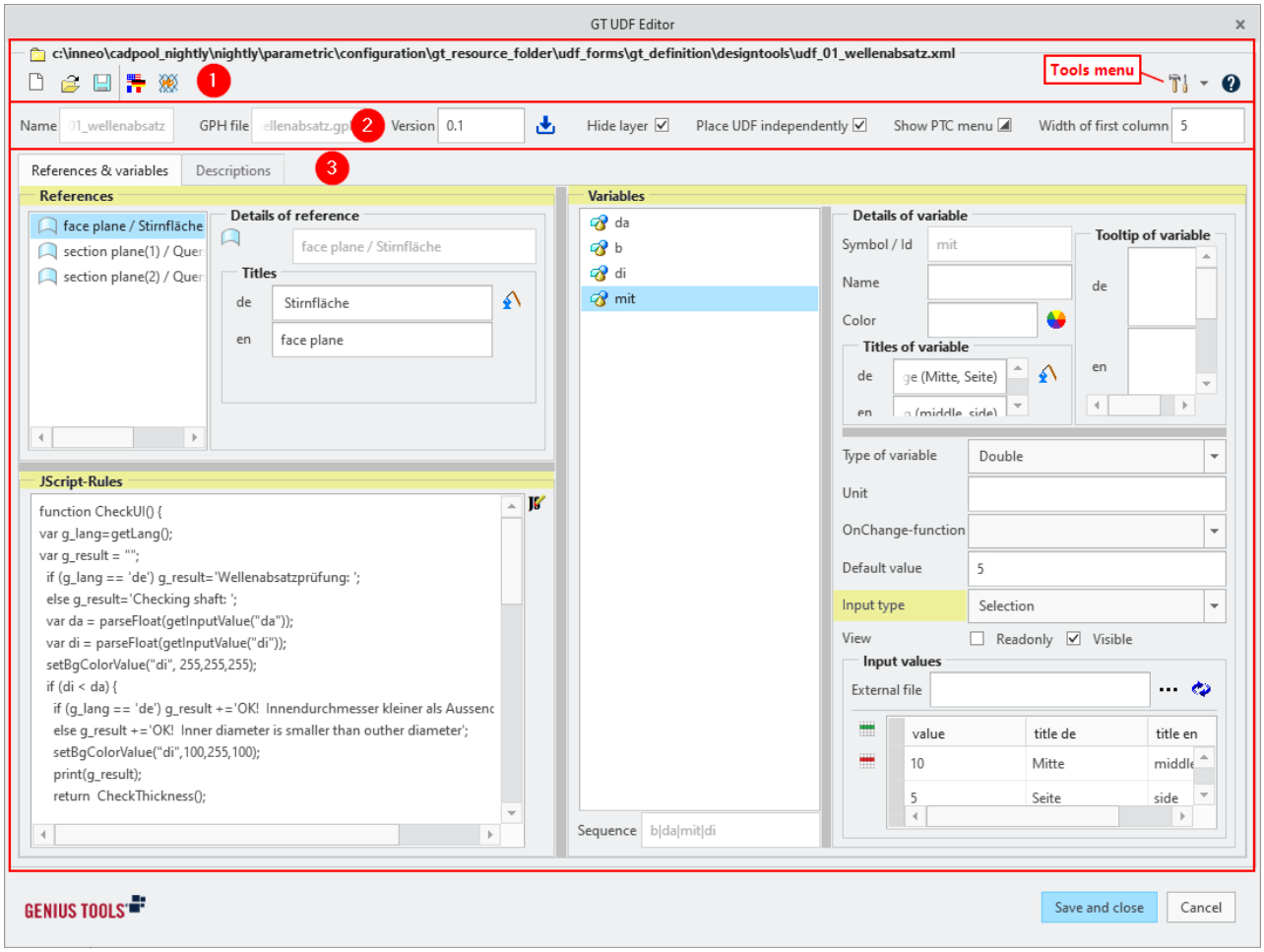
To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.


© 2026 INNEO Solutions GmbH

## 23.3.1.1 User interface

The user interface of *UDF Editor* consists of the following elements:




1. Command bar[615] with tools menu
2. Details of selected UDF[617]
3. Tabs:
   – References & Variables[618] with the segments: References[619] – JavaScript-Rules[619] – Variables[621] – Field types[623]
   – Descriptions[627]

## 23.3.1.2 Command bar

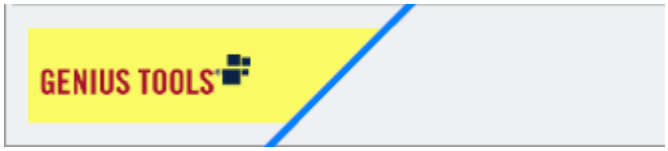
The following buttons are included in the command bar:

| Icon | Name | Description |
|---|---|---|
| | Create new definition from GPH file | Creates a new UDF definition from the selected UDF in XML format. |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
| | Open from XML file | Opens a UDF definition from the file system. |
| | Save | Saves the current UDF definition. |
| | Display variant list | Displays a list of available variants of the current UDF. |
| | Edit Languages | Opens the dialog for managing the languages of a UDF definition. |
| | Automatic standard texts | Opens the dialog for automatically setting standard texts for all variables and references. The table is also divided into these two areas.<br>The identifier of the GPH file is used as the key for the translation for the references and the name for the variables.<br>See Set standard texts for multiple elements |
| | Tools menu | The tools menu contains various supporting functions. |
| | Help | Opens the help. |

The current location of the UDF definition is displayed above the command bar. If an internal UDF definition is edited, the footer turns yellow.




*The footer changes color and indicates that an internal UDF definition is being edited.*

**Warning**: Make sure to edit the correct definition!

## Tools menu

Supporting functions are started from the tools menu.

**Open a definition according to selected UDF model**: Opens an external UDF definition for the model.


© 2026 INNEO Solutions GmbH

**Create a new definition according to a selected UDF**: Creates a UDF definition from the current selected UDF.

**Open from GPH file**: Opens a UDF definition from a UDF file.

**Open from model**: Opens a UDF definition from the current model.

**Save definition as XML file**: Saves the current configuration to an external XML file.

**Write definition into the model**: Saves the current configuration into the UDF definition in the current model.

**Delete a definition in the model**: Deletes the selected UDF definition from the current model.

## 23.3.1.3 Details of UDF

The segment UDF Details contains the following information.




**1. Name of UDF definition**
Shows the name of the selected XML file, which defines the form of the UDF.

**2. Name of GPH file**
The GPH file stores the UDF.

**3. Version**

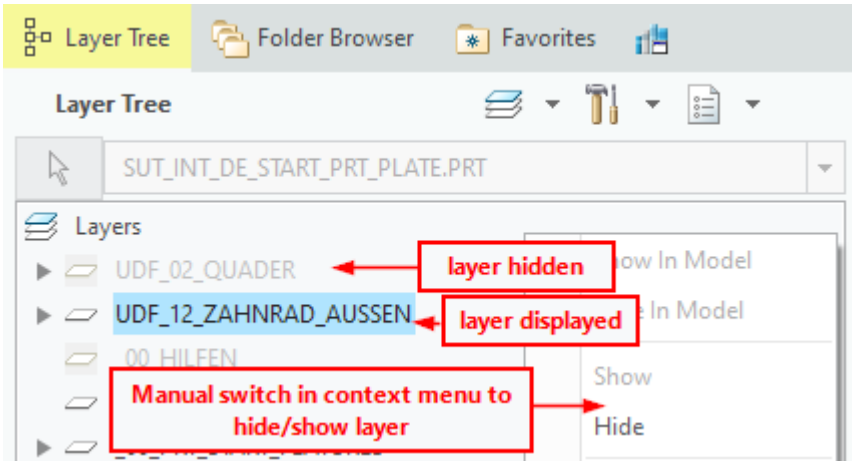
Displays the current version of the UDF definition (XML).

**4. Update**

The button  opens the dialog that updates the UDF definition with the current GPH file, see %TARGETTITLE%>629.

**5. Hide layer**
A slide with the same name is created for each built-in UDF. All auxiliary elements are located on this slide. If you create geometries with a UDF, these are displayed, but not the auxiliary elements required for the construction (default setting with check mark set). If you create auxiliary elements with UDF, then it is recommended not to set the check mark, so that the slide is displayed and you can see the implemented UDF.


© 2026 INNEO Solutions GmbH




**6. Place UDF independently**

Define the default setting for this checkbox in the tools menu of the *UDF Forms* dialog.

– Place UDF independently (default): Changes to the UDF form (the GPH file) will not be applied to the UDF.

– Place UDF dependently: Changes to the UDF form (the GPH file) will be applied to the UDF after updating.

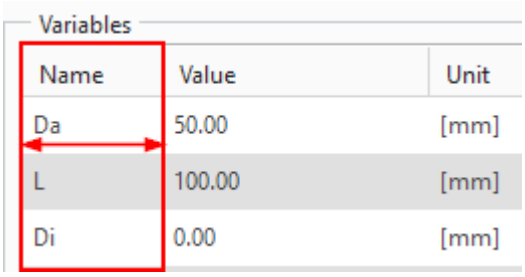
**7. Show PTC menu**

This checkbox is used to determine whether the PTC menu for selecting the assembly properties is displayed: *On* ☑ / *Off* ☐ / *Conditionally On* ◪. *Conditionally on* ◪ is selected by default. With this function, the menu for selecting assembly properties is displayed only if Creo Parametric decides that it is required for the particular UDF.

**8. Width of first column**

Defines the width of the first column in the Variables area. Default is 5. Enter -1 if you want the first column to enlarge automatically when the dialog window is enlarged.

The first column can be defined in the Description tab under Column order[628].




## 23.3.1.4 References and variables tab

The References and Variables tab consists of the sections:

– References:[619] for managing the placement references of a UDF.

– Javascript rules[619]: executable code for a UDF form

– List of available variables[621] of the UDF with details and field types


© 2026 INNEO Solutions GmbH

– Instances: This section is only displayed if a UDF contains variants or instances. Use the drop-down list *Function after instance change* to select a JavaScript OnChange function. Use the *Visible* checkbox to show or hide the variant selection in UDF Forms.

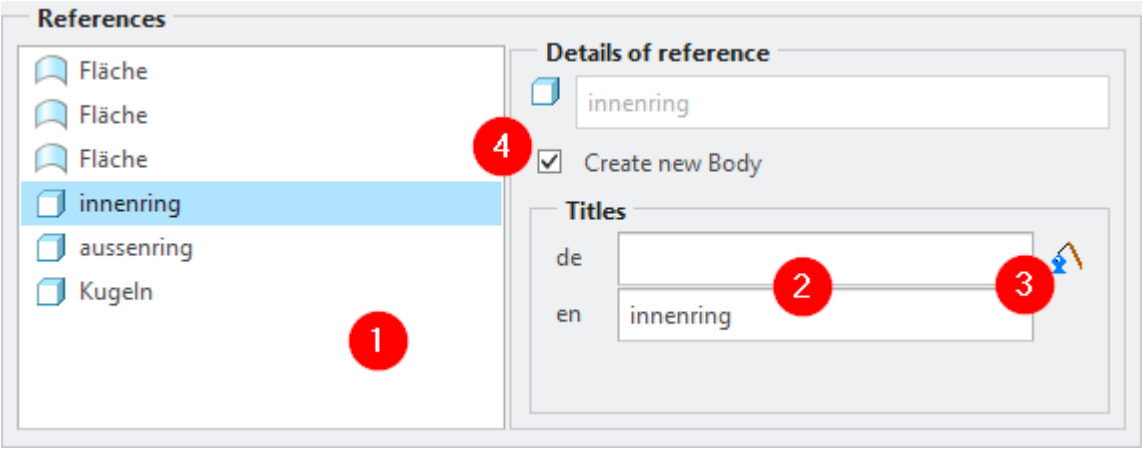
## References

The placement references of a UDF are managed here.

The references of a UDF are displayed on the left. This references cannot be edited, since UDFs are not changed by the editor.

If references in a UDF are named in the `English text / translation text` format (with spaces before and after the slash), the reference titles are automatically filled in.

Reference details are displayed on the right. Here, localized titles of the references are defined for the UDF form. Select a reference and enter the language-dependent text directly in the input fields. Click the flag icon to edit the localized titles, alternatively.




*UDF references*

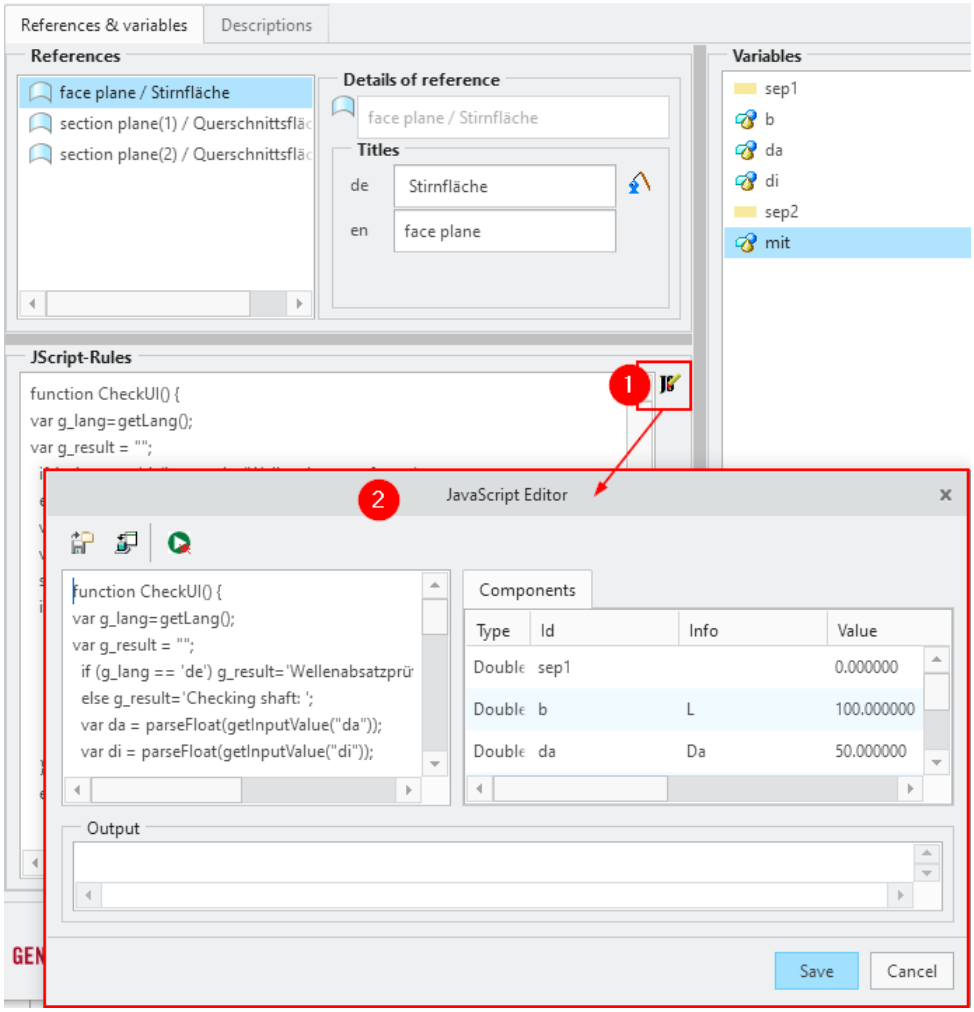
1. Select a reference
2. Enter the localized reference directly or
3. Select a standard text via (Select default text[678])
4. For body references only, a pre-selection can be made whether to create a new body.

## JavaScript in UDF Forms

In the section *JScript Rules* in the tab *References and Variables*, executable JavaScript code for a UDF form is stored.

Enter JavaScript directly into the input field or use the *JavaScript Editor*[784]. Click on the JavaScript icon to open the editor. Go here[788] for an explanation of the JavaScript functions and short examples.


© 2026 INNEO Solutions GmbH




*Javascript symbol (1) opens the editor (2)*

**Tip**: Use JavaScript to adjust the visibility and background color of separators.

# Time of execution

Javascript functions can be executed at different times. You can add as many as required to your JavaScript code.

| Time of execution | Function |
|---|---|
| After loading a UDF group | PostLoad |
| After loading a value table (CSV/XML file) | PostLoadFromFile |
| Before saving a UDF group | PreSave |
| After saving a UDF group | PostSave |


© 2026 INNEO Solutions GmbH

| Time of execution | Function |
|---|---|
| After clicking the button *Check values* in the UDF Forms dialog | CheckUI |
| After changing a variable value | OnChange – is activated in the variables details[623] area |

The names of *PostLoad, PostLoadFromFile, PreSave* and *CheckUI* are fixed. Functions of the type OnChange can have any name.

**Please note:** Use the JavaScript function *creoMapkeyAddToStack* only as a PostSave function, because it executes mapkeys and mapkeys usually close windows.

# Variables

The *Variables* section manages the available variables for a UDF. UDF parameters as well as dimensions can be variables.

**Please note**: Parameters from a UDF are also displayed under Variables. If parameters are to be controlled with UDF Forms, they must be created at the first feature of the UDF (GPH file). The parameters are written to the UDF group in the model later.

## Creating variables and separators

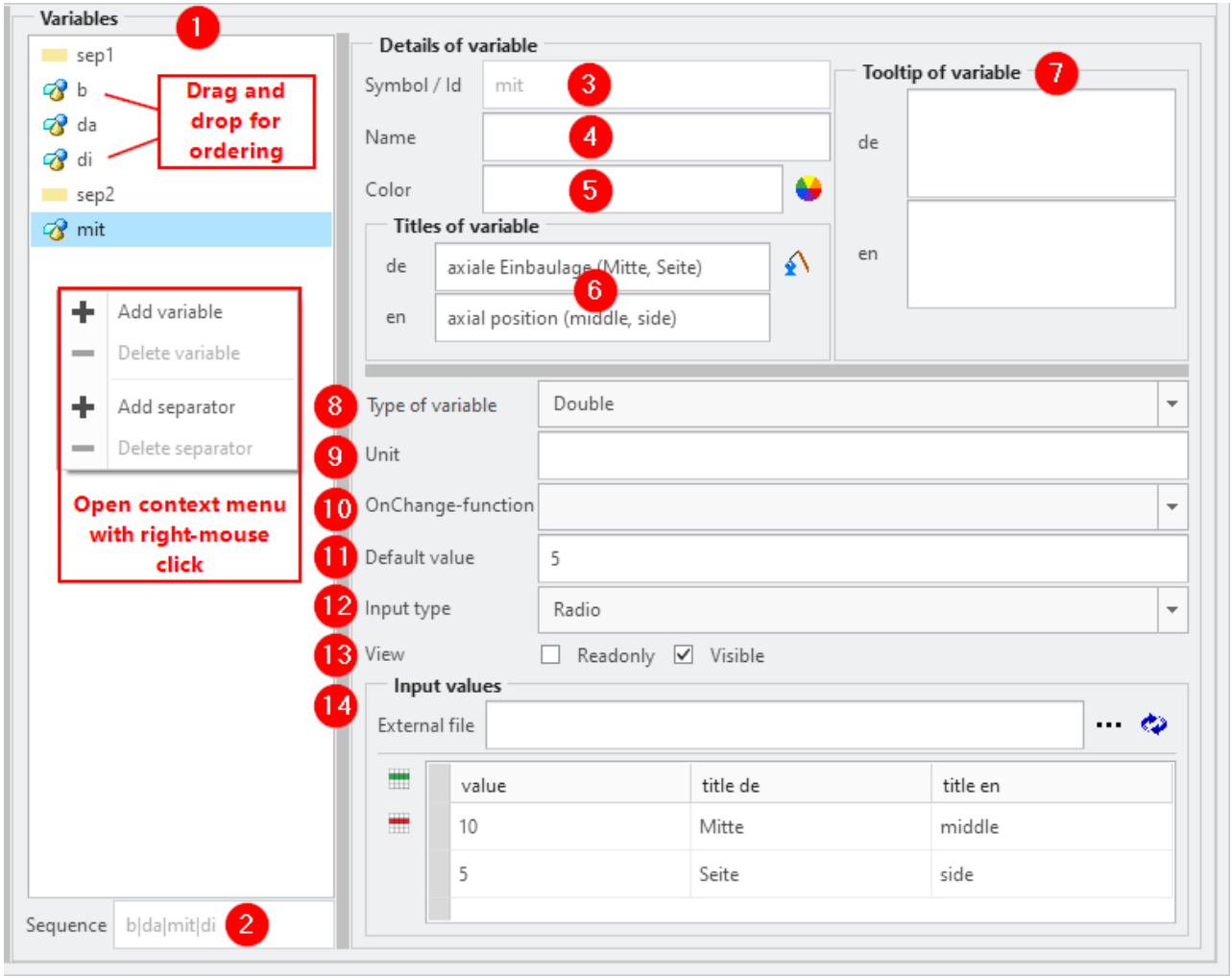
In the variable list (1) on the left, two types of objects are created: Variables and Separators.

– Use separators to logically organize the variables in the UDF Form, for example as section headings.

– Variables come directly from the UDF or are first defined in the UDF Form. Variables are needed to customize the UDF using the form area, but can also be used in JavaScript as auxiliary variables for calculation.

You can use drag-and-drop to reorder the variables. The original variable sequence of the GPH file is displayed below the variable list in the Sequence field (2).


© 2026 INNEO Solutions GmbH




*Variables area in the References & Variables tab*

## Variable details

Click on a variable to display its properties in the details area.

**3. Symbol**: The internal name of a variable or separator. The symbol can be changed for manually created objects.

**4. Name**: Defines the name of a variable. The name is displayed in the third column of the form.

**5. Color**: Defines the background color of the UI element. The color can be selected via the color dialog[796].

**6. Titel:** Defines the title of a variable or separator. The title is displayed in the fourth column of a UDF form. Standard texts can be selected in the default text dialog[678].

**7. Tooltip**: Defines localized tooltips. These are displayed when the cursor is placed over the variable or separator in UDF form.

**8. Type of variable**: Defines the data type of a variable. Text, Integer, Boolean and Double are available.


© 2026 INNEO Solutions GmbH

**9. Unit**: Defines the unit of a variable.

**10. OnChange-Function:** OnChange functions are executed when the value of a form element is changed automatically (e. g. by value tables) or manually by leaving the input field or pressing Enter. The available options are CheckUI, PreSave, CheckThickness, see Explanations and examples[788] .

**Please note:** Avoid endless loops through functions.

**11. Default value**: The value that is displayed by default in the input field of the form.

Defines the value of a variable that is entered into the mask when the surface is reloaded and updated. This value is also entered for value tables. Behind a value of a value table there is always a set of variables with one value each. When saving the UDF definition, the system checks whether the default value of the value table matches the default values of the variables of the associated record. Instances are not taken into account!

**12. Input type**: Defines the type of field for a variable: Text field, Selection, Radio (radio button), Value table and Extended Selection, see Selection fields and value tables[623] . Boolean parameter will automatically be displayed as a checkbox.

**13. View:** Set a variable that users should not be allowed to change to *Readonly* and to *Visible* if the variable is t o be displayed to users.

**14. Input values**: Define all possible values for the UDF Form or import values from an external file, see Selection fields and value tables[623] . This area appears only for the input types Selection, Radio, Value Table and Extended Selection.

# Selection fields and value tables

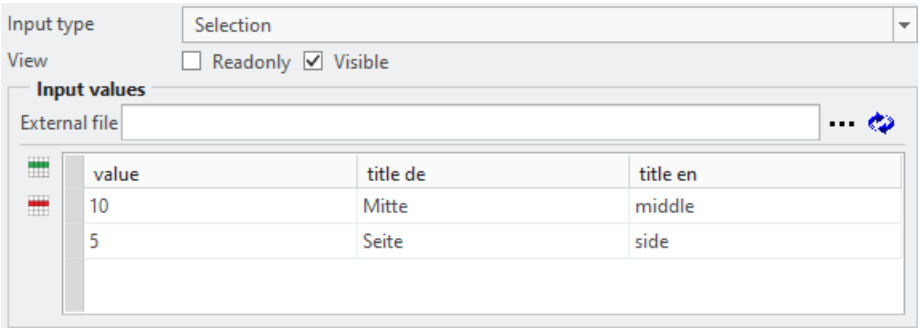
In the drop down menu Input type in the Variables area of *UDF Forms Editor* you can select different field types in forms:

Text field, selection, radio (option field), value table and extended selection field. With the exception of text fields, the values to be available for selection in the UDF form are specified here.

## Selection fields and radio buttons

Selection fields and radio buttons are controlled by a table, whose values are defined here or can be imported from an external file.


© 2026 INNEO Solutions GmbH




**Manual entry**

– Use the buttons on the left to add (green) and remove (red) rows.

**Import values from external file**

– The values from the CSV file are saved in the UDF form.

– When the form is opened, the software checks for the linked CSV file and adopts any updates.

– Select the required file in the field *External file* with the button (...). By default the file explorer opens in the folder *%GT_RESOURCE_FOLDER%udf_forms\gt_data* by default. You can define another folder with the option `gtuf_external_data_folder` from version 8.0.2. on.

– If you want to upload changes to a CSV file while working in the UDF Forms Editor, click the refresh button *(Updates the value table)*.

**Create external file**

The CSV file has to contain the selection values in the first column and the labels in the other columns. The header cell of the first column is ignored. The header cells of the other columns have to contain the two-digit codes for the corresponding languages.

| | A | B | C |
|---|---|---|---|
| 1 | | de | en |
| 2 | 10 | Mitte | middle |
| 3 | 5 | Seite | side |

The CSV file can be created in the system folder *gt_data* or another folder (`gtuf_external_data_folder`). Up to version 8.0.2. the CSV file is searched in the working directory.

**Reading the external file**

As of version 8.0.2. there are several ways to link an external CSV file for selection fields.

1. Filename with extension, no path: the file must be located in the folder defined in the `gtuf_external_data_folder` configuration option. Default is *%GT_RESOURCE_FOLDER %udf_forms\gt_data*.
2. Relative to the folder defined in the `gtuf_external_data_folder` configuration option. Use this if the CSV file should be located in another folder, e. g. together with XML files.

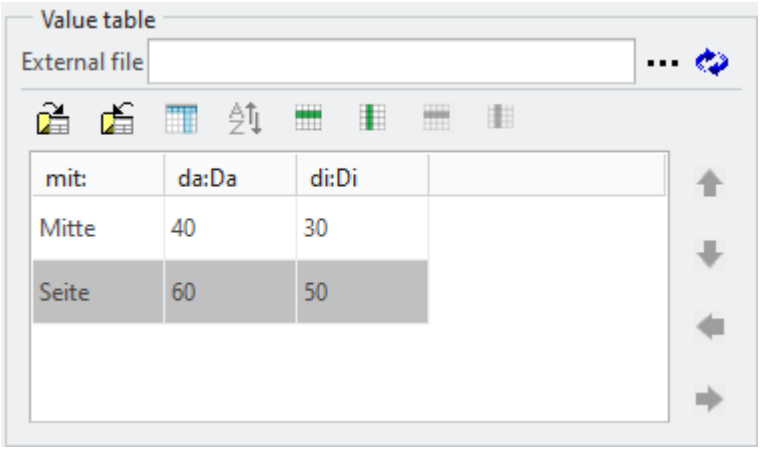

© 2026 INNEO Solutions GmbH

Example: the files *din13.csv* and *din13.xml* are located in *%GT_RESOURCE_FOLDER% udf_forms\gt_definition\company*. The specification of the external file is then .. *\gt_definition\company*

3. Absolute path: this should not be used, since changes cannot be transferred.

## Value tables

Value tables control several variables simultaneously. Each table line is a single variant. Use the arrow buttons on the right to move rows or columns.




*Use the arrow buttons on the right to move rows and columns in the table.*

– The values from the CSV file are saved in the UDF form.

– When the form is opened, the software checks for the linked CSV file and adopts any updates.

– Select the required file in the field *External file* with the button (...). By default the file explorer opens in the folder *%GT_RESOURCE_FOLDER%udf_forms\gt_data* by default. You can define another folder with the option `gtuf_external_data_folder` from version 8.0.2. on.

– If you want to upload changes to a CSV file while working in the UDF Forms Editor, click the refresh button *(Updates the value table)*.

**Create external file**

To create the header rows for a CSV file, first create the required columns in the value table. Then, export the table as a CSV file. The exported file contains the required table header.

| | A | B | C |
|---|---|---|---|
| 1 | mit: | da:Da | di:Di |
| 2 | mit | da | di |
| 3 | Mitte | 40 | 30 |
| 4 | Seite | 60 | 50 |


© 2026 INNEO Solutions GmbH

**Read external file**

There are three ways to specify an external CSV file, see section under Selection and option fields[624].

**Editing value tables**

The following buttons are available for editing.

| Icon | Name | Description |
|---|---|---|
|  | Import value table from file | Imports a value table from the following file formats:<br>– xls (Excel 97-2003)<br>– xslx (Excel 2003-2016)<br>– csv (Comma-separated values)<br>– txt (text files with UTF-16LE/Unicode formatting) |
|  | Export value table to file | Exports a value table to the formats:<br>– xls (Excel 97-2003)<br>– xslx (Excel 2003-2016)<br>– csv (Comma-separated values) |
|  | Edit value table in spreadsheet software | The value table is opened in a spreadsheet program (depending on the client computer) and can be edited. Then it is reimported. |
|  | Sort selected column | Sorts the rows by the active column. Three sorting orders are available:<br>– 0-Z<br>– Z-0<br>– initial |
|  | Add/Remove row | Adds a new line under the current cursor position. To remove, select a row and click *Remove row*. |
|  | Add/Remove column | Adds a new column to the right of the current cursor position. To remove, |


© 2026 INNEO Solutions GmbH

| Icon | Name | Description |
|---|---|---|
| | | select a column and click *Remove column*. |

**Please note**: Any changes to a value table made when editing the table in a spreadsheet software (deleting, moving or editing rows or columns) are adopted as-is in GENIUS TOOLS UDF Forms.

## 23.3.1.5 Description tab

In the Description tab, general information for a UDF form is stored.

**Titles**
Define the localized names of the UDF. They will be displayed in the UDF Form. Standard texts can be used via the button (Description of the standard text selection dialog678).

**Help documents**
Add language-dependent documents here. Enter the file path directly into an input field or select a file using the file selection.

Help documents have to be accessible and openable on client computers! The applications to use are specified by the client computer.

Supported path formats:

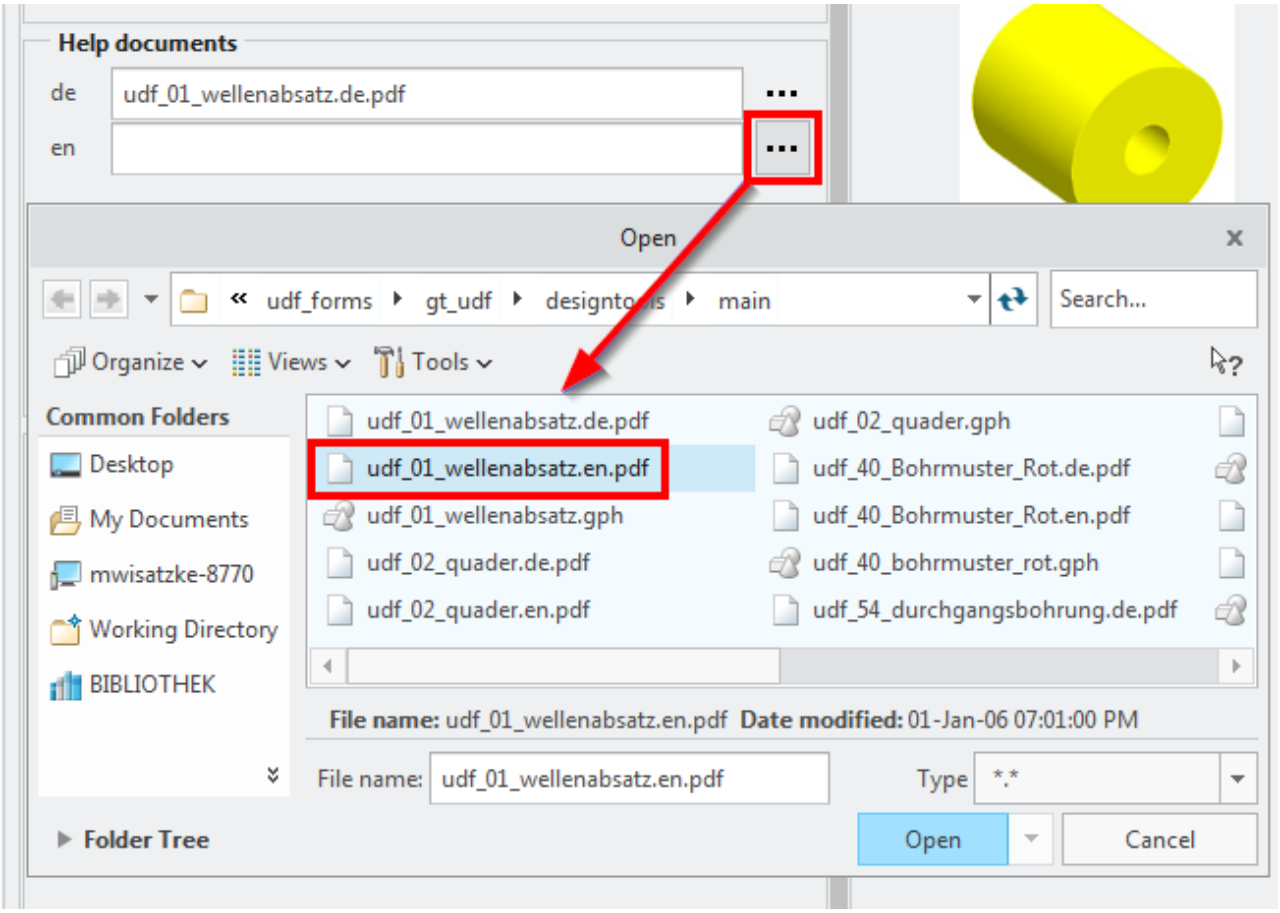
– absolute paths

– UNC paths

– HTTP paths

– If you enter the file name only, the help document is searched in the group directory and in the directory configured as `gtuf_data_folder`.


© 2026 INNEO Solutions GmbH




*Open the file selection and select a help document*

**Descriptions**
Enter a localized description of the UDF here. The description is displayed in the UDF Form dialog next to the placement references, depending on the Creo language.

**Images**
Preview images are configured in this area. Since images can contain localized information, they can be stored individually for each language.

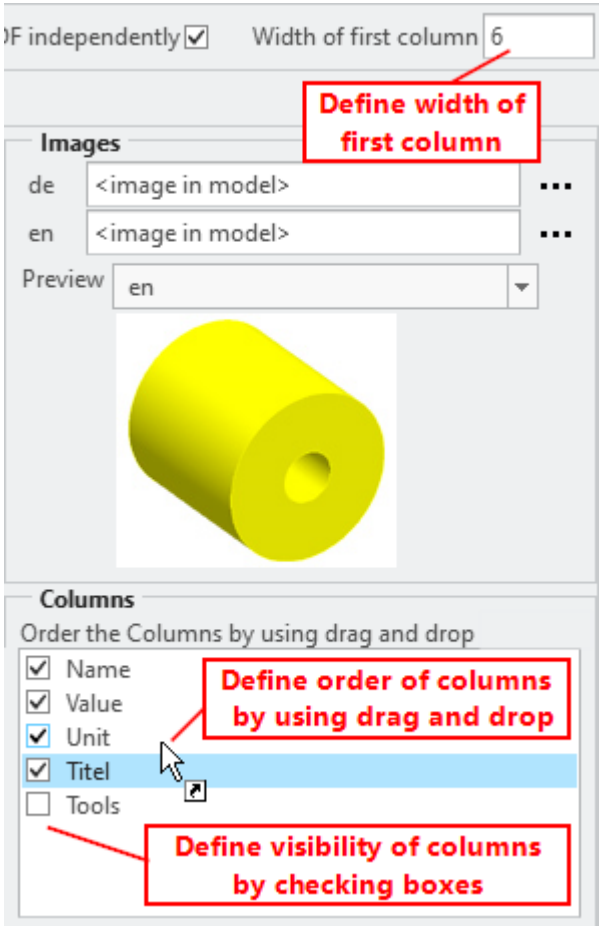
Enter the file path directly into an input field or select a file using the file selection.

Use the drop-down list *Preview* to switch between the localized preview images.

**Order of columns**
In UDF Forms Editor you can arrange the columns in the *Variables* area of a UDF Form by moving them per drag-and-drop and hide them by unchecking the boxes.

For the first column you can define the width in the upper Details area[618].


© 2026 INNEO Solutions GmbH




The Tools column displays value tables[625] and Javascript functions[619], if available.

## 23.3.2 Updating UDF definitions

The GPH file on which a UDF is based is being modified several times during the development process.

Creo and therefore also *UDF Forms* always uses the current GPH file.

The UDF definition used by the module *UDF Forms* is read from the directory and then no longer matches the GPH.

If a new UDF definition is created for the current GPH, the descriptions, help documents, JavaScript code, etc. are no longer included in the definition.

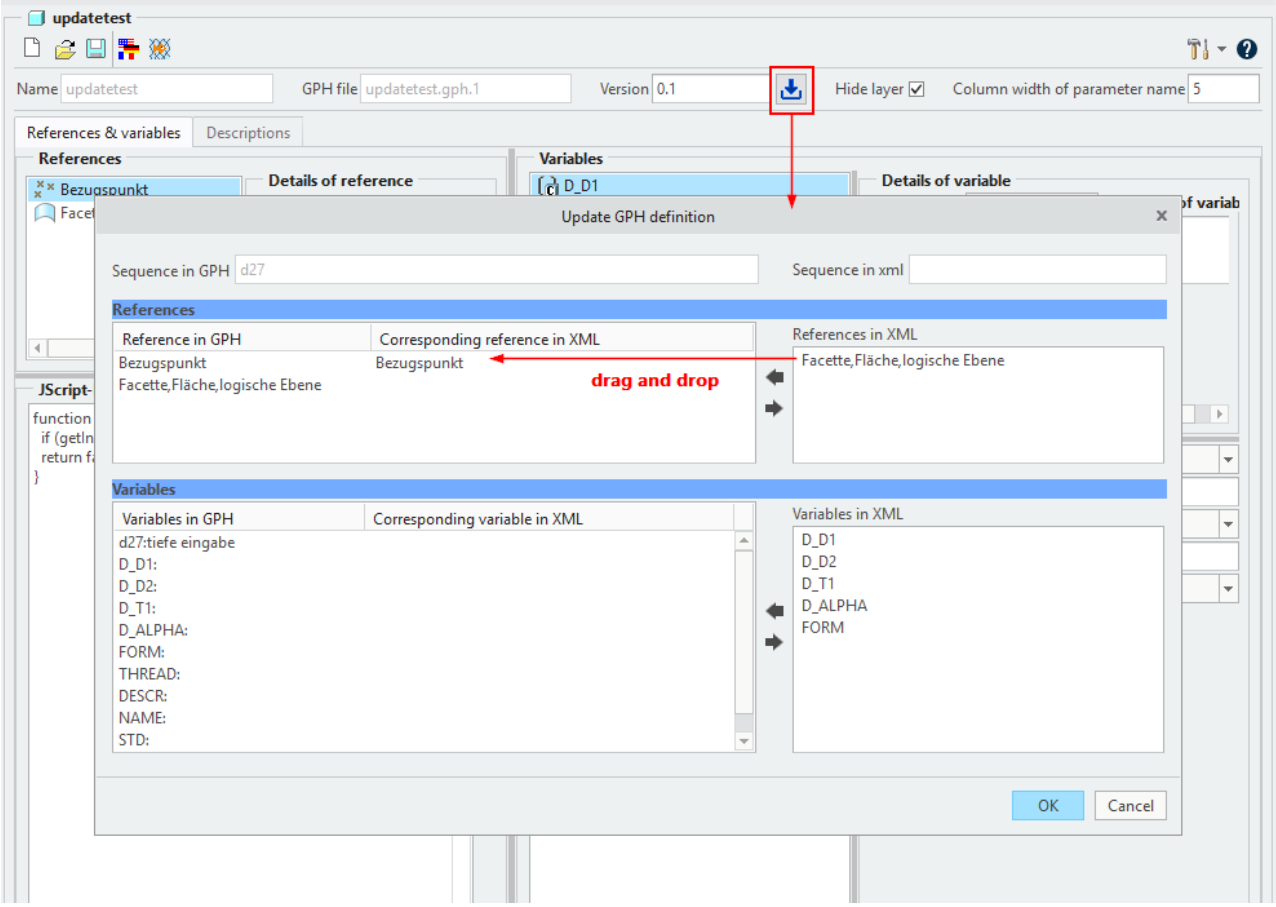
To solve the problem, there is now an update mechanism in *UDF Forms Editor*.

In this context, a version is stored in the UDF definition. When creating the UDF, the parameter `UDF_VERSION` is added to the created group. Several UDF definitions of the same name are now also supported in the model.

**Warning**: GPH files of older versions must not be in the session during the update process!


© 2026 INNEO Solutions GmbH

The update mechanism is started in the editor:

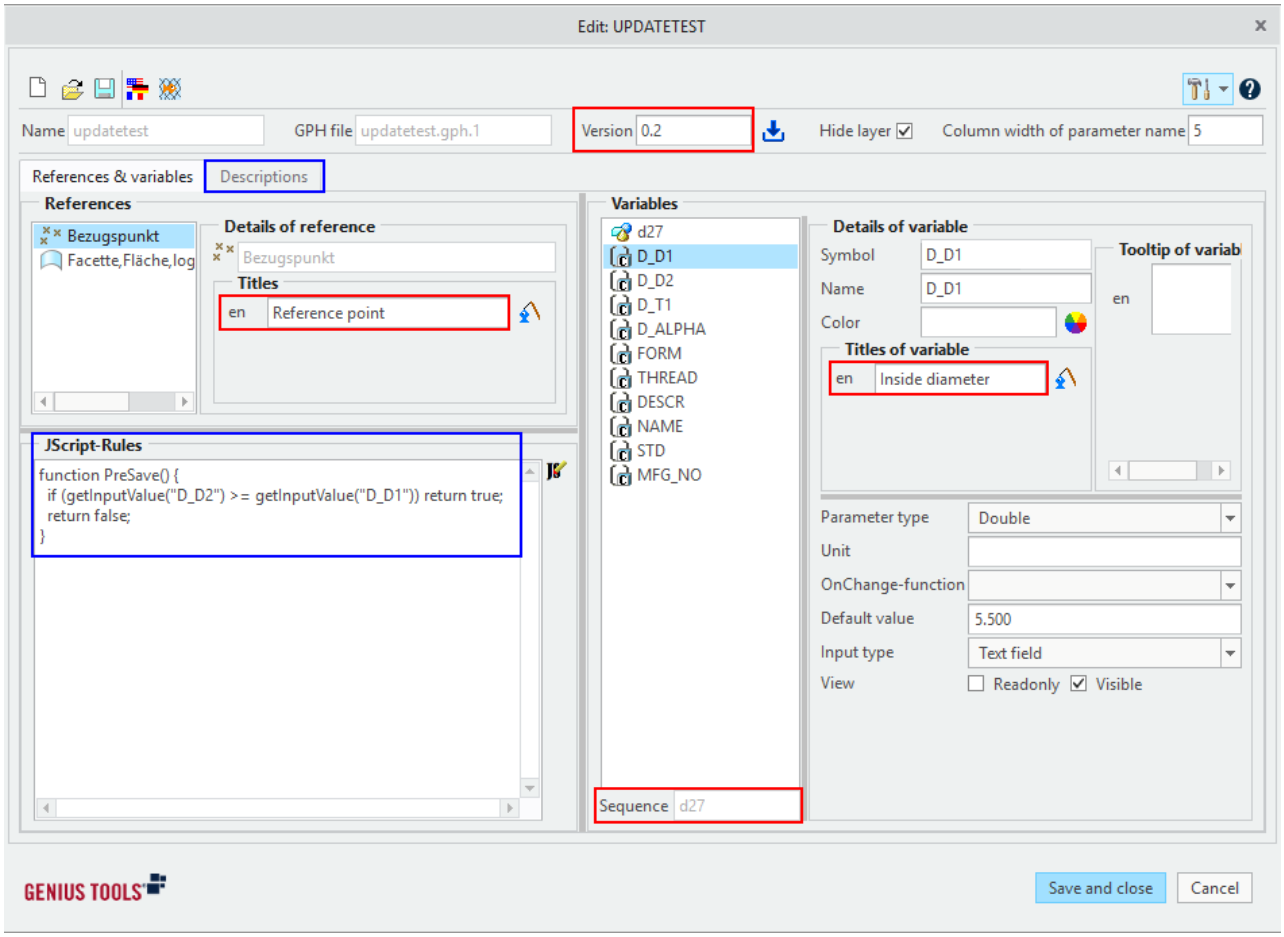



*Update Dialog*

In the update dialog, the individual references and variables of the current GPH are now compared with the references and variables of the UDF definition. The program creates an automatic assignment. Further assignments can be created using DragAndDrop. The assignment can also be done with the arrow keys. The table cells of both areas must be selected.

After saving, the version of the UDF definition is automatically incremented by 1, references that are no longer needed are removed and former variables from the sequence are also removed. Descriptions, help, JavaScript code etc. are retained.


© 2026 INNEO Solutions GmbH

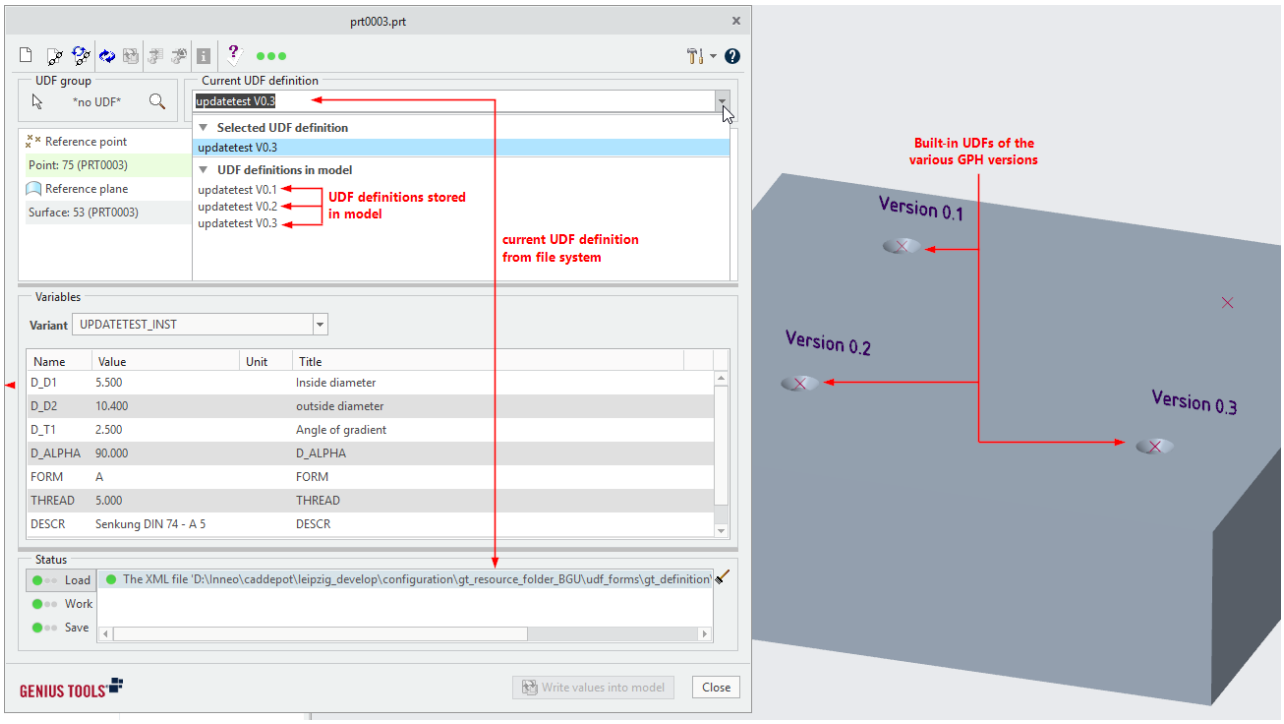



*Result of the update*

The blue areas show the adjusted or accepted values and the green areas show the values retained.


© 2026 INNEO Solutions GmbH

# Model with UDFs in three different versions




After the update, UDF groups of older versions can also be edited with the corresponding UDF definitions.

This way, dimensions and parameters can be adjusted. The current GPH is always used for exchanging instances or creating a new UDF group.


© 2026 INNEO Solutions GmbH

# 24 Utilities

*Utilities* consist of several individual applications. The following applications are included in *Utilities* and can be used in different modes:

| Function | Mode | | | |
|---|---|---|---|---|
| | Creo Standby | Assembly | Part | Drawing |
| 3D Note Form 635 | | ✓ | ✓ | |
| Annotation Info 642 | | ✓ | ✓ | |
| Annotation Transfer 649 | | ✓ | ✓ | |
| Close All Other Windows 652 | ✓ | ✓ | ✓ | ✓ |
| Combined View Gallery 653 | | ✓ | ✓ | |
| Command Control 655 | ✓ | ✓ | ✓ | ✓ |
| Component Parameter 656 | | ✓ | | |
| Convert Materials 663 | ✓ | | | |
| Copy Component Parameter To Substitution Component Parameter 663 | | ✓ | | |
| Create point and coordinate systems 671 | | ✓ | ✓ | |
| Create Search.pro 665 | ✓ | | | |
| Create Tolerance Table 669 | | | | ✓ |
| CS Assembler 673 | | ✓ | | |
| Default Text Editor 676 | ✓ | | | |
| Export Points 679 | | ✓ | ✓ | |
| Export Table 1:1 to Excel 686 | | ✓ | | ✓ |


© 2026 INNEO Solutions GmbH

| Function | Mode | | | |
|---|---|---|---|---|
| | Creo Standby | Assembly | Part | Drawing |
| Export Table to CSV[687] | | | | ✓ |
| Export Table to Excel[689] | | ✓ | | ✓ |
| Extend Relations[702] | | ✓ | ✓ | |
| Extended Dimensions Functions[707] | | ✓ | ✓ | |
| Find Contact Surfaces[709] | | ✓ | | |
| Find References[710] | | | ✓ | |
| Full Backup[712] | ✓ | ✓ | ✓ | ✓ |
| GenTOL References[717] | | ✓ | ✓ | |
| GTOL Text[719] | | ✓ | ✓ | |
| Inspect Instance Note[727] | | ✓ | ✓ | |
| Load Save Converter[730] | ✓ | | | |
| Open Base Model[731] | | Features with reference models | | |
| Open / Create Drawing[732] | | ✓ | ✓ | |
| Select Contact Surfaces[734] | | | ✓ | |
| Select Surfaces by Color[737] | | | ✓ | |
| Set Component Combined View[738] | | ✓ | | |
| Show Information[741] | ✓ | | | |
| Show Pitch[742] | | ✓ | ✓ | |
| Show Thread Size[745] | | | ✓ | |
| Sort Combined Views[747] | | ✓ | ✓ | |


© 2026 INNEO Solutions GmbH

| Function | Mode | | | |
|---|---|---|---|---|
| | Creo Standby | Assembly | Part | Drawing |
| Toggle Symbol Variants[749] | | | | ✓ |
| Work Dir Manager[751] | ✓ | ✓ | ✓ | ✓ |

Also refer to the chapter Configuration of GENIUS TOOLS for Creo[799].

**Please note:** Many functions are only available with subscription licenses for GENIUS TOOLS for Creo. For a list, see License-dependent functions[26].

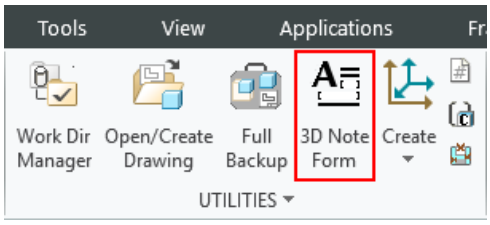
# 24.1 3D Note Form

Available in the ribbon menu GENIUS TOOLS

The component *3D Note Form* allows you to modify dimension and parameter values of all 3D notes of a model quickly and comfortably in one dialog box, i. e. without having to open each 3D note separately.

## Starting the program: in assembly mode

*3D Note Form* can only be started in assembly mode, or if notes exist in a part. To start *3D Note Form*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

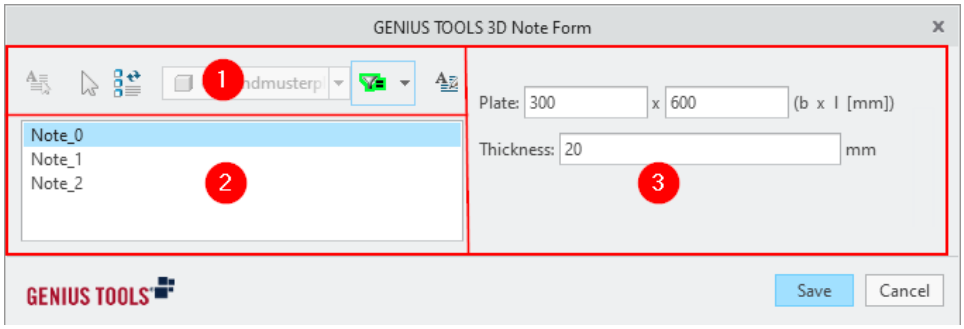

© 2026 INNEO Solutions GmbH

## Configuration of the ribbon menu

Use the configuration option `gtu_start_3d_note_form` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

### 24.1.1 User interface

The user interface of the *3D Note Form* consists of the following areas:



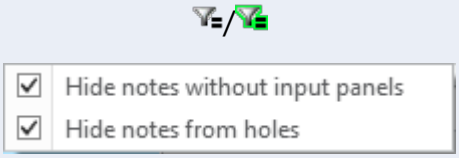

1. Command bar
   The command bar includes the selection list. In assembly mode this drop-down menu displays the most previously selected assemblies and parts.
2. Notes list
3. Current note

## Command bar

The following buttons are included in the command bar:

| Icon | Name | Description |
|---|---|---|
|  | Note selection | Allows to select notes directly in the Creo main window. |
|  | Model selection | Enables direct selection of models in the model tree or in the graphics window.<br>**Note:** The function is only available in assembly mode! |


*© 2026 INNEO Solutions GmbH*

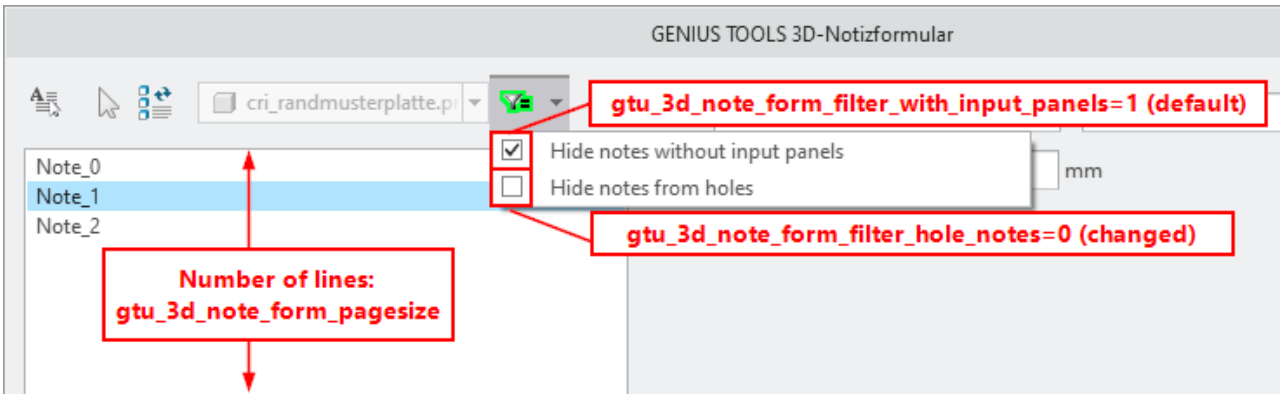
| Icon | Name | Description |
|---|---|---|
|  | Save and regenerate | Saves the current note and regenerates the model with the current values. |
|  | Filter | Hides notes with input panels and/or notes from holes (green) or displays them. |
|  | Text editor | Opens and closes the dialog for text editing of a note. |

## Configuration options for the dialog box

**gtu_3d_note_form_filter_hole_notes**
Defines whether notes on holes are hidden when the dialog box is opened (1, filter is enabled) or not (0). Default value: 1

**gtu_3d_note_form_filter_with_input_panels**
Defines whether notes without input panels are hidden when the dialog box is opened (1, filter is enabled) or not (0). Default value: 1

**gtu_3d_note_form_pagesize**
Defines the number of lines shown per page. Default: 15





© 2026 INNEO Solutions GmbH

## 24.1.2 Displaying and editing notes

Notes are displayed with static and dynamic content. Static contents of a note are texts. Dynamic contents like parameters or dimensions are displayed as editable input fields. If you use static texts, they can be changed in the *3D Note Form* *text editor*[637] .

Enter parameters and dimensions into notes in following notation to make them editable with *3D Note Form*:
Parameter: `&ParameterName`
Dimensions: `&DimensionName`

Modify the values in the input fields as desired. After saving and regenerating the new values are transferred to the model.

**Please note**: If notes are longer than 15 lines, they are displayed in page mode. In this case, a note is split up and two buttons to switch pages are displayed.

### Notes with calculations

The *3D Note Form* supports arithmetic operations and mathematical functions.

They differ in notation from default arithmetic operations in Creo. To use an arithmetic operation the expression must begin with the equals sign (=).

**Example**: "=12-d4" or "=Math.pow(d2,3)"

**Please note**: Dimension units will be ignored in calculations, only the values are used.

| Mathematical function | Description | Input example |
|---|---|---|
| + | Addition | =d27+5 |
| - | Subtraction | =12-d4 |
| / | Division | =d42/23 |
| * | Multiplication | =d31*3 |
| Math.sqrt(x) | Square root | =Math.sqrt(9) |
| Math.pow(x,y) | Power (x to the power of y) | =Math.pow(d2,3) |
| Math.abs(x) | Absolute value | =Math.abs(-5) |


© 2026 INNEO Solutions GmbH

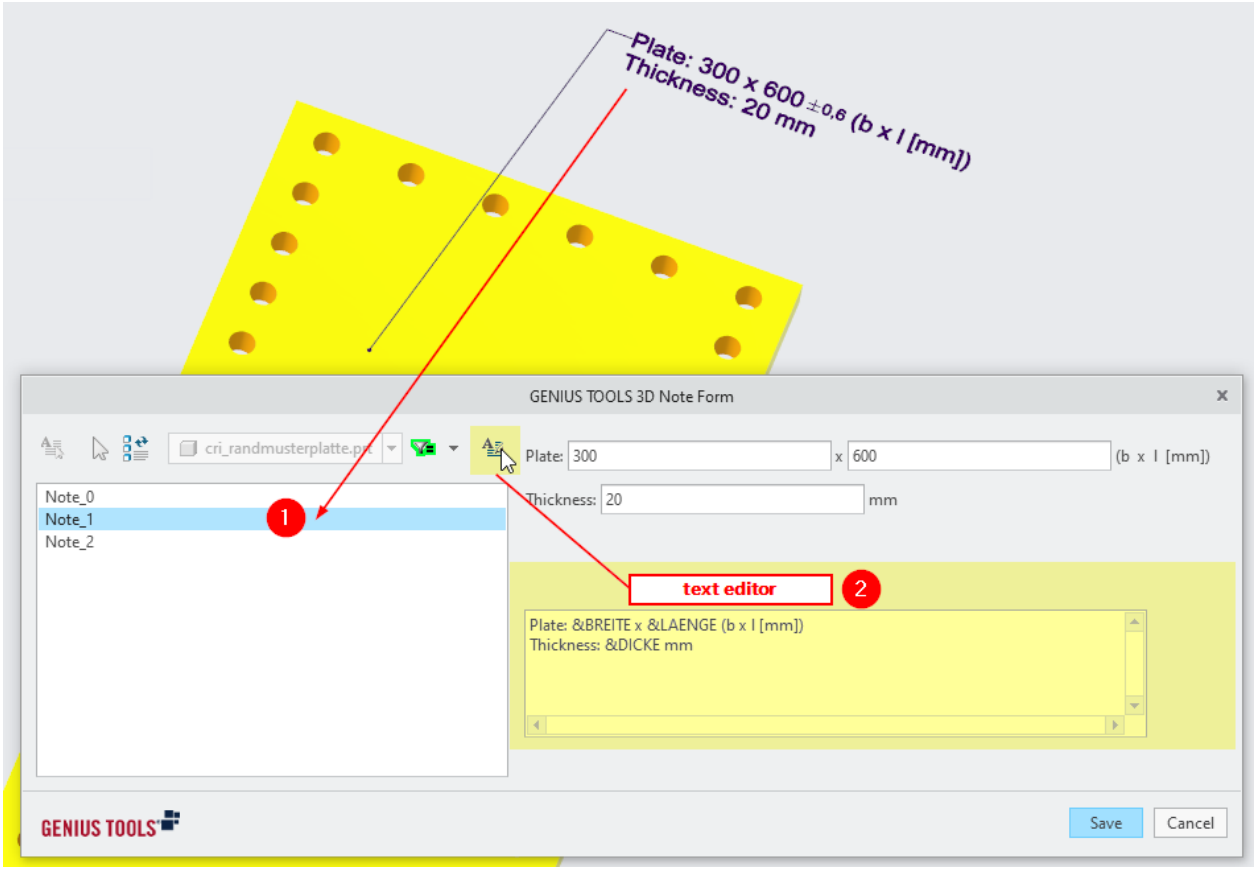
| Mathematical function | Description | Input example |
|---|---|---|
| Math.round(x) | Commercial rounding | =Math.round(2.565) |
| Math.ceil(x) | Round up on next integer | =Math.ceil(3.6) |
| Math.floor(x) | Round down on next integer | =Math.floor(3.4) |

**Please note**: Calculation formulas will not be saved.

## 24.1.3 Usage example

This example illustrates working with notes in the 3D Note Form with static text, dynamic content and calculations. Various notes are stored in this part.

1. Open the *3D Note Form*. Select the note *Note_1*.
2. Go to the text editor.




*Display of the values of all 3D notes in the note form*

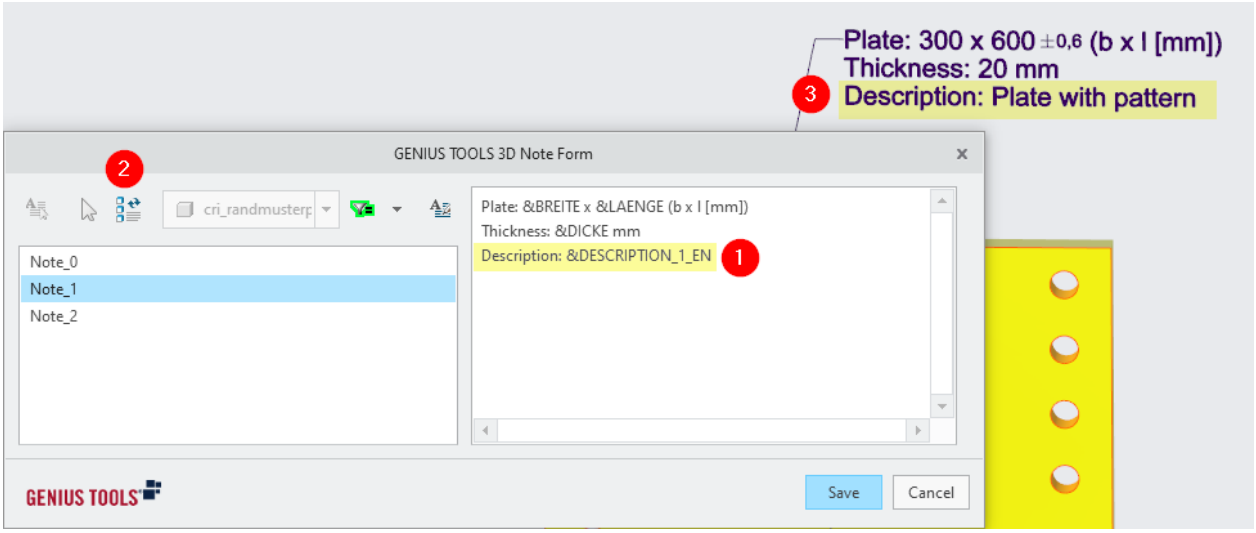

© 2026 INNEO Solutions GmbH

# Editing a note

In the text editor you will see the breakdown of the displayed note according to this scheme:

| Note | Text deposited | Content of the note |
|---|---|---|
| Plate: | &WIDTH x &LENGTH (w x l [mm]) | 300 x 600 (w x l [mm]) |
| Thickness: | &THICKNESS mm | 20 mm |

1. Add a new line in the text editor:
   `Description: &DESCRIPTION_1_EN`
2. Confirm the entry with *Save and Regenerate model*.
3. The result is applied to the note.

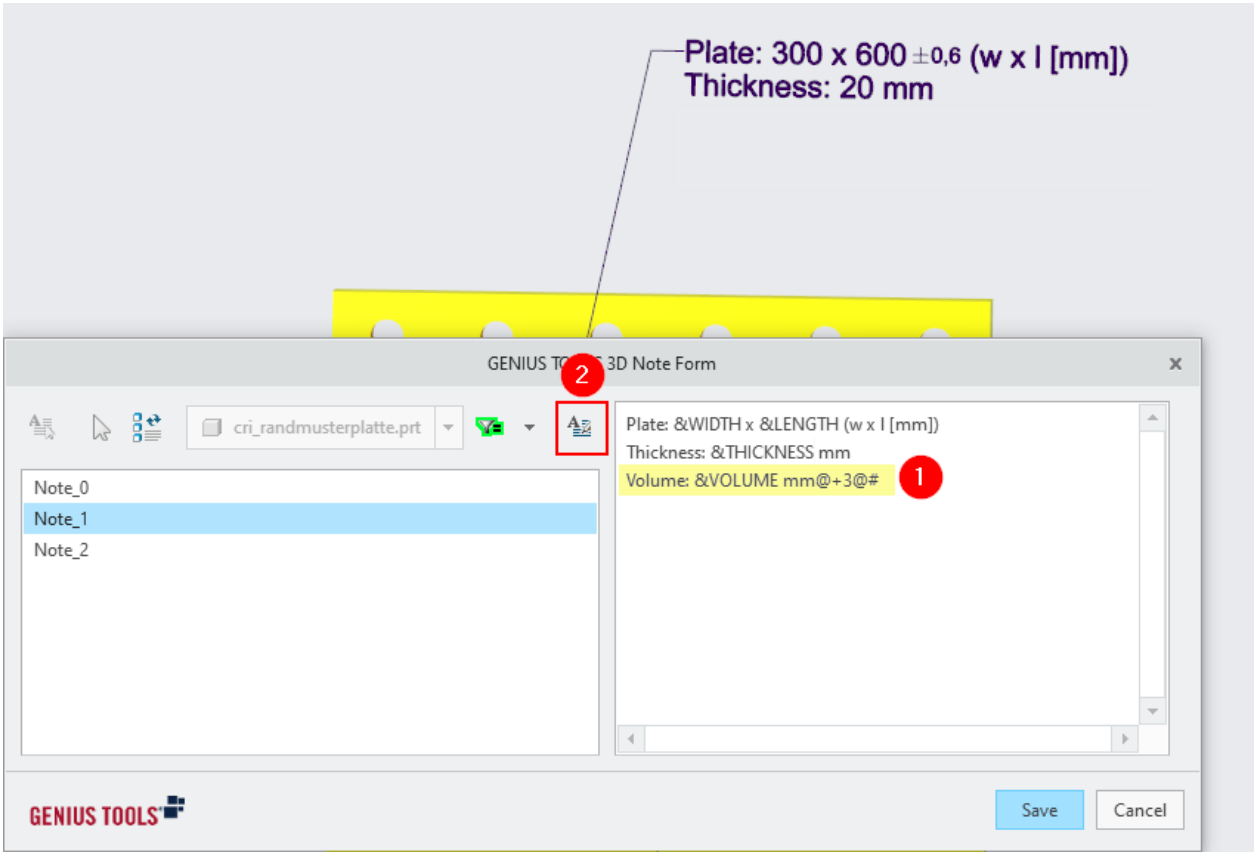



# Defining note content via calculation

The `VOLUME` parameter is needed for the next step. Create this parameter if it does not already exist.

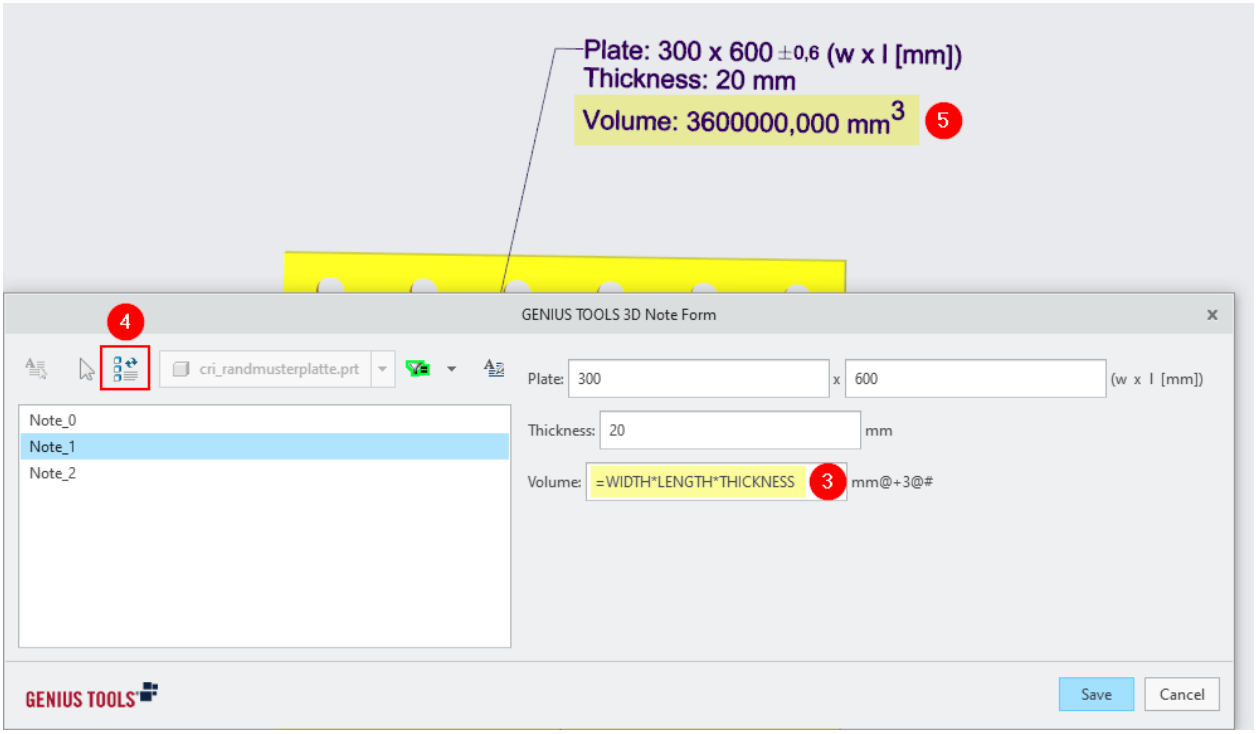
1. Add the following line to the note:
   `Volume: &VOLUME mm@+3@#`
2. Exit the text editor by clicking the text editor button  again. Confirm the saving of the changes made with *Yes*.


© 2026 INNEO Solutions GmbH




3. If you now want to fill this parameter with the 3D note form, enter `=WIDTH*LENGTH*THICKNESS` in the input field.
4. Confirm the input with Save and Regenerate. The volume is automatically calculated and transferred to the parameter.
5. The result is then displayed in the note.


© 2026 INNEO Solutions GmbH




# 24.2 Annotation Info

Available in the ribbon menu GT MBD

The module *Annotation Info* is used to display information about all types of annotation elements in combined views. *MBD Info*[413] is an enhancement to Annotation Info that offers additional features and appears in the navigation pane. This module lists non-visible information to help you find errors and redundancies. Annotation elements can be displayed in parts, assemblies and in multi-body models. It is an information module only and cannot be used to edit annotations.

*Annotation Info* is available in part mode and assembly mode with these features:

1. Viewing occurrence and frequency of annotations in a clear and concise table format
2. Finding annotations and thus dependencies that are not visible in any combined state

Use the following modules to further process the information obtained with *Annotation Info*:

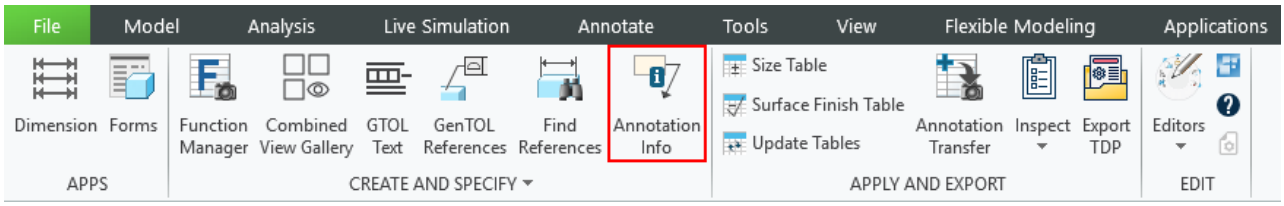
– to manage combined views, see *Function Manager*[167]

– to transfer annotations, see *Annotation Transfer*[649]


© 2026 INNEO Solutions GmbH

# Starting the program: in part and assembly mode

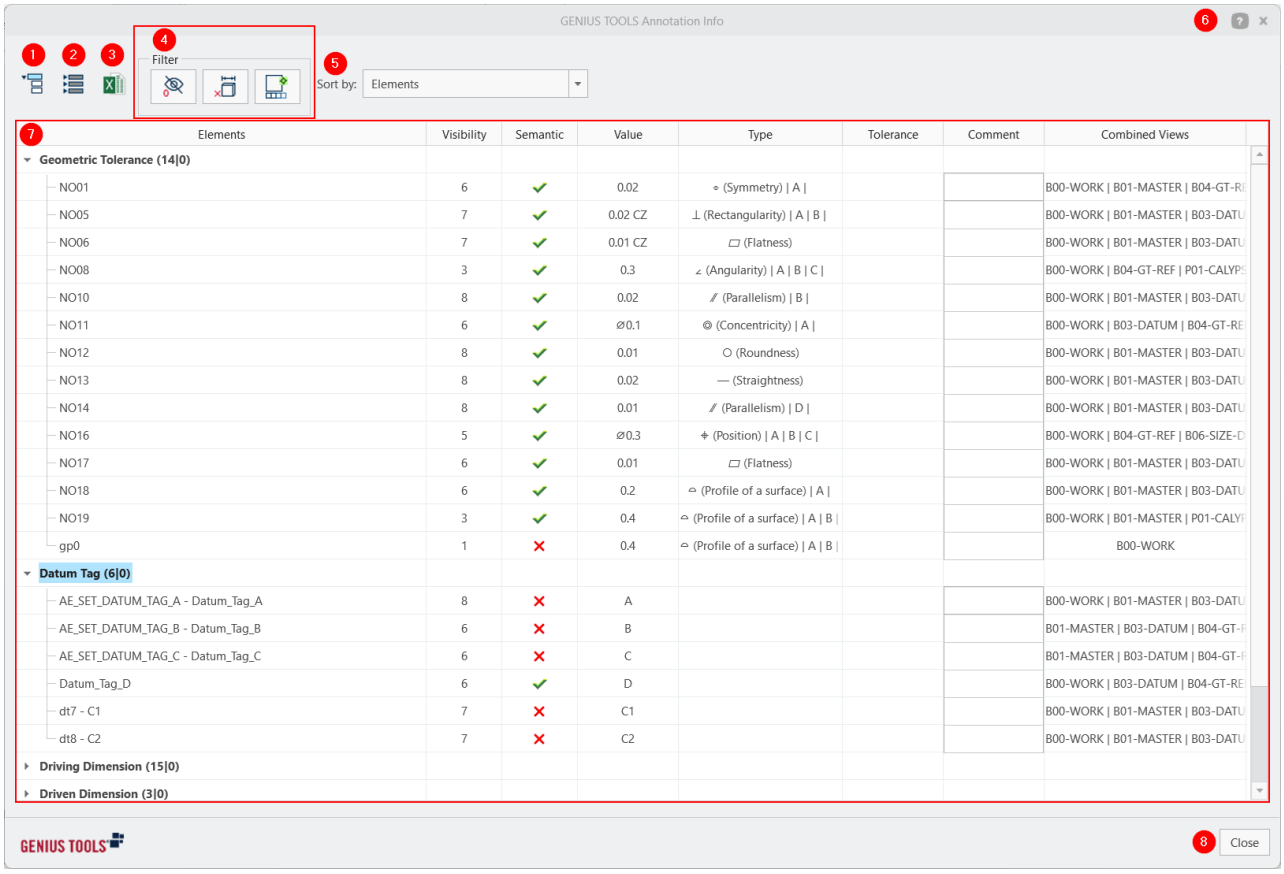
To start *Annotation Info*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




*Starting via the GT MBD in the ribbon menu*

# User interface

When opening the program, you see the following user interface:




1. Expanding the table
2. Collapsing the table
3. Exporting the table, see *Export Table to Excel*[689]

**Please note:** The name of the export template is stored in the configuration option `gtu_annotation_info_excel_template`. Further settings can be made using the configuration options that begin with the following prefix: `gtu_table_to_excel_`


© 2026 INNEO Solutions GmbH

4. **Filter**
   *Show only invisible annotations*
   Existing annotation elements that are not part of a combined view are highlighted in red. Clicking on such an annotation element displays its position in the model.
   *Show only annotations without semantics*
   *Show only annotations from the active combined view*
5. **Set the column by which the table is sorted in ascending order**
   By default, the table is sorted by the column *Items*.
6. **Help**
   Opens this help.
7. **Tabular listing of all annotation elements in a model (across all combined views):**
   Left-click in a row to highlight the corresponding annotation element in the model tree. Right-clicking an annotation opens a sub-menu showing all the combined views in which the annotation is present. Left-click on a combined view to view the model in that combined view.
   *Types of annotation elements:*
   - *Geometric Tolerance*
   - *Datum Tag*
   - *Driven Dimension*
   - *Surface Finish*
   - *Symbol*
   - *Note*
   - *Tables*
8. Close the user interface

# Displaying annotations

By default, the table is sorted by the column *Items*. The following columns are displayed:

**Please note:** The *Comment* column can only be used in Creo version 11 or later.

| Column | Remark | Table can be sorted by this column |
|---|---|---|
| Items<br>e.g.<br>▸ Geometric Tolerance (14\|1) | The display is structured according to the following scheme:<br>`Annotation element type (number of these annotation elements in combined views \|` | ✓ |

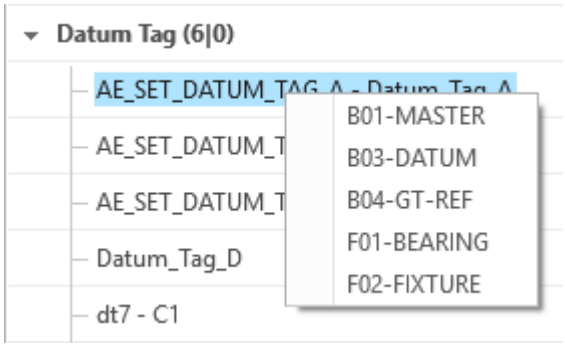

© 2026 INNEO Solutions GmbH

| Column | Remark | Table can be sorted by this column |
|---|---|---|
| | number of these annotation elements not present in any combined view) | |
| Visibility | 0 = invisible<br>≥ = visible | ✗ |
| Semantic | Indicates if the annotation element is linked. If there is no link, this may cause problems in subsequent processes based on that annotation element.<br>✓ = semantic link exists<br>✗ = semantic link does not exist<br>T = TED dimension that is not semantically linked | ✗ |
| Value | | ✓ |
| Type | | ✓ |
| Comment | Additional optional information is stored in an annotation parameter. | ✗ |
| Tolerance | | ✓ |
| Combined Views | Combined view containing the annotation element | (✓: table filtered by number of combined views) |

## Displaying combined views for an annotation

Right-clicking an annotation element opens a submenu with the combined views in which the annotation element is contained. Left-clicking on a combined view switches to that view and displays the annotation element in that view.


© 2026 INNEO Solutions GmbH




## Displaying automatically generated hole notes

The configuration option `gtu_annotation_info_show_hole_notes=1` allows you to display automatically generated hole notes by Creo Parametric. Default: 0=off.

## Displaying automatically generated hole notes

The configuration option `gtu_annotation_info_mandatory_ref_number_for_linear_dimensions` allows you to define the minimum number of semantic references a linear dimension must have in order to be marked as semantically correct in the *Annotation info*. Default: 2

This applies to:

Driving dimensions

Driven dimensions

Reference dimensions

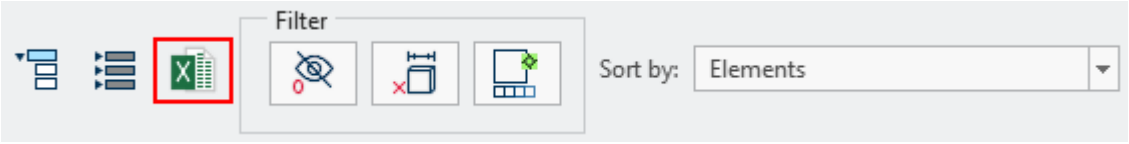
## Configuration of the ribbon menu

Use the configuration option `gtu_start_annotation_info` to switch off the function, so no button is displayed in the *GT MBD* ribbon menu. (Default is 1=On/not hidden)

# 24.2.1 Export to Excel

## Exporting annotation elements to Excel

To export annotation elements to Excel, click on the command *Export to Excel*. The dialog Export table to Excel[690] opens, in which you select the Excel template, export file and configuration.





© 2026 INNEO Solutions GmbH

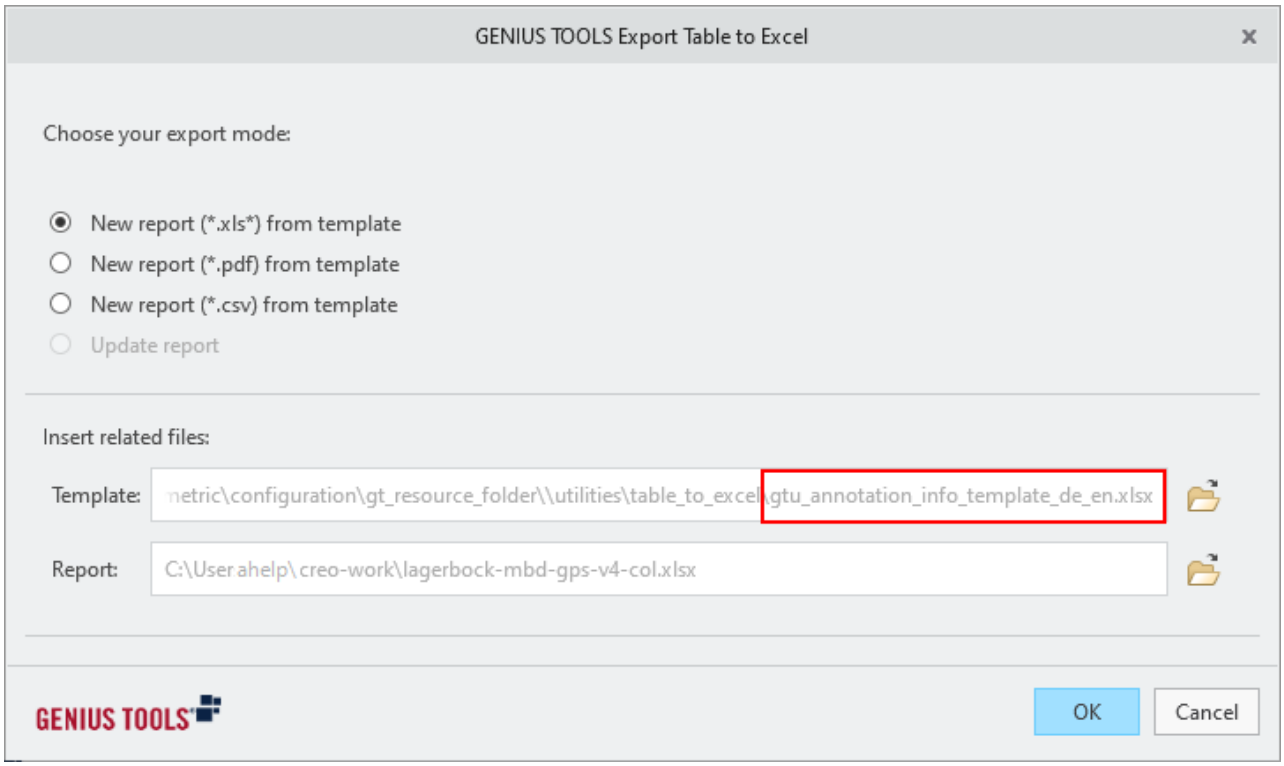
Starting the Excel export can take some time because the Excel template has to be checked. Once the export is complete, the Excel file will open automatically.

**Warning:** Close all Excel windows before starting the export.

## Export dialog

By default, the out-of-the-box template is selected. The file name is generated from the name of the model. You can also design an individual Excel template[648].

The selected configuration determines which annotations are exported. Annotations will be exported if the selected configuration contains the annotation.




## Configuring path and name of the template

The selection of the template and the name of the template are determined by these configuration options:

**gtu_annotation_info_excel_template**
Defines the name of the basic Excel template. Default: *gtu_annotation_info_template_de_en.xlsx*

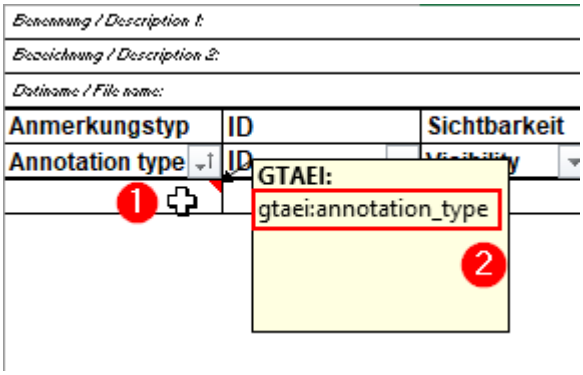
**gtu_table_to_excel_template_folder**
Defines the path to the template directory for "Table to Excel" (.xlsx-files). Default: *%gt_resource_folder%\utilities\table_to_excel*


© 2026 INNEO Solutions GmbH

# Customizing the Excel template

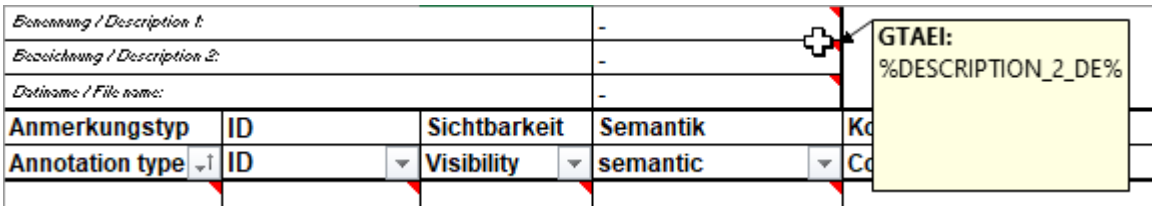
If you want to create your own template, you can use the supplied template *gtu_annotation_info_template_de_en.xlsx* from the directory *%gt_resource_folder%\utilities\table_to_excel* as a basis and add a comment (2) to the first value cell (1) of the desired column. The chapter Create template[693] describes how to set up an export template step by step.

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.




*Comment used for column assignment*

The text of the comment determines which values are to be written into the column. Head parameters allow variables without module abbreviations, e. g. *%DESCRIPTION_1_DE%*.




*Head parameter forAnnotation Info*

The text in the comment must consist of the component abbreviation gtaei: and a keyword:

| Comment text | Column name |
|---|---|
| gtaei:<columnName> | Output of additional user-defined parameters |
| gtaei:annotation_type | Annotation type |
| gtaei:combined_views | Combined views |
| gtaei:semantic | Semantic |
| gtaei:tolerance_value | Tolerance value |
| gtaei:type | Type |


© 2026 INNEO Solutions GmbH

| Comment text | Column name |
|---|---|
| gtaei:value | Value |
| gtaei:visibility | Visibility |
| gtaei:comment | Comment |

## Creating a template with multiple spreadsheets

You can use other module acronyms besides *gtaei*. This allows you, for example, to export data from *MBD Info* and *Inspect 3D* together, see Export data from several GENIUS-TOOLS-components[701] for an example.

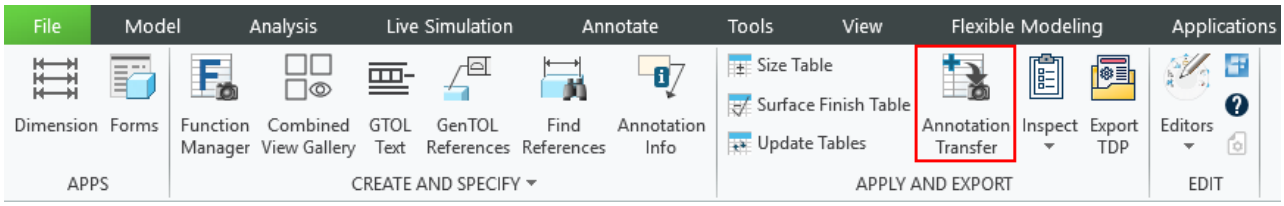
# 24.3 Annotation Transfer

Available in the ribbon menu GT MBD

You can use the module *Annotation Transfer* to transfer annotations from one or more combined views to another combined view.

## Starting the program: in part and assembly mode

To start *Annotation Transfer*, click *GT MBD* in the ribbon menu and go to *APPLY AND EXPORT* section.

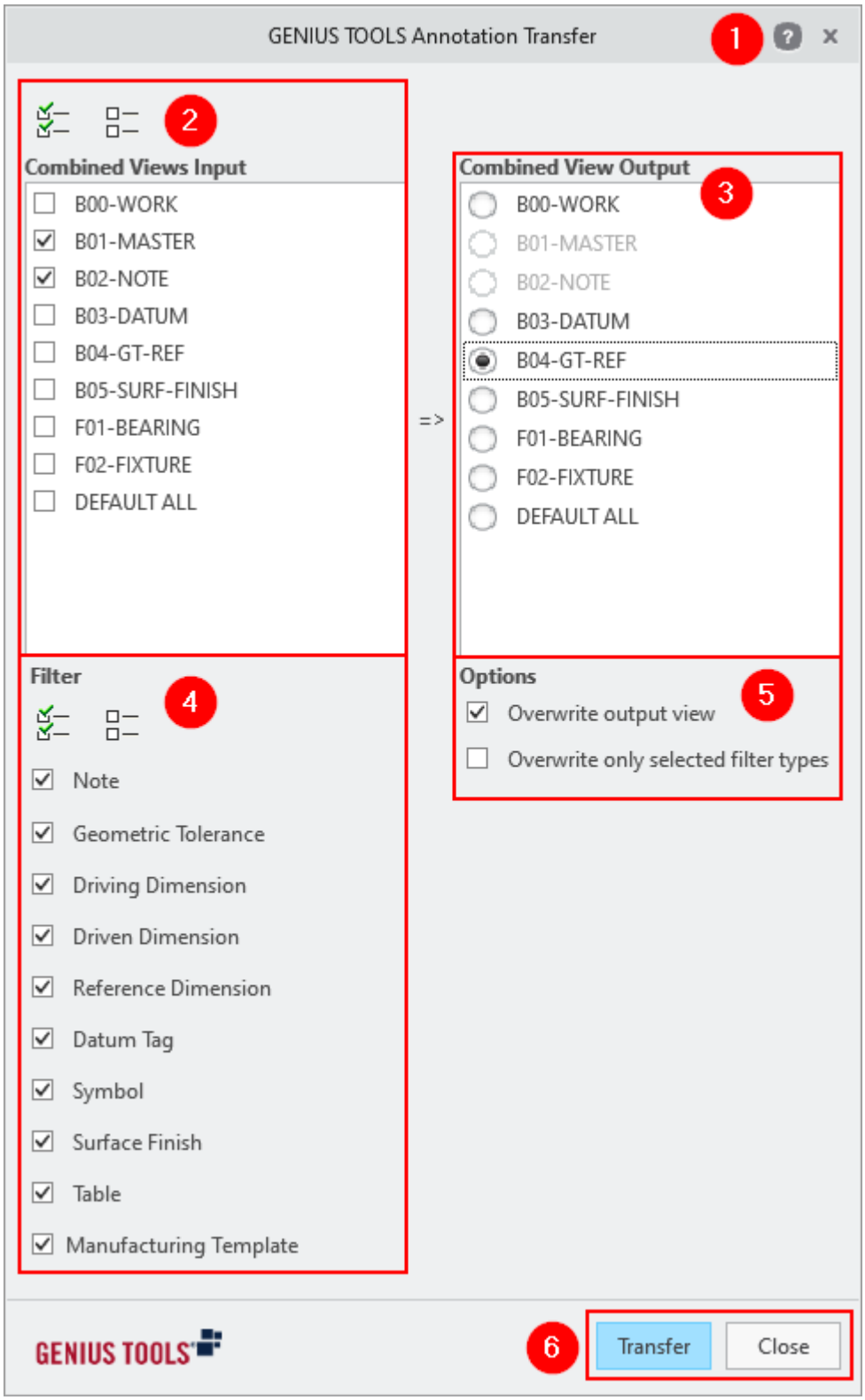



*Starting via the GT MBD in the ribbon menu*

## Transferring Annotations

The user interface provides the following transferabilities:


© 2026 INNEO Solutions GmbH




1. **Help**
   Opens the help.

2. **Selecting combined views to copy**
   All combined views are deselected by default. All combined views can be selected with


© 2026 INNEO Solutions GmbH

the button and selected again with the button . The combined views can also be selected individually.

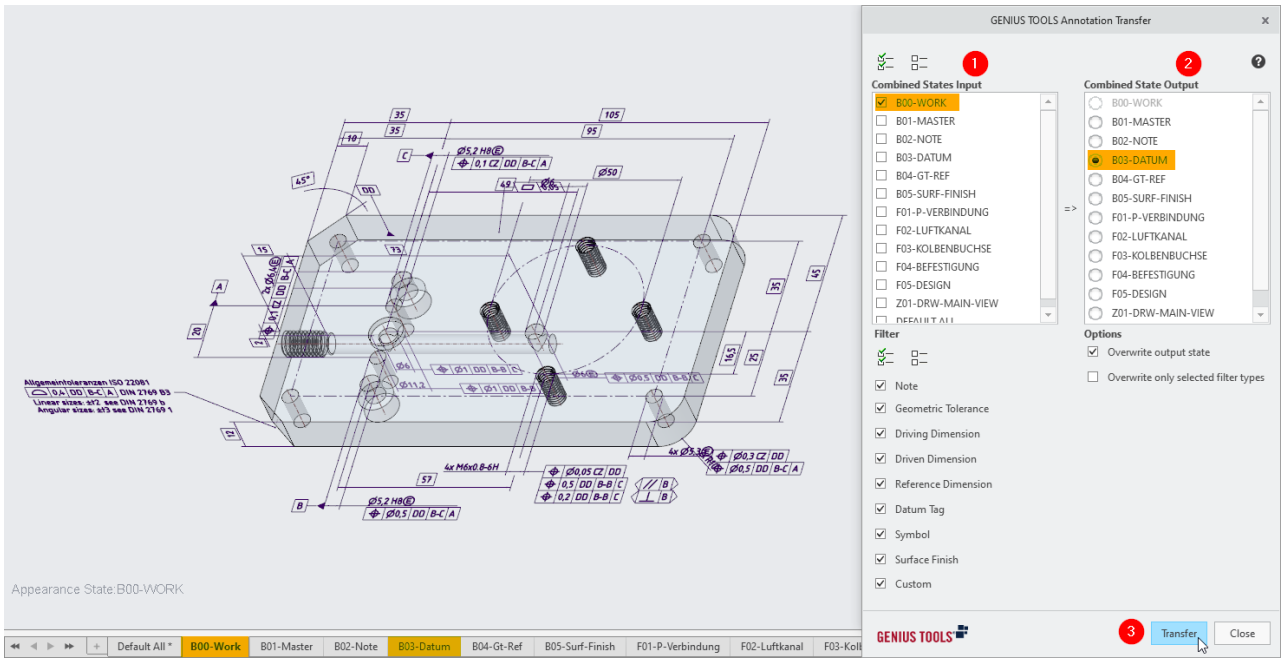
3. **Selecting copy destination**
   Select a combined view to which the elements to copy will be transferred.
4. **Selecting elements to copy**
   Under *Filter*, select the items you want to transfer. By default, all values are selected. All filters can be deselected with the button and selected again with the button .

**Please note:** Groups of annotations are transferred in their entirety. For example, if you selected *Reference Dimension*, all reference dimensions are transferred, including attached annotations of other types, such as *Datum Tag*: 

.

5. **Setting options**
   The following options are available and can be enabled by checking the appropriate box:
   - **Overwrite output state (Set by default)**
     Removes all annotations from the selected destination view before transferring.
   - **Overwrite only selected filter types**
     Only elements set under *Filter* are removed from the selected destination view before transferring.
6. **Starting transfer and closing window**
   - **Transfer**
     Click *Transfer* to copy the annotations according to your settings. To keep the transferred annotations, save the part / assembly in Creo Parametric.
   - **Close**
     Closes the user interface. Adjusted settings are not saved.


© 2026 INNEO Solutions GmbH




*Example of transferring annotation elements*

# 24.4 Close All Other Windows

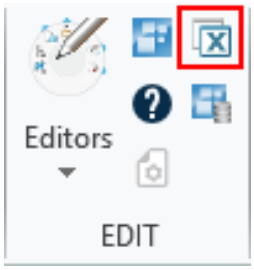
Available in the ribbon menu GENIUS TOOLS

This function closes all Creo windows except the current and the main window.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the function: in all modes

To start *Close all other windows*, click *GENIUS TOOLS* in the ribbon menu in the *EDIT* section or by using the *Quick Access* (key [<]).




*Start via the ribbon menu*



*Start via Quick Access*


© 2026 INNEO Solutions GmbH

## Configuration of the ribbon menu

Use the configuration option `gtu_start_close_all_windows` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## Configuration

You can define, if the models will be saved before closing the window with the configuration option `gtu_close_all_windows_save_changed_models` (default is 0=No automatic saving).

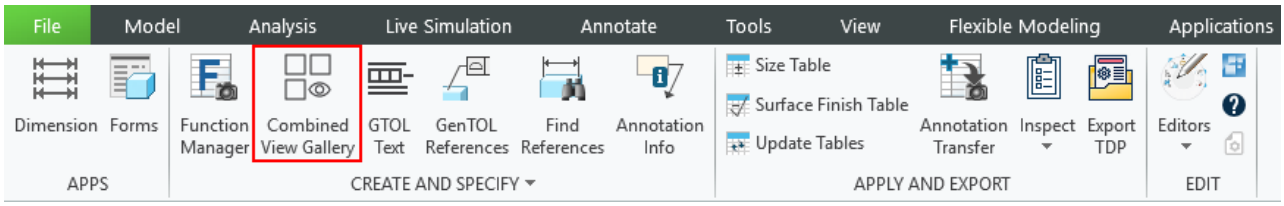
# 24.5 Combined View Gallery

Available in the ribbon menu GT MBD

*Combined View Gallery* allows you to view, compare and browse through all the combined views available in a model.

## Starting the program: in part and assembly mode

To start *Combined View Gallery*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY section*.



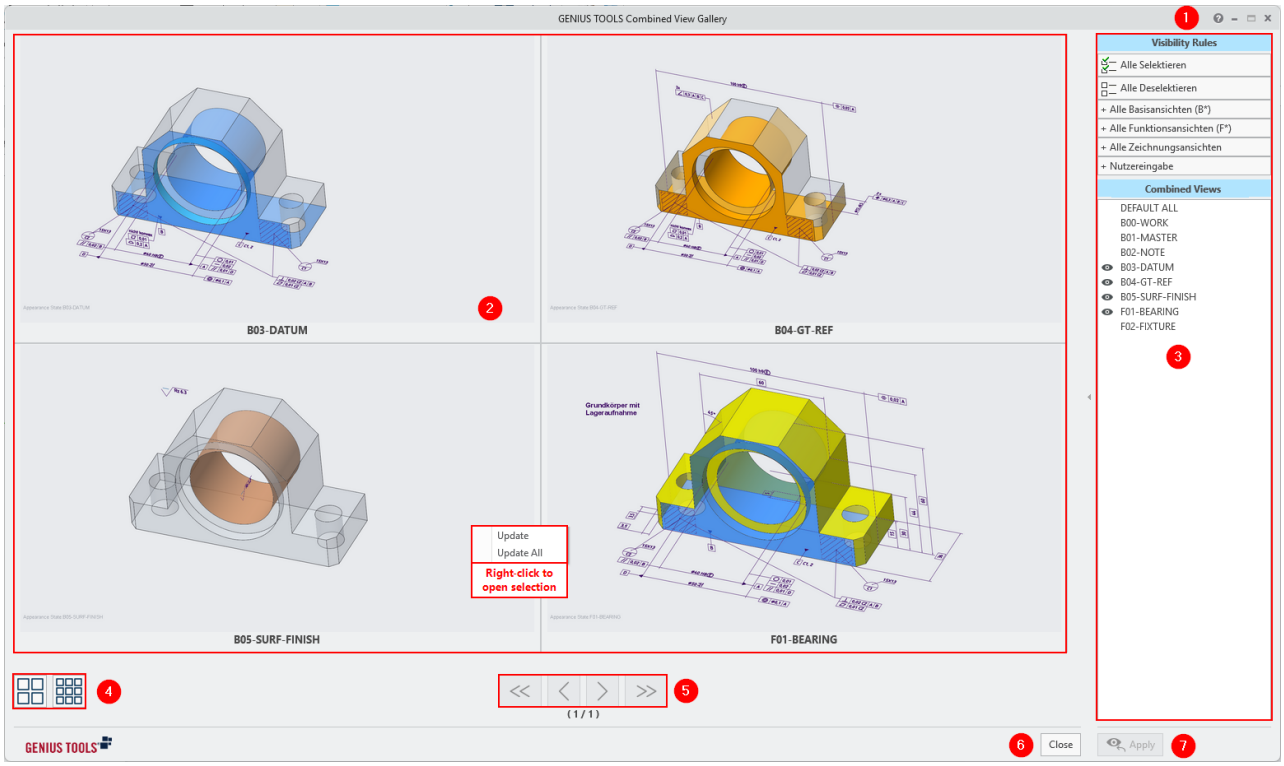

*Starting via the GT MBD in the ribbon menu*

## User interface

When you open the program, the current combined view is displayed in the gallery by default. If several views have to be created, the configuration option *gtu_combined_view_gallery_creation_rule* can be used to define a rule as to which views should be added when the gallery is opened.

Example: *gtu_combined_view_gallery_creation_rule = ^F*; this rule states that an image is created for all views beginning with "F".


© 2026 INNEO Solutions GmbH




1. Help
   Opens this help.

2. Overview of added combined views
   By default, you see one combined view. If several combined views were added to the gallery, you can switch to the single view by clicking on a combined view. With another click on the combined view, you switch to the last used multiple view.

   Update: return the view to its original position
   Update All: return all views to their original position

3. Visibility rules allow you to select which combined views should be displayed. You can create your own rules in the XML file using RegEx. All views are listed under *Combined Views*. Use the eye icon to show or hide the combined views.

4. Selecting the view mode
   2x2 view
   3x3 view

5. Navigation elements
   The arrows are grayed out if only one page is available.

6. Close the user interface

7. Apply changes


© 2026 INNEO Solutions GmbH

## Configuring visibility rules

Create an XML file for each visibility rule. A maximum of 8 visibility rules can be displayed. The structure is as follows:

```
<?xml version="1.0"?>
<gps-cvg version="1.0">
      <rule name="All functions" value="^F.*" clear_before="0" image="">
            <title>
                  <en>Add all function views (F*)</en>
                  <de>+ Alle Funktionsansichten (F*)</de>
            </title>
            <tooltip>
                  <en>select all combined views with beginning "F"</en>
                  <de>Selektiert alle kombinierten Ansichten die mit "F"
beginnen</de>
            </tooltip>
      </rule>
</gps-cvg>
```

`rule name`: Specify the rule. RegEx[487] or a search text can be used for this purpose.

`title`: Specify the title of the visibility rule.

`tooltip`: Specify a tooltip.

## Configuration

**gtu_combined_view_gallery_creation_rule**
Defines a rule, in the form of a regular expression, to find combined views whose screenshot is displayed when the program is started.

# 24.6 Command Control

Available in the ribbon menu GENIUS TOOLS

With Command Control you can hide or disable Creo Parametric ribbon menu commands (all commands).


© 2026 INNEO Solutions GmbH

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

The utility is activated with the configuration option `gtu_start_command_control=1`.

The configuration option `gtu_command_control_configuration` defines the commands to be changed. The value of the option has the following syntax:

```
NAME_1:MODUS_1|NAME_2:MODUS_2|NAME_3:MODUS_3
```

`NAME_x is a Creo Parametric Command Name.`

`MODUS_x` can take on one of the three values: `UNAVAILABLE, INVISIBLE, DISALLOW.`

**Example:** The following configuration options hide the two commands "Automatic check in" and "Change working directory":

```
gtu_start_command_control=1
gtu_command_control_configuration=ProCmdMdlTreeWfChkInExp:INVISIBLE|
ProCmdSessionChangeDir:INVISIBLE
```

## Mode values

UNAVAILABLE: Command is grey and cannot be selected.

INVISIBLE: Command is hidden.

DISALLOW: Command is visible, but is not executed.

## Creo Parametric Command Name

To find the name of a Creo Parametric Command you can record a mapkey with this command and copy the command out of the mapkey text. (See also Use cases[565].)

**Example:** What is the name of the command “Change working directory”?

Recorded Mapkey:

```
mapkey cd @MAPKEY_NAMEChange directory;@MAPKEY_LABELWorking directory;\
mapkey(continued) ~ Close `main_dlg_cur` `appl_casc`;~ Command
`ProCmdSessionChangeDir` ;\
mapkey(continued) ~ Trail `UI Desktop` `UI Desktop` `DLG_PREVIEW_POST`
`file_open`;
```

Result: The name for the command is „ProCmdSessionChangeDir“.

# 24.7 Component Parameter

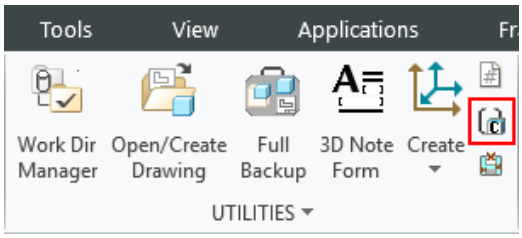
Available in the ribbon menu GENIUS TOOLS


© 2026 INNEO Solutions GmbH

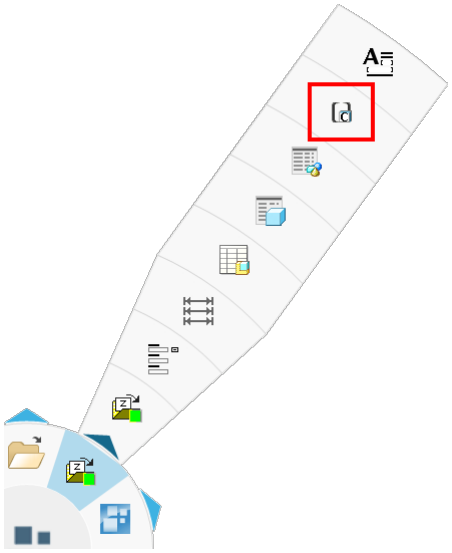
*Component Parameter* creates component parameters in assemblies. A component parameter allows to assign and edit values for models of same name. With component models of same name it is possible to assign different component parameter values.

## Starting the program: in assembly mode

To start *Component parameter* in assembly mode, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using the *Quick Access* (key [<]).



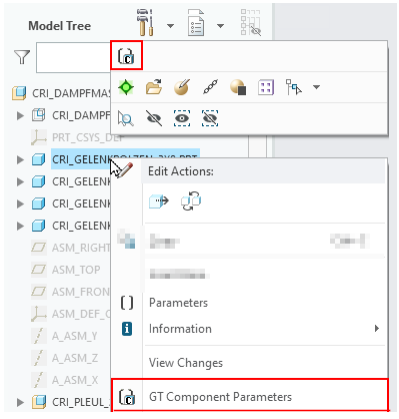

*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

*Component Parameter* starts with an empty selection. Select a subassembly in the model tree to start with a selection.

The area for editing component parameters[659] can also be opened from the context menu of a model (right-clicking).



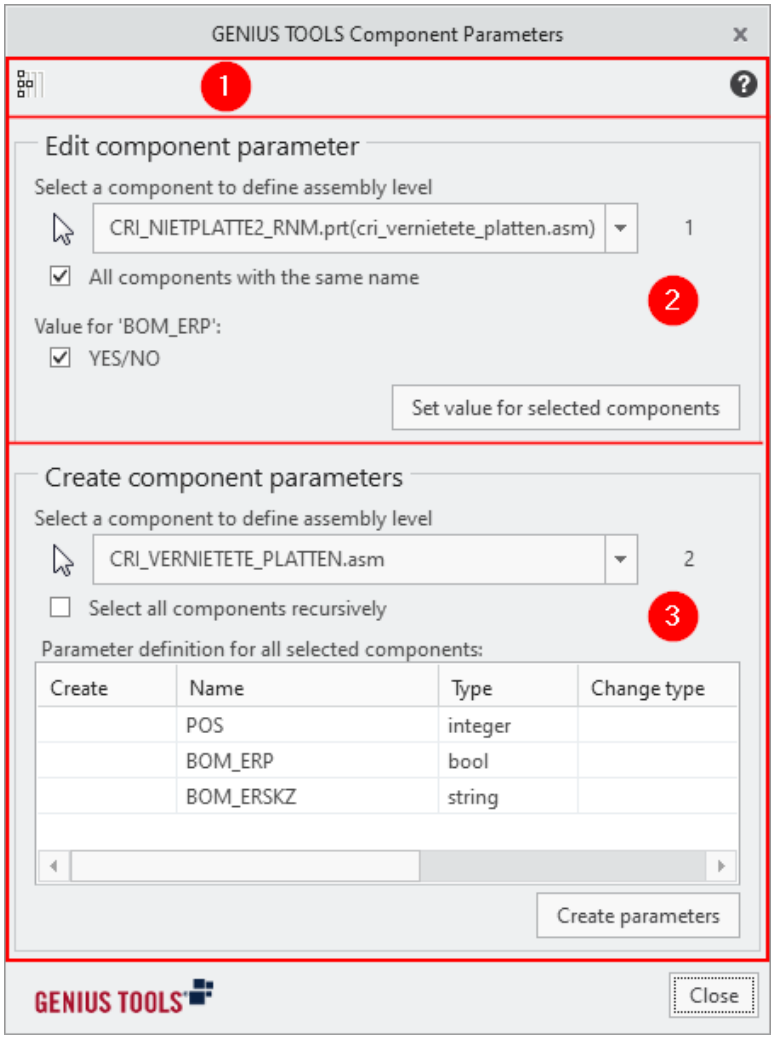

## Configuration of the ribbon menu

Use the configuration option `gtu_start_component_params` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)


© 2026 INNEO Solutions GmbH

## 24.7.1 User interface

The user interface of *Component Parameter* consists of the following elements:



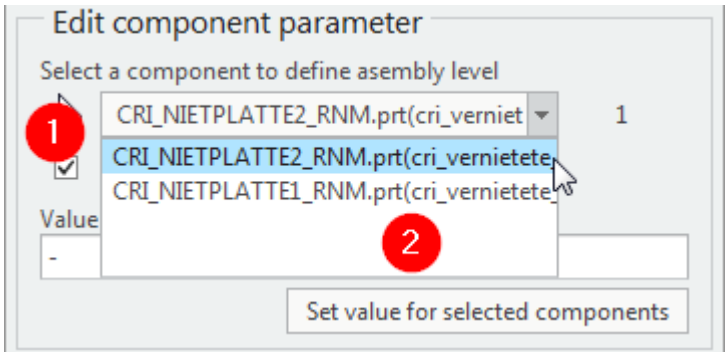

1. Command bar with button to show parameters in model tree and help.
   Use the button left to display the model tree. The component parameters are automatically displayed as an additional column in the model tree. This can be configured individually for each parameter.
2. Configuration of the position number parameter[659]
   This area can be opened by right-clicking on a part in the model tree.
3. Area to create the component parameters[660]


© 2026 INNEO Solutions GmbH

## 24.7.2 Editing position number parameters

The parameter values for assembly elements are set in the editing area.

Use the model selection to select a subassembly or an element from it. Models already selected via the object selection are displayed in the list, too.




*The model selection with object selection (1) and drop-down list (2).*

The number of assembled instances of the selected element is displayed right to the model selection. Click on the number to open a new window displaying the IDs of the assembled elements.

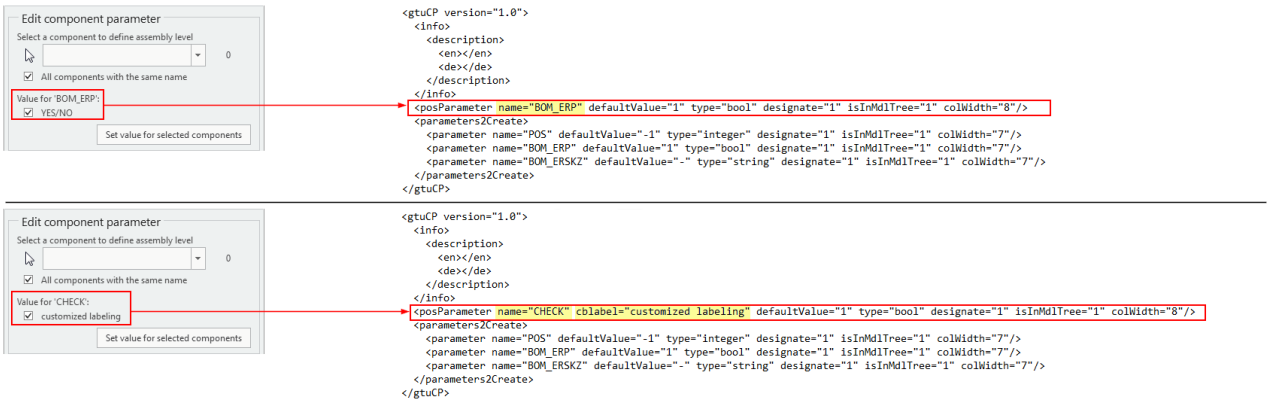
The *All assembly elements with same name* option aggregates repeatedly assembled assemblies or parts. This way it is possible to pass the same position numbers to all elements.

If the option is deactivated and multiple assemblies of an element have been detected, an individual value can be entered into the *Value for POS* table for each element.

**Please note**: The name (POS) for the component parameter for position numbers may vary due to the configuration on your system.

**Boolean parameter**
In the editing area you can select a boolean parameter. The display of this option is customizable. You do it with the help of the XML file *gtu_component_parameters.xml*. It is located under the path *<WorkingDirectory>\gtstarter\cadpool\INNEO\parametric\configuration\gt_resource_folder\utilities\component_parameters*.


© 2026 INNEO Solutions GmbH




*Definition of the label "Value for" in the XML file. Default setting (see above) and customization (see below).*

## 24.7.3 Creating component parameters

The *Create component parameters* area is used to create parameters across an entire assembly or subassemblies.

**Please note**: You can create any parameters. Prior to this, the parameters have to be created in the configuration.

Select the desired editing layer via the model selection. The number of affected elements is displayed right to the model selection. Click on the number to open a new window displaying the IDs of the assembled elements.

**Example**
If a part on the layer below the main assembly is selected, the main assembly will be displayed. All elements on the same layer and in the same assembly (in this case, the main assembly) as the selected part will be affected by the edit.

Activate the *Recursive selection of all assembly elements* option to select all elements underneath an assembly. Deactivate the option to select only those on the first layer.

The table displays the component parameters to be created. Select or deselect individual parameters using the *Create* option.

**Please note**: Parameter properties can only be modified in the configuration of the XML file.

Click on *Create parameters* to create all selected parameters in the assembly elements.

## 24.7.4 XML configuration

The position number parameter and the component parameters to be created are specified in the *gtu_component_parameters.xml* file in the *component_parameter* subfolder of *gt_resource_folder*:


© 2026 INNEO Solutions GmbH

```
<gtuCP version="1.0">
  <info>
...
  </info>
  <posParameter    name="POS"
                   defaultValue="5"
                   type="string"
                   designate="1"
                   isInMdlTree="1"
                   colWidth="8"/>
  <parameters2Create>
...
  </parameters2Create>
</gtuCP>
```

The XML file to be used is specified via the `gtu_comp_file_name` configuration option.

## Position number parameter

The following attributes are required for the position number parameter:

| XML attribute | Description |
|---|---|
| **name** | Specifies the name of the position number parameter to be created. |
| **defaultValue** | The default value displayed in the text box. |
| **type** | Specifies the parameter type. Available are *string*, *integer* and *double*. |
| **designate** | Should the parameter be designated when used with Windchill (0: no, 1: yes). |
| **isInMdlTree** | Specifies whether the parameter should be displayed in the model tree when clicking the associated button. |
| **colWidth** | If *isInMdlTree* is active, this attribute determines the width of the column for this parameter in the model tree in characters. |

**Example**

```
<posParameter      name="POS"
             defaultValue="5"
             type="string"
             designate="1"
             isInMdlTree="1"
             colWidth="8"/>
```

**Warning**: Only one position number parameter will be processed.


© 2026 INNEO Solutions GmbH

## Component parameter

Also creation and initial filling of the component parameters is controlled via this XML file. Any desired target component parameters with different properties can be specified:

| XML attribute | Description |
|---|---|
| **name** | Specifies the name of the parameter to be created. |
| **defaultValue** | The default value entered when creating the parameter. |
| **type** | Specifies the parameter type. Available are *string*, *integer*, *double* and *bool*. |
| **designate** | Should the parameter be designated when used with Windchill (0: no, 1: yes). |
| **isInMdlTree** | Specifies whether the parameter should be displayed in the model tree when clicking the associated button. |
| **colWidth** | If *isInMdlTree* is active, this attribute determines the width of the column for this parameter in the model tree in characters. |
| **isType2Change** | If *isType2Change* is "1", the selection box in the interface for the type change is active. If the parameter already exists in the component and the parameter type is different from the default type, the parameter is deleted and a new parameter with this name, the correct type, the original value and the original parameter description is created again. |

Changing the type of component parameters is necessary, for example, if parameters with an incorrect type were added via Assembly Report. The value of the parameter and the description of the parameter are retained.

**Example**

```
<parameters2Create>
...
<parameter
      name="CompParam"
      defaultValue="-"
      type="string"
      designate="0"
      isInMdlTree="1"
      colWidth="4"
      isType2Change="1"/>
```


© 2026 INNEO Solutions GmbH

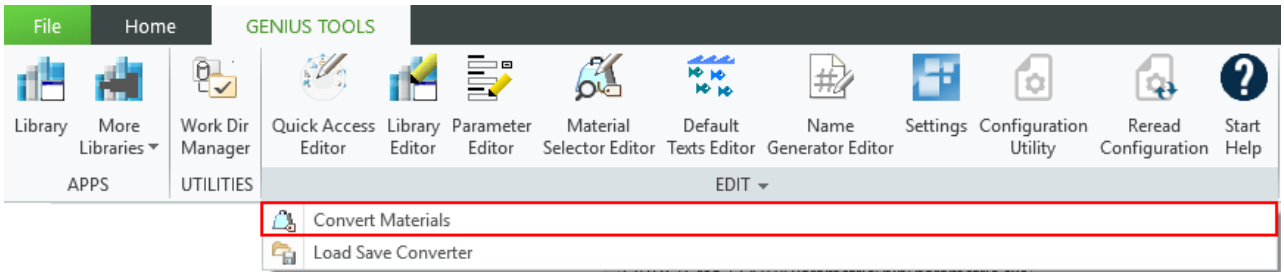
```
...
</parameters2Create>
```

# 24.8 Convert Materials

The function *Convert Materials* converts all MAT files into MTL files. MTL files can contain much more information than MAT files.

## Starting the function: in standby mode

To start *Convert Materials* , click *GENIUS TOOLS* in the ribbon menu and go to *EDIT-HELP* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

The Creo selection dialog for selecting the material directory opens. Select the directory containing the material files to be converted. The converted MAT files are moved to the directory *old* within the selected directory.

## Configuration of the ribbon menu

Use the configuration option `gtu_start_convert_materials` to switch off the function, so no button is displayed in the ribbon menu. (Default is 1=On/not hidden)

# 24.9 Copy Component Parameter To Substitution Component Parameter

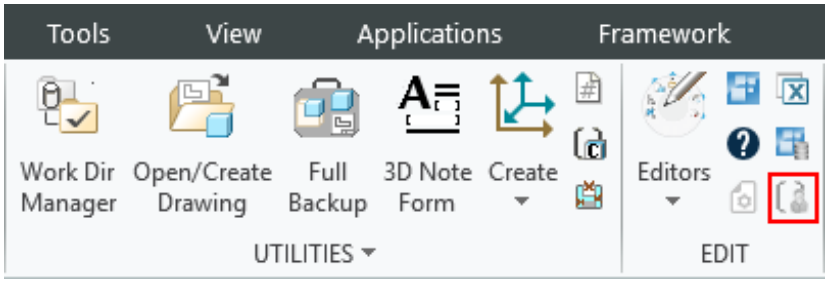
Available in the ribbon menu GENIUS TOOLS

This function copies all component parameters from the master representation to the substitution component representation in simplified representations.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

# Starting the program: in assembly mode (simplified representation)

The button can be found in the *GENIUS TOOLS* ribbon menu in the *EDIT* section, if an assembly is opened as simplified representation, which was created by substitution.

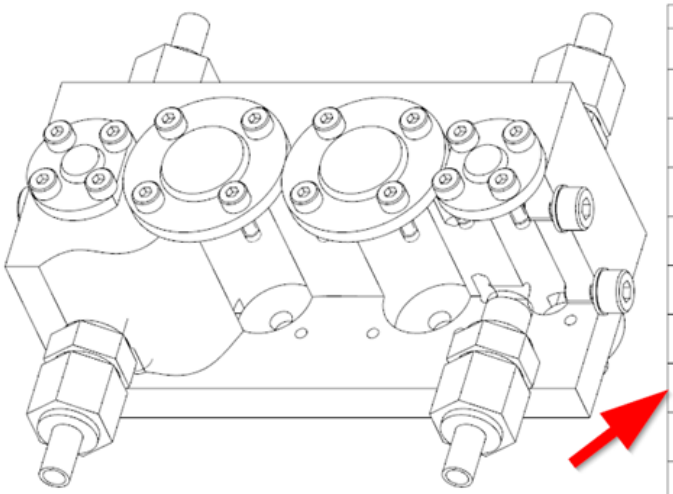



*Starting via the GENIUS TOOLS in the ribbon menu*

# Procedure

The substituted components will be enhanced by the component parameters from the component from the master representation at function runtime. If the parameters are not in existence yet, they will be created.

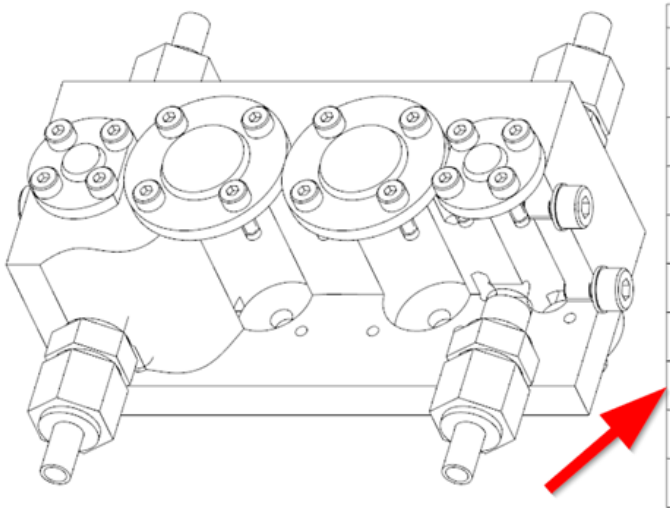
If you use the simplified representation inside a drawing for a report (Pro/Report), all component parameters are filled after executing the function. You can see the behavior described in the following images.



| Artikel: | | 002021268 | Baukastenstückliste | |
|---|---|---|---|---|
| Pos. | Anz. | Artikelnummer Halbzeugartikel-Nr. | Rev. | Benennung Modelldateiname |
| 10 | 4 | 002021274 | C | Anschlussverschraubung CRI_UEBERWURF_U_NIPPEL.ASM |
| 35 | 2 | 002021222 | C | ob.Schieberdeckel CRI_DECKEL_SCHIEB_2000.PRT |
| 40 | 2 | 002021226 | C | ob.Zylinderdeckel CRI_DECKEL_ZYL_2000.PRT |
| 15 | 4 | S002021239 | B | Scheibe CRI_D433T1D3_2.PRT |
| 45 | 2 | 002021262 | C | unt.Schieberdeckel CRI_DECKEL_SCHIEB_U_2000.PRT |
| 50 | 2 | 002021217 | C | unt.Zylinderdeckel CRI_DECKEL_ZYL_U_2000.PRT |
| | 1 | 0002021253 | C | Zylinderblock 2000 CRI_ZYLINDERBLOCK_2000.PRT |
| 25 | 32 | S002021250 | B | Zylinderschraube CRI_D912M2L6.PRT |
| 30 | 4 | S002021265 | B | Zylinderschraube CRI_D912M3L5.PRT |

*Before using the function Copy Component Parameter To Substitution Component Parameter*


© 2026 INNEO Solutions GmbH



| Artikel: | | 002021268 | Baukastenstückliste | |
|---|---|---|---|---|
| Pos. | Anz. | Artikelnummer Halbzeugartikel-Nr. | Rev. | Benennung Modelldateiname |
| 10 | 4 | 002021274 | C | Anschlussverschraubung CRI_UEBERWURF_U_NIPPEL.ASM |
| 35 | 2 | 002021222 | C | ob.Schieberdeckel CRI_DECKEL_SCHIEB_2000.PRT |
| 40 | 2 | 002021226 | C | ob.Zylinderdeckel CRI_DECKEL_ZYL_2000.PRT |
| 15 | 4 | S002021239 | B | Scheibe CRI_D433T1D3_2.PRT |
| 45 | 2 | 002021262 | C | unt.Schieberdeckel CRI_DECKEL_SCHIEB_U_2000.PRT |
| 50 | 2 | 002021217 | C | unt.Zylinderdeckel CRI_DECKEL_ZYL_U_2000.PRT |
| 20 | 1 | 0002021253 | C | Zylinderblock 2000 CRI_ZYLINDERBLOCK_2000.PRT |
| 25 | 32 | S002021250 | B | Zylinderschraube CRI_D912M2L6.PRT |
| 30 | 4 | S002021265 | B | Zylinderschraube CRI_D912M3L5.PRT |

*After*

## Configuration of the ribbon menu

Use the configuration option `gtu_start_copyCParamToSubsCParam` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

# 24.10 Create Search.pro

Available in the ribbon menu GENIUS TOOLS

When opening an assembly (ASM file) you can only access parts that are saved in other directories, if these directories are written into a separate search path file.

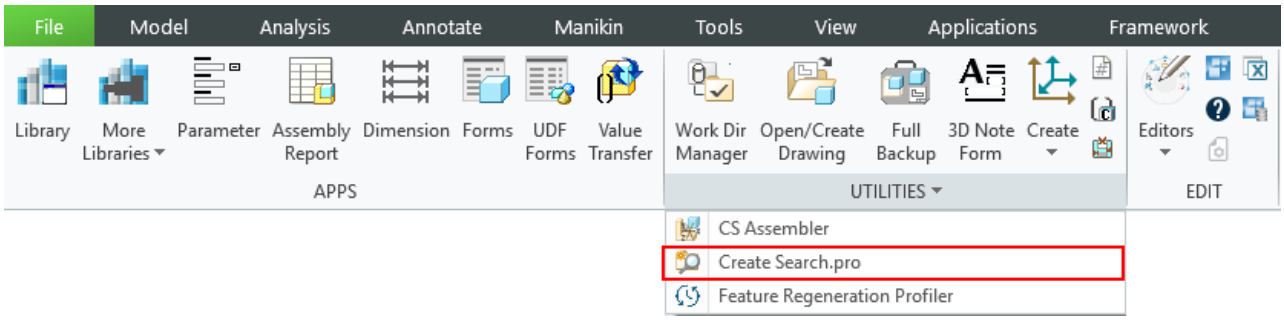
The function *Create Search.pro* creates and configures a project or assembly-specific *search.pro* file.

**Search.pro**
Search.pro is a file which includes the search paths for all parts of an assembly. It is especially useful in settings where a PDM system is not available.

## Starting the program: in assembly mode

To start *Create Search.pro*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.


*© 2026 INNEO Solutions GmbH*




*Starting via the GENIUS TOOLS in the ribbon menu*

A Save dialog opens after starting. After selecting a storage location the file will be saved and the *Create Search.pro* function is completed.

The content of the search path file depends on the configuration.

## Hiding the button from the menu

With configuration option `gtu_start_create_search_pro` you can hide the display of the button in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## Configuring search.pro file

**gtu_create_search_pro_standard_save_name**
Defines the suggested file name of the search path file in the Save dialog. Default: search.pro

**gtu_create_search_pro_exclude_current_path**
Defines whether the current working directory is included in the search.pro file (0) or not (1). Default: 1

**gtu_create_search_pro_exclude_file**
Defines a file containing search paths which are not to be included in the new search.pro file.

> **Tip:** All paths that point to the standard library should be specified here. We recommend entering the file that is defined in the configuration option `search_path_file`.

**gtu_create_search_pro_line_start**
Defines characters which are inserted at the beginning of each line.

> **Tip:** You should enter the config.pro format `search_path` here. Thus, the search.pro file can be loaded to the configuration settings of Creo.

**gtu_create_search_pro_standard_save_folder**
Defines the standard storage location.

0 – Creo standard directory (opens the File dialog)

1 - directory of the current object

2 – current working directory


© 2026 INNEO Solutions GmbH

# Creating a mapkey for loading and reloading a search path file (seach.pro file)

In order to guarantee a correct presentation of an assembly, the corresponding search path file should be loaded into the configuration settings of Creo before opening the assembly file. You can automate this process by creating a mapkey which saves the sequences of commands.

This mapkey can be used as a key macro and can also be added to the *Quick Access* ring menu.

The following process describes how to load the search.pro file from the working directory. You can also directly copy and paste the mapkey from the section Content of the mapkey file 668 below.

**Procedure**

For Creo to be able to read the search paths, the Creo configuration option `search_path` must be placed at the beginning of each line of the Search.pro file:

1. You are in standby mode of Creo.
2. Open the *Configuration Utility* editor from the *GENIUS TOOLS* ribbon.
3. Go to the configuration option `gtu_create_search_pro_line_start` and enter: search_path
4. Close the dialog and save changes in the main window with  before closing Configuration Utility.
5. In the *GENIUS TOOLS* ribbon, click the *Reread Configuration* button.

Create a new search path file for the assembly in question or overwrite the old one:

6. In the *GENIUS TOOLS* ribbon menu, go to *Utilities* and click on *Create Search.pro* (See screenshot above).
7. In the following Save dialog, save the file in the working directory.

Record the mapkey as in step 8 to 16 or copy and paste the content from the section below:

8. In the Creo menu, go to *File* > *Options* > *Environment* > *Mapkey Settings*
9. In the *Mapkeys* dialog box, click *New*.
10. In the *Record Mapkeys* dialog box, enter a key combination and a name, e.g. “sp” and „Load Search.pro “
11. Click on *Record*.
12. In the Creo menu, go to *File* > *Options* > *Configuration Editor*.
13. Under Options, click *Import/Export* > *Import configuration File*.
14. Select the working directory and open the Search.pro file.


© 2026 INNEO Solutions GmbH

15. Click *OK* and in the following *Creo Parametric Options* dialog box, click *No*. (The settings should not be saved to a configuration file).
16. In the *Record Mapkey* dialog box, click the *Stop* button and click *OK*.

Save the mapkey as a file in the working directory:

17. In the *Mapkeys* dialog box, click *Save changed*.
18. In the following *Save* dialog, enter a file name, e. g. mapkey, and click *OK*.

mapkey.pro

```
mapkey  sp @MAPKEY_LABELSearch.pro;\
mapkey(continued) ~ FocusIn `main_dlg_cur` `EMBED_BROWSER HOME`;\
mapkey(continued) ~ Select `main_dlg_cur` `appl_casc`;~ Close `main_dlg_cur` `appl_casc`;\
mapkey(continued) ~ Command `ProCmdRibbonOptionsDlg` ;\
mapkey(continued) ~ Select `ribbon_options_dialog` `PageSwitcherPageList` 1 `ConfigLayout`;\
mapkey(continued) ~ Select `ribbon_options_dialog` `ConfigLayout.ImportExportBtn`;\
mapkey(continued) ~ Close `ribbon_options_dialog` `ConfigLayout.ImportExportBtn`;\
mapkey(continued) ~ Activate `ribbon_options_dialog` `ConfigLayout.Open`;\
mapkey(continued) ~ Trail `UI Desktop` `UI Desktop` `DLG_PREVIEW_POST` `file_open`;\
mapkey(continued) ~ Trail `UI Desktop` `UI Desktop` `PREVIEW_POPUP_TIMER` \
mapkey(continued) `file_open:Ph_list.Filelist:<NULL>`;\
mapkey(continued) ~ Move `file_open` `file_open` 2 2.358632 4.834028;\
mapkey(continued) ~ Activate `file_open` `Current Dir`;\
mapkey(continued) ~ Select `file_open` `Ph_list.Filelist` 1 `creo-work`;\
mapkey(continued) ~ Select `file_open` `Ph_list.Filelist` 1 `creo-work`;\
mapkey(continued) ~ Activate `file_open` `Ph_list.Filelist` 1 `creo-work`;\
mapkey(continued) ~ Select `file_open` `Ph_list.Filelist` 1 `search.pro`;\
mapkey(continued) ~ Command `ProFileSelPushOpen_Standard@context_dlg_open_cmd` ;\
mapkey(continued) ~ Activate `ribbon_options_dialog` `OkPshBtn`;\
mapkey(continued) ~ FocusIn `UITools Msg Dialog Future` `yes`;\
mapkey(continued) ~ FocusIn `UITools Msg Dialog Future` `no`;\
mapkey(continued) ~ Activate `UITools Msg Dialog Future` `no`;\
mapkey(continued) ~ FocusIn `main_dlg_cur` `EMBED_BROWSER HOME`;
```

*Content of the mapkey file*

19. Copy the contents of the file into a Config.pro file that is loaded when Creo is started.

**Result**

The key combination (here: "sp") is available as mapkey in every Creo session.

The mapkey can also be included in *Quick Access*. This is described in the chapter Inserting existing mapkeys.[559]

**Content of the mapkey file for the key combination "sp"**

mapkey sp @MAPKEY_LABELLoad search.pro;~ Select `main_dlg_cur` `appl_casc`;\
mapkey(continued) ~ Close `main_dlg_cur` `appl_casc`;~ Command `ProCmdRibbonOptionsDlg` ;\
mapkey(continued) ~ Select `ribbon_options_dialog` `PageSwitcherPageList` 1 `ConfigLayout`;\
mapkey(continued) ~ Select `ribbon_options_dialog` `ConfigLayout.ImportExportBtn`;\
mapkey(continued) ~ Close `ribbon_options_dialog` `ConfigLayout.ImportExportBtn`;\
mapkey(continued) ~ Activate `ribbon_options_dialog` `ConfigLayout.Open`;\
mapkey(continued) ~ Trail `UI Desktop` `UI Desktop` `DLG_PREVIEW_POST` `file_open`;\
mapkey(continued) ~ Activate `file_open` `Current Dir`;\
mapkey(continued) ~ Trail `UI Desktop` `UI Desktop` `PREVIEW_POPUP_TIMER` \
mapkey(continued) `file_open:Ph_list.Filelist:<NULL>`;\


© 2026 INNEO Solutions GmbH

mapkey(continued) ~ Select `file_open` `Ph_list.Filelist` 1 `search.pro`;\
mapkey(continued) ~ Command `ProFileSelPushOpen_Standard@context_dlg_open_cmd` ;\
mapkey(continued) ~ Activate `ribbon_options_dialog` `OkPshBtn`;\
mapkey(continued) ~ FocusIn `UITools Msg Dialog Future` `no`;\
mapkey(continued) ~ Activate `UITools Msg Dialog Future` `no`;

# 24.11 Create Tolerance Table

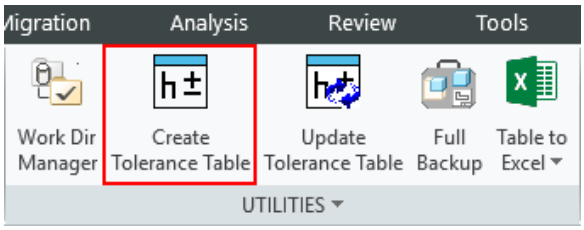
Available in the ribbon menu GENIUS TOOLS

This function creates tables of dimensions in Creo Parametric drawing mode. The set tolerance dimensions are used.

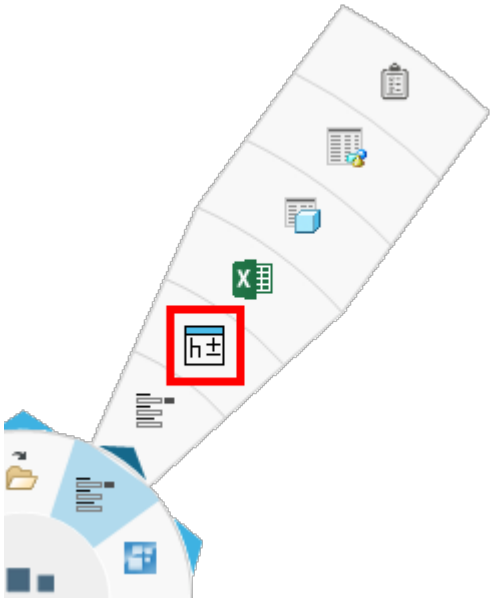
**Please note**: Only fitting tolerances already displayed will be output.

## Starting the program: in drawing mode

To start *Create tolerance table* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

The function creates a tolerance table at a freely selectable position on a drawing. As a default the tolerance table will always be created for all views displayed on the drawing. It can be selected via the configuration whether views of the current sheet only or if all views should be considered.

The following table formats are supported:


© 2026 INNEO Solutions GmbH

| Tolerance | Fitsize |
|---|---|
| ∅2.20 H6 | 0.006 |
| | 0.000 |
| ∅2.20 H7 | 0.010 |
| | 0.000 |
| ∅12.00 H7 | 0.018 |
| | 0.000 |

*Table type A: Dimensions*

| Tolerance | Minimum | Maximum |
|---|---|---|
| ∅2,20 H6 | 2,200 | 2,206 |
| ∅2,20 H7 | 2,200 | 2,210 |
| ∅12,00 H7 | 12,000 | 12,018 |

*Table type B: Minimum/maximum dimension*

The default table type is type A. You can change the format by setting the configuration option `gtu_tol_table_form` to B.

## Update tolerance table

If a tolerance is changed or a new one is added, the tolerance table has to be recreated by pressing the *Update Tolerance Table* button  in the ribbon menu.

## Configuration of the ribbon menu

Use the configuration option `gtu_start_tolerance_table` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## Configuration

The configuration option `gtu_tol_table_creo_insert` defines, whether the table is inserted without preview (0) or with preview (1).

## Configuration of the table

**gtu_tol_table_decimal_marker_follow_dtl**
Changes the representation of numerical values from period to comma as decimal separator, if the DTL option `decimal_marker = = COMMA` is set. Define, whether the DTL file should be read (1) or not (0). (Default is 1.)

**gtu_tol_table_form**
Defines the format type of the tolerance table. Type A: Dimensions, Type B: Minimum / maximum (see screenshots above)

**gtu_tol_table_sort_order**
Defines, whether the table is sorted in ascending (ASC) or descending (DESC) order.


© 2026 INNEO Solutions GmbH

**gtu_tol_table_origin_at_bottom_right**
Defines the origin and the growth direction of the table.
0 - The origin of the table is at the top left and it is built up to the bottom right.
1 - The origin of the table is at the bottom right and it is built up to the top left.

**Please note:** If the table is placed at the bottom of the drawing, reversing the growth direction is helpful. If the table is placed at the top edge, it makes sense to turn the growth direction downwards.
The selected growth direction of the table cannot be displayed in the preview (`gtu_tol_table_creo_insert=1`).

## Configuration for use with ModelCheck

When using with ModelCheck, it may be necessary to change the font for the first line. Set the following option:

**gtu_tol_table_hidden_font**
Defines the font for the hidden table header. If no font is defined, the header will receive the drawing font. Default: Isofont

For more configuration options to define the table view refer to configuration settings[901].

# 24.12 Create Point and Coordinate systems

Available in the ribbon menu GENIUS TOOLS

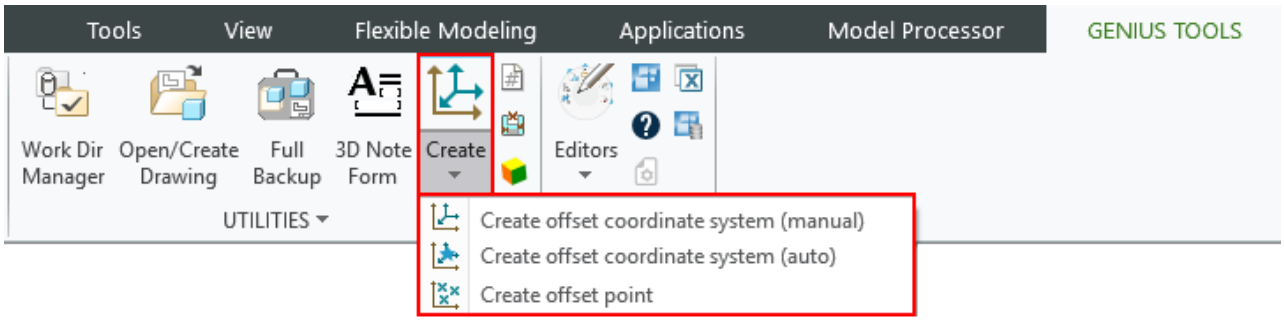
These functions make it possible to create a coordinate system or a point at the same location (point, vertex or coordinate system) within a part or an assembly, but with reference to a newly selected reference coordinate system.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Opening the function: in part and assembly mode

To start a function, click *GENIUS TOOLS* in the ribbon menu, go to *UTILITIES* section and click on *Create*  to open the drop-down menu.


© 2026 INNEO Solutions GmbH




*Starting via the GENIUS TOOLS in the ribbon menu*

# Functions

– **Create offset coordinate system (manual)**
These functions make it possible to create a coordinate system at the same location (point, vertex or coordinate system) within a part or an assembly, but with reference to a newly selected reference coordinate system. If an existing coordinate system is copied, its rotation is also applied.

– **Create offset coordinate system (auto)**
If a point or corner point is selected, a coordinate system is automatically placed at this spot if the defined coordinate system *%gtu_datum_creation_std_placement_cs* is found. If no coordinate system is found, the user must select a coordinate system.

– **Create offset point**
These functions make it possible to create a point at the same location (point, vertex or coordinate system) within a part or an assembly, but with reference to a newly selected reference coordinate system.

# Configuration of the ribbon menu

Use the configuration option `gtu_start_datum_creation` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

# Configuration Create offset coordinate system (Auto)

**gtu_datum_creation_std_placement_cs**
Defines the name for the reference coordinate system. If the coordinate system can't be found, a selection is requested. (Default is X0Y0Z0)

# Name definition for configuration options for offset coordinate system

**gtu_datum_creation_cs_prestring**

Defines the name of the created feature.


© 2026 INNEO Solutions GmbH

MCS_%gtu_datum_creation_start_feature_numbers%,

MCS_%gtu_datum_creation_start_feature_numbers% + 1,

...

## Name definition for configuration options for offset points

**gtu_datum_creation_pnt_prestring**
Defines the name of the created feature.

PNT_%gtu_datum_creation_start_feature_numbers%,

PNT_%gtu_datum_creation_start_feature_numbers% + 1,

...

## Configuration for the start value of the generated construction elements

**gtu_datum_creation_start_feature_numbers**
Defines the start value for created features.

# 24.13 CS Assembler

Available in the ribbon menu GENIUS TOOLS

*CS Assembler* is a tool for automatically adding a number of components to an assembly. CS stands for coordinate system.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

The target assembly and the component models have to contain specific coordinate systems. In the target assembly, the relevant coordinate systems are specified by component parameters. In the component models, the relevant coordinate systems are specified by a common name, which is defined in a configuration option.

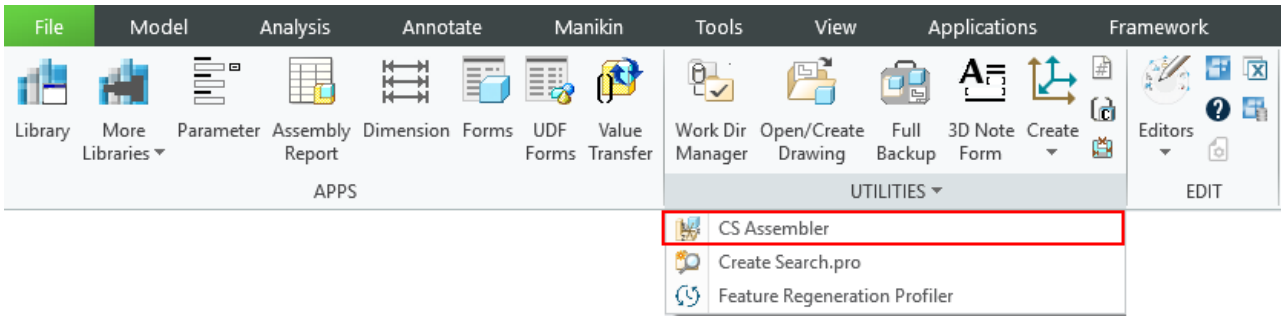
The assembling process works by making the corresponding coordinate systems coincident to each other.

The assembling instructions are defined in an XML file which assigns a target coordinate system in the assembly to each component model. For each component model, the file defines which target coordinate system or which target coordinate systems of the same name should be used.


© 2026 INNEO Solutions GmbH

# Starting the program: in part mode

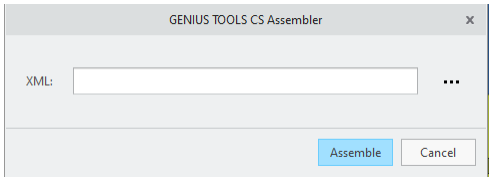
Open the assembly that you want to add components to. To start *CS Assembler* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

# Procedure

Select the XML file that contains the assembling instructions. The models specified in the XML file will be added to the assembly, if all requirements are fulfilled.




# Requirements

1. In order to be able to use *CS Assembler*, the configuration option `gtu_start_csassembler` has to be active, that is, set to 1.
2. Each model that is to become a component of the target assembly has to have an assembling coordinate system, which has to have the name specified in the configuration option `gtu_csassembler_component_cs_name`.
3. The models to be assembled have to be located in the Creo search path.
4. The assemby that is to receive the components has to have a target coordinate system for each model to be assembled.
   - The target coordinate system can be identified by the component parameters *CLASSIFICATION* and *SUBTYPE*. These parameters are used to assign a target coordinate system to each component model in the XML assembling instruction. If there are multiple target coordinate systems in the assembly that fit the assembling instructions, the component model is added multiple times.
   - Alternatively, the name of the target coordinate system can be entered in the attribute *ics_name*.


*© 2026 INNEO Solutions GmbH*

This provides for several possibilities to find a coordinate system. You can either define *CLASSIFICATION* or define *ics_name* or use one of the following combination of attributes:

`classification` and `subtype`

`classification` and `ics_name`

You cannot combine `classification`, `subtype` and `ics_name`.

## XML assembling instructions

The assembling instructions have to be available in a specified XML format and in UTF-8 encoding. This section describes the required XML format.

The root element `root` can include additional information:

- `std_ics_name="ICS1"`: ics stands for include coordinate system
- `std_pcs_name="PCS1"`: pcs stands for placement coordinate system

For example, ICS1 and PCS1 stand for the feature names of the coordinate systems. Both entries are optional.

Each model to be assembled is defined by an element `assemble` with the following attributes.:

- `file`: file name of the model, with extension
- `classification`: value of the *CLASSIFICATION* parameter on the target coordinate system
- `subtype` (optional): value of the *SUBTYPE* parameter on the target coordinate system. You can use different values for the *SUBTYPE* parameter in order to use multiple target coordinate systems with the same *CLASSIFICATION*.
- `ics_name`: name of feature of the target coordinate system

`assemble` is an empty element.

## Example

```
<?xml version="1.0" encoding="utf8"?>
<root std_ics_name="MEOW" std_pcs_name="ROW">
    <assemble file="PRT001.prt" classification="001" subtype=""/>
    <assemble file="PRT002.prt" classification="002" subtype=""/>
</root>
```


© 2026 INNEO Solutions GmbH

## Configuring coordinate systems: multi-level assembly and maximum assembly

**gtu_csassembler_multi_level**
Defines whether coordinate systems from newly assembled components should be used as target coordinate system inside the same assemble task (1) or not (0). Default: 1

**gtu_csassembler_maximal_cs_count**
Defines the maximal number of coordinate systems that can be assembled into a model. Default: 5000

## Configuration of the ribbon menu

Use the configuration option `gtu_start_csassembler` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

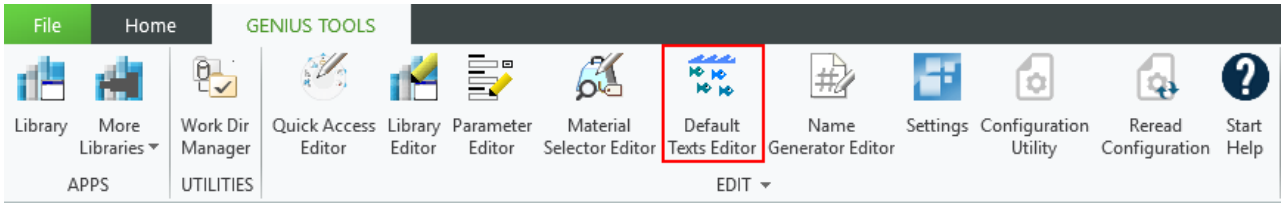
# 24.14 Default Texts Editor

Available in the ribbon menu GENIUS TOOLS

Create a database of words which can be used in multiple *GENIUS TOOLS* extension modules to fill dialog boxes with multilingual descriptions partly automated.

## Starting the program: in Creo standby mode

To start *Default Texts Editor*, click *GENIUS TOOLS* in the ribbon menu and go to *EDIT-HELP* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

*Default Texts Editor* starts with the database and table defined by the configuration.

## Configuration of the ribbon menu

Use the configuration option `gtu_start_edit_default_texts` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

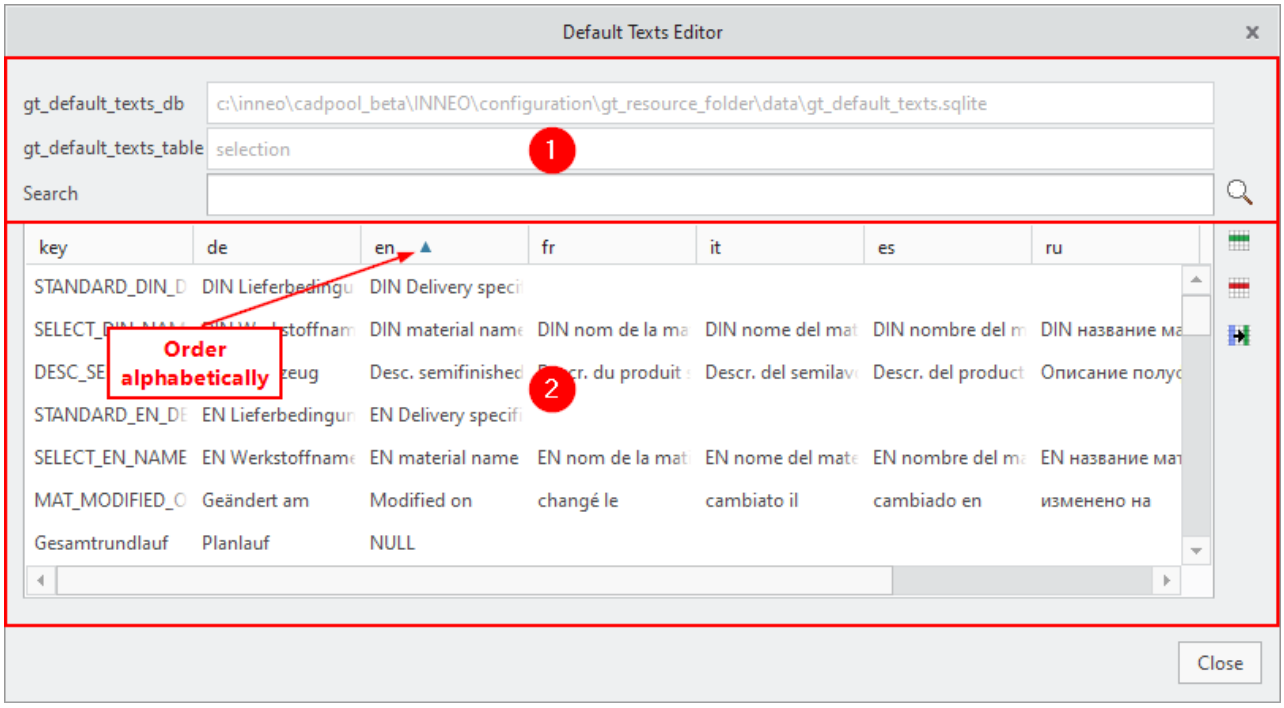

© 2026 INNEO Solutions GmbH

# Configuration

Define the path to the database with the configuration option `gt_default_texts_db` and the table to be used from that database with `gt_default_texts_table`.

## 24.14.1 Create default text

The user interface of *Default Texts Editor* is divided into two areas.




1. Configuration options and filters
   – use the search function to filter down the list of editable elements
2. Tabular editor
   – edit words and phrases directly inside the table
   – add elements/rows with the button
   – remove elements/rows with the button
   – copy values from one column to another with the button

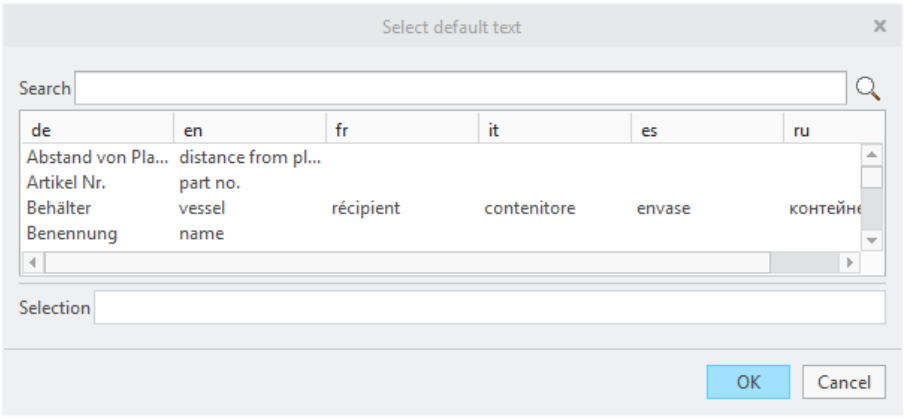
To add or delete languages (columns) the database must be changed accordingly.

**Tip**: You can change and add words while you select standard text[678] in other *GENIUS TOOLS* extension modules. Already set values will not be affected by these changes.


*© 2026 INNEO Solutions GmbH*

## 24.14.2 Select default text

In the editors of many *GENIUS TOOLS* extension modules it is possible to add multilingual text for a single element, e.g. parameter in *Parameter* or measurement in *Forms*. This function is opened with the button which opens the dialog *Select default text*.

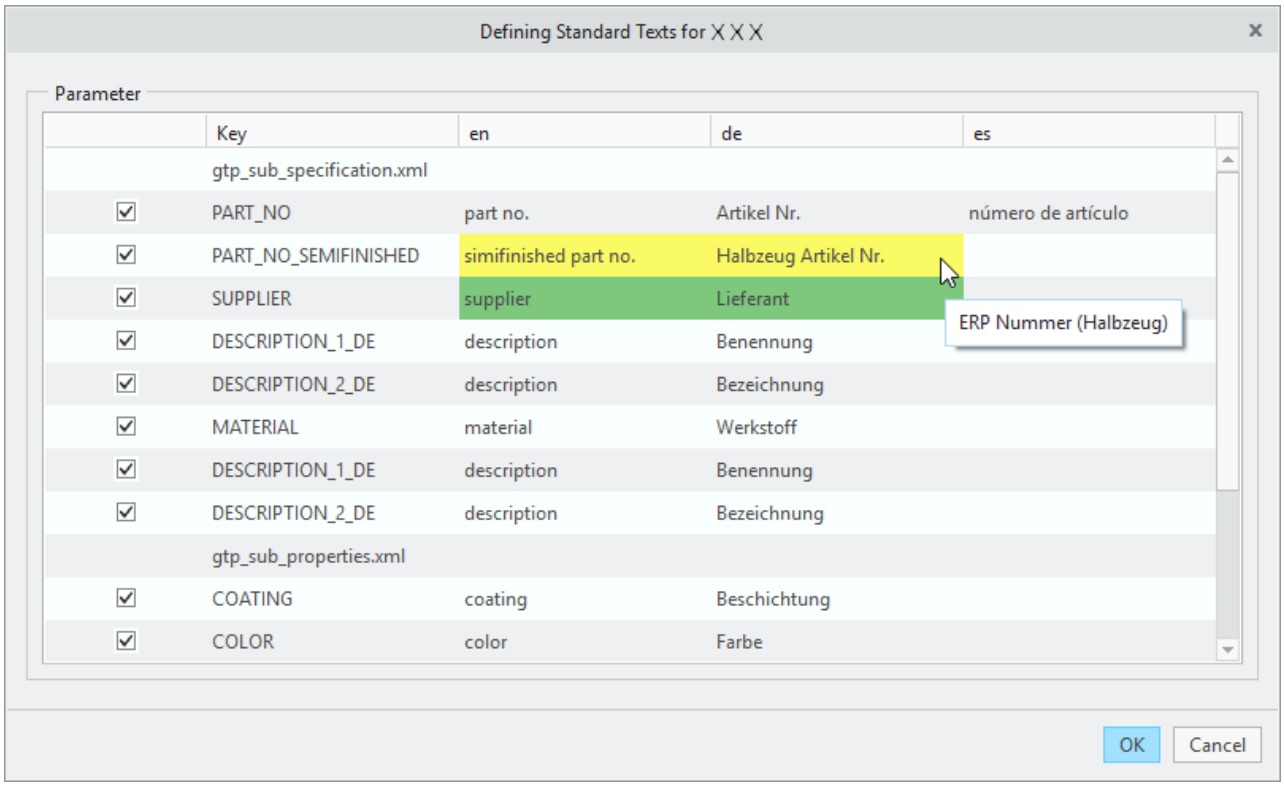



Select a standard text and click OK. The selected text is transferred to the language dependent fields. You can use the search field to filter the text list.

## Select default text for multiple elements

In addition, standard texts for several elements can also be stored in some editors. The automatic setting of standard texts is called up via the button . The program collects all suitable elements and transfers them to the user interface for automatic setting of standard texts. An attempt is made to find a suitable translation on the basis of a keyword in the database. The result is displayed in a table. The table lists only those elements for which the program has found a suitable key.


© 2026 INNEO Solutions GmbH




You cannot enter your own values here.

The values in the column *Key* depend on the module. For example, *Parameter* contains the names of the parameters and separators.

The language columns depend on the displayed columns in the calling editor.

The background colors of the individual cells have the following meaning:

– white: The translation of the element in the editor matches that in the database.

– green: The element has no translation in the editor yet.

– yellow: The translation of the element in the editor is different from the translation in the database. The translation of the editor appears as a tooltip.

If the dialog is confirmed with OK, all checked lines are transferred to the editor.

# 24.15 Export Points

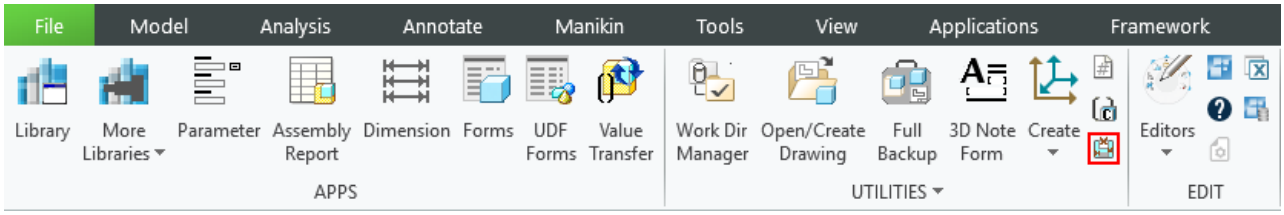
Available in the ribbon menu GENIUS TOOLS

This functions allows you to export single points or an array of points as well as dynamically created points on curves (X-Y-Z-values) to an ASCII file, which can either have the format of a Creo point file (PTS file) or a user defined template.


© 2026 INNEO Solutions GmbH

# Starting the program: in part and assembly mode

To start *Points* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.



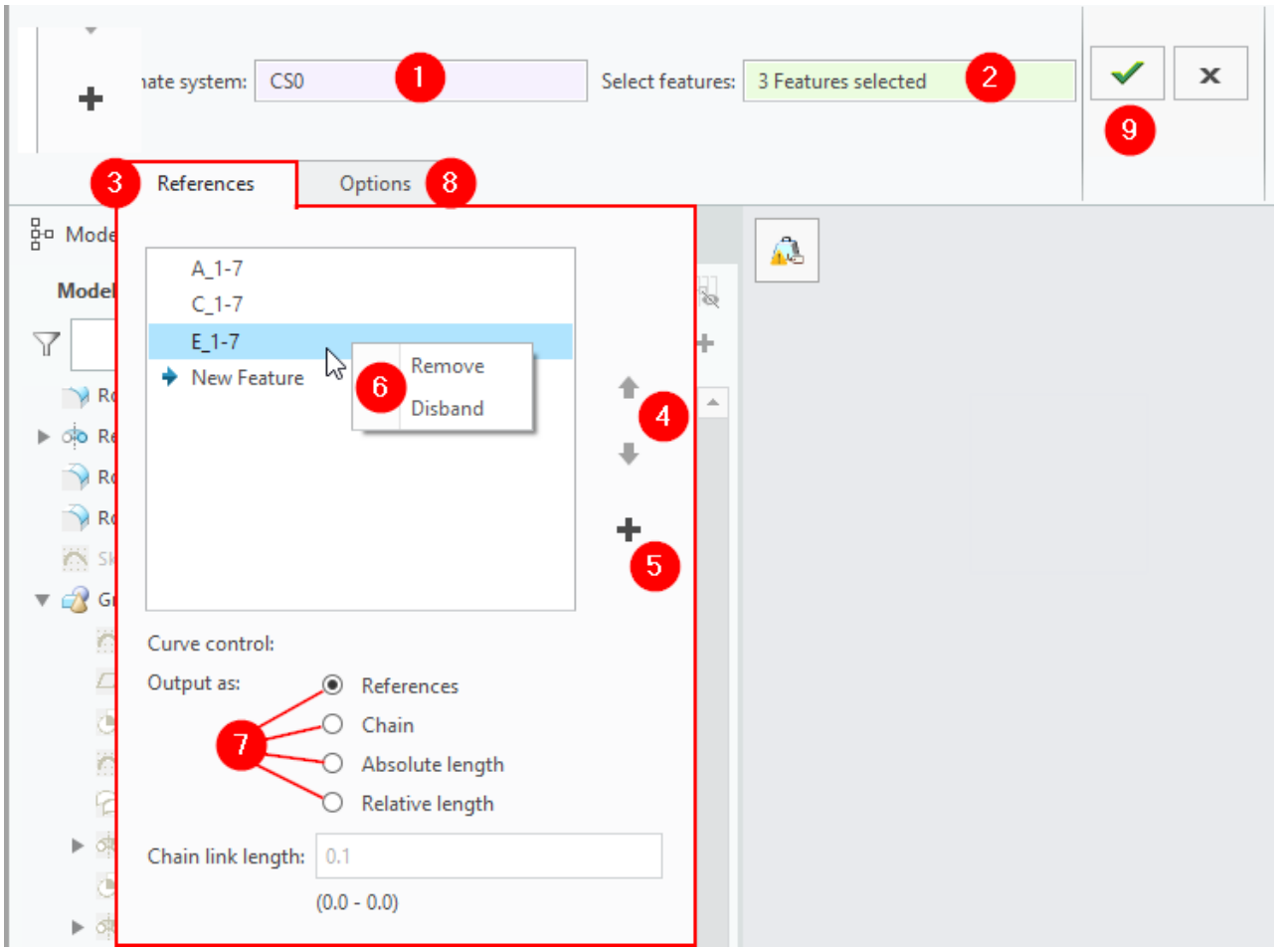

*Starting via the GENIUS TOOLS in the ribbon menu*

# Configuration of the ribbon menu

Use the configuration option `gtu_start_points` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## 24.15.1 Exporting points

With GENIUS TOOLS Points, you can export single points, curves and point groups as PTS files with different settings.




*Dialog Points with opened References tab*


© 2026 INNEO Solutions GmbH

## Steps

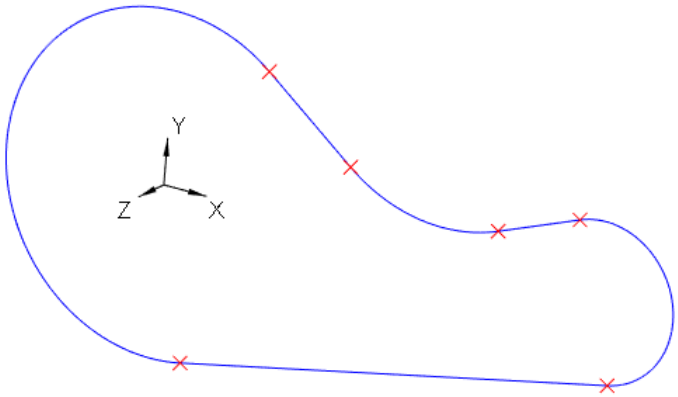
1. Click in the field Select coordinate system (1).
2. Select a coordinate system that serves as the base for the coordinates of the points. The name of the coordinate system is displayed – in above example: CS0.
3. Click in the field *Select features (2)*.
4. Select Creo features such as points, curves or groups of points. You can also select single-level patterns. Multilevel patterns are not supported. (In above example: Features A_1-7, C_1-7 and E_1-7.)
   - The quantity of selected features is displayed in the main window in the *Select Features* field (2)
   - All names of selected features are displayed in the tab *References (3)*.
3. In the *References* tab (3):
   - Check the order of features and adjust if necessary by using the arrows on the right (4).
   - Click *Add points by name* (5) to select points by their name if the exported points conform to certain naming conventions.
   - You can delete features or disband groups of points by using the context menu by right-clicking on a feature (6).
4. Define the output options of curves (7). (For more information see next chapter 681 .)
5. In the tab *options (8)* choose an export format. (For more information see Export options 682 .)
6. Confirm the dialog by clicking the green check mark button (9).

## 24.15.2 Settings for points on curves

In the segment *Curve control* (7) in the *References* tab you can select the method of how to define points on a curve. There are four possibilities.

**References:** Start and end points of each segment of the curve are output.


© 2026 INNEO Solutions GmbH




**Chain:** The value of “segment length” is the distance between two points, not the segment length of the curve.




**Absolute length:** The value of “segment length” is the segment length of the curve that defines the coordinates of the points.

**Relative length:** The value of “segment length” is the relative segment length of the curve that defines the coordinates of the points.

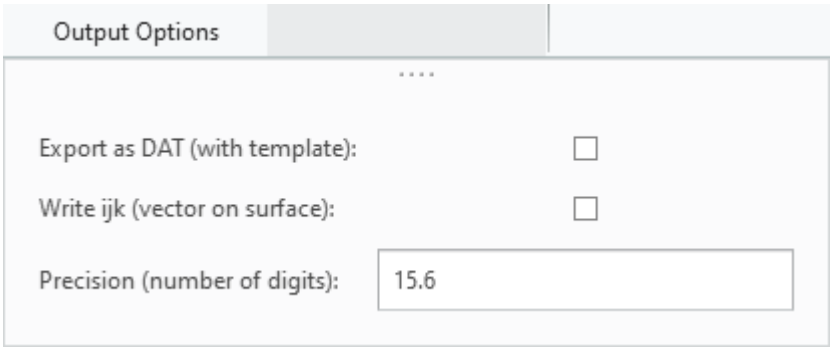
## Configure preselection

You can change the preselection of the curve control with the configuration option `gtu_points_curve_output_type`: 0 = References, 1 = Chain, 2 = Absolute length, 3 = Relative length (Default: 0)

## 24.15.3 Export options

Settings in the options tab define the output format of the points values file.


© 2026 INNEO Solutions GmbH




*Options tab in Points*

**Export as DAT (with template):** exports data of points values into a template.

**Write ijk (vector on surface):** The ijk-values describe the perpendicular vector (normal) if a point through a surface. The surface that is next to the point would be used automatically.

**Precision (number of digits):** defines the total number of digits and the digits after the decimal point of each point value.

- Input format ".X": number of digits after the decimal point (e.g.: *.4* = four digits after the decimal point)
- Input format "X.X": number of total digits including the number of digits after the decimal point (e.g.: *10.4* = 10 digits of which four after the decimal point)

# Configure preselection in the options tab

You can change the preselection of the *Export as DAT* checkbox with the configuration option `gtu_points_write_use_templates`. (Default is 0=off)

You can change the preselection of the *Write ijk* checkbox with the configuration option `gtu_points_write_ijk`. (Default is 0=off)

You can set the predefined value in the *Precision* field with the configuration option `gtu_points_precision`. (Default value is 15.6)

# Output formats and examples

## 1. Standard output

No template and no ijk values

File format: Creo point file *.pts

Example of an output with default precision of 15.6:

```
 96.090000      -0.320000      50.000000
130.690000      -5.660000      50.000000
123.440000     -33.600000      50.000000
 89.940000     -36.160000      50.000000
 83.610000      -8.040000      50.000000
```


© 2026 INNEO Solutions GmbH

### 2. Output with ijk values

The perpendicular is added to a surface at the position of a point.

File format: *.dat

Example of an output with default precision of 15.6.

```
  96.090000      -0.320000      50.000000      0.000000      1.000000      0.000000
 130.690000      -5.660000      50.000000      0.000000      1.000000      0.000000
 123.440000     -33.600000      50.000000      0.000000      1.000000      0.000000
  89.940000     -36.160000      50.000000      0.000000      1.000000      0.000000
  83.610000      -8.040000      50.000000      0.000000      1.000000      0.000000
```

### 3. Output with template

You can customize the output file by using a template. See next chapter 684 on how to create it.

File format: *.dat

## 24.15.4 Creating a template

You can set up a template by defining the following configuration options.

**gtu_points_write_use_template**
Defines whether a template should be used for outputting points values to a DAT file (1) or not (0). Presets the checkbox *Use templates* in the *Options* tab in *Points*. Standard: 0

**gtu_points_write_filename**
Defines the filename (without extension). It would be extended by .pts (without template) or with .dat (with template). Should the template use another extension this could be set by gtu_points_write_template_extension.

**gtu_points_write_template_extension**
Defines the file extension for a file with template. If the option is not set "dat" is used. The option has no impact on writing a file without a template.

**gtu_points_write_template_names**
Defines whether point names should be written to the output file. Default: 0

**gtu_points_write_template_names_fill_up_length**
Fills up the point name to a defined number of characters. Default: 20

**gtu_points_write_template_modelnames**
Defines whether model names should be written to the output file. Default: 0

**gtu_points_write_template_filenames_fill_up_length**
Fills up the model name to a defined number of characters. Default: 20


© 2026 INNEO Solutions GmbH

**gtu_points_write_template_header**
Defines a header for the DAT file.

**gtu_points_write_template_footer**
Defines a footer for the DAT file.

**gtu_points_write_template_line_left**
Defines the start of a line in the DAT file.

**gtu_points_write_template_line_right**
Defines the end of a line in the DAT file.

**gtu_points_write_template_names_split**
Replaces the defined string in the file name by a space character: Default: _

**gtu_points_write_template_seperator**
Defines the separator that is used for separating the values. Standard is space " ".

**gtu_points_autoselect_points_with_rule**
Points are preselected at module startup and can be filtered by name. The preselection at assemblies would be done over all parts and subassemblies. The definition has to be done in RegExp (e.g. ^.*$).

## 24.15.5 Example template

A file with points is to be output as a .csv file. This should have a header.

set options:

`gtu_points_autoselect_points_with_rule=^.*$`
All points are preselected.

`gtu_points_write_filename=@mdln@`
The output should be named after the model (not the selected coordinate system).

`gtu_points_write_template_extension=csv`
The file extension is .csv.

```
gtu_points_write_template_modelnames=1
gtu_points_write_template_names=1
gtu_points_write_template_filenames_fill_up_length=1
gtu_points_write_template_names_fill_up_length=1
```
Model names and point names should be written, not padded.

`gtu_points_write_template_header=File name,Point name,X,Y,Z,`
The following information will be written as the first line in the .csv.

`gtu_points_write_template_seperator=,`
The cells are to be separated by a ; character.

`gtu_points_write_use_template=1`


© 2026 INNEO Solutions GmbH

When opened, the use of the template is enabled.

Outcome:

```
File name,Point name,X,Y,Z,
cri_rl-2.asm,APNT0,      28.000000,      75.500000,     -43.000000,
cri_rl-1.asm,APNT0,      28.000000,      75.500000,      43.000000,
cri_loetnippel.prt,PNT0,      65.500000,      60.000000,     -40.000000,
cri_loetnippel.prt,PNT0,     105.500000,      60.000000,      -0.000000,
cri_loetnippel.prt,PNT0,      65.500000,      60.000000,      40.000000,
cri_loetnippel.prt,PNT0,     -28.000000,      75.500000,     -43.000000,
cri_loetnippel.prt,PNT0,      28.000000,      75.500000,     -43.000000,
cri_loetnippel.prt,PNT0,     -28.000000,      75.500000,      43.000000,
cri_loetnippel.prt,PNT0,      28.000000,      75.500000,      43.000000,
cri_dampfmaschine_2000_skel.prt,PNT0,      11.000000,       6.062178,
3.500000,
cri_dampfmaschine_2000_skel.prt,PNT1,     -11.000000,      -6.062178,
3.500000,
cri_dampfmaschine_2000_skel.prt,PNT2,      11.000000,      35.857312,
0.000000,
cri_dampfmaschine_2000_skel.prt,PNT3,     -11.000000,      23.732956,
0.000000,
cri_dampfmaschine_2000_skel.prt,PNT4,      28.000000,      30.734190,
0.000000,
cri_dampfmaschine_2000_skel.prt,PNT5,     -28.000000,      35.410727,
0.000000,
cri_dampfmaschine_2000_skel.prt,PNT6,      28.000000,      -2.338269,      -
1.350000,
cri_dampfmaschine_2000_skel.prt,PNT7,     -28.000000,       2.338269,      -
1.350000,
```

# 24.16 Export Table One-To-One to Excel

Available in the ribbon menu GENIUS TOOLS

The function creates an Excel file without making use of a template, i.e. tables are always exported exactly as displayed. The Excel file without macros (XLSX file) can be generated from an assembly report (in assembly mode) or a drawing table (in drawing mode).

**Warning**: The Excel export requires Excel 2016 or later on the executing computer. Make sure that no Excel instance is running when you export a table.

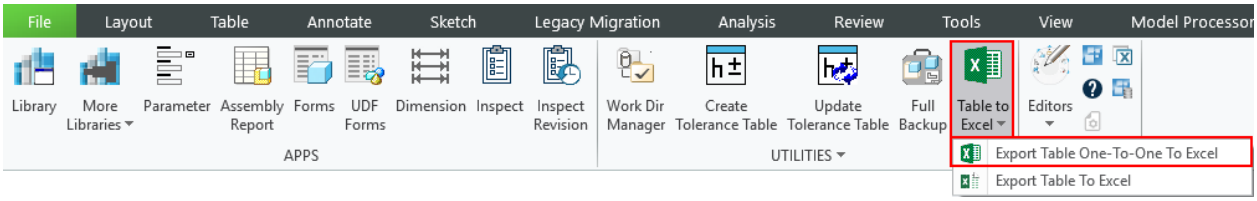
## Starting the program: in drawing and assembly mode

*Export table One-To-One to Excel* via can be started:

– For report tables: via the tools menu of the dialog window *Assembly Report*[45].


© 2026 INNEO Solutions GmbH

– For drawing tables: in drawing mode via the *GENIUS TOOLS* in the ribbon menu in the *UTILITIES* section.




*Starting via the GENIUS TOOLS in the ribbon menu (drawing mode)*

After starting the function, select a table in your current drawing. Merged cells will not be considered, the contents of the individual cells will be adopted as-is.

## Configuring export settings

The Excel file is saved by default in the first level of the working directory with the name of the assembly or the drawing. The name can be changed.

Further export settings can be set with these configuration options:

**gtu_table_to_excel_open_export**
Defines whether the excel file will be opened after the export (1) or not (0). Default: 1

**gtu_table_to_excel_run_check**
Defines whether to check if an Excel session is open (1) or not (0). Default: 1

## Configuring export settings of drawing tables

For the export of drawing tables the following additional configuration options can be set:

**gtu_table_to_excel_autosave**
Defines whether the Excel table to be created is written to the defined report directory without a save dialog (1). Default: 0

**gtu_table_to_excel_report_folder**
Defines the directory where Excel reports are saved from *Export Table One-to-One to Excel* and from *Export Table to Excel*.

# 24.17 Export Table to CSV

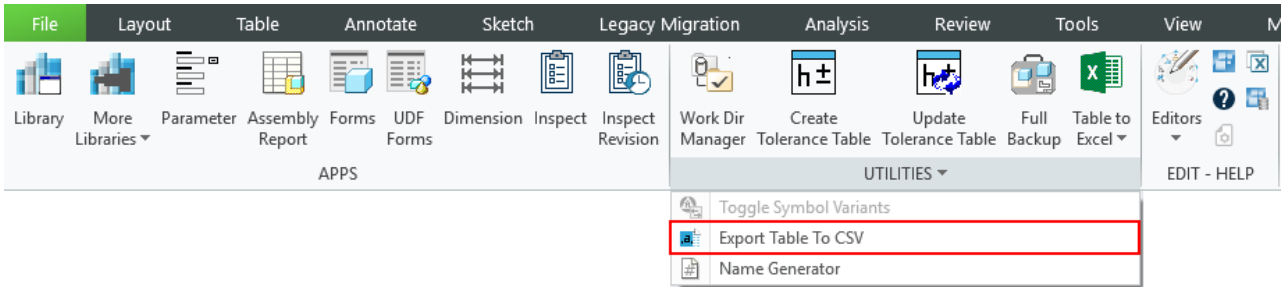
Available in the ribbon menu GENIUS TOOLS

*GENIUS TOOLS Export Table to CSV* exports a selected drawing table, or a table template with data from a drawing. In both cases, the result is written to a CSV file.


© 2026 INNEO Solutions GmbH

# Program start: in drawing mode

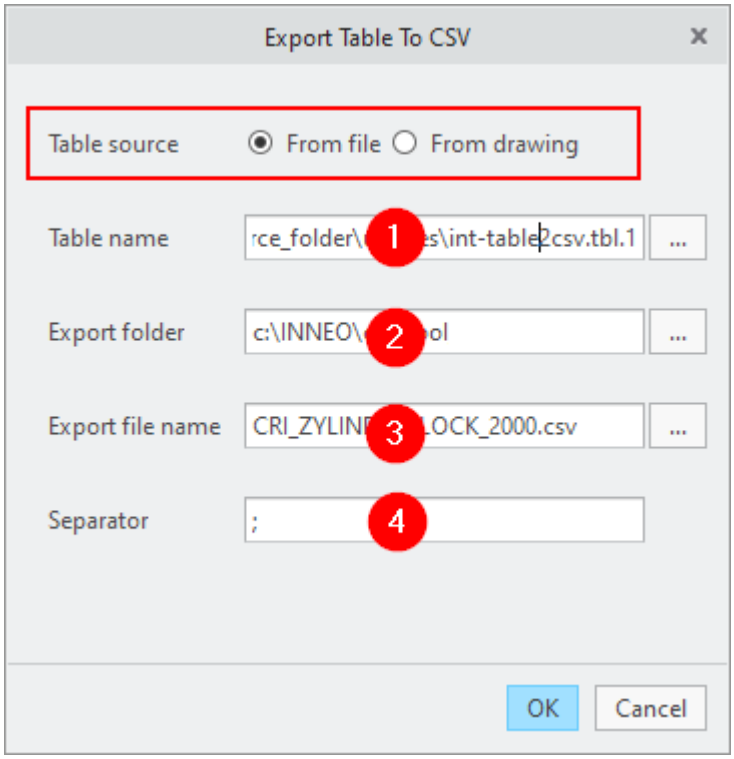
To start *Export Table to CSV*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section when having opened a drawing.




*Starting via the GENIUS TOOLS in the ribbon menu*

# Create report

You can choose from two sources to retrieve the data you want to export: from a TBL file (as a template) or from a table that has already been placed on a drawing.




*Selecting a table source*

– **Report from file:** adds a template table (TBL file) to the current drawing which is filled using the table rules and exported to a CSV file. After completion of the export, the table is removed from the drawing.

– **Report from drawing:** exports the selected drawing table to a CSV file The table is not changed.


© 2026 INNEO Solutions GmbH

## Configuring export settings

Both options can be adapted in the following configuration options. (See screenshot.)

1. Name of the table, only when selecting "From file": **gtu_table_to_csv_table_template**
   Specifies the TBL file that will be used as default for data export. Default path: *%GT_RESOURCE_FOLDER%/utilities/int-table2csv.tbl*
2. Export folder: **gtu_table_to_csv_export_folder**
   Specifies the directory to which the exported CSV file will be written.
3. Export file: **gtu_table_to_csv_export_file**
   Defines the name of the exported CSV file. Can contain variables. Default: file name of model
4. Separator: **gtu_table_to_csv_export_sep**
   Defines the separator between the values. Default: ; (semicolon)

Furthermore, you can define the way the CSV file will be encoded:

**gtu_table_to_csv_write_file_as_utf8**
Defines whether the CSV file is written as ASCII (0) or UTF8 (1). Start value: 1

**gtu_table_to_csv_write_file_as_utf8_with_bom**
Defines whether a CSV file written with UTF8 is additionally encoded with Byte Order Mark (BOM) (1) or not (0). Initial value: 0. For this, above option `gtu_table_to_csv_write_file_as_utf8` must be set to 1.

## Configuration of the ribbon menu

Use the configuration option `gtu_start_table_to_csv` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

# 24.18 Export Table to Excel

Available in the ribbon menu GENIUS TOOLS

The function *Export Table to Excel* exports a report table, e.g. a Creo drawing table into an Excel file, CSV file or PDF file. The data to be exported is specified by using an Excel template.

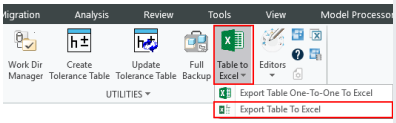
**Warning**: The Excel export requires Excel 2016 or later on the executing computer. Make sure that no Excel instance is running when you export a table.

## Starting the program: in drawing and assembly mode

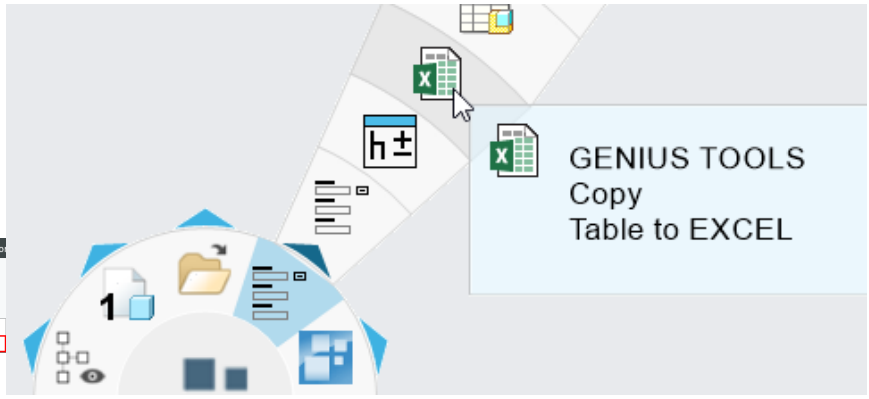
Start *Export Table to Excel* in the different modes.


© 2026 INNEO Solutions GmbH

– For drawing tables (drawing mode):
via the *GENIUS TOOLS* ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).



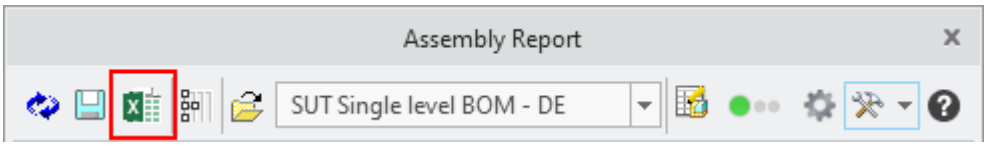

*Starting via the GENIUS TOOLS in the ribbon menu*




*Starting via Quick Access*

– For report tables in *Inspect*[190] and *Inspect Revision* (drawing mode):
in the command bars of the dialog windows

– For report tables in *Assembly Report*[45] (assembly mode):
in the command bar of the dialog window




– For the overview table in *Annotation Info*[642] :
in the command bar of the dialog window

**Please note:** When exporting via the utility *Annotation Info*, the name of the export template is read from the configuration option `gtu_annotation_info_excel_template`. For all other use cases, the name of the export template is read from the configuration option `gtu_table_to_excel_std_template_name`.

## Configuration

The configuration option `gtu_table_to_excel_open_export` defines, whether the Excel file opens after export (1) or not (0) (default is 1=On/not hidden).

## 24.18.1 Creating reports

Follow the dialog to start the export of drawing tables or other report tables.

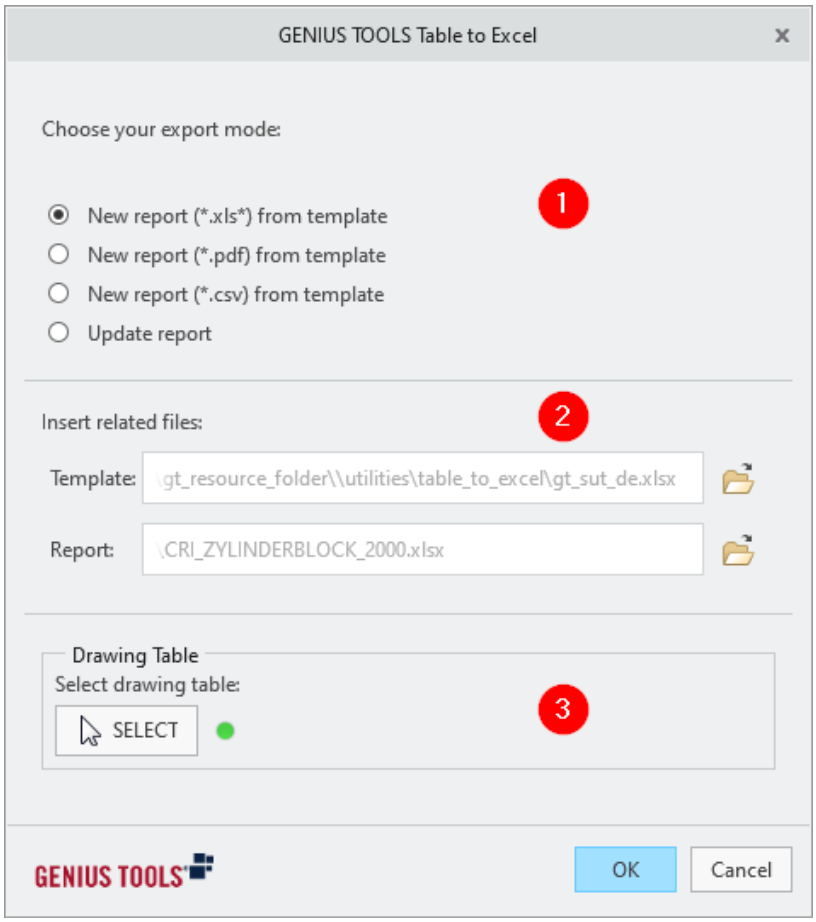



*Table to Excel dialog*

## 1. Choosing an export mode

Choose from these options:

**New report (*.xls*) from template:** This option fills an Excel template with the contents of a drawing table or a report table and saves the new Excel document. In an Excel document with macros (.xlsm), the macros are not included in the output document.
**New report (*.pdf) from template:** This option fills a PDF template with the contents of a drawing table or a report table and saves the new PDF document.
**New report (*csv) from from template:** This option fills a CSV template with the contents of a drawing table or a report table and saves the new CSV document.

**Update report:** This option updates an existing Excel document by writing values to the column defined by the comment line. To update, select the Excel document.

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.

**Please note**: An existing Excel file can only be updated if the cell comments have not been removed (manually, or automatedly via the configuration options).


© 2026 INNEO Solutions GmbH

**Deactivating the update function:** You can gray out and deactivate the function *Update report* by setting the configuration option *gtu_table_to_excel_erase_replaced_comments* to 1. Default: 0=Function is activated.

### 2. Specifying the files

**Template:** Select a configured Excel template[693], PDF template or CSV template for the report. The default paths vary according to the various *GENIUS TOOLS* extension modules, see section below.

**Please note:** Images saved in a CSV template cannot be exported and displayed in the CSV report.

**Report:** Specify the destination directory for the report to be created.

### 3. Select drawing or report table

This area is divided into the following *GENIUS TOOLS* extension modules:

– in drawing mode:

Click *Select* and then select a drawing table in Creo.

– in *Inspect*:

For inspection symbols: Select a configuration file (XML)[207].

For revision / snapshot histories: Select a configuration file with history (XML).[233]

– in *GENIUS TOOLS* Assembly

Select a report definition (XML).[42]

– reports from different *GENIUS TOOLS* extension modules:

Several selection fields will open, if you select a template that contains more than one component acronym, e.g. *gti* and *gti_rev*, see Export data from several GENIUS-TOOLS-components.[701]

## Configuring default paths

Default paths in the dialog can be changed in the various *GENIUS TOOLS* extension modules as follows.

**For the template file**

– in drawing mode:
*<gt_resource_folder>\utilities\table_to_excel*
Can be changed with `gtu_table_to_excel_template_folder`.

– in *Inspect*:

For inspection symbols: *%gt_resource_folder%\inspect\gti_inspection_template_de_en.xlsx*
Can be changed with `gti_excel_template`.


© 2026 INNEO Solutions GmbH

For revision histories: *%gt_resource_folder%\inspect*
Can be specified with `gti_revision_folder`.
– The default template *gti_revision_template_de_en.xlsx can* be changed with `gti_revision_excel_template`.

– in *GENIUS TOOLS* Assembly:
*%gt_resource_folder%assembly\gt_assembly.xlsx*
Can be changed with `gta_export_template`.

**For the output file**
– In drawing mode:
Can be set with `gtu_table_to_excel_report_folder`.

– in *Inspect*, for inspection characteristics:
Can be set with `gti_excel_export_path`.

– in *Inspect Revision*:
Can be set with `gti_revision_excel_export_path`.

– In *GENIUS TOOLS* Assembly:
Can be set with `gta_export_path`.

**gtu_table_to_excel_run_check**
Specifies whether to check if an Excel session is open. Default: 1=yes

## 24.18.2 Create template

Exporting a report to Excel requires an Excel template in XLSM or XLSX format (Excel file with and without macro functions). This function is available in the components *Assembly Report* and *Inspect*.

An export template is an excel table, where certain cells contain comments that consist of a component acronym and a fill command. By using various component acronyms you can create templates with data from different *GENIUS TOOLS* extension modules.[701]

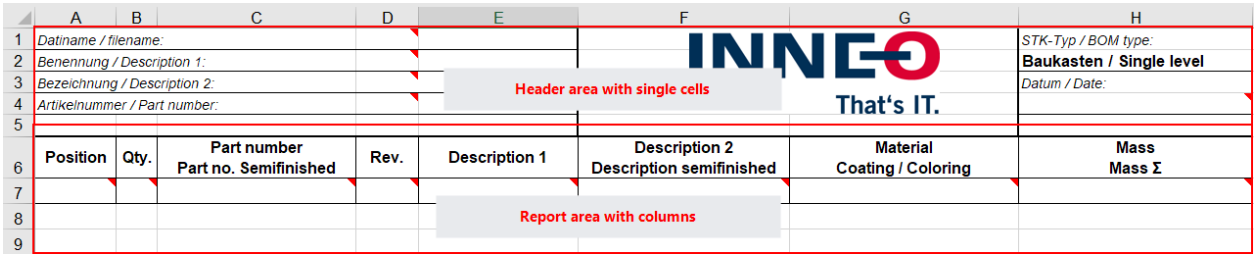
The report output format is the same as the template format but can be changed.

### 24.18.2.1 Step-by-step guide

Create a separate excel file for each report template.

**Design table**
1. Open a new Excel file.
2. Design a table by formatting cells and rows and fill in the headings as needed. Be careful to distinguish between the header and the report area:
   – head parameters and variables fill a single cell
   – report parameters fill a column


© 2026 INNEO Solutions GmbH

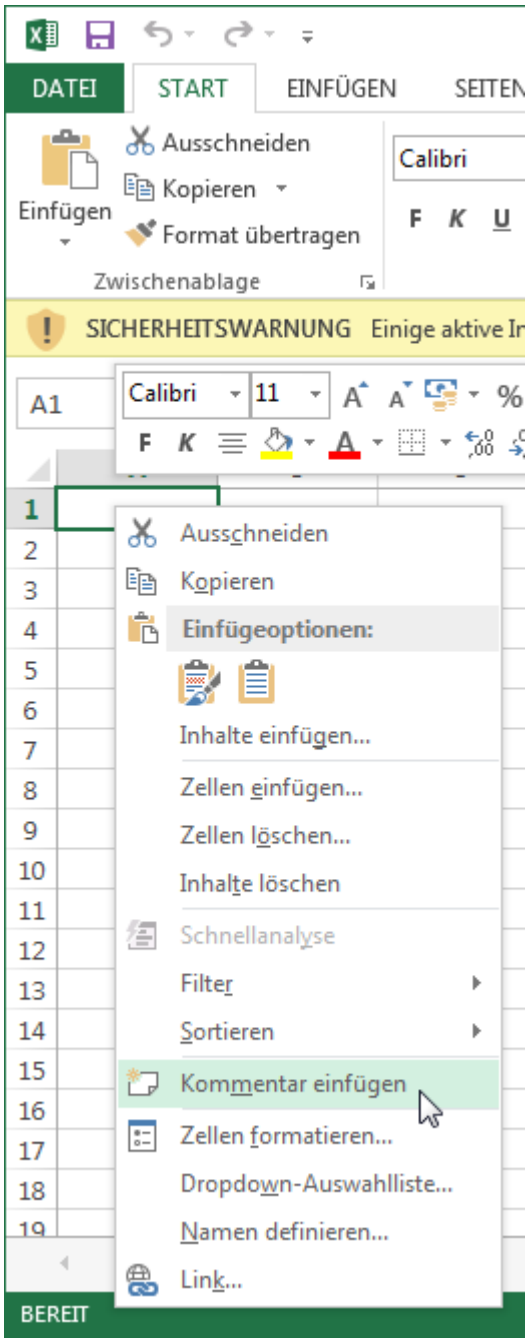


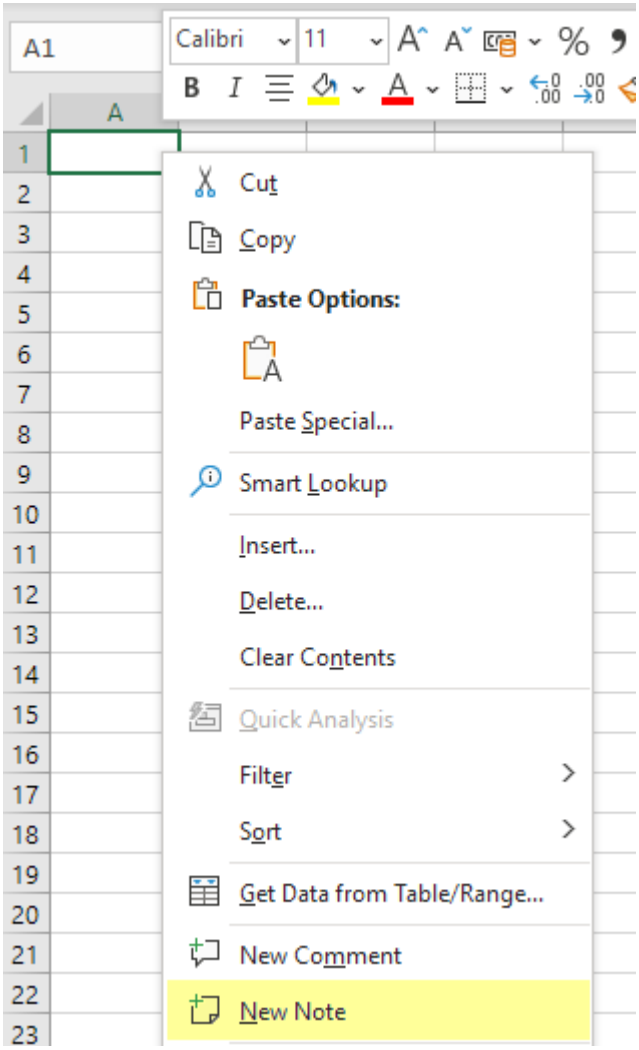

### Creating comments

3. Create comments by using the Excel context menu. The text of the comment determines the value that will be copied to a single cell or the cells of a column.

**Please note:** This function is called *Comment* until Excel 2019, from Microsoft 365 onwards it is called *Note*.

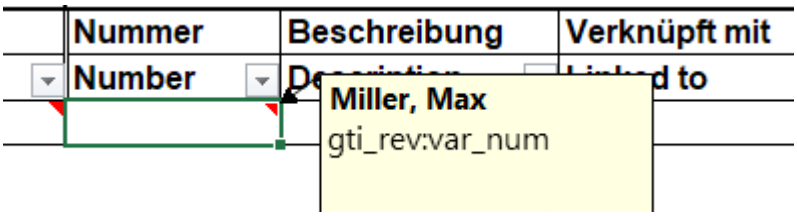

© 2026 INNEO Solutions GmbH




**Content of comments**

4. The first line of a comment is the editor name and will be ignored. The second line will be analyzed and fills the cell.




5. The following content is possible in comments:

   – for a single cell: head parameters and variables,


© 2026 INNEO Solutions GmbH

– for a column: report parameter

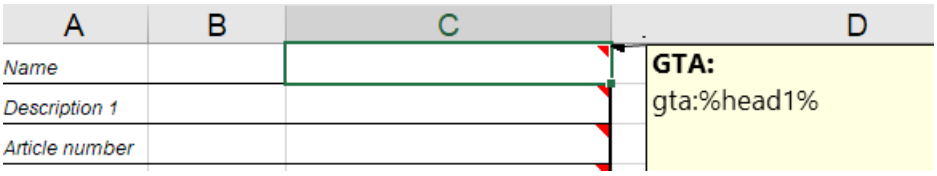
**Head parameters**

Head parameters are model and drawing-specific parameters in the notation `%ParameterName%`, e.g. `%DESCRIPTION_1_EN%`. Head parameters are written into a single field.

6. Enter the text of the comment in a cell of the head area

– for *Assembly Report*: in the notation *gta:%head*%* (Replace * with the position of the head parameter in the editor.)

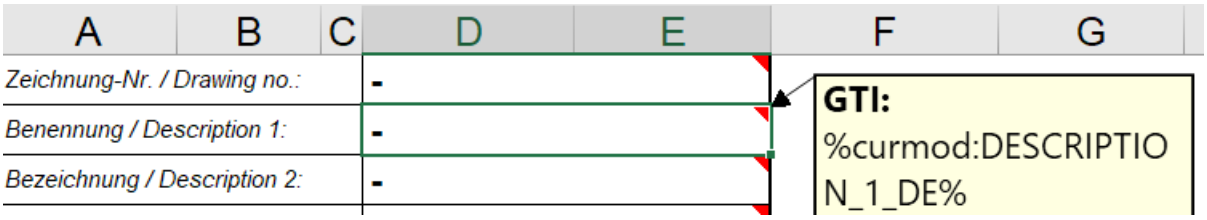
Example: Head parameter 1 = *gta:%head1%*, head parameter 2 = *gta:%head2%* etc.




*Head parameters with numbering for Assembly Report*

– for *Inspect / Inspect Revision*: Variables without component acronym

Example: *%DRAWING_NO%*




*Head parameter for Inspect*

**Report parameters**

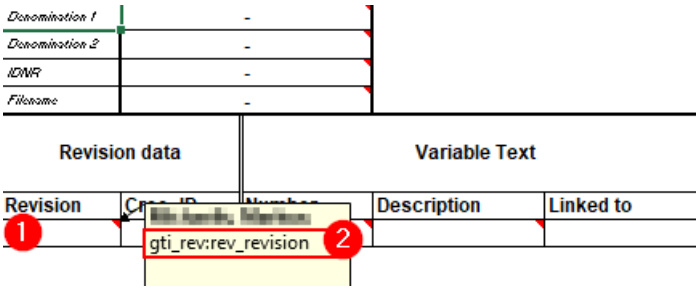
Report parameters fill cells in a column. The length of the Excel report depends on the number of rows displayed.

Report parameters consist of the name of the component (acronym) and the fill command.

| Components | Acronym | Fill command |
|---|---|---|
| *Assembly Report* | gta: | %ruleParameter% to give out position parameter<br>%col*% for each column, e.g. %col1% |
| *Inspect* | gti: | See table.[203] |
| *Inspect Revision* | gti_rev: | See table.[241] |
| *Annotation Info* | gtai: | See table[646]. |

7. Enter the comment text (2) for report parameters in the first value line (1) of the desired column.


© 2026 INNEO Solutions GmbH

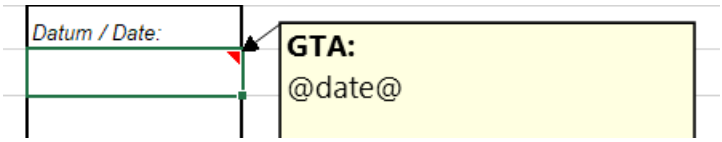



*Comment for column filling in Inspect Revision*

8. Define a report parameter column by column.
   – for *Assembly Report*:
   *gta:%ruleParameter%* as well as *gta:%col*%* for each column, e.g. *gta:%col1%*. Without the display of the position parameter, the counting of the columns starts at 0.
   – for *Inspect / Inspect Revision*:
   Use the acronyms and commands as shown in the table of step 5.
   See Table for GTI[203] and Table for GTIR[241]

**Variables**
Variables fill a single cell with general information.

9. Use the available variables of GENIUS TOOLS for Creo if you wish, e.g. date and time information as well as Creo object information, in the notation `@variable@`. See Usage of variables[913].




**Please note**: The variables *@feat_id@*, *@selmdln@* and *@selmdlpath@* are not supported!

10. Save the template
    – for *Assembly Report* under:
    *%gt_resource_folder%\assembly*
    – for *Inspect / Inspect Revision* under:
    *%gt_resource_folder%\inspect*

## 24.18.2.2 More options for templates

### Output drawing table rows

Output one or more drawing table rows.

`T:RowNumber(row):ColumnNumber(column):NumberFollowingRows:NumberHeaderRows` (for split tables)


© 2026 INNEO Solutions GmbH

Always pay attention to the table orientation in Creo Parametric (the table direction in the table properties)! The first cell of a drawing table is always the initial point of the table.

**Examples**

`T:1:1` - Copies the cell 1:1 (first row:first column (the counter always starts at 1) into the Excel cell.

`T:1:1:all` - Copies the cell 1:1 and all cells 1 of the following rows (2:1, 3:1...) to the associated Excel table rows (each in the column below).

`T:1:1:3` - Copies the cell 1:1 and the first cell of the three following rows (2:1, 3:1, 4:1 - if so many exist) to the following Excel table rows.

`T:1:1:all:3` - Copies all cells of position 1 after the first three rows to the associated Excel table rows.

## Output empty cells

Furthermore, it is possible to copy empty cells that function as placeholders into the report table. In this case, only the frame is taken over.

`empty:NumberOfHeaders:NumberOfValueLines:NumberOfFootlines`

**Examples**

`empty:1` - Copies a line with a border from the Excel template.

`empty:6` - Copies 6 lines with border from the Excel template.

`empty:2:4` - Copies 2 lines with border from the Excel template.

`empty:1:all:2` - Copies all cells except the last two rows with borders from the Excel template.

## Rules for replacing and expanding cell contents

An additional worksheet can be added to the Excel template file, in which rules for replacing and / or expanding cell contents are defined. The name of this worksheet is stored in the configuration option `gtu_table_to_excel_rule_sheet_name` (Default: *Rules*). Use the configuration option `gtu_table_to_excel_export_rule_sheet` to define whether this worksheet is exported (Default: 1=yes).

All rules must be filled in according to the following scheme, starting at line 1:

| | A | B | C |
|---|---|---|---|
| 1 | **Column** | **Search text** | **Rule** |
| 2 | G | Ⓔ | A:+.1\|G:replace("Ⓔ"," Max "col(K)" (GP)") |
| 3 | G | Ⓔ | A:+.2\|G:replace("Ⓔ"," Min "col(J)" (LP)") |

*Example rules for changing cell contents in the Excel worksheet "Rules"*

– **Column A**
Specify the column in which to search for the search text, here: column G


© 2026 INNEO Solutions GmbH

– **Column B**
  Specify the search text, here: Ⓔ

– **Column C**
  Definition of the rule(s) for changing the cell contents. You can add further information to the cell text and / or replace cell contents.

**Please note:** The replacement text within the quotation marks must not contain a **|** or a **:** as these characters are used as separators.

## Rule: Expanding cell contents

Additional information can be added to a table cell by applying a rule with the following structure:

A:+.1

**Effects of the rule**
– The entry in column A is extended by the value .1.

## Rule: Replacing cell contents

Information in a table cell can be replaced by information from other cells by applying a rule with the following structure:

G:replace("Ⓔ"," Max "col(K)" (GP)")

**Effects of the rule**
Depending on whether the search text (here: Ⓔ) is found in the column to be searched (here: column G), the rule can have the following effects:

– **Case 1:** In a cell in column G, the value Ⓔ is found and replaced by: Text Max `[for maximum]`, value from column K on this line and the text (GP)

– **Case 2:** The value Ⓔ is not found in a cell in column G. The cell contents are not changed.

## Combining rules

**Expanding cell contents** and **Replacing cell contents** can be used as one rule. To do this, the two rules must be connected with a vertical bar |.

A:+.1|G:replace("Ⓔ"," Max "col(K)" (GP)")

**Effects of the rule**
Either both rules or neither of the two rules are executed. Depending on whether the search text (here: Ⓔ) is found in the column to be searched (here: column G), the rule has the following effect:

– **Case 1:**

  · The entry in column A is extended by the value .1.


© 2026 INNEO Solutions GmbH

- In a cell in column G, the value Ⓔ is found and replaced by: Text Max `[for maximum]`, value from column K on this line and the text (GP)

– **Case 2:**

- The entry in column A is not changed.
- The value Ⓔ is not found in a cell in column G. The cell contents are not changed.

## Multiple rules on a line

Multiple rules can be defined for a line. In this case, the line is duplicated in the cell.

A:+.1|G:replace("Ⓔ"," Max "col(K)" (GP)")
A:+.2|G:replace("Ⓔ"," Min "col(J)" (LP)")

**Effects of the rule**
Either both rules or neither of the two rules are executed. Depending on whether the search text (here: Ⓔ) is found in the column to be searched (here: column G), the rule has the following effect:

– **Case 1:**

- The entry in column A is extended by the value .1.
- In column A, the initial value is additionally extended by .2 on a newly added line.
- In a cell in column G, the value Ⓔ is found and replaced by: Text Max `[for maximum]`, value from column K on this line and the text (GP)
- The text Min `[for minimum]`, value from column J on this line and the text (LP) are added to this cell in the newly added line.

– **Case 2:**

- The entry in column A is not changed.
- The value Ⓔ is not found in a cell in column G. The cell contents are not changed.

## Result

| | A | B | C |
|---|---|---|---|
| 1 | Column | Search text | Rule |
| 2 | G | Ⓔ | A:+.1\|G:replace("Ⓔ"," Max "col(K)" (GP)") |
| 3 | G | Ⓔ | A:+.2\|G:replace("Ⓔ"," Min "col(J)" (LP)") |

*Rules to be applied in the "Rules" worksheet*

| | A | B | C | G |
|---|---|---|---|---|
| 1 | Zeichnung-Nr. / Drawing no.: | | | |
| 2 | Benennung / Description 1: | | | |
| 3 | Bezeichnung / Description 2: | | | |
| 4 | Dateiname / File name: | | | |
| 5 | Nr. | Blatt | Raster | Bemaßungstext |
| 6 | No. | Shee | Grid | Dimension text |
| 7 | 019 | 1 | C5 | 8.5Ⓔ |
| 8 | 020 | 1 | D4 | 4.5 |

*Output without application of rules*

| | A | B | C | G |
|---|---|---|---|---|
| 1 | Zeichnung-Nr. / Drawing no.: | | | |
| 2 | Benennung / Description 1: | | | |
| 3 | Bezeichnung / Description 2: | | | |
| 4 | Dateiname / File name: | | | |
| 5 | Nr. | Blatt | Raster | Bemaßungstext |
| 6 | No. | Shee | Grid | Dimension text |
| 7 | 019.1 | 1 | C5 | 8.5 Max 8.7 (GP) |
| 8 | 019.2 | 1 | C5 | 8.5 Min 8.3 (LP) |
| 9 | 020 | 1 | D4 | 4.5 |

*Output after applying the rules*


© 2026 INNEO Solutions GmbH

# 24.18.2.3 Export data from several GENIUS TOOLS extension modules

By using the component abbreviations you can create templates with different areas or spreadsheets. This is useful in *Inspect*, for example.

Fill fields and columns from different *GENIUS TOOLS* extension modules by creating comments with the corresponding component abbreviations (gta, gti, gti_rev). Otherwise, set up the template according to the general procedure[693].

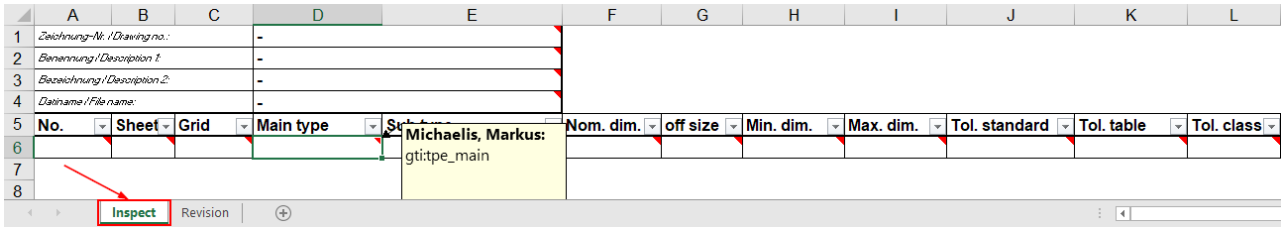
If you select a template with several component abbreviations, the selection fields belonging to the components will open in the lower area.

**Example for the common export of report tables**

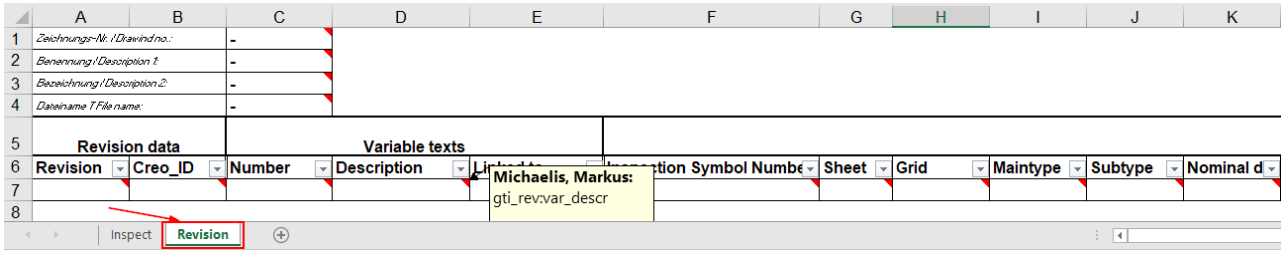
– of inspection characteristics from *Inspect* and

– of Revision histories from *Inspect Revision*

1. Create several Excel tables with different component abbreviations in the comments, here: gti and gti_rev.




*Report data from Inspect with the acronym gti:*

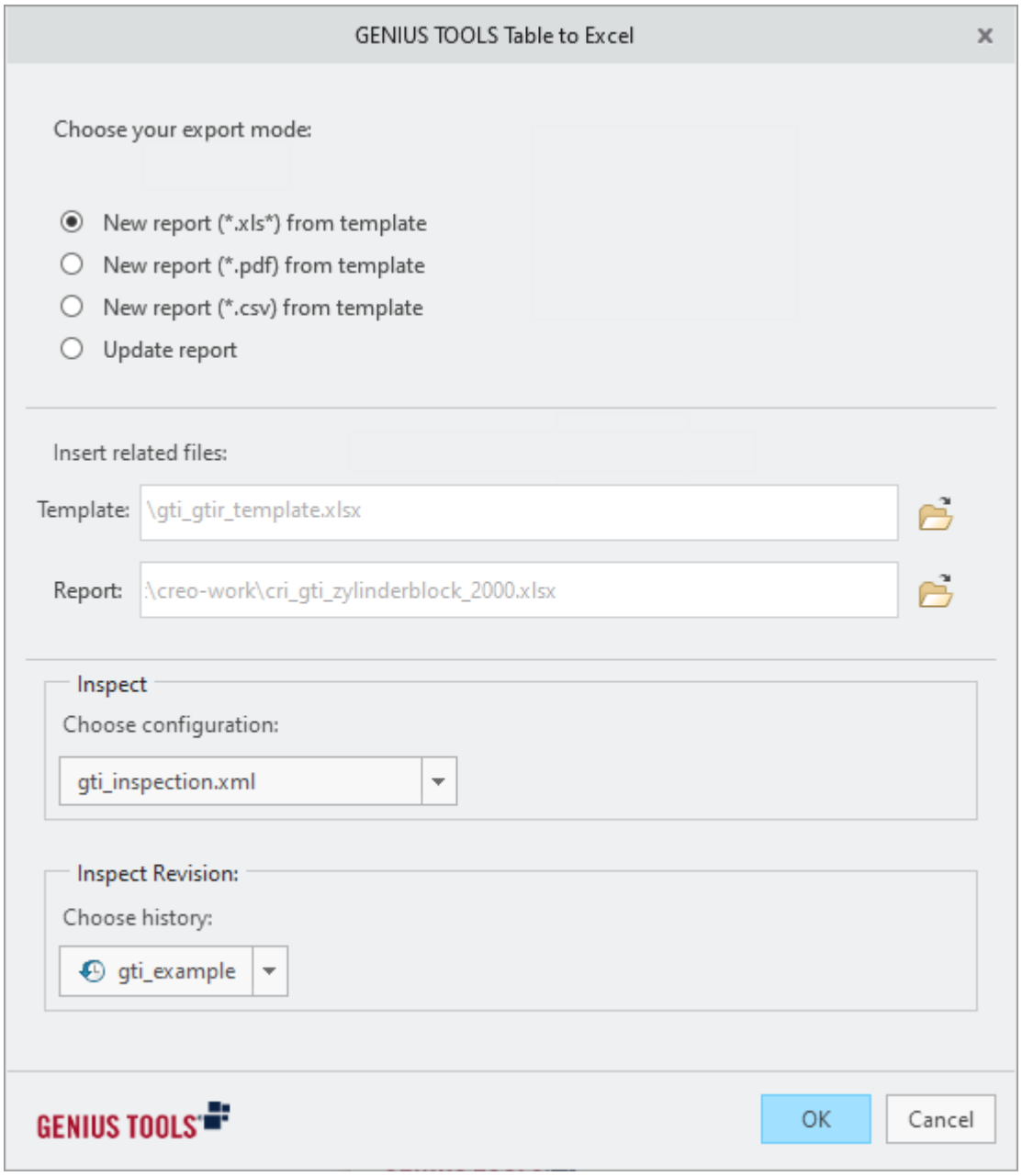



*Report data from Inspect Revision with the acronym gti_rev:*

2. Save the file as *gti_gtir_template.xlsx*

3. Result: When selecting this file as a template, the dialog box opens with selection fields for *Inspect* and *Inspect Revision*.


© 2026 INNEO Solutions GmbH




# 24.19 Extend Relations

Available in the ribbon menu GENIUS TOOLS

*Extend Relations* is a set of extra functions to model relations that define parameters for models and bodies.

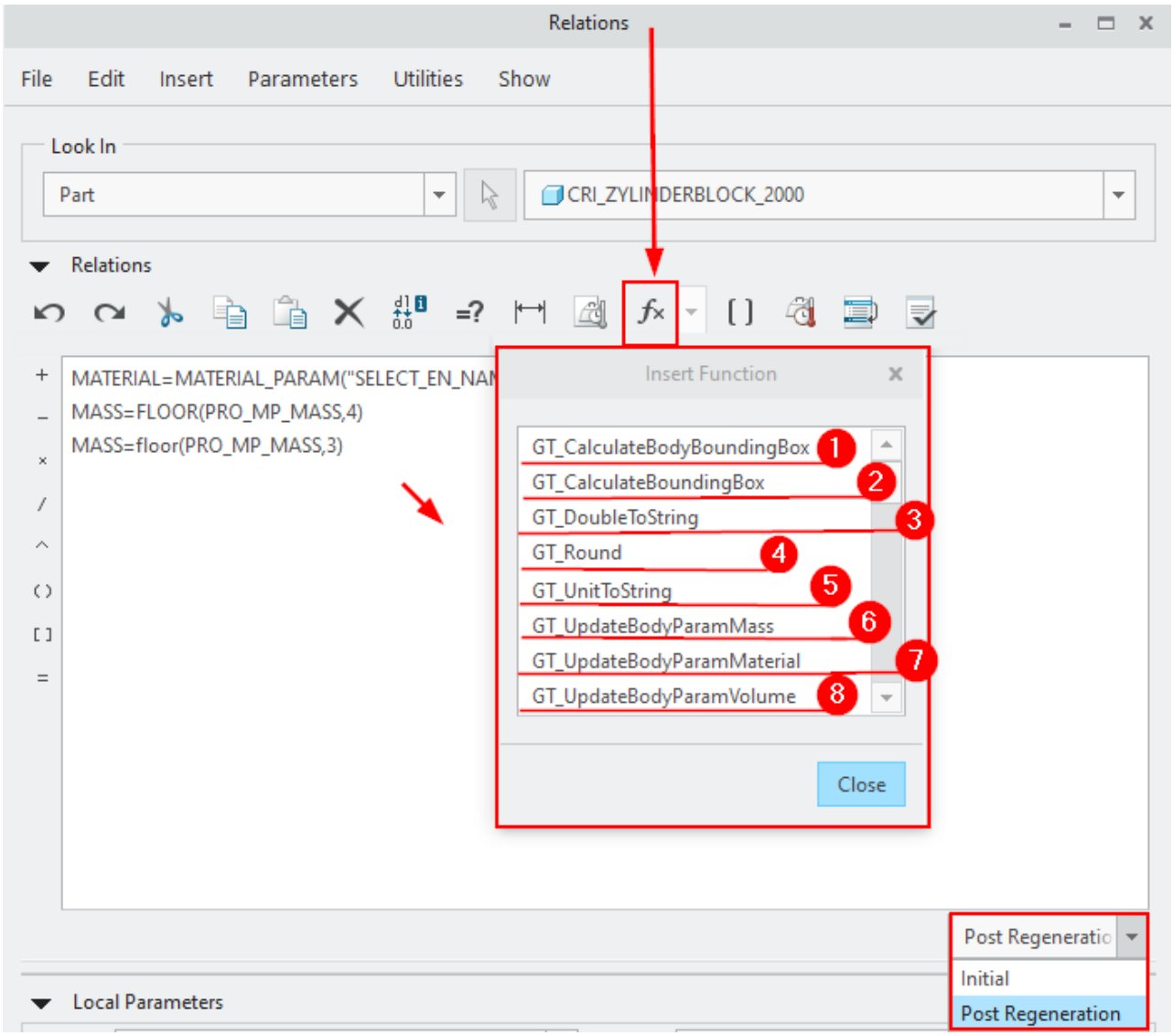
The additional functions are executed at (initially) or after (post) regeneration guaranteeing that all values are up-to-date.


© 2026 INNEO Solutions GmbH

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Adding functions: in part and assembly mode

The additional functions can be found on the *Tools* tab in the Creo's *Relations* dialog. In this Creo dialog, click fx. Take care to select the proper kind of relations – you can choose between *Initial* relations and *Post Regeneration* relations (Below: Post Regeneration). This selection allows you to control the timing of the execution of the relations and thus the regeneration time. *Initial* relations are executed whenever changes are made, but before the model is regenerated, while *Post Regeneration* relations are executed only after the entire model is regenerated.




*Functions that extend relations in the Creo dialog Relations > Insert Function*


© 2026 INNEO Solutions GmbH

# Functions

Use these functions for parameters for bounding boxes (functions 1, 2), for bodies (functions 5, 6, 7), and for rounding values (functions 3, 4):

**1. GT_CalculateBodyBoundingBox("BodyParameter","optFormat")**
A parameter is generated and updated for each body in a model with the values of a bounding box.

"optFormat" is optional: Write in the notation "x: %.Numberf y: %.Numberf z: %.Numberf"

Example: `GT_CalculateBodyBoundingBox("DIMENSION", "x: %.1f y: %.1f z: %.1f")`

**2. GT_CalculateBoundingBox("ModelParameter","optFormat")**
A parameter is generated and updated for the model with the values of the bounding box of the model..

"optFormat" is optional: Write in the notation "x: %.Numberf y: %.Numberf z: %.Numberf"

Example: `GT_CalculateBoundingBox("DIMENSION", "x: %.1f y: %.1f z: %.1f")`

**3. GT_DoubleToString()**
Function rounds parameter values and converts the received numbers into strings. See Rounding. 706

**4. GT_Round()**
Function rounds parameter values. See Rounding. 706

**5. GT_UnitToString()**
Function converts the unit of a parameter into strings.

Example: `MASS_UNIT = GT_UnitToString("MASS")`

**6. GT_UpdateBodyParamMass ("BodyParameter",optMultiplier,optDecimalPlaces)**
A body parameter with the value of the body mass is created and updated for each body.

The entries *optMultiplier* and *optDigitsAfterComma* are optional. If not defined *optMultiplier* = 1.0 and *optDigitsAfterComma* is not rounded.

Example: The multiplier 0.001 is used to convert the default setting "kg" to "g".

| Input | Result |
|---|---|
| GT_UpdateBodyParamMass ("MASS", 0.001, 2) | multiplies the body mass by 0.001<br>rounds to 2 decimal places<br>writes the result into the parameter "MASS" |

**7. GT_UpdateBodyParamMaterial("BodyParameter", "MaterialParameter")**
A parameter is generated and updated for each a body with the value of the material parameter of the body's material.


© 2026 INNEO Solutions GmbH

A body parameter with the value of the body mass is created and updated for each body.

Example: `GT_UpdateBodyParamMaterial("MATERIAL","SELECT_EN_NAME")`

**8. GT_UpdateBodyParamVolume ("BodyParameter",optMultiplier,optDecimalPlaces)**

A body parameter with the value of the body volume is created and updated for each body.

The entries *optMultiplier* and *optDigitsAfterComma* are optional. If not defined *optMultiplier* = 1.0 and *optDigitsAfterComma* is not rounded.

Example:

| Input | Result |
|---|---|
| GT_UpdateBodyParamVolume ("VOLUME") | writes the result into the parameter "VOLUME" |

# Configuring display

Use the following configuration options to change the standard display of the functions in the *Insert function* dialog.

**gtu_start_relationExtension**
Defines whether all commands of the *Extend Relations* utility are shown (1) or not (0). Default: 1.

**gtu_relationextension_calculateBoundingBox**
Defines whether the commands *GT_CalculateBodyBoundingBox()* and *GT_CalculateBoundingBox()* are shown (1) or not (0). Default: 1.

**gtu_relationextension_doubleToString**
Defines whether the rounding command *GT_DoubleToString* is shown (1) or not (0). Default: 1.

**gtu_relationextension_round**
Defines whether the rounding command *GT_Round* is shown (1) or not (0). Default: 1.

**gtu_relationextension_unitToString**
Defines whether the converting command *GT_UnitToString* is shown (1) or not (0). Default: 1.

**gtu_relationextension_updateBodyParamMass**
Defines whether the body mass command *GT_UpdateBodyParamMass* is shown (1) or not (0). Default: 1.

**gtu_relationextension_updateBodyParamMaterial**
Defines whether the body material command *GT_UpdateBodyParamMaterial* is shown (1) or not (0). Default: 1.

**gtu_relationextension_updateBodyParamVolume**


© 2026 INNEO Solutions GmbH

Defines whether the body volume command *GT_UpdateBodyParamVolume* is shown (1) or not (0). Default: 1.

# Rounding

The functions *GT_Round* and *GT_DoubleToString* round numerical values according to IEEE 754. *GT_DoubleToString* converts a number into a string.

## GT_Round(Value,RoundDigits)

Enter the name of the parameter (e.g. LENGTH) or a numerical value in the parentheses of the function as well as the decimal places to be rounded.

Examples for parameter LENGTH=17.3845:

| Input | Results |
|---|---|
| A=GT_Round(17.3845,1) | 17.400000 |
| B=GT_Round(LENGTH,1) | 17.400000 |
| C=GT_Round(LENGTH,2) | 17.380000 |
| D=GT_Round(LENGTH,3) | 17.385000 |

## GT_DoubleToString(value,"%[WIDTH][.PRECISION][f]"",")

Function rounds parameter values according to IEEE 754 and converts the obtained numbers into strings. Enter in the parenthesis of the function:

**Value:** name of the parameter or numerical value

**[WIDTH]:** This number specifies the minimum number of all characters of the string. Characters can be digits, commas and spaces. (Example: The number 17.38 has 5 characters).

– if this specification is less than or equal than the number of all digits and the comma, the specification has no effect on the representation of the string. (In the example: E, F)

– if this specification is greater than the number of all digits and the comma, empty spaces are inserted at the beginning. (In the example: J, I, K)

**[.PRECISION]:** Number of digits after the comma

**[f]:** Formatdefinition:

– f: float-point definition

– e: Exponential definition

– g: Shortest definition (f or e)

**",":** A replacement character for the . (dot) can be specified here, e.g. a comma.
Examples of parameters LENGTH=17.3845:


© 2026 INNEO Solutions GmbH

| Input | Result | Explanation |
|---|---|---|
| A=GT_DoubleToString(LENGTH,"%f") | 17.384500 | Number as floatpoint with pending zeroes (6 places) |
| B=GT_DoubleToString(LENGTH,"%e") | 1.738450e+01 | Number as exponential with pending zeroes (6 places) |
| C=GT_DoubleToString(LENGTH,"%12f") | 17.384500 | Number with WIDTH of 12 |
| D=GT_DoubleToString(LENGTH,"%.1f") | 17.4 | **1**: One digits after dot |
| E=GT_DoubleToString(LENGTH,"%.2f") | 17.38 | **2**: Two digits after dot |
| F=GT_DoubleToString(LENGTH,"%.2f",",") | 17,38 | **2**: Two digits after comma as separator |
| H=GT_DoubleToString(LENGTH,"%7.2f") | 17.38 | **7**: WIDTH of 7 and 2 digits after dot<br>**2**: Two places after dot |
| I=GT_DoubleToString(LENGTH,"%.1e") | 1.7e+01 | **1**: One digits after dot |
| J=GT_DoubleToString(0.000001,"%g") | 1e-07 | Shortest definition |

*G is not generated because it is a PTC constant (gravitational constant 9.8)*

# 24.20 Extended Dimension Functions

Available in the ribbon menu GENIUS TOOLS

In the dialog *Extended Dimension Functions* a selected dimension can be quickly increased or decreased by a defined value. The modification is made on the nominal value.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Opening the function: in part and assembly mode

The following conditions must be fulfilled:

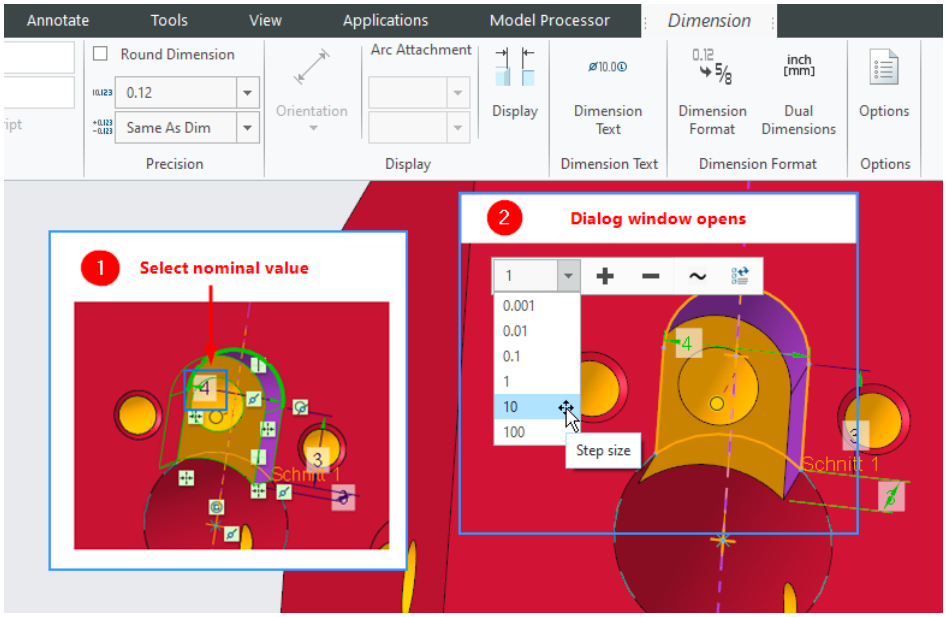

© 2026 INNEO Solutions GmbH

– a dimension must be selected,

– the selected dimension must not be a measured dimension,

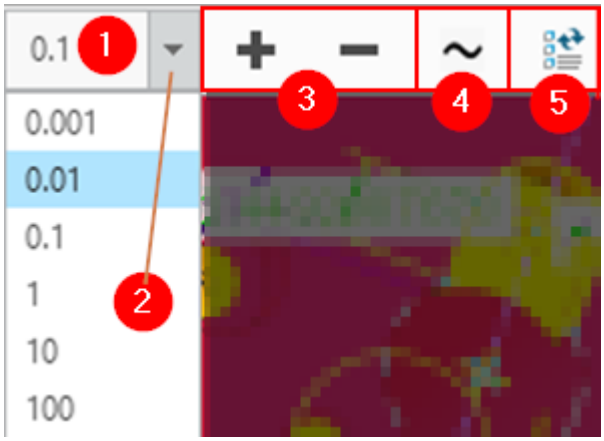
– the selected dimension must not be controlled by a relationship.

The dialog opens above the selected value.




**Please note:** If the dimension is set by hand in the *Dimension* tab in the *Value* group, this can collide with the advanced dimension functions.

## User interface




*Dialog Extended Dimension Functions*

1. Current step size
2. Dropdown window for selecting step size
3. Increasing/ decreasing the value by one in the selected step size
4. Round to the digit of the selected step size


© 2026 INNEO Solutions GmbH

5. Regenerate model

## Configuration of the ribbon menu

Use the configuration option `gtu_start_extendedDimensionFunctions` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

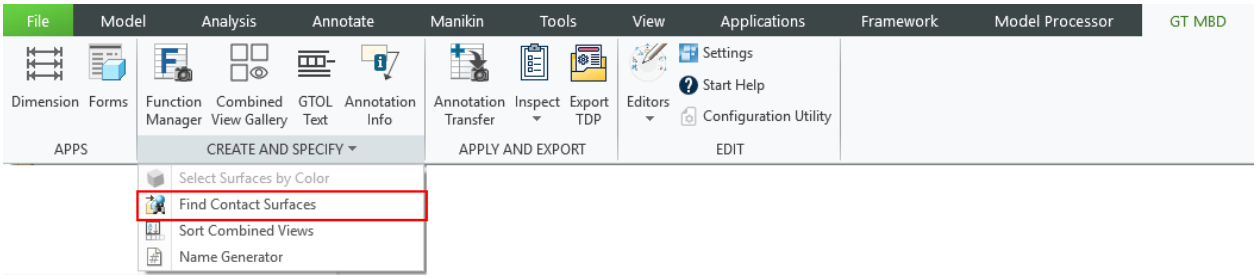
# 24.21 Find Contact Surfaces

Available in the ribbon menu GT MBD

The function *Find Contact Surfaces* is a pure analysis tool that searches for the adjacent surfaces to a selected surface (= contact surfaces). These contact surfaces depend on the assembly in which the part is used. Open *Find Contact Surfaces* and then click on the surface you want to find the contact surfaces for.

## Starting the program: in assembly mode

1. To start *Find Contact Surfaces*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




*Starting via the GT MBD in the ribbon menu*

2. The Creo selection dialog opens. Select the part in the assembly whose contact surfaces you want to see. The contact surfaces are searched for. How long this takes depends on the size of the assembly. The progress is displayed. When the search is complete, the window closes. No other windows appear.

**Please note:** How long this takes depends on the size of the assembly in which the part is being inspected.

## Displaying found contact surfaces

3. To view the found contact surfaces, open the part in part mode.


© 2026 INNEO Solutions GmbH

4. Go on to open the function *Select Contact Surfaces*[734]. This function is the equivalent to *Find Contact Surfaces* in part mode. Use *Select Contact Surfaces* to view the found contact surfaces and select and color them as desired.

# 24.22 Find References

Available in the ribbon menu GT MBD

*Find references* enables the automatic assignment of references to dimensions. The function attempts to find the corresponding semantic references for displayed dimensions. Any references that are not found can be assigned manually. *Find references* generates the semantics for Creo Parametric model dimensions that represent ISO-GPS dimensioning.

The following requirements must be fulfilled for a dimension to be included in the list:

– Type of annotation: driving, driven or reference dimension

– Type of dimensioning: linear, radial, diameter

– Dimensions must be visible in the model

– no (reference) ordinate dimension

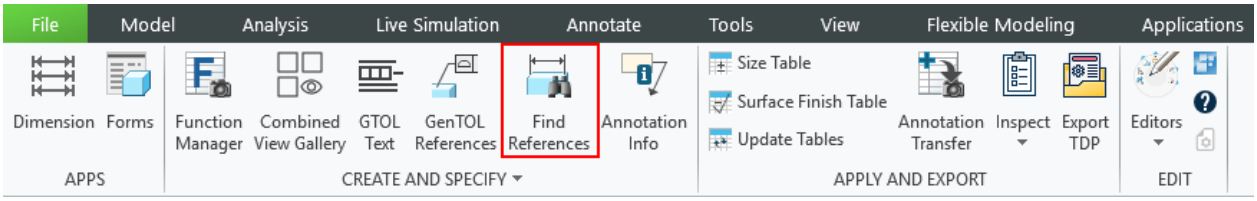
– no TED dimensions

– no angular dimensions

– no semantic references

## Starting the program: in assembly mode

To start *Find References*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




*Starting via the GT MBD in the ribbon menu*

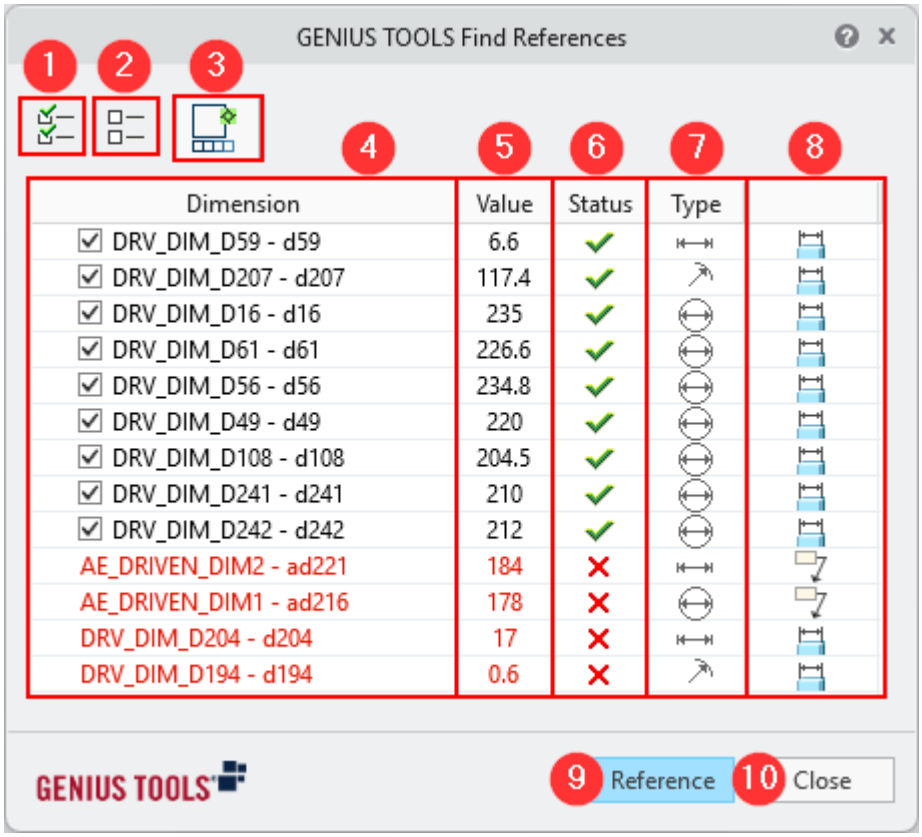
## Hiding the button from the menu

Use the configuration option `gtu_start_find_references` to switch off the function, so no button is displayed in the *GT MBD* ribbon menu. (Default is 1=On/not hidden)

## Find references

When you open *Find references*, you will see the following user interface:


© 2026 INNEO Solutions GmbH




1. Select all dimensions
2. Deselect all dimensions
3. Shows only dimensions that are visible in the currently active combined view.
4. Displays the dimension and can be activated manually by checking the box.
5. Displays the value of the respective dimensions.
6. Shows whether suitable semantic references were found (green tick) or not (red cross).
7. Shows the type of dimension (linear, radial, diameter).
8. Management of the dimension reference for the respective dimension; the symbol shows whether the dimension is assigned to an annotation feature.
9. Found semantic references are automatically assigned to dimensions with a green tick.
10. Close window.

## Procedure

1. Entries can be selected to highlight the respective dimensions in the model and can be manually checked or unchecked with a green tick. A green tick indicates that suitable semantic references have been found - the corresponding surfaces are highlighted in the model.
2. Assign dimensions


© 2026 INNEO Solutions GmbH

– Automatically: Click on *Referencing* and semantic references found are automatically assigned to the checked dimensions.
– Manual: Dimensions with a red cross must be referenced manually. This is done by clicking on the corresponding symbol in the dimension line. The window *References* or *Annotation Feature* opens, where the semantic references can be assigned manually and confirm with *OK*.

# 24.23 Full Backup

Available in the ribbon menu GENIUS TOOLS

*Full Backup* allows you to back up an object and objects that are dependent on it, as well as back up drawings with the same name and suppressed objects.

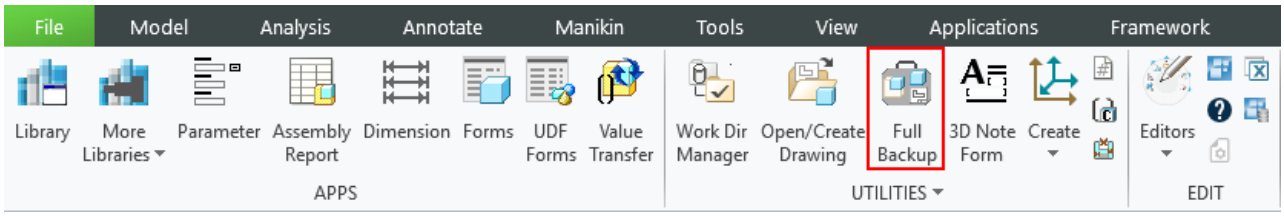
Thus, *Full Backup* extends the Creo Parametric function "create backup".

The objects to be saved can be saved in a directory or a ZIP file.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: all modes

To start *Full Backup* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

## Configuration of the ribbon menu

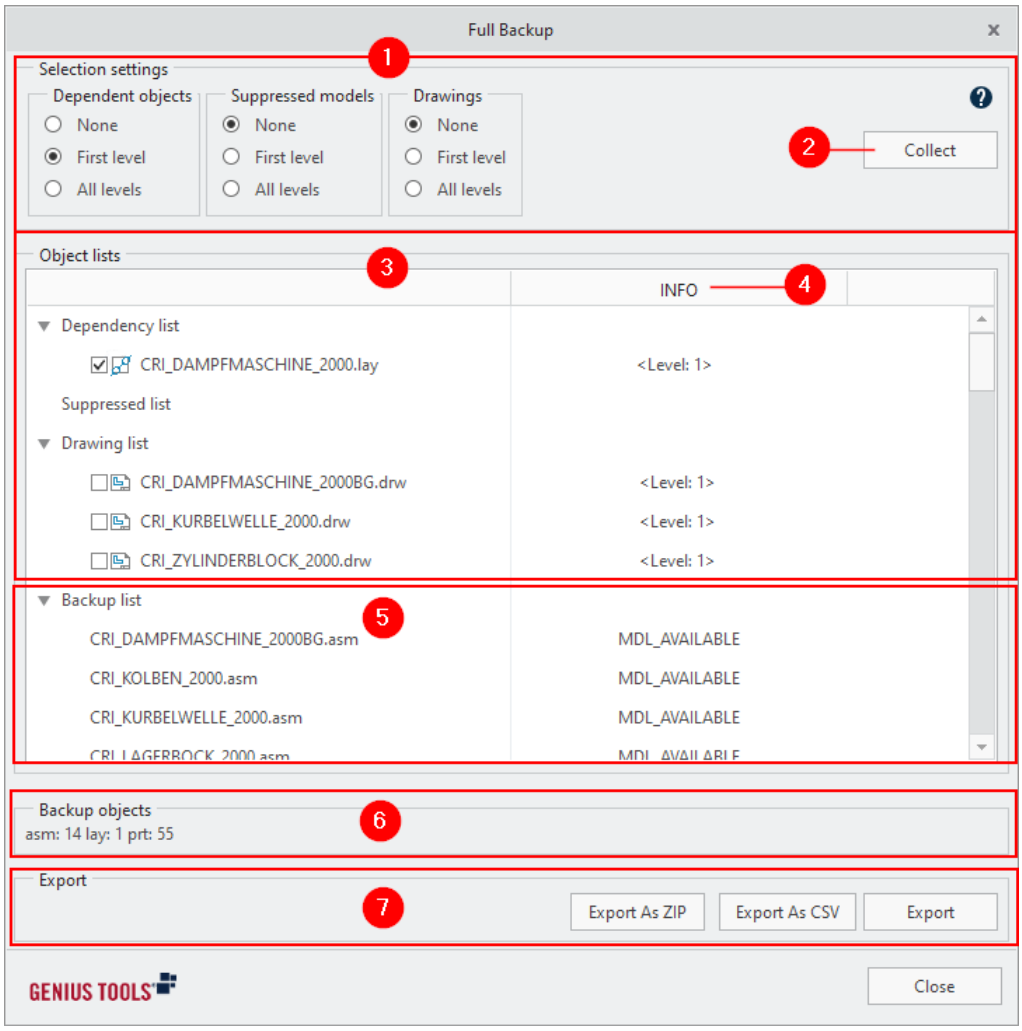
Use the configuration option `gtu_start_fullbackup` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## 24.23.1 Collect objects for backup list

You can decide whether and how dependent objects are added to the backup list and also include other objects.

Before the backup, you can select and deselect objects in the corresponding object lists.


© 2026 INNEO Solutions GmbH




*GT Utilities Full Backup - Dialog*

Decide in the selection settings (1) which elements to include in the backup list.

- None: Collects the current model and makes all drawings with the same name available for manual selection.
- First / All levels: *Full Backup* searches for dependent objects, drawings with the same name and suppressed models in different levels, see next chapter 714 for details.

The *Collect* action (2) transfers to the object lists (3):

- all objects for the backup list: these are already checked.
- objects of the next level: are displayed with an empty checkbox and can thus be transferred manually to the backup list.
- details of the level are included in the info column (4).
- a model-parameter value defined by the configuration option `gtu_full_backup_column_parameter`.

All elements that are checked in the object list are transferred to the backup list.

The backup list (5) contains all backup objects to be backed up, sorted by their file extension:


© 2026 INNEO Solutions GmbH

– drawings (drw), assemblies (asm), parts (prt): are available for selection / deselection in the object lists
– drawing frames (frm) and layout (lay): are bound to the drawings in the backup list.

Objects that cannot be found are displayed in the *Failed Models* list.

**Please note:** the backup list may contain objects that are not available for selection, e.g. drawing frames.

The status bar (6) shows the number of objects in the backup list, listed according to the file extension.

The backup list can be exported (7):

– as a ZIP file: File name and path of the location are defined in the file selection dialog. Afterward fb__ and the current datum get added to the file name, e. g. *fb__17-06-2022_cri_kurbelwelle_2000*.
– as a CSV file: File name and path of the location are defined in the file selection dialog.
– in a folder: a subfolder is created with date and object name, e.g. *fb__17-06-2022_CRI_KURBELWELLE_2000*. All objects of the backup list are saved in this folder. If the folder is already existing an index is added.

**Warning:** After the commands *Export As Zip* and *Export,* the storage paths of all backup objects no longer point to their original directories.
Before further editing with Creo Parametric, all objects must be removed from the session and must be called up again from their original directories. It is the same behavior as the standard Creo Parametric command *Save Backup*.

## 24.23.2 Choose levels of dependencies

*Full Backup* searches for dependent objects, drawings with the same name and suppressed models in different levels.

Drawings are analyzed to provide you with a checkbox for additional drawing models for the backup list.

Searching and analyzing dependent objects can take a long time for large assemblies or for data storage locations that have poor network connectivity. In these cases, it is recommended to select none as the search level and manually add the files that should be exported.

Overview of the selection settings:

| **Dependent objects** |
| --- |
| **Search: None** |


© 2026 INNEO Solutions GmbH

| Dependent objects | |
|---|---|
| Object to backup | Inclusion in backup list |
|  | The current part.<br>– If it is a variant, the generic is saved automatically.<br>– If a drawing with the same name exists, it is offered for selection. |
|  | The current assembly and all active sub-assemblies.<br>– If it is a variant, the generic is automatically saved. This also applies to all sub-components.<br>– If drawings with the same name (sub-models) exist, they are offered for selection. |
|  | Manual selection for: the current drawing. All drawing frames and all drawing models that control the selected drawings are saved automatically.<br>– Generics are automatically saved for all drawing models that are a variant. This also applies to all sub-components.<br>– If drawings with the same name (sub-models) exist, they are offered for selection. |
| **Search: First level** | |
| Object to backup | Inclusion in backup list |
|  | All objects that control elements of the current part. |
|  | All objects that control elements of the current part or its subcomponents. |
|  | All objects that control elements of the drawing models or their subcomponents. (drawing frames and drawing models). |
| **Search: All levels** | |
| Object to backup | Inclusion in backup list |
|  | All objects that are dependent on parts of subsequent levels. |


© 2026 INNEO Solutions GmbH

| | |
|---|---|
| **Dependent objects** | |
|  | All objects that are dependent on parts or subcomponents of subsequent levels. |
|  | All objects that control elements of the drawing models or their subcomponents of the subsequent levels. (drawing frames and drawing models). |
| **Suppressed objects** | |
| Object to backup | Inclusion in backup list |
| **Search: None** | |
|  | No suppressed objects of the current assembly are included.<br>- The suppressed objects would be offered for selection. |
| **Search: First level** | |
|  | All suppressed objects of the assembly that controls the part will be included.<br>- The second level of suppressed objects would offered for selection. |
|  | All suppressed objects of the current assembly are included.<br>- The second level of suppressed objects would offered for selection. |
|  | All suppressed objects of the assembly that controls the drawing will be included.<br>- The second level of suppressed objects would offered for selection. |
| **Search: All levels** | |
|  | All suppressed objects of the found dependent objects are included in the backup list. The search depends on the selected search level for dependent objects. |
| **Drawings** | |
| **Search: None** | |
| Object to backup | Inclusion in backup list |


© 2026 INNEO Solutions GmbH

| Drawings | |
|---|---|
|  | The current part.<br>– If it is a variant, the generic is saved automatically.<br>– If a drawing with the same name exists, it is offered for selection. |
|  | The current assembly and all active sub-assemblies.<br>– If it is a variant, the generic is automatically saved. This also applies to all sub-components.<br>– If drawings with the same name (sub-models) exist, they are offered for selection. |
|  | Manual selection for: the current drawing. All drawing frames and all drawing models that control the selected drawings are saved automatically.<br>– Generics are automatically saved for all drawing models that are a variant. This also applies to all sub-components.<br>– If drawings with the same name (sub-models) exist, they are offered for selection. |
| **Search: First level** | |
| Object to backup | Inclusion in backup list |
|  | Drawings with the same name are selected immediately.<br>Additional drawing models are added to the backup list. If these have drawings with the same name, they are offered for selection. |
| **Search: All levels** | |
|  | Drawings with the same name are selected immediately.<br>Additional drawing models are added to the backup list. If these have drawings with the same name, they are offered for selection. |

# 24.24 GenTOL References

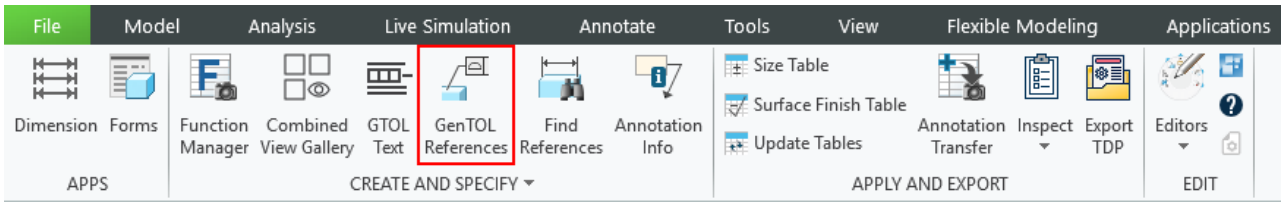
Available in the ribbon menu GT MBD


© 2026 INNEO Solutions GmbH

The module *GenTOL References* [abbreviation for General Tolerances] references all available surfaces to the general tolerance. The number of referenced surfaces is listed in the message log.

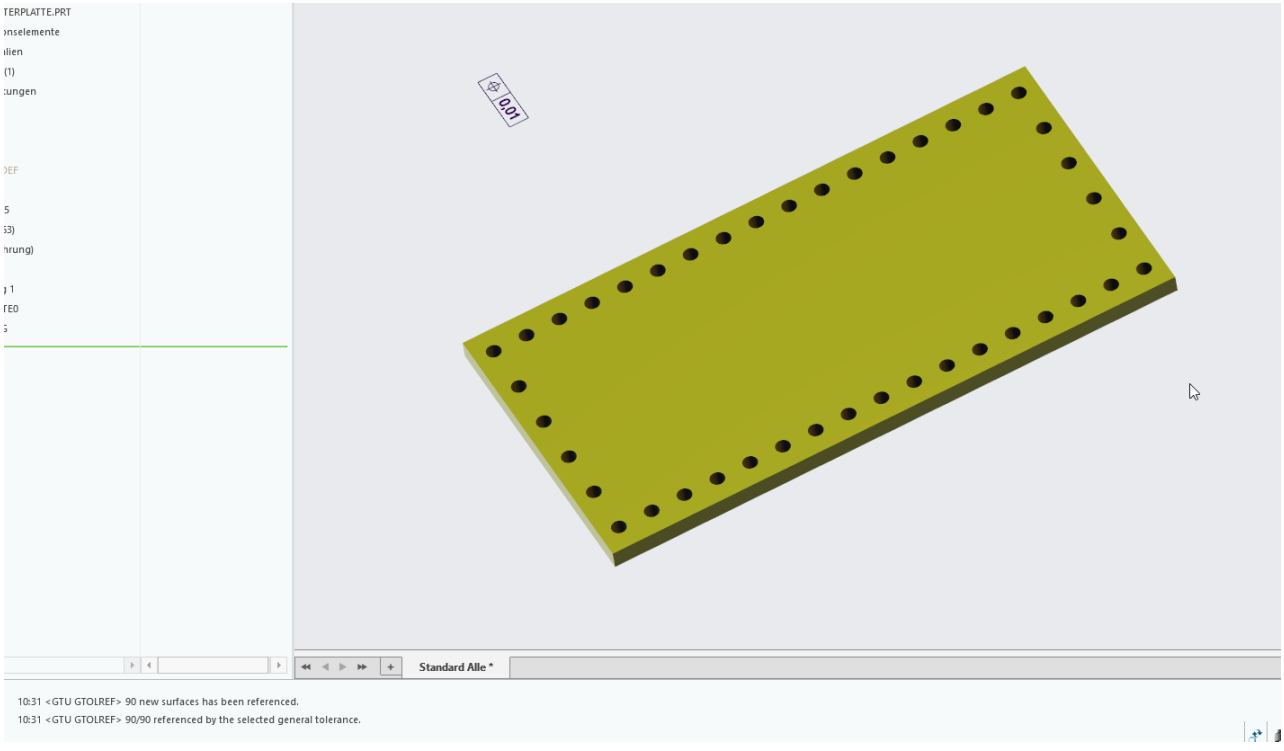
## Starting the program: in part mode

To start *GenTOL References* , click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




*Starting via the GT MBD in the ribbon menu*

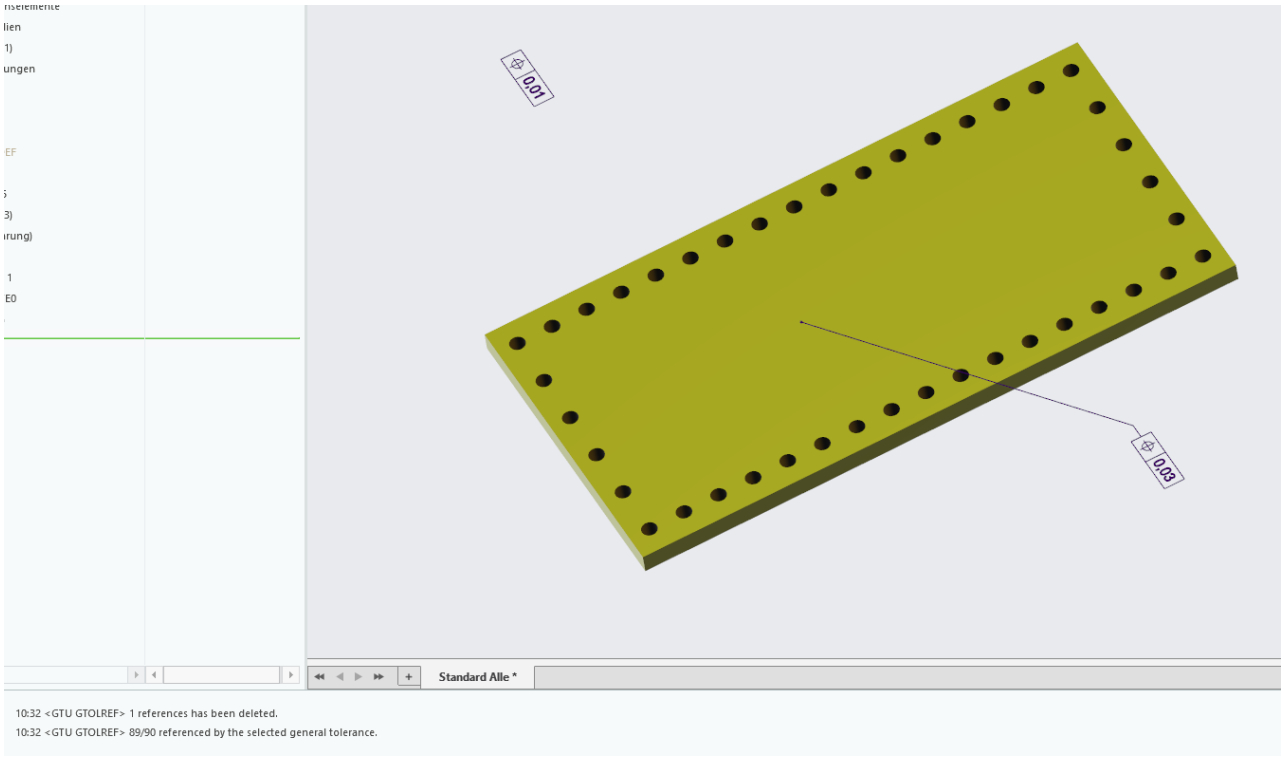
The first time you call this function in a model, the message log asks you to select a general tolerance. The module will then remember the selected general tolerance. After selection, all surfaces that are not referenced separately are referenced by the general tolerance. The number of referenced surfaces is displayed in the message log.




*Initially referenced general tolerance*

If you make changes to the geometric tolerances, such as deleting or adding a reference, call the function again. The surfaces referenced with the general tolerance are updated.


© 2026 INNEO Solutions GmbH




*Newly added geometric tolerance that has resulted in a change to the referenced general tolerance*

**Please note:** Using this utility creates the parameter *GT_GENERAL_TOLERANCE*. If you want to change the selected general tolerance, delete the parameter *GT_GENERAL_TOLERANCE*.

# 24.25 GTOL Text

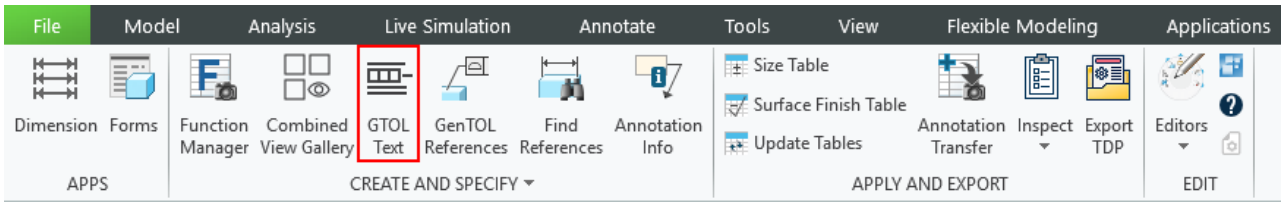
Available in the ribbon menu GT MBD

The module *GTOL Text* allows you to edit texts of existing shape and position tolerance annotations. Set up templates for these texts and links to more information using the *GTOL Text Editor*.

## 24.25.1 Starting the program: in part mode and in assembly mode

To start *GTOL Text*, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section. The button opens a Creo selection dialog where you can select an annotation to edit its texts. Alternatively, click on the annotation to edit and then open the module.


© 2026 INNEO Solutions GmbH



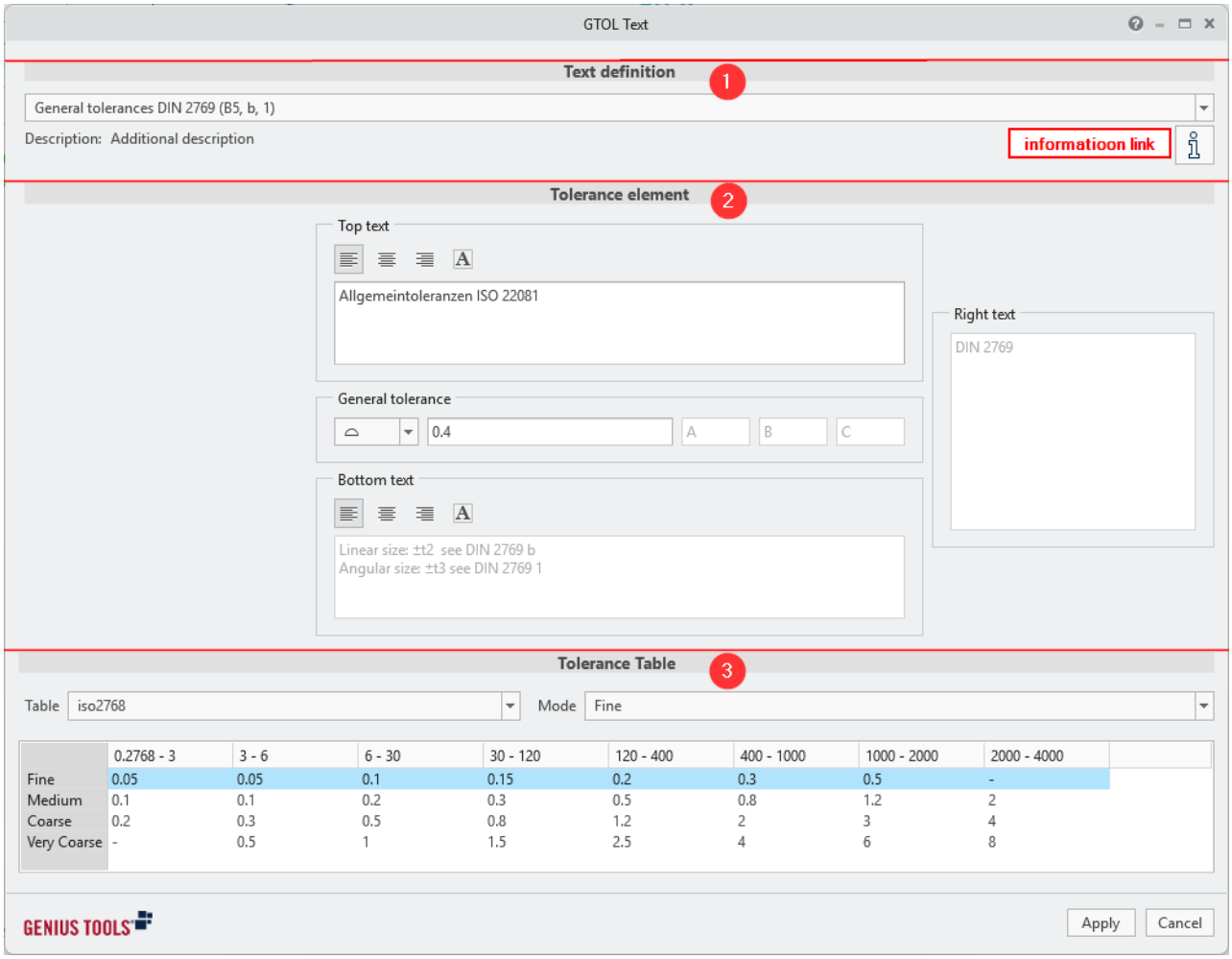

*Starting via the GT MBD in the ribbon menu*

## 24.25.2 Usage

This section contains information on the use of *GTOL Text*. The general structure of the program is explained.

### 24.25.2.1 User interface

The user interface of *GTOL Text* consists of the following elements:




1. Select text definition
   Templates are available under text definition that can be used to pre-fill the comment fields. Depending on the template, the individual fields can be written in, expanded or blocked. Templates can be added and edited as XML files in the *GTOL Text Editor*[722].


© 2026 INNEO Solutions GmbH

Under the selection text definition, further information may be available in the form of a description and a link, e.g. to a standard.

2. Edit tolerance element
3. Select tolerance table

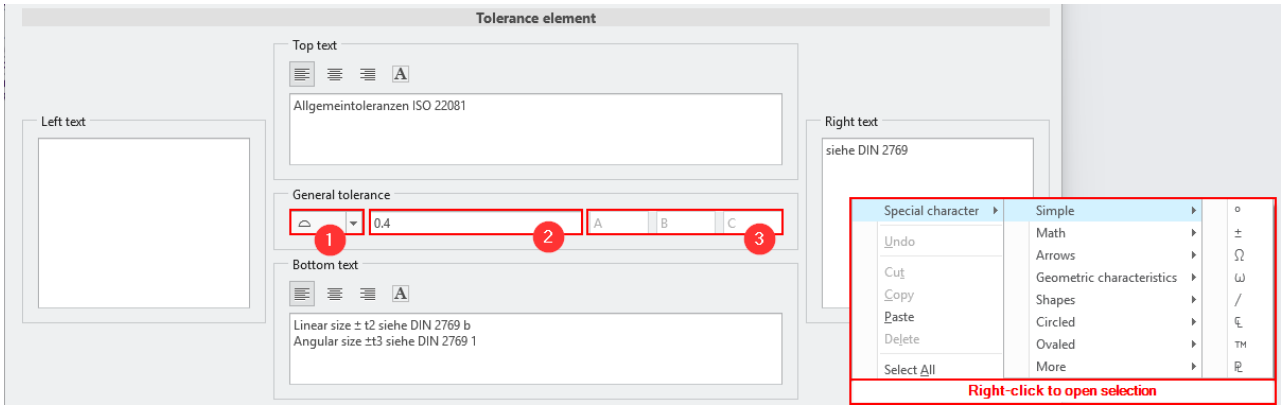
## 24.25.2.2 Edit tolerance element

In the section Tolerance element, you can select the general tolerance and edit the annotations.

### Select reference system

The specification of the basic reference system is composed of:

You can edit the frame specification as follows:




1. Select reference symbol
2. Enter value
3. Automatic reading of references from the model (these fields cannot be edited manually)

### Editing annotations

Depending on the settings in the *GTOL Text Editor*, the text fields are fully or partially available for editing. If a text field is editable, text can be added and deleted. Grayed out text cannot be edited.

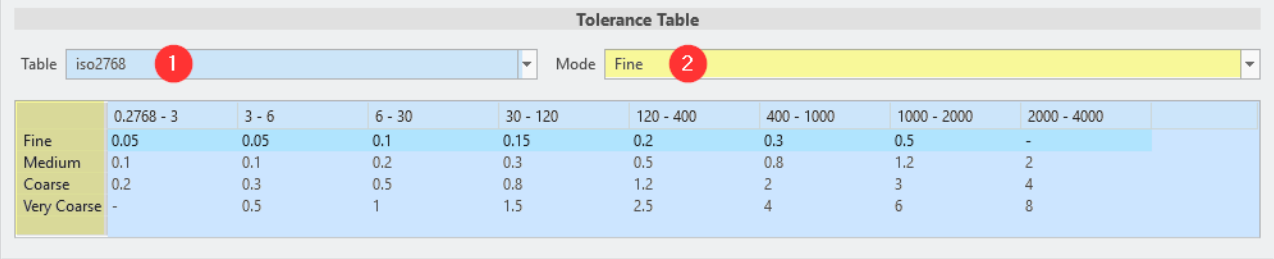
**Insert special characters**
Right-clicking in a text field opens the context menu from which you can select special characters.

**Please note:** The currently set annotation texts can be displayed in the overview table in *Inspect 3D*[215] and exported as an Excel[202]


© 2026 INNEO Solutions GmbH

## 24.25.2.3 Select tolerance table

The tolerance table can be selected in the menu to automatically transfer standard-compliant tolerances to dimensions and simplify dimensioning according to standards.




| | 0.2768 - 3 | 3 - 6 | 6 - 30 | 30 - 120 | 120 - 400 | 400 - 1000 | 1000 - 2000 | 2000 - 4000 |
|---|---|---|---|---|---|---|---|---|
| Fine | 0.05 | 0.05 | 0.1 | 0.15 | 0.2 | 0.3 | 0.5 | - |
| Medium | 0.1 | 0.1 | 0.2 | 0.3 | 0.5 | 0.8 | 1.2 | 2 |
| Coarse | 0.2 | 0.3 | 0.5 | 0.8 | 1.2 | 2 | 3 | 4 |
| Very Coarse | - | 0.5 | 1 | 1.5 | 2.5 | 4 | 6 | 8 |

1. Select the tolerance table from the list.
2. Select the mode defined in the tolerance table.

## 24.25.3 Configuration

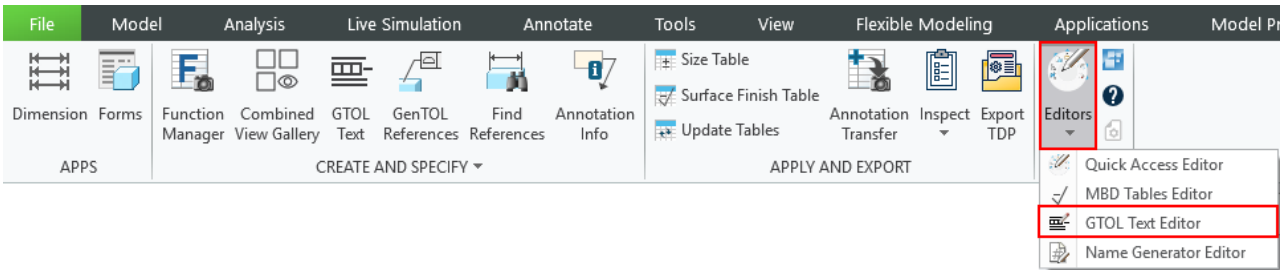
This section contains information about *GTOL Text Editor* and XML template creation.

## 24.25.3.1 GTOL Text Editor

With *GTOL Text Editor*, you can create and edit templates as custom XML files, and define the extent to which these templates can be modified with *GTOL Text*.

### Starting the program

To start *GTOL Text Editor*, open the *GT MBD tab in* the ribbon menu and go to the section *APPLY AND EXPORT* to open the drop-down list.




*Starting via the GT MBD tab in the ribbon menu*

### Configuration of the ribbon menu

Use the configuration option `gtu_start_gtol_text_editor` to switch off the function, so no button is displayed in the ribbon menu. (Default is 1=On/not hidden)


© 2026 INNEO Solutions GmbH

**Please note:** If the Editor button is not visible, access was disabled by the administrator. Settings for user permissions can be made in the *Configuration Utility* and, if you are working with *Startup TOOLS*, in the *GENIUS TOOLS Project Configurator*, see Set access to program editors 781.

## Saving changes

The changes made in the Editor are saved in the resource directory.

When working with *Startup TOOLS*, the data of the local resource folder (in the Cadpool directory) is synchronized with the resource folder of the administrator computer (Caddepot).

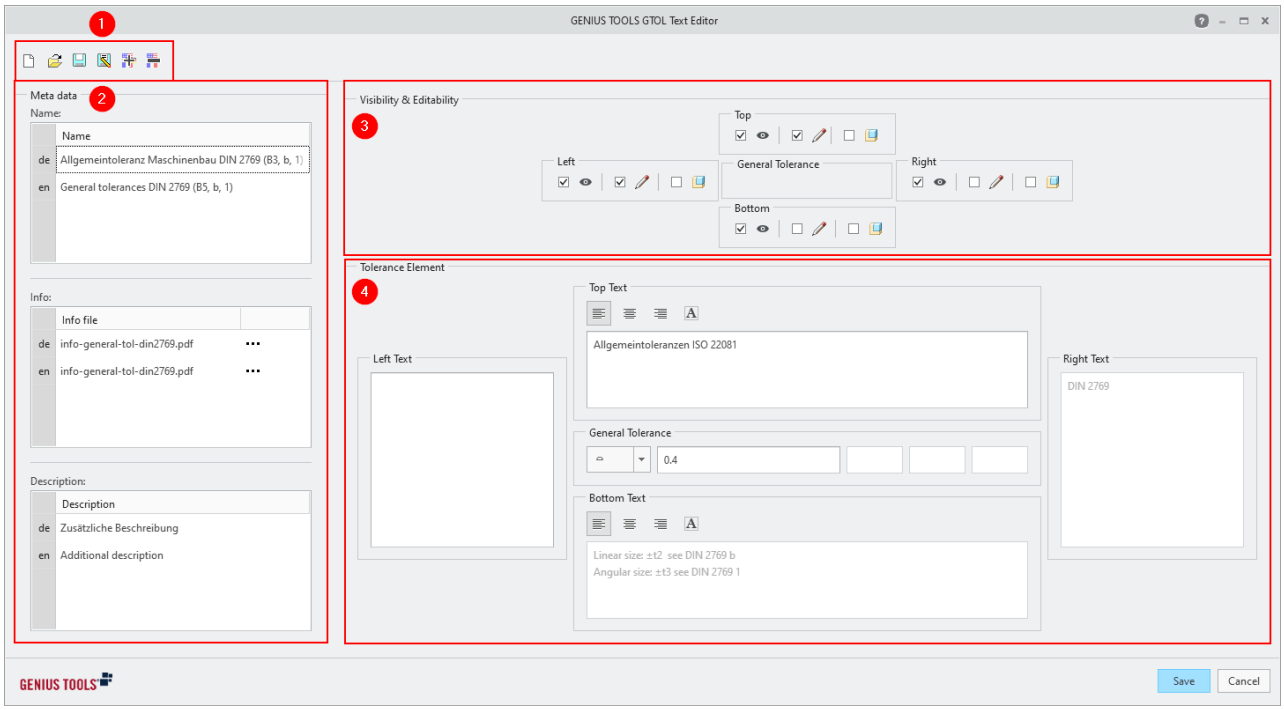
SUT-Path: <*operatingenvironment*>/*parametric/configuration/gt_resource_folder*.

To ensure that the changes are applied, you should work directly in Caddepot.

**Warning:** If you are using satellites, note that the Caddepot of a satellite is overwritten by the Caddepot of the main server.

## User interface

The user interface of *GTOL Text Editor* consists of the following elements:




1. Command bar 724
2. Meta data 726
3. Visibility & editability 724
4. Tolerance Element 725


© 2026 INNEO Solutions GmbH

## Command bar

| Symbol | Name | Description |
|---|---|---|
|  | Creates a new configuration file | Creates a new configuration as an XML file. |
|  | Opens a configuration file | Opens an existing XML file. |
|  | Save configuration | Saves the XML file under the specified name and in the preconfigured location. The location is defined by the configuration option `gtol_text_folder`. |
|  | Save configuration as XML | Opens the Save dialog box, which allows you to save the XML file under a different name and in a different location. |
|  | Dialog to add a language | Adds a new empty row per language to the metadata fields. English is already set up as a fallback language. |
|  | Dialog to delete a language | Deletes already created languages line by line. English is the default language. It cannot be deleted. |
|  | Help | Opens the help. |

## Setting visibility and editability

Text can be added to all four sides of a shape and position tolerance. The text can be edited individually on each side and is controlled by these buttons:

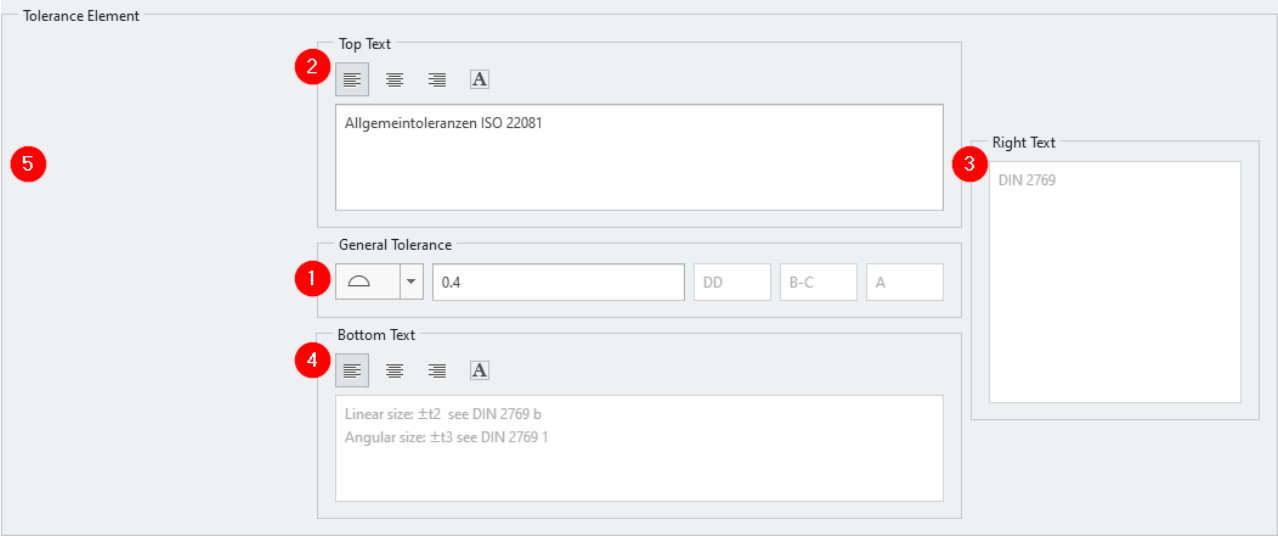
| Symbol | Name | Description |
|---|---|---|
|  | Visibility | Specifies whether the annotation text is visible. The check mark is set by default. To create an annotation text, the check mark must be set. |


© 2026 INNEO Solutions GmbH

| Symbol | Name | Description |
|---|---|---|
|  | Editability | Specifies whether the annotation text can be edited. The check mark is set by default. To create an annotation text, the check mark must be set. If the text in *GTOL Text* is not to be edited, uncheck the box after entering the text. |
|  | Get text from model | Specifies the text source if text is already stored in the model: The check mark is unchecked by default. (checked = text is taken from the model if text is stored in the model, unchecked = text is taken from the configuration XML file, even if text is stored in the model) |

# Editing annotation texts

Text can be added to all four sides of a shape and position tolerance. The Visibility & Editability[724] settings determine the options for entering annotation text under *Tolerance Element*:




1. **Main Symbol**
   Geometric symbol and tolerance value can be entered. The reference systems are read. These fields cannot be edited.

2. **Text Top**
   Additional text can be added to the existing text in *GTOL Text*. The text can be aligned and framed, e.g. for TED dimensions.


© 2026 INNEO Solutions GmbH

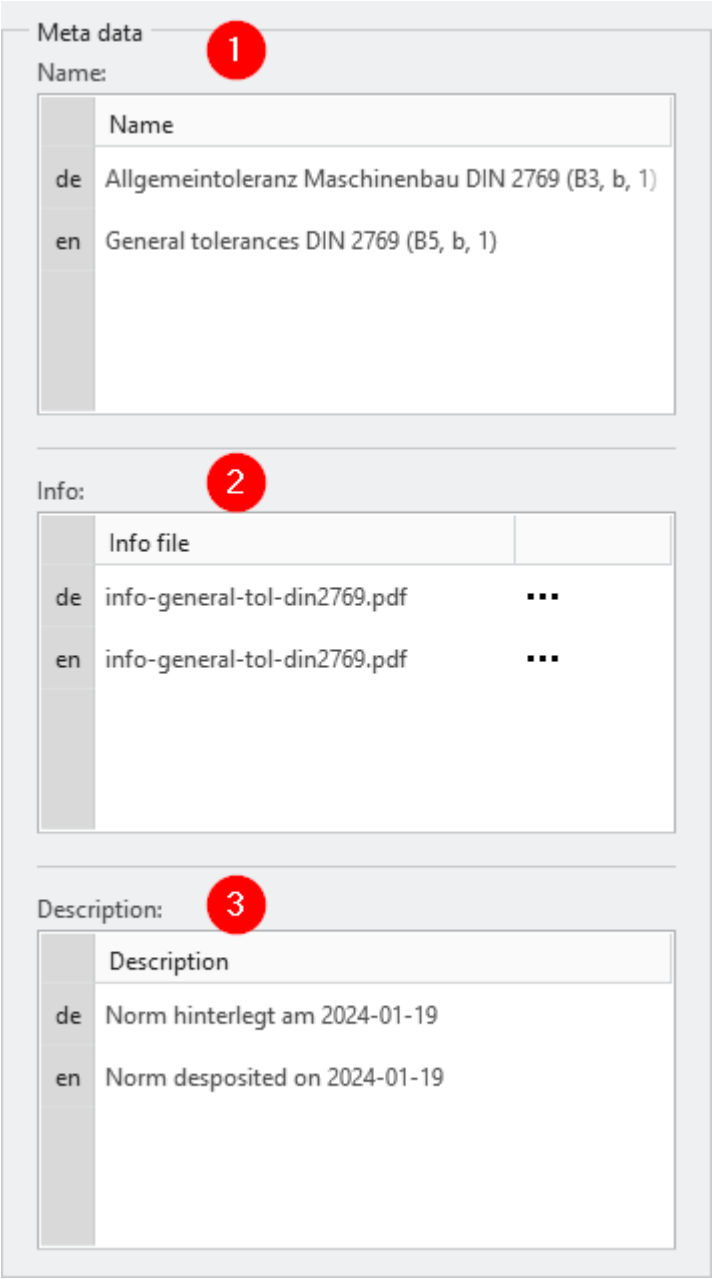
3. **Text Right**
   The entered text was made uneditable by removing the check mark for .
   The text on the left and right cannot be aligned or framed.
4. **Text Bottom**
   In *GTol Text*, no more text can be added to the existing text because the check mark for was removed. The text can still be aligned and framed in the Editor, e.g. for TED dimensions.
5. **Text Left**
   The field is made invisible by clearing the checkbox . When checked, the same editing options are available as for *Text Right*.

## Adding further information

Under *Metadata*, you give the template a name by which it will be referenced in a model. You can also add additional information to the template, which is displayed in the template under *Description*, as well as a link to a document or website.

1. **Name of the text definition**
   Name that summarizes the saved annotation texts, e.g. according to DIN 2769.
2. **Info**
   Additional information can be stored as a document (e.g. PDF, DOC) or URL, e.g. a standard or web link (in the format http://www.inneo.com). In order to open the files, an appropriate program must be installed on the PC.
   The location of the documents to be stored is defined by the configuration option `gtol_text_info_folder`.
3. **Description**
   In addition to the template name, you can add a note describing the template, e.g. the date or the author.


© 2026 INNEO Solutions GmbH




Lines for an entry in English are predefined for all information. English is a default language and cannot be removed. Lines for other languages can be added using  and deleted using  .

## 24.26 Inspect Instance Note

Available in the ribbon menu GT MBD

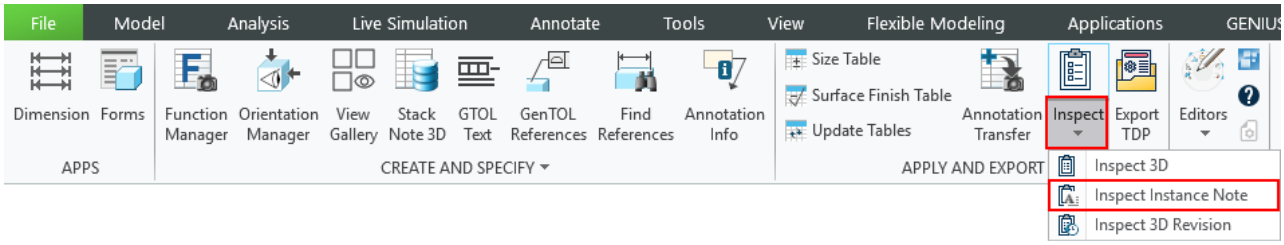
With *Inspect Instance Note*, inspection symbols and notes can be selected to add a note. This shows which inspection symbols belong to the respective pattern.


© 2026 INNEO Solutions GmbH

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: in part and assembly mode

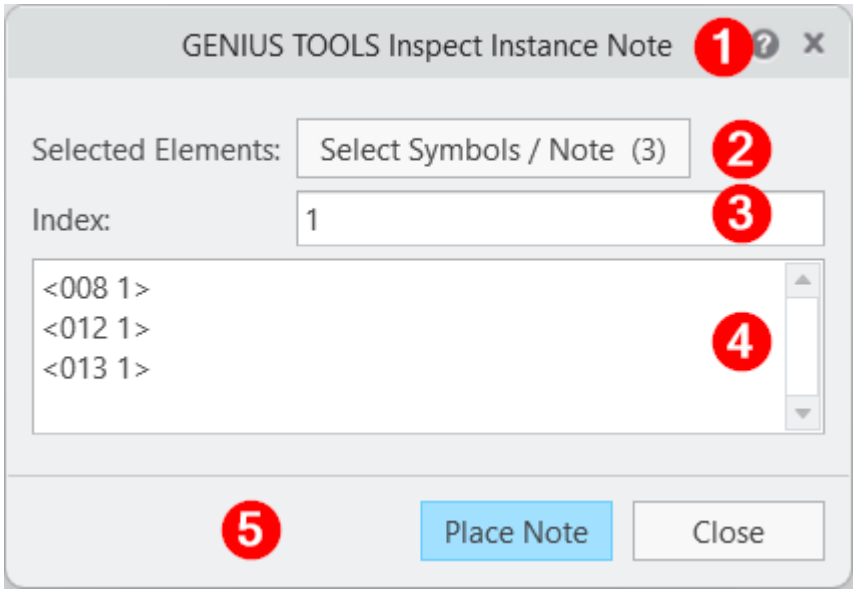
To start *Inspect Instance Note*, click *GT MBD* in the ribbon menu, go to *APPLY AND EXPORT* section and click on the drop-down menu under *Inspect*.




*Starting via the GT MBD in the ribbon menu*

## User interface

The user interface of *Inspect Instance Note* consists of the following elements:



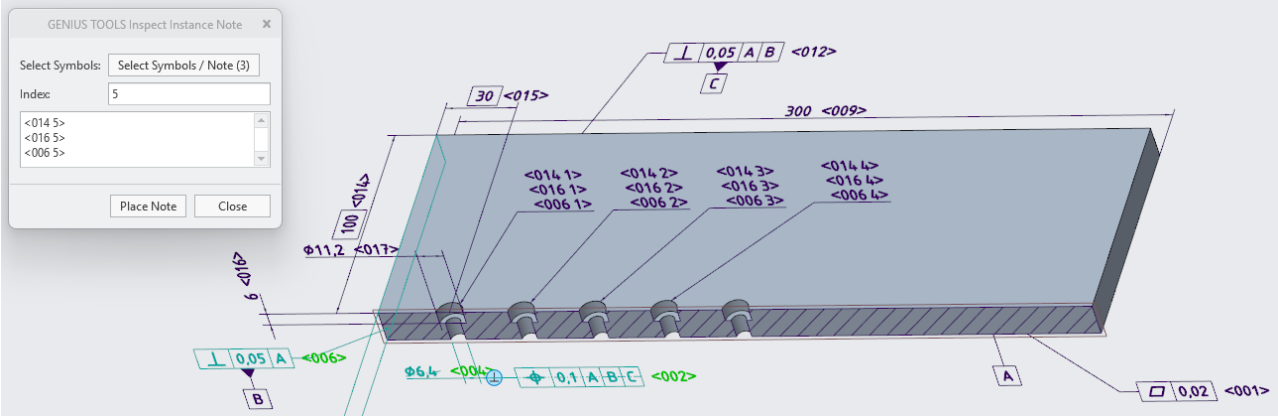

1. Opens this help.
2. Enables the selection of symbols or a previously created note.
3. Consecutive numbering of notes, starting with 1.
4. Displays the currently selected inspection symbols. This list determines which inspection symbols are listed in the note.
5. Creates the note for placing at the desired positions. All entries from the list are output with the current index. Quit the function by clicking *Close*.


© 2026 INNEO Solutions GmbH

# Creating notes

Prerequisite: Inspection characteristics have already been assigned to the pattern variant.

1. Click on the button *Select Symbols / Note*.
2. Select the desired inspection symbols and confirm with *OK*.
3. Click on *Place Note* to place the note at the desired location.




**Creating additional notes after extending the pattern variant**

Prerequisite: A note has already been created and the pattern has been extended, e. g., if additional holes have been added subsequently.

1. Click on the button *Select Symbols / Note*.
2. Select the desired note so that all inspection symbols are displayed in the list.
3. Click on *Place Note* to place the note at the desired location.

# Configuration

You can use the configuration option `gtu_inspect_instance_note_line_definition=</num/ @index@>` to configure inspection number and index number for each line:

– `/num/` stands for the variable text from the inspection symbol.

– `@index@` is managed by the function but can be changed in the user interface.

# Configuration of the ribbon menu

Use the configuration option `gtu_start_inspect_instance_note` to switch off the function, so no button is displayed in the *GT MBD* ribbon menu. (Default is 1=On/not hidden)


© 2026 INNEO Solutions GmbH

# 24.27 Load Save Converter

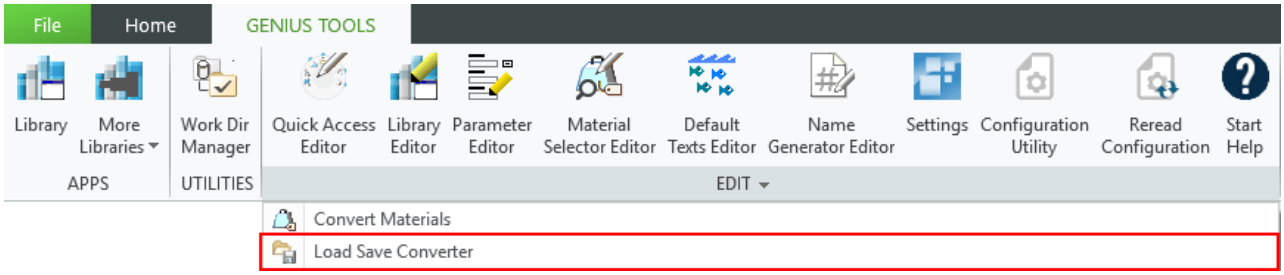
Available in the ribbon menu GENIUS TOOLS

This function allows you to convert objects, which have been created in older versions of Creo, Wildfire or Pro/engineer, into the currently used Creo version. Also, you can convert to the student format.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: in Creo standby mode

To start *Load Save Converter* , click *GENIUS TOOLS* in the ribbon menu and go to *EDIT-HELP* section.

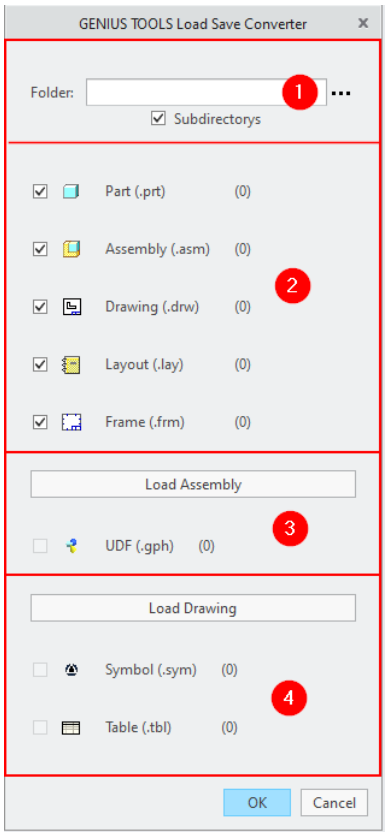



*Starting via the GENIUS TOOLS in the ribbon menu*

## Steps

1. When using the *Load Save Converter* function Creo objects are uploaded from a directory and saved again. This converts them into the currently used Creo format.


© 2026 INNEO Solutions GmbH




*Dialog box of Load Save Converter*

2. Choose the directory (1) that contains the dated objects.
3. Check the object types (2) you want to convert.
4. If you want to convert UDF (user defined features), check the box in the segment *Load Assembly* (3). This uploads a stored assembly and converts the UDF that it contains.
5. If you want to convert symbols or tables, check the box in the segment *Load Drawing* (4). This uploads a stored drawing and converts the symbols and tables that it contains.
6. Click *OK*.
7. The conversion of the files is displayed in the message area.

## Configuration of the ribbon menu

Use the configuration option `gtu_start_loadSaveConverter` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

# 24.28 Open Base Model

Available in the ribbon menu GENIUS TOOLS

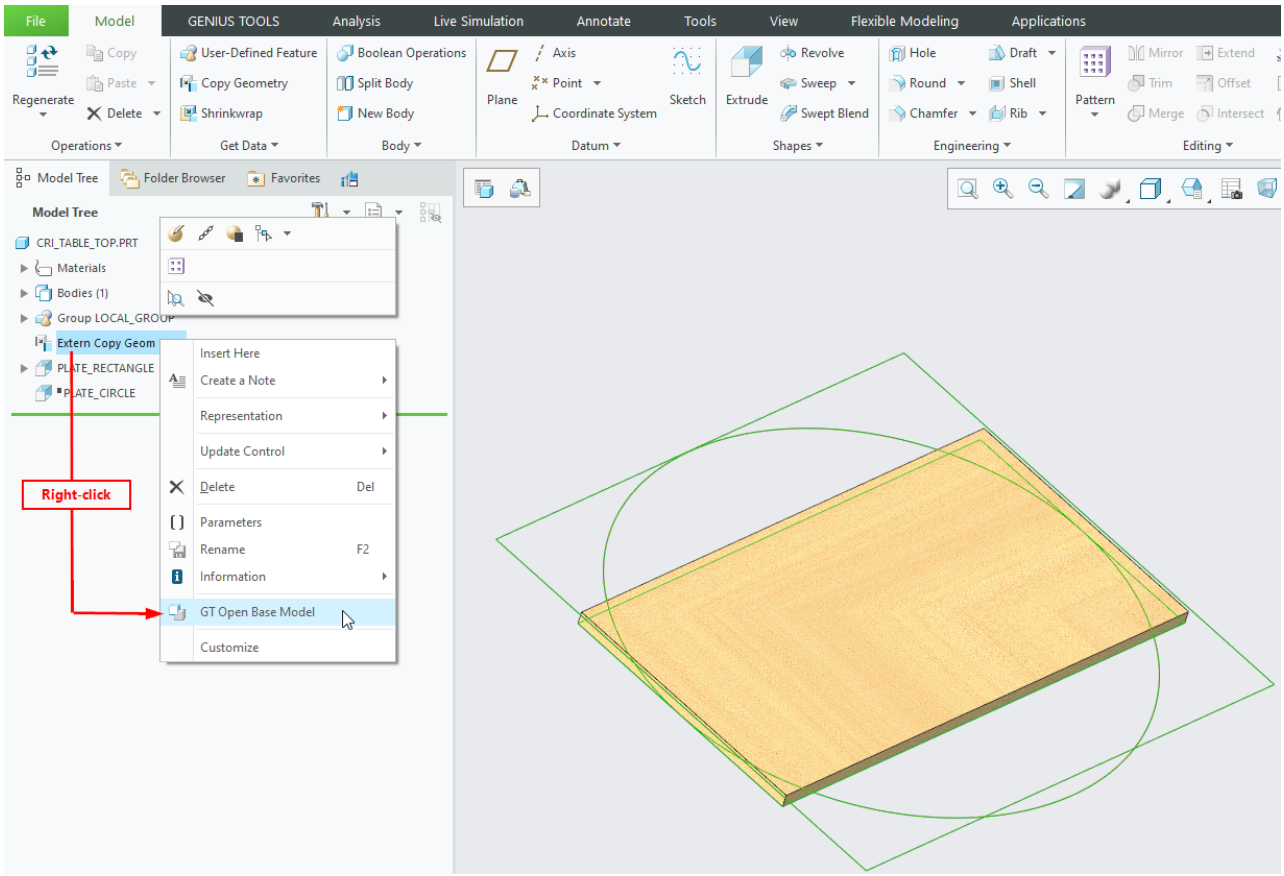

© 2026 INNEO Solutions GmbH

This function allows you to quickly open geometric base models that are reference sources for features.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: in part and assembly mode

The function *GT Open Base Model* can be found in the context menu – which opens when right-clicking – on all features that have a reference source. (Here: "Extern Copy Geom")




*Funktion GT Open Base Model in context menu*

## Configuration of the ribbon menu

Use the configuration option `gtu_start_openGeomOrigin` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

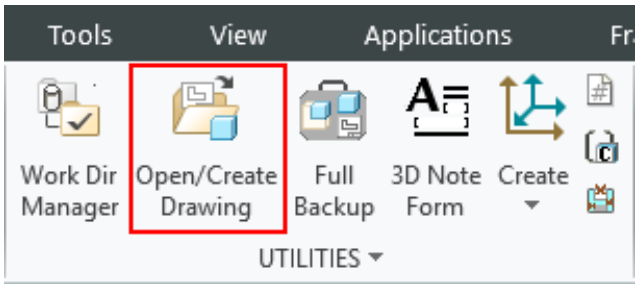
# 24.29 Open/Create Drawing

Available in the ribbon menu GENIUS TOOLS

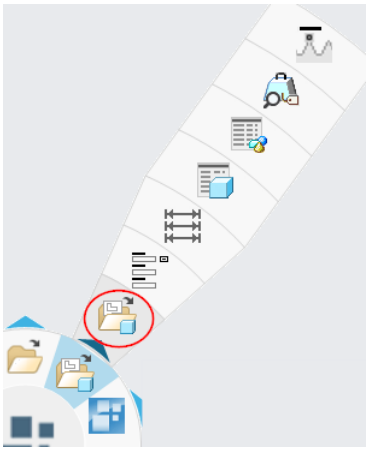
This function opens or creates a drawing of the current model depending on whether a drawing with the model name already exists.

## Starting the program: in parts and assembly mode

To start *Open/Create Drawing*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



*Starting via Quick Access*

The function searches for a drawing named: <*PREFIX*><*Model name*><*SUFFIX*>.*drw*.*, or like defined in the configuration option `gtu_ord_drw_name`, in the default search paths of Creo Parametric.

Example: `gtu_ord_drw_name = @mdlname@`

The model name is used. (Default setting)

The configuration option `gtu_ord_pre_drw_name` defines the prefix and the configuration option `gtu_ord_post_drw_name` defines the suffix for the file names of the drawings.

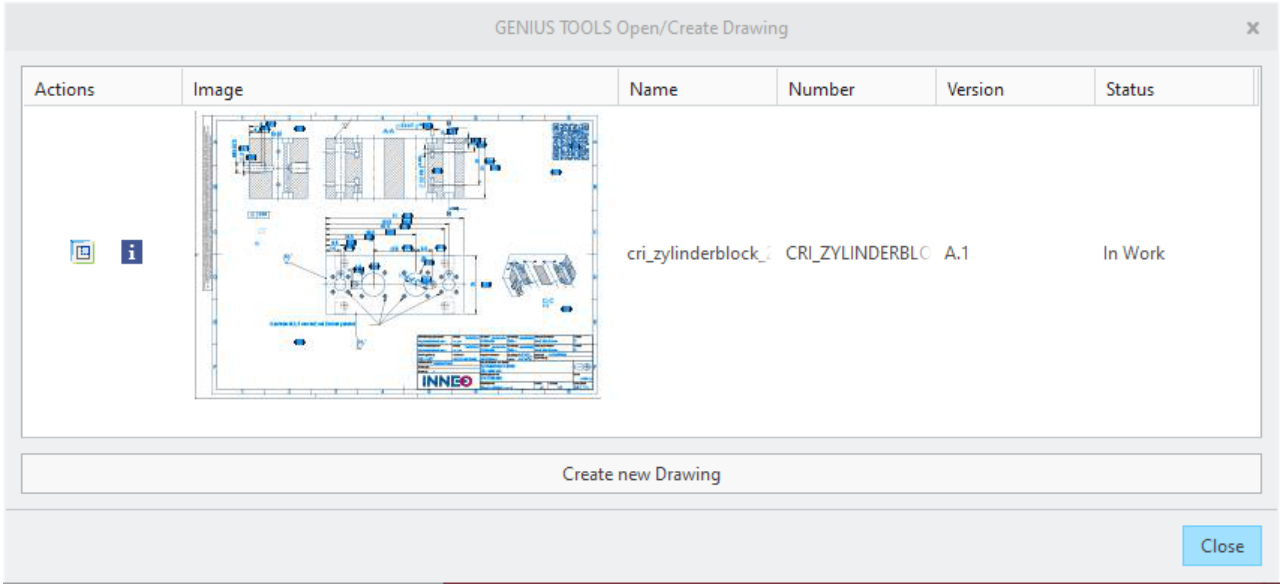
**Without Windchill with activated REST API or (gtu_ord_use_pdm_server = 0):**
If a drawing with the defined name is found inside the search path, it would be opened. If no drawing is found, the dialog for the creation of a new drawing would be shown. Inside this dialog, the file name and other options are already filled.

Example for the file name `gtu_ord_drw_name = $$repl$MO-$DW-$@mdlname@$$`

Within the filename, `mo-` is replaced by `dw-` for retrieval and creation.

**With Windchill with activated REST API and (gtu_ord_use_pdm_server = 1):**
If one or more drawings are found by the Windchill REST API, these would be shown for selection. If not, or by the dialog, the dialog for the drawing creation can be opened. In this dialog the file name and other options are already filled.


© 2026 INNEO Solutions GmbH




*Selection of a drawing from Windchill*

Example for the configuration `gtu_ord_drw_name =`

If the entry is empty, the WT number generator will be used to generate a new drawing.

## Configuration

**Provide function in assembly mode before selecting a part**
By default, the function creates a drawing *after* a part has been selected in an assembly. Set the configuration option `gtu_ord_try_to_use_selected_part_if_inside_asm` to 0 to open the function first and select the part or parts in the 3D model in the second step, i. e. there is an additional dialog "Select 1 element".

**Fill dialog with customized mapkey**
The creation dialog for a new drawing can be filled / controlled with a mapkey. This can be defined using `gtu_ord_createdrw`. Inside the configuration option % -signs need to be doubled up.

Example: `gtu_ord_createdrw=%%mymapkey`).

If `gtu_ord_createdrw` is not defined, the Mapkey %createdrw would be run if defined.

Also refer to the Configuration[755] section.

# 24.30 Select Contact Surfaces

Available in the ribbon menu GT MBD

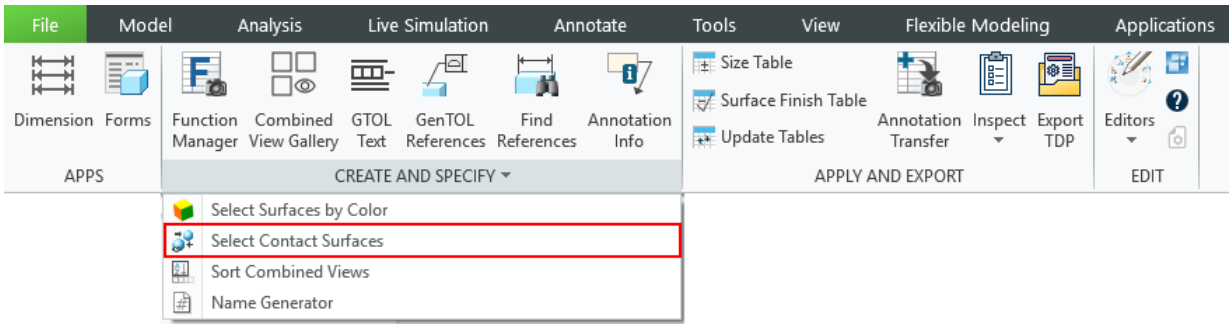
After searching for a part's contact surfaces, the identified contact surfaces can be viewed and selected with *Select contact surfaces* and colored with *Function Manager*[167].

# Starting the program: in part mode and in assembly mode

The function *Select contact surfaces* can be opened in several ways:


© 2026 INNEO Solutions GmbH

1. To Start Select Contact Surfaces, click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.

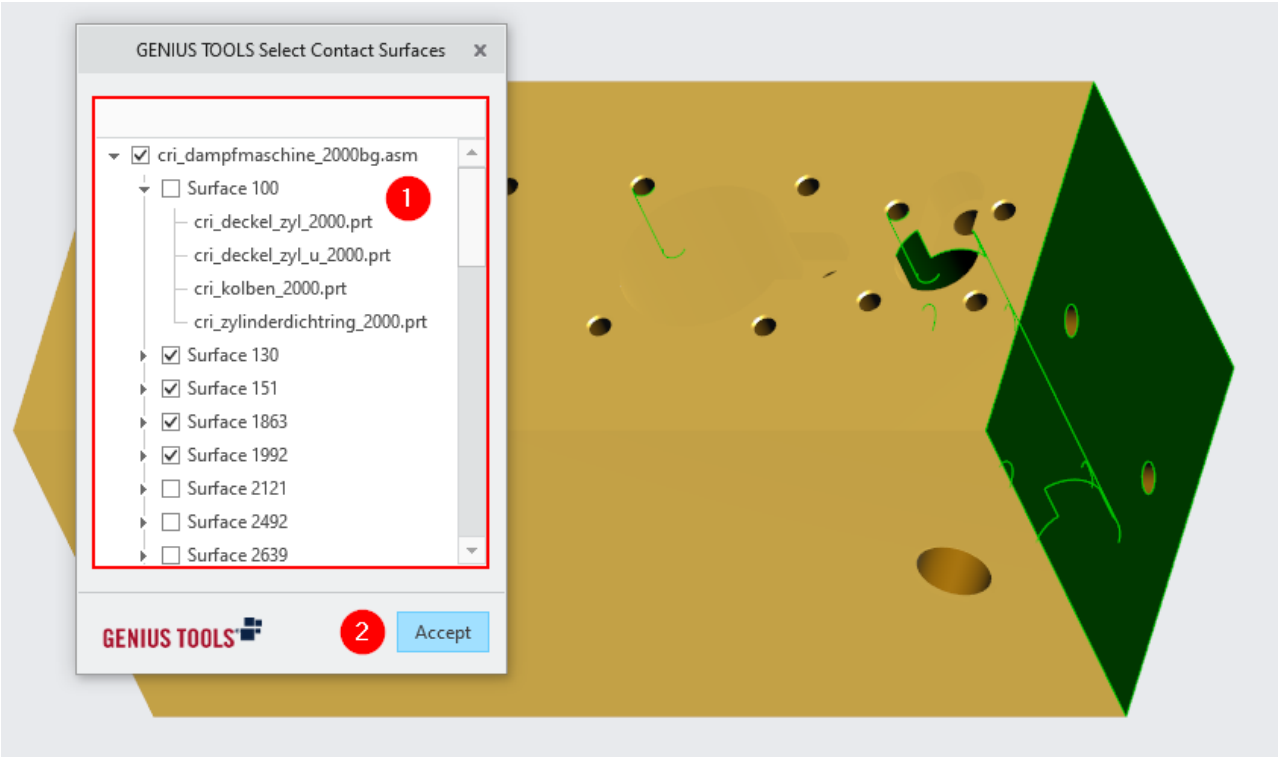



*Starting via the GT MBD in the ribbon menu*

2. Opening the module from the *Function Manager*[167]
   The contact surfaces are selected and colored at the same time.

# Displaying Contact Surfaces

When opening the program, you see the following user interface:




1. **Overview and selection**
   You will see a display of the found contact surfaces. You can select the contact surfaces you want to edit. The selected contact surfaces are highlighted in the Creo model.
2. **Accept selection**
   Confirms the selection of contact surfaces and closes the user interface of *Select Contact Surfaces*. The surfaces remain selected after closing the module and can be further edited, e.g. colored, using the Function Manager[167].


© 2026 INNEO Solutions GmbH

**Please note:** If no contact surfaces are displayed in the user interface, first use *Find Contact Surfaces*[709] to find the contact surfaces that the part has in the appropriate assembly.

# 24.31 Select Surfaces by Color

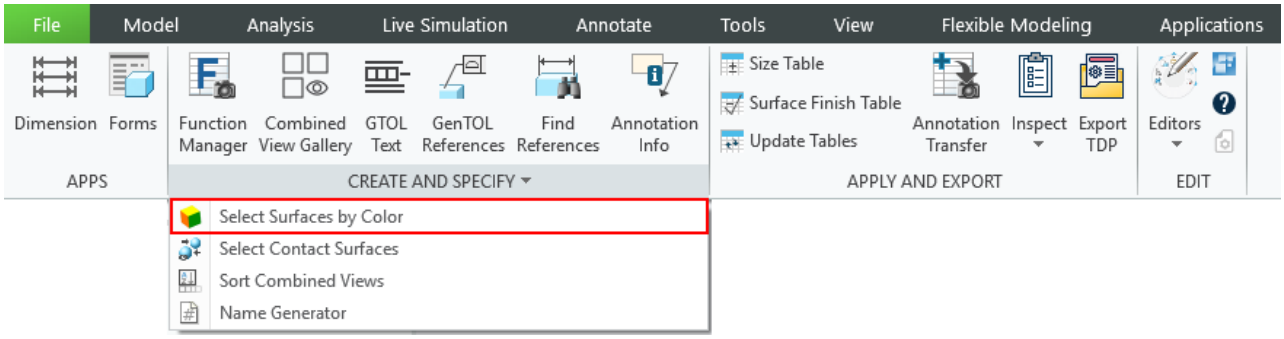
Available in the ribbon menus GENIUS TOOLS | GT MBD

Use this function to select surfaces of the same color or all uncolored surfaces. This works for both body surfaces and quilts. Selected surfaces are highlighted in the model and can thus easily be given a new color or be otherwise modified.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: in part mode

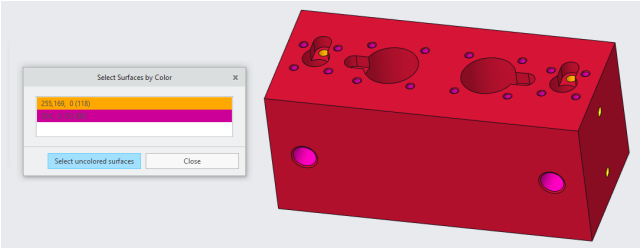
To start *Select Surfaces by Color* , click *GENIUS TOOLS* in the ribbon menu and go to the *UTILITIES* section or click *GT MBD* in the ribbon menu and go to the *CREATE AND SPECIFY* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

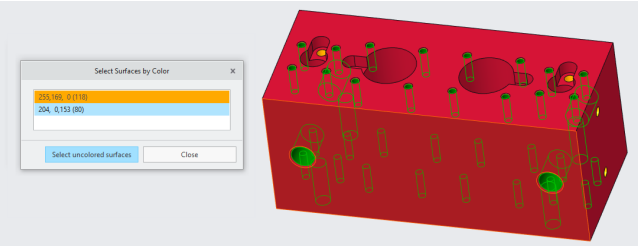
## Steps

1. Open the dialog *Select Surfaces by Color*.



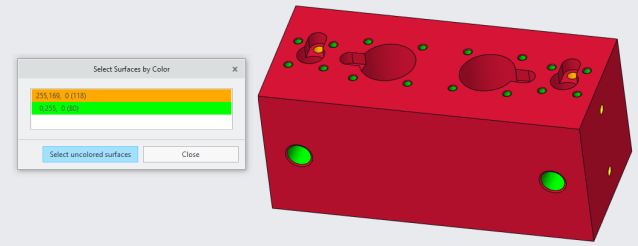

© 2026 INNEO Solutions GmbH

2. Select a color from the dialog. (Here: Pink). The line of the selected color turns light blue. Selected surfaces are highlighted in the model.



**Please note:** Surfaces with the color of the part (here: red) cannot be selected.

3. In the Creo dialog modify the selected surfaces as usual. (Here: a new color has been chosen in Creo > View > Appearances: green.)



## Configuration of the ribbon menu

Use the configuration option `gtu_start_selectSurfaceByColor` to switch off the function, so no button is displayed in the *GENIUS TOOLS* and *GT MBD* ribbon menus. (Default is 1=On/not hidden)

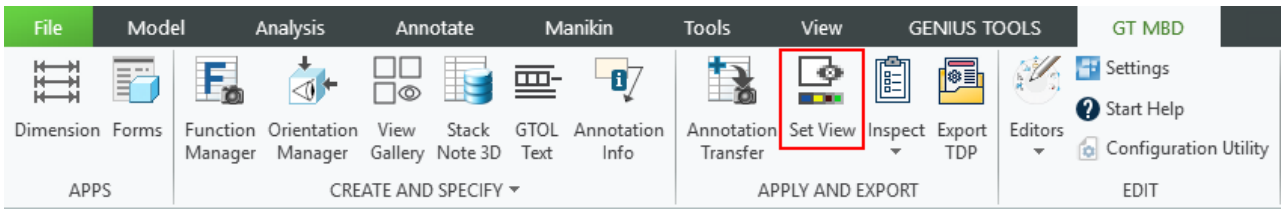
# 24.32 Set Component Combined View

Available in the ribbon menu GT MBD

Use the utility *Set Component Combined View* to temporarily set the combined views of the components in assembly mode.

## Starting the program: in assembly mode

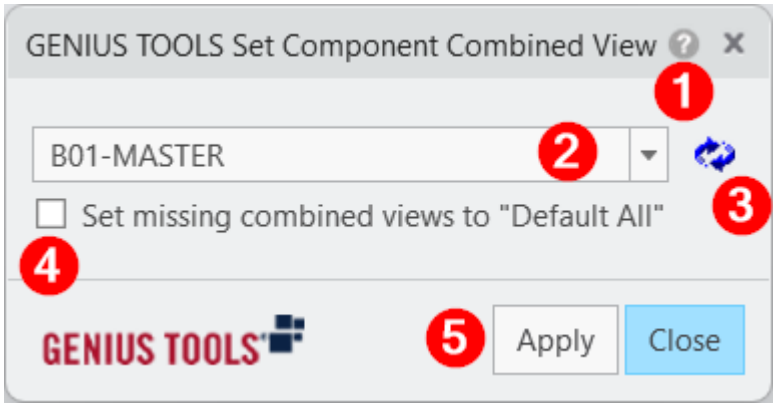
To start *Set Component Combined View* , click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section.




## Setting the Combined View of a Component in the Model

When opening the program, you see the following user interface:


© 2026 INNEO Solutions GmbH



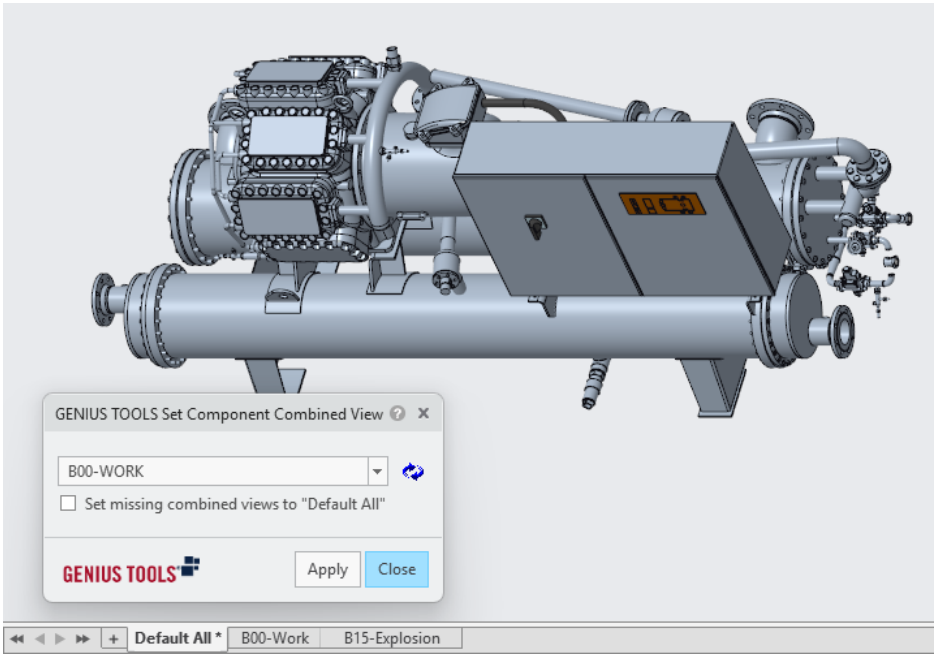

1. **Help**
   Opens the help.
2. **Overview and selection**
   This drop-down menu lists all combined views available in this model, including their frequency in parentheses. When you open the user interface, you are presented with an overview of the main model's combined views.
3. **Load available combined views**
   Loads all combined views available in this model and its components—including their frequency indicated in parentheses—for display under 2.
4. **Set missing combined views to 'Default All'**
   If the selected combined view is not available for at least one component, the combined view for that component is set to “Default All.”
5. **Applying and closing window**
   - **Apply**
     Click *Apply* to display the model in the selected combined view.
   - **Close**
     Closes the user interface. Adjusted settings are not saved.

## Use case

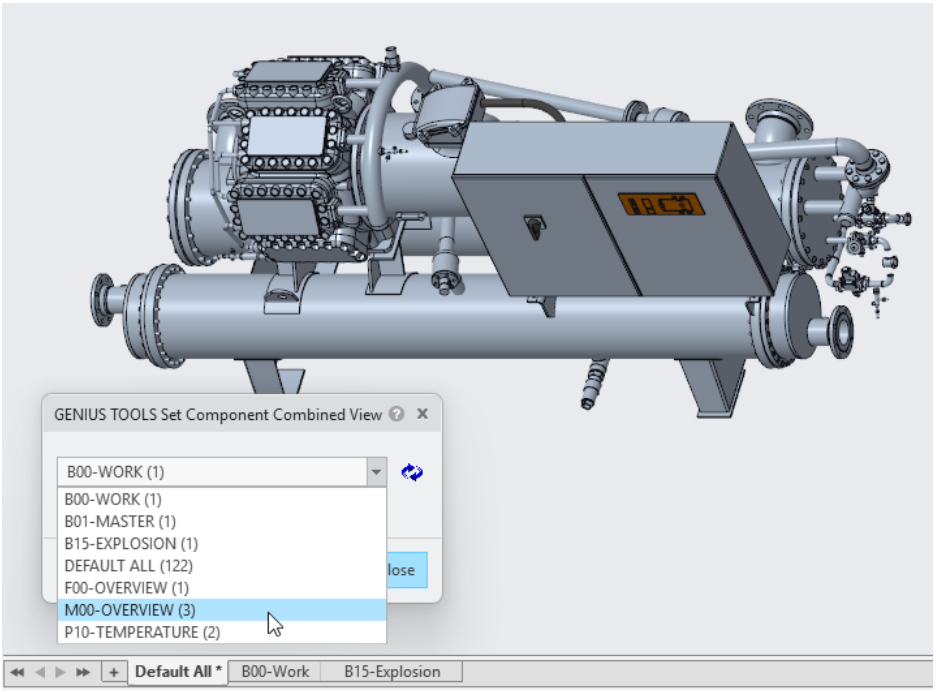
This utility is explained using a cooling unit as an example. The main model contains three combined views: *Default All*, *B00-Work* and *B15-Explosion*.

1. Open *Set Component Combined View*.


© 2026 INNEO Solutions GmbH

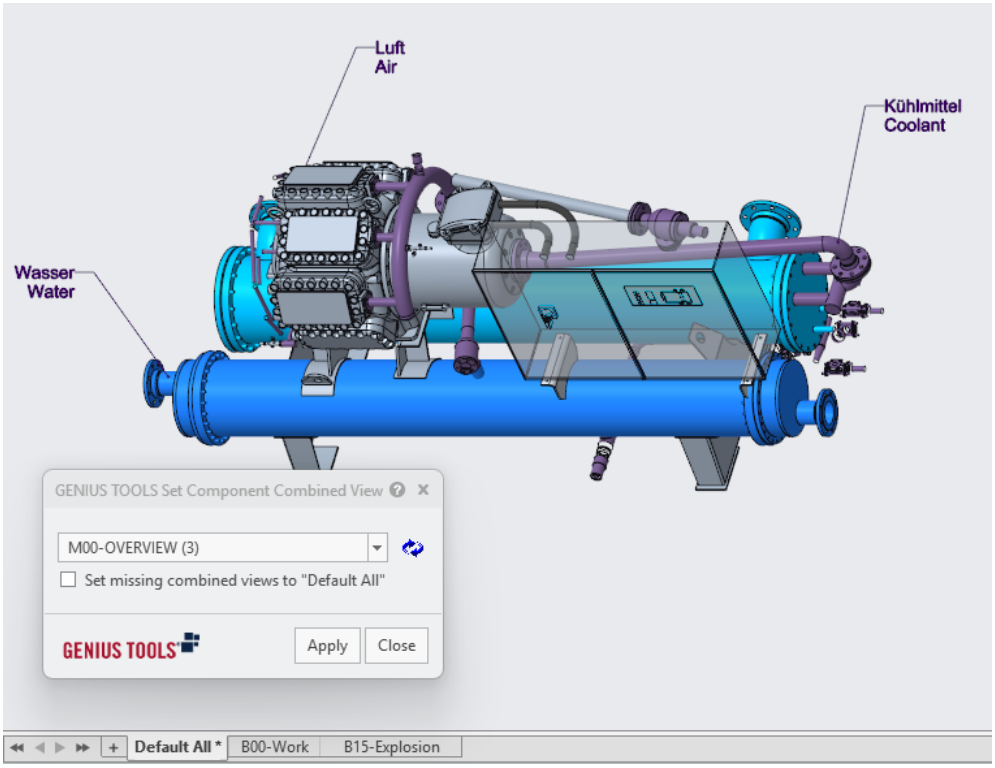



2. Klicken Sie auf *Load available combined views*. A drop-down menu will appear displaying the available combined views and their frequency.



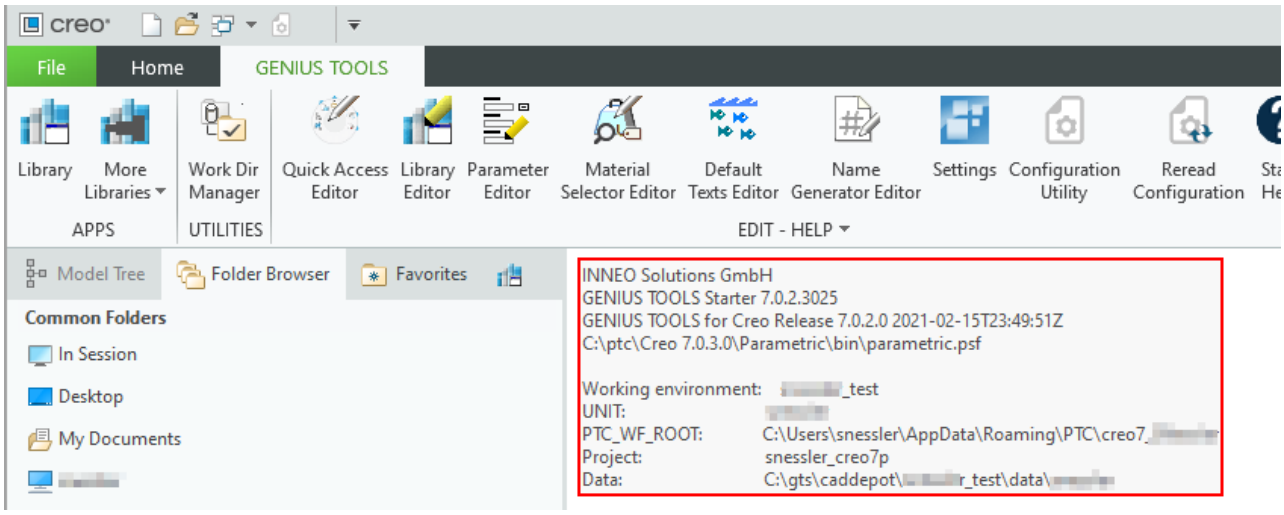

3. For example, select M00-OVERVIEW to see an overview of the media, and then click *Apply*. With just one click, you can access this combined view without having to open the corresponding submodel.


© 2026 INNEO Solutions GmbH




# 24.33 Show Information

Textual information can be customized and displayed in the Creo Parametric main window.




## Configuration

The following configuration options set up the information on display.

**gtu_ui_change_show_info**
Switches the function on. (Default is 1=on)


© 2026 INNEO Solutions GmbH

**gtu_ui_change_show_info_text**
Defines the text to be displayed.

To create a line break, write `<br>` or `</br>`.

Variables, which are defined by %, are used differently from the standard way described under Variables[913] . Variables are searched and replaced in the following order. If no value is found in one step, the next step is tried:

- Creo configuration option
- environment variable
- *GENIUS TOOLS* configuration option
- the variable is not replaced

The replacement happens at GENIUS TOOLS for Creo start. Other variables are replaced at runtime (e.g. `$ENVIRONMENT_VARIABLE$` or string replacements). Please also refer to Configuration of the GENIUS TOOLS for Creo[873] .

Example: `%GT_VERSION_STRING%<br/><br/>%PROE_START%<br/><br/>INNEO Solution GmbH`

**gtu_ui_change_show_info_color**
Defines the color of the letters. (hexadecimal color code)

**gtu_ui_change_show_info_background**
Defines the color of the background. (hexadecimal color code)

# 24.34 Show Pitch

Available in the ribbon menu GENIUS TOOLS

This function extends the displayed dimensions of a metric fine thread by the pitch value of the thread. The value *∅ 10* thus turns into *M10 x 0.75*, whereby the nominal value of the pitch is taken from the properties of the thread and the "M" for metric threads (thread type: ISO) is taken from the *iso.hol* file.

There are two ways of constructing a metric thread:

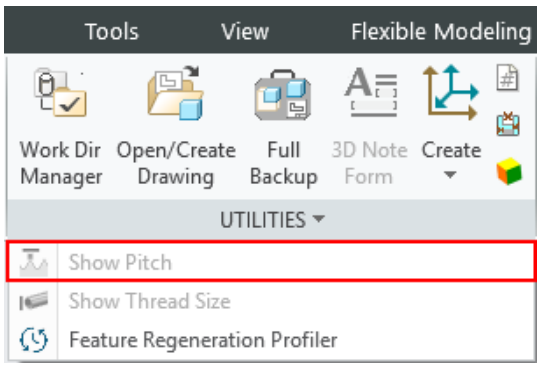
- separate feature *Cosmetic thread* for internal and external threads
- internal thread as integrated cosmetic thread within a hole feature

*Show Pitch* is based on the same operation principle as *Show Pipe Thread Size*[745] .

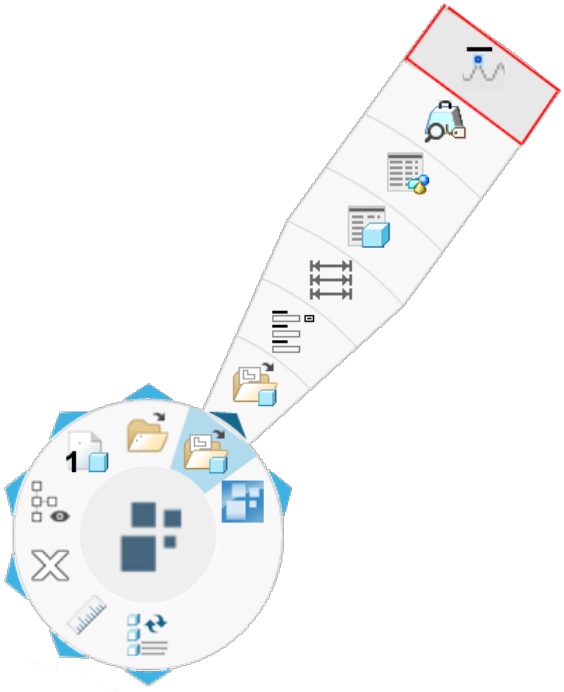

© 2026 INNEO Solutions GmbH

# Starting the program: in part mode (Dimension function)

Select a dimension of cosmetic thread feature or a threaded hole in Creo. To start *Show pitch* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*




*Starting via Quick Access*

# Procedure to show pitch

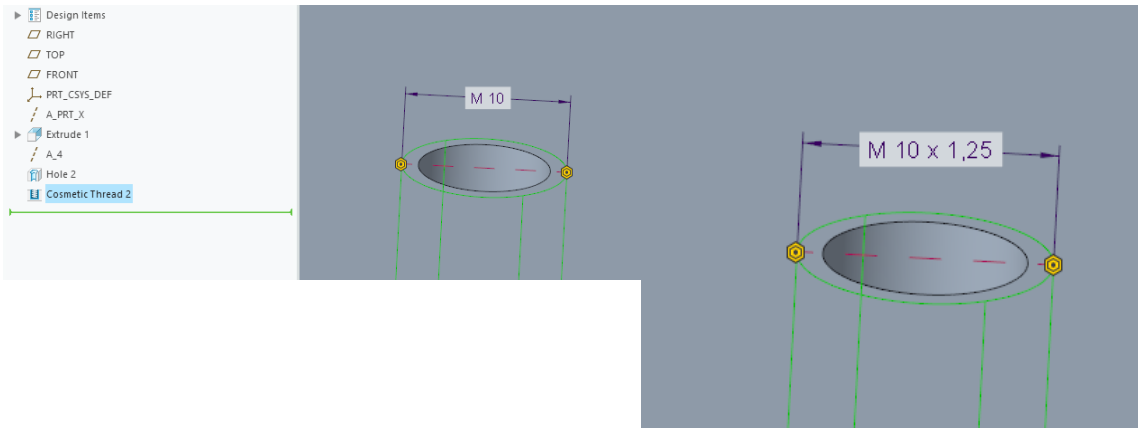
Precondition: If you start Creo Parametric in English, make sure that `gtu_show_pitch_text_definition` is set to `PITCH.` (See configuration options below.)

1. Select the dimension on which the pitch of a metric thread is to be displayed:
   – for internal threads: dimension of the thread's radius at the cosmetic element, here with ∅ *10*.
   – for external threads: dimension at the cylindrical feature
2. Open the function *Show Pitch*.
3. Special case: If multiple outer threads are contained within a volume feature, select a cosmetic thread feature.

Result: The function replaces the dimension by `M @D x &PITCH:@feat_thread_id@[.2]` and thus displays *M 10 x 1.00*

The displayed text can be configured (see below).


© 2026 INNEO Solutions GmbH




*Display of the dimension before and after using the function Show pitch on an internal thread of a Cosmetic thread feature*

# Configuration of the ribbon menu

Use the configuration option `gtu_start_show_pitch` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

# Configuring function

**gtu_show_pitch_text_definition**
If Creo Parametric is started in English, this option has to be changed from `STEIGUNG` to `PITCH`. Default: `M @D x &STEIGUNG:FID_feat_thread_id@[.2]` In a German installation the configuration has to be `STEIGUNG`.

The *feat_thread_id* variable must be used to insert the feature ID.

**Please note:** The older variable *feat_no* was renamed to *feat_thread_id*, but is still recognized.

**gtu_show_pitch_text_definition_fallback**
This value has to be customized for *Show Pitch* to work with the set Creo language. The fallback is used if `gtu_show_pitch_check_param` can not be found. The default setting is `M @D x &PITCH:FID_@feat_thread_id@[.2]`

**gtu_show_pitch_check_param**
Defines a language-dependent feature parameter. If defined, it is checked if the parameter exists. If the parameter does not exist, the function is unavailable. The default setting is `STEIGUNG`.

**gtu_show_pitch_check_param_fallback**
Defines a language-dependend feature parameter. If defined, it is checked if the parameter is existing. If the parameter does not exist, the function is unavailable. Default is `PITCH`.

See also chapter Configuring GENIUS TOOLS for Creo[755] .


© 2026 INNEO Solutions GmbH

# 24.35 Show Thread Size

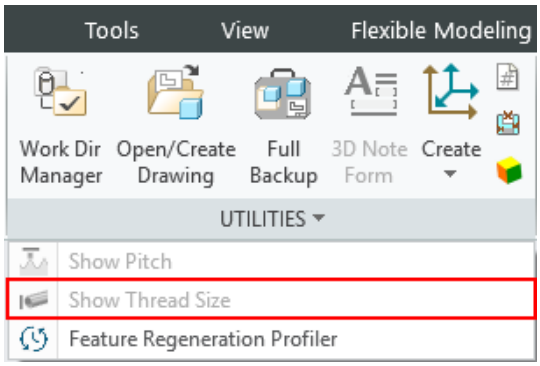
Available in the ribbon menu GENIUS TOOLS

This function extends the displayed dimensions of a bore thread by the pipe thread size. For example, the dimension ∅ *15* becomes *G3/8 (∅ 15)*, so that the thread size from the file *iso.hol* is prefixed to the numerical value. This works only for bore threads (Type: Standard, Profile: Straight, Tapped).

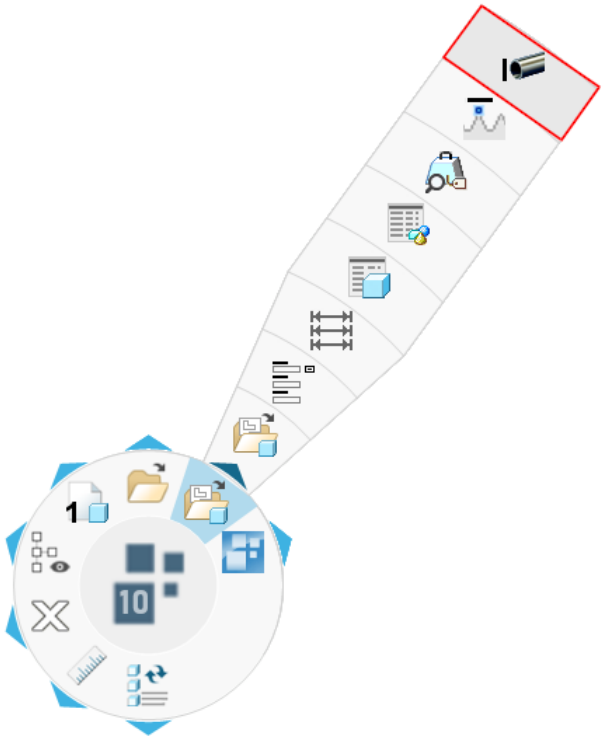
*Show Pipe Thread Size* is based on the same operation principle as *Show Pitch*[742].

## Starting the program: in part mode (Dimension function)

To start *Show pitch*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section or by using *Quick Access* (key [<]).




*Starting via the GENIUS TOOLS in the ribbon menu*



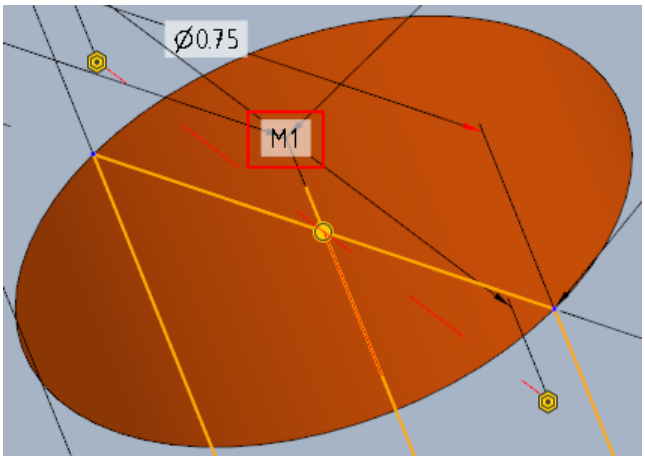

*Start via Quick Access*

## Procedure to show pitch

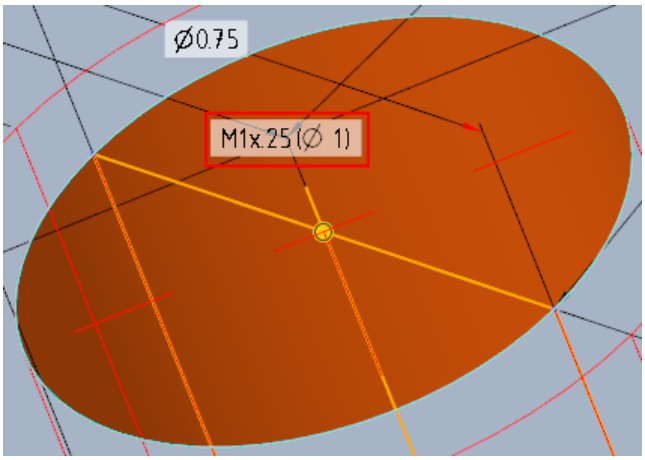
Precondition: If you start Creo Parametric in English, make sure that `gtu_show_thread_size_text_definition` is set to `SCREW_SIZE`. (See configuration options below.)

1. Select a dimension on which to display the size of the bore thread.


© 2026 INNEO Solutions GmbH




2. Open the function *Show Thread Size*.
3. The selected dimension is directly extended by the thread size.




## Configuration of the ribbon menu

Use the configuration option `gtu_start_show_thread_size` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menu. (Default is 1=On/not hidden)

## Configuring function

**gtu_show_thread_size_check_param**
Defines a language-depended feature parameter. If defined, it will be checked if the parameter exists. If the parameter does not exist, the function is unavailable.

**gtu_show_thread_size_check_param_fallback**
Defines a language-depended feature parameter. If defined, it will be checked if the parameter exists. If the parameter does not exist, the function is unavailable.

**gtu_show_thread_size_text_definition**
This value has to be customized for "*Show Thread Size*" to work with the set Creo language.

**Please note:** The older variable *feat_no* was to *feat_thread_id*, but is still recognized.


© 2026 INNEO Solutions GmbH

**gtu_show_thread_size_text_definition_fallback**
This value has to be customized for "*Show Thread Size*" to work with the set Creo language. The fallback is used if gtu_show_thread_size_check_param can not be found.

See also chapter Configuring GENIUS TOOLS for Creo[755] .

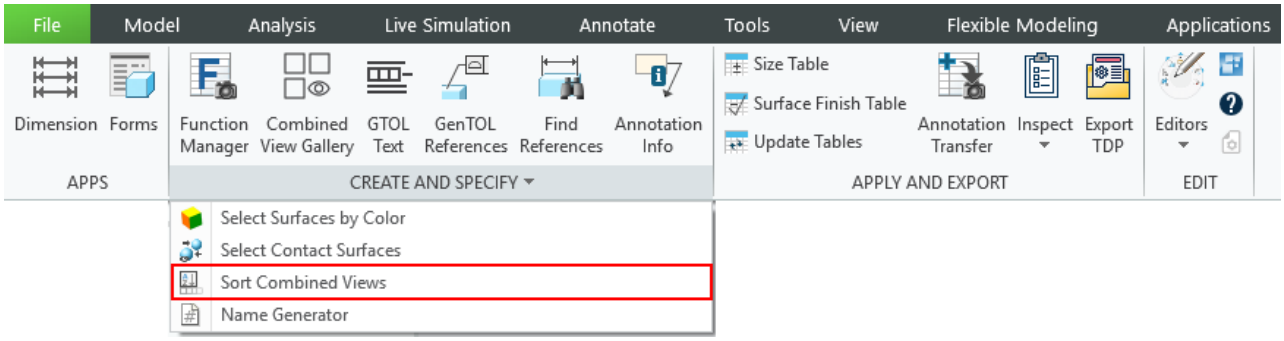
# 24.36 Sort Combined Views

Available in the ribbon menu GT MBD

The function *Sort Combined Views* sorts the combined states alphabetically and automatically displays them in the new sort order. If you create combined states with the module *Function Manager*[167] , they are automatically sorted alphabetically. If you create combined states separately in Creo, you can sort them later using the function *Sort Combined Views*.

## Starting the program: in part mode and in assembly mode

To start *Sort Combined Views* , click *GT MBD* in the ribbon menu and go to *CREATE AND SPECIFY* section. Click the button to start sorting immediately.




*Starting via the GT MBD in the ribbon menu*

Save the part / assembly to retain sort order.

# 24.37 Split Table

Available in the ribbon menu GENIUS TOOLS

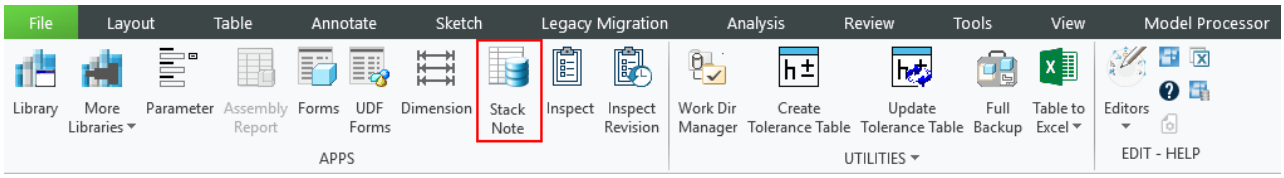
With *Split Table* you can split a table by placing it to the right or left of the table and repositioning it. Furthermore, this function can be opened via *Stack Note*[568] , see the following chapter Splitting a note table[576] .


© 2026 INNEO Solutions GmbH

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

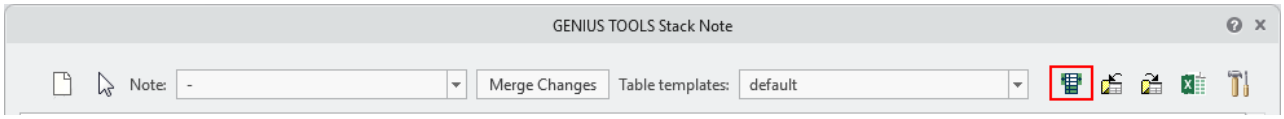
# Starting the program: in drawing mode

To start *Stack Note*, click *GENIUS TOOLS* in the ribbon menu and go to *APPS* section.




*Starting via the GENIUS TOOLS in the ribbon menu*

Click in the command bar on *Split Table*:

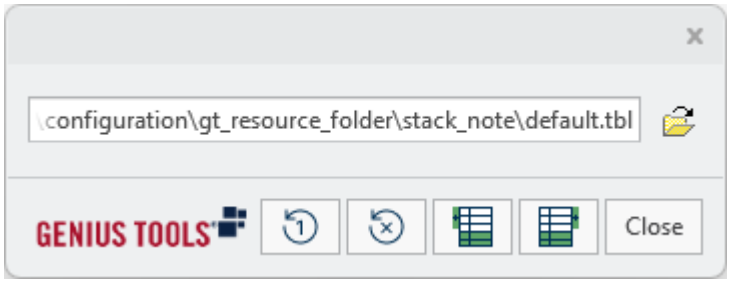



# Configuration of the ribbon menu

Use the configuration option `gtu_start_split_table` to switch off the function, so no button is displayed in the ribbon menu. (Default is 1=On/not hidden)

# Split table

When you open *Split Table* you will see the following user interface:

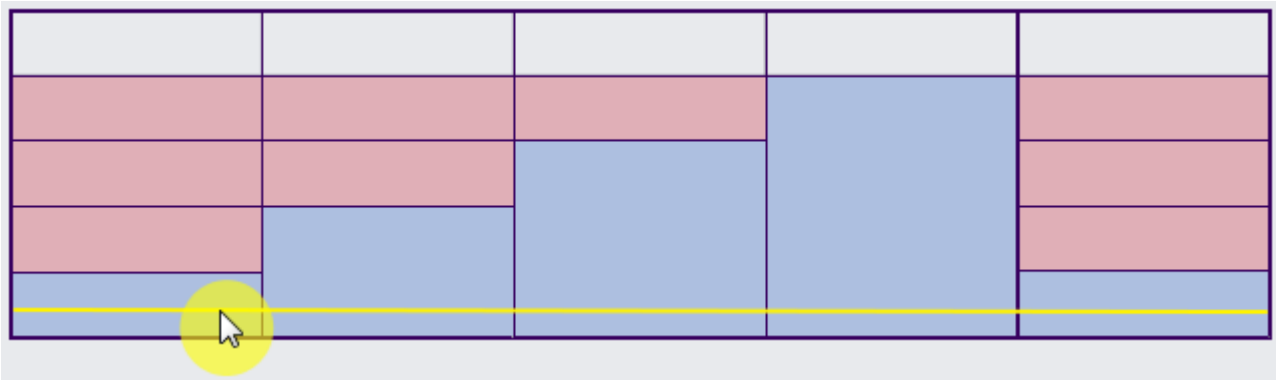



1. Select the table template by clicking on the folder icon. This template must have the same number of columns as the table to be split.

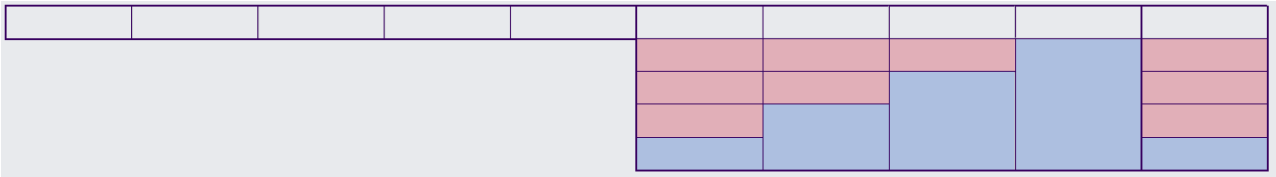
**Please note:** *Split Table* can only be opened in *Stack Note* in Creo version 10. From Creo version 11.0.1.0 onwards, it is also a standalone function.

2. Click either *Split table to left side* or *Split table to right side*.


*© 2026 INNEO Solutions GmbH*



3. Click in the table cell where you want to split the table. Based on the click coordinates, all table cells (marked in blue) are selected horizontally on the same x-axis (yellow line). Merged table cells are also taken into account.



4. You can select the table with the mouse pointer to reposition it.
5. You can use the *Revert last split* button to undo the last split, while the *Undo all table splits* button returns the table completely to its original state.

# 24.38 Toggle Symbol Variants

Available in the ribbon menu GENIUS TOOLS

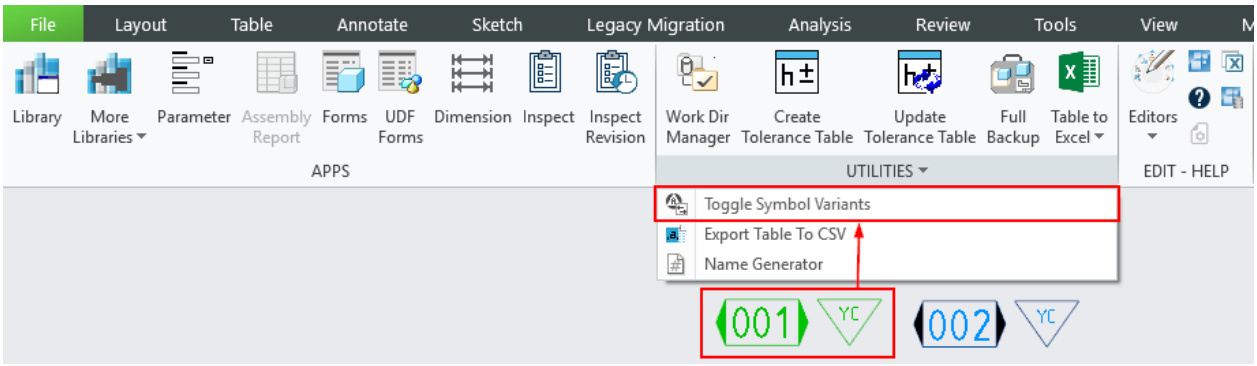
This function allows you to quickly switch between all variants of a grouped symbol. The order of the variants corresponds to the order in the SYM file that defines the symbol group.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: in drawing mode

To start *Toggle Symbol Variant*, click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.


© 2026 INNEO Solutions GmbH

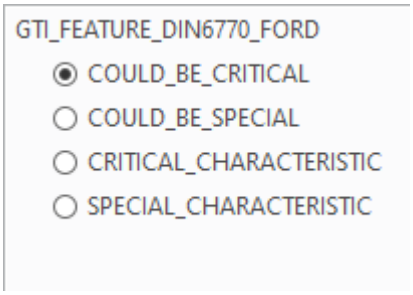



*Starting via the GENIUS TOOLS in the ribbon menu*

The following conditions must be fulfilled:

– a symbol must be selected,

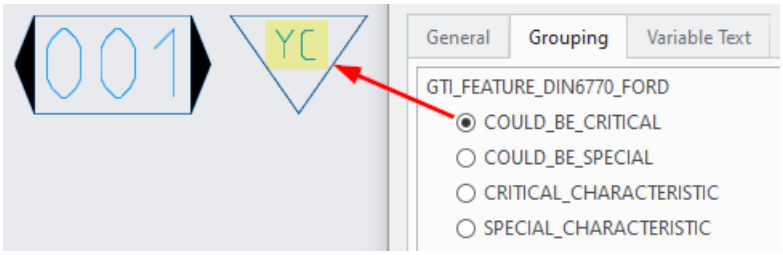
– the selected symbol must only have one variant level, i.e. no sub groups. For symbols with sub groups the button is greyed.



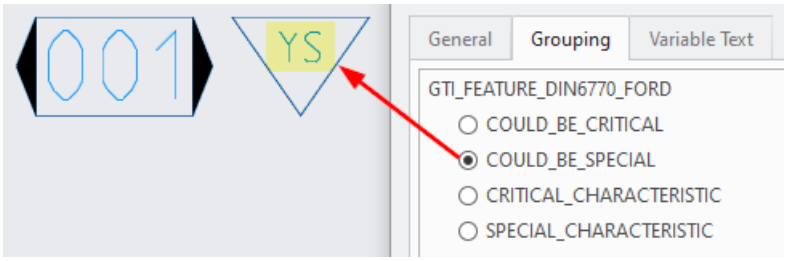

*Symbol with one variant level*

## Example

1. Initial state:

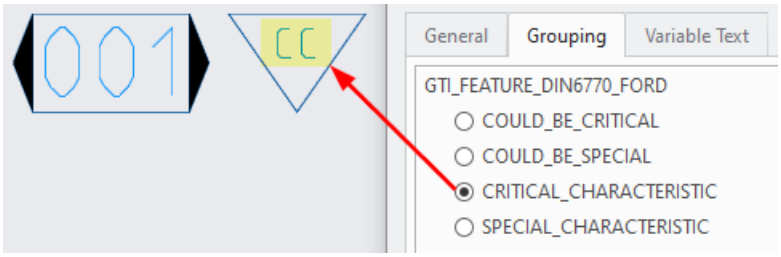



2. Click on to select the next grouping variant:




3. By clicking again on the next grouping variant is selected:


© 2026 INNEO Solutions GmbH




## Configuration of the ribbon menu

Use the configuration option `gtu_start_toggleSymbolGroups` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menus. (Default is 1=On/not hidden)

# 24.39 Work Dir Manager

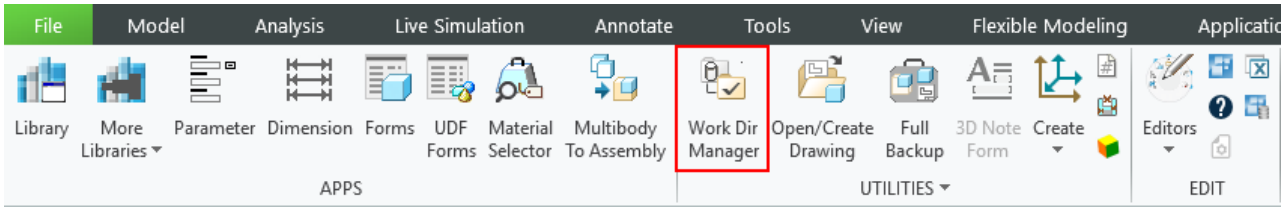
Available in the ribbon menu GENIUS TOOLS

With *Work Dir Manager* you can change the current working directory quickly. The function collects all used directories (outside of WT PDM) automatically during the regular working process.

**Please note**: This function is only available with a subscription license for *GENIUS TOOLS* for Creo.

## Starting the program: all modes

To start *Work Dir Manager* , click *GENIUS TOOLS* in the ribbon menu and go to *UTILITIES* section.

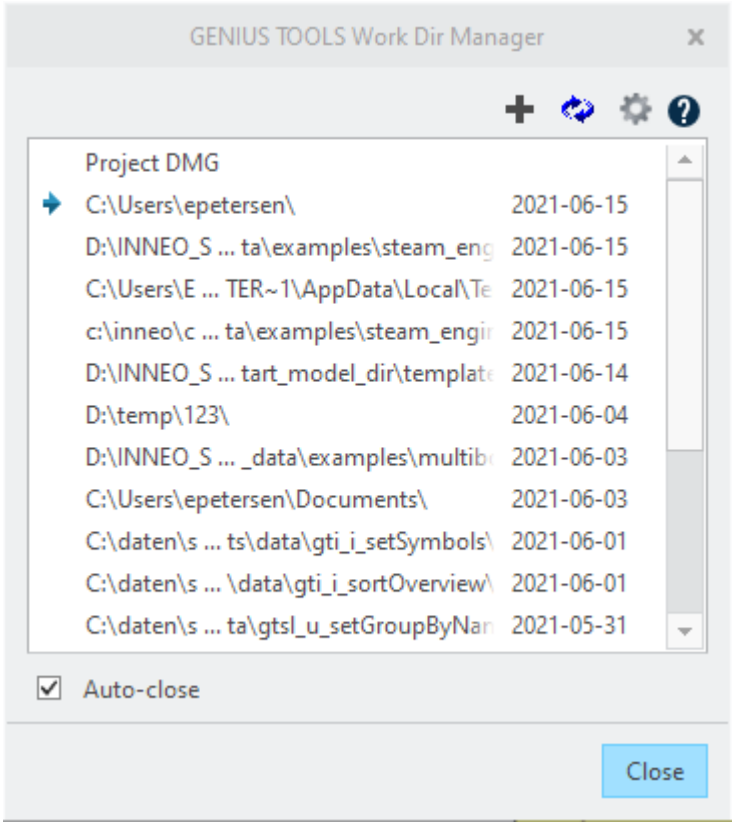



*Starting via the GENIUS TOOLS in the ribbon menu*

## Procedure

The dialog box lists the paths of all directories of the displayed models. These paths are automatically recorded.


© 2026 INNEO Solutions GmbH




*Work Dir Manager Dialog*

The current working directory can be changed by double-clicking on another path.

The paths are sorted by the last time stamp. Paths that are not used for a certain time, will be automatically deleted from the list. The time period is defined by the option `gtu_work_dir_manager_autodelete_after_days`. Paths that should not be deleted must be marked as favorites in the context menu.

## Command bar

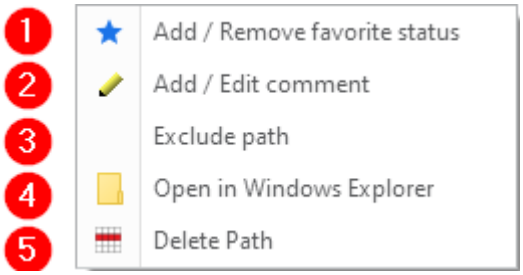
| Symbol | Name of function | Description |
|---|---|---|
|  | Add path | Add new working directory. Newly added path is automatically set as the current working directory. |
|  | Reload | Reloads Work Dir Manager |
|  | Edit excluded paths | Opens the CFG file containing all excluded paths, i. e. those that have been permanently removed from being displayed in the Work Dir Manager window (see function 3). |


© 2026 INNEO Solutions GmbH

If you check the *Auto-close* box, the dialog box will close automatically after clicking on a path.

## Context menu

Open the context menu by clicking the right mouse button in the dialog window.




*Context menu with right-mouse click*

| Function | Name | Description |
|---|---|---|
| 1 | Favorite status | Path will not be deleted. |
| 2 | Comment | A comment is shown instead of the path name. Delete the comment again by emptying the input field (empty comment). |
| 3 | Exclude path | Deletes the path permanently from the list, so that it will not be added in the future. Excluded paths are written into a CFG file which can be edited with the command bar function *Edit excluded paths*. |
| 4 | Open in Windows Explorer | Opens the selected path in Windows Explorer |
| 5 | Delete path | Deletes the path from the list until a model is opened which contains a file with this path. |

All settings are saved in the directory defined by the configuration option `gtu_work_dir_manager_save_path`. (Default: *%appdata%\INNEO\GENIUS_TOOLS\for_Creo\work_dir_manager*)

## Configuration of the ribbon menu

Use the configuration option `gtu_start_work_dir_manager` to switch off the function, so no button is displayed in the *GENIUS TOOLS* ribbon menus. (Default is 1=On/not hidden)


© 2026 INNEO Solutions GmbH

## Configuration

The configuration option `gtu_work_dir_manager_always_at_front=1` always switches the dialog to the foreground.


© 2026 INNEO Solutions GmbH

# 25 Configuring GENIUS TOOLS for Creo

*GENIUS TOOLS* for Creo can be customized on several levels, i. e. each additional function for Creo Parametric can be adapted differently for various departments (units), projects and users, which guarantees maximum flexibility.

The individual modules are controlled via configuration files (CFG files). These files can be located in different places on the network or on the local system. They are read in according to a defined logic when Creo Parametric is started. This allows you to implement many different configurations.

In this section you will find information about the startup mechanism and the configuration concept[755] as well as an overview of all configuration options of the individual modules[799] and their control via the graphical user interface of *Configuration Utility*[765].

In addition, you will find information about the *JavaScript Editor*[783], which can be called from various modules.

## 25.1 Configuration and start

The extension modules of *GENIUS TOOLS* for Creo are controlled in configuration files with the file extension *.cfg*, which can be created in different layers.

### Configuration files

A configuration is stored in a main configuration file (*gt_main.cfg*) and in a file for module settings (*gt_modules.cfg*).

Information in the configuration files can be commented with a semicolon.

**Main configuration file**
The file *gt_main.cfg* contains

– start switches of the individual modules, e. g. *gt_start_assembly*

– start switches of editors for individual modules, e. g. *gt_start_library_editor*

– options for setting the project and unit directories, see section below[758].

– further configuration options starting with *gt_*, e. g. *gt_lang*


© 2026 INNEO Solutions GmbH

```
;*** gt_main.cfg ***

gt_conf_unit=default
gt_conf_project=project
gt_resource_folder=R:\cad\creo\gt_resource_folder
gt_licpath=7766@cadserver

gt_start_assembly=1
gt_start_library=1
gt_start_material=1
gt_start_name_generator=1
gt_start_parameter=1
gt_start_quick_access=1

;*** Not in use/Nicht verwendet: ***

;gt_start_library_editor=0
;gt_start_material_editor=0
;gt_start_name_generator_editor=0
;gt_start_parameter_editor=0
;gt_start_quick_access_editor=0
```

Information for project and unit directory, GT resource folder and license server

Start switch for various modules of GTfC: access allowed

Commented start buttons for different editors of the modules

*Example of a gt_main.cfg file*

**Module configuration file**

The *gt_modules.cfg* file contains the configuration options of the individual modules, i. e. configuration options starting with *gt*_*, e.g. *gta_export_file* for GENIUS TOOLS Assembly Report.

```
;*** gt_modules.cfg ***

gta_clear_pos_parameter=1
gta_default_file=R:\cad\creo\gt_resource_folder\gta_single_level.xml
gta_export_template=R:\cad\creo\gt_resource_folder\gta_single_level_de_en.xlsx
gti_excel_template=gti_inspection_template_de_en.xlsx
gti_revision_excel_template=gti_revision_template_de_en.xlsx
gti_start_file=gti_inspection.xml
gtl_downsync=0
gtl_detail_window_udf_forms_height=1024
gtl_home_db=home.db
gtl_img_size=40
gtl_show_object_names=0
gtl_start_db=home.db
```

Set configuration options for

gta - GENIUS TOOLS Assembly Report

gti - GENIUS TOOLS Inspect

gtl - GENIUS TOOLS Library

*Example of a gt_modules.cfg file*

# Configuration concept

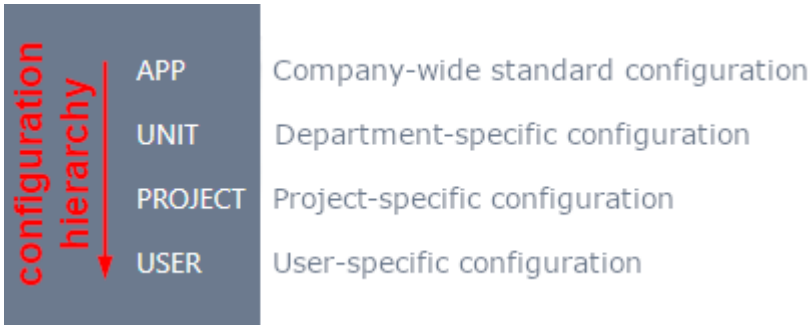
For a multi-layer configuration, the configuration files *gt_main.cfg* and *gt_modules.cfg* can be stored at different locations in the network. Such a configuration should be used if employees of a project or department are to be given configurations that differ from other projects or departments.

GENIUS TOOLS for Creo provides four different layers for a multi-layer configuration. Which folders you use for this can be controlled through the use of environment variables.

The APP layer represents the company-wide standard configuration of the software. It can be extended, modified and adapted by separate settings in the layers UNIT (department-specific), PROJECT (project-specific) and USER (user-specific). The layers are arranged hierarchically in ascending order. A set value in a subsequent layer overwrites the set value


© 2026 INNEO Solutions GmbH

in the previous layer. For example, if a different directory is set in the user settings (USER) than under PROJECT, the user's directory is used.




*Read-in of a multi-level configuration*

This section describes the path specifications that apply to working without *Startup TOOLS*. If you work with *Startup TOOLS*, consult the section at the end of the chapter.

**APP**
All values under APP are the basic configuration. This configuration is overwritten each time *GENIUS TOOLS* for Creo is updated. Data from the default path will be overwritten with the Startup TOOLS during an update. An information dialog box appears with *GENIUS TOOLS* for Creo.

Default path: <*GTfCInstallationDirectory*>*conf*

The environment variable *gt_conf_dir* can be used to redirect the first configuration directory (APP) to another defined path.

**UNIT**
Under UNIT department-specific configuration settings can be defined.

Default path: <*gt_resource_folder*>*configuration**units**default*

The default software start path can be overwritten with the *gt_conf_unit* configuration option.

**PROJECT**
All configuration settings defined under PROJECT are a project-specific configuration.

Default path: <*gt_resource_folder*>*configuration**projects**default*

The default software start path can be overwritten with the *gt_conf_project* configuration option.

**USER**
User-specific configuration settings are defined under USER.

Default path: *%appdata%**INNEO**GENIUS_TOOLS**for_Creo**configuration*

The default start path of the software can be overwritten with the configuration option *gt_conf_user*.


© 2026 INNEO Solutions GmbH

## Define project and unit directory

If you want to change the start paths to the directories of the project and unit layers, the specifications must be made in the general configuration file (*gt_main.cfg*). The specifications should only be contained in the *gt_main.cfg* file of the APP layer.

Path changes are made using the configuration options *gt_conf_unit* and *gt_conf_project* mentioned above.

## Startup mechanism

GENIUS TOOLS for Creo goes through the following startup mechanism to determine the correct configuration.

1. Read configuration file *gt_main.cfg* coming from the APP layer.
2. Determine the to be used directories UNIT, PROJECT and USER from the *gt_main.cfg* file via the configuration options *gt_conf_unit* and *gt_conf_project* and *gt_conf_user*. (These specifications should be made at the APP layer).
3. Determine and set the resource directory to use from the *gt_main.cfg* files under APP, UNIT, PROJECT, and USER using the *gt_resource_folder* configuration option. (Determining and setting the resource directory follow the configuration concept.)
4. Determining the modules to be started via the configuration option *gt_start_** (start switch).
5. Reading in the configuration settings for the individual modules. Reading in the configuration of the individual modules also follows the configuration concept.
6. Start *GENIUS TOOLS* for Creo with the determined configuration.
7. Evaluating the *GENIUS TOOLS* licenses via the *gt_licpath* configuration option.

## Reading in outdated CFG files

Up to version 6 of *GENIUS TOOLS* for Creo, there was a separate CFG configuration file for each module. These are still read in for compatibility reasons. By using the editors in the modules, CFG files are automatically transferred to the new configuration structure, i. e. only *gt_main.cfg* and *gt_modules.cfg* exist in this case.

Alternatively, *Configuration Utility* can be called. The first save process automatically adapts to the new structure.

The old CFG files are read into gt_modules.cfg in the following order.

| Reading order | Module |
|---|---|
| 1 | Library (gt_library.cfg) |


© 2026 INNEO Solutions GmbH

| Reading order | Module |
|---|---|
| 2 | Forms (gt_forms.cfg) |
| 3 | Name Generator (gt_name_generator.cfg) |
| 4 | Quick Access (gt_quick_access.cfg) |
| 5 | Material (gt_material.cfg) |
| 6 | Parameter (gt_parameter.cfg) |
| 7 | Dimension (gt_dimension.cfg) |
| 8 | Utilities (gt_utilities.cfg) |
| 9 | Assembly Report (gt_assembly.cfg) |
| 10 | UDF Forms (gt_udf_forms.cfg) |

## Language configuration

The language of the user interface is defined by the Creo-language. It is German if you have a German Creo, otherwise, it is English.

There are different configuration options for the display of content, which determine these for each module. Some of the configuration options are preset with the language of the user interface and can be overwritten later by the configuration files.

| Option | Default |
|---|---|
| gt_lang | UI language |
| gta_lang | UI language |
| gtf_def_lang<br>gtf_lang | en, de (ger), fr<br>UI language |
| gti_def_lang<br>gti_lang | en<br>UI language |
| gtl_def_lang<br>gtl_lang | en<br>UI language |


© 2026 INNEO Solutions GmbH

| Option | Default |
|---|---|
| gtm_db_def_lang<br>gtm_db_lang | en<br>UI language |
| gtp_lang | UI language |
| gtqa_db_def_lang<br>gtqa_db_lang | en<br>UI language |
| gtu_lang | UI language |
| gtuf_lang | UI language |

The module contents are displayed in the defined language (`*_lang`). If this is not defined in the data to be displayed, the fallback language is used (`*_def_lang`), if available. If this is not available, or defined in such a way that it is also not available in the data to be represented, it falls back to *en*.

An exception is the `gtf_def_lang` option, which can contain several language abbreviations. These are automatically added when a new form is created in GENIUS TOOLS Forms. The first specification (en by default) determines the fallback language.

For information on the effective range of the configuration options, please refer to configuration settings[928].

# Working with Startup TOOLS

If you are working with *Startup TOOLS*, you have installed *GENIUS TOOLS Starter* (GTS) and in this case you should use the corresponding GTS variables. This ensures that the configuration layers that apply in *GENIUS TOOLS Starter* largely correspond to the GTfC layers. The GTS environment variables are set by using *GENIUS TOOLS Starter*.

| GTS environment variable | Description |
|---|---|
| GTS_CONFIGURATION_DIR | Application-specific reference to the selected configuration directory. Path: *<Caddepot>\<OperatingEnvironment>\<application>\configuration* |
| GTS_PROJECT_DIR | Environment variable for project-specific configurations. |

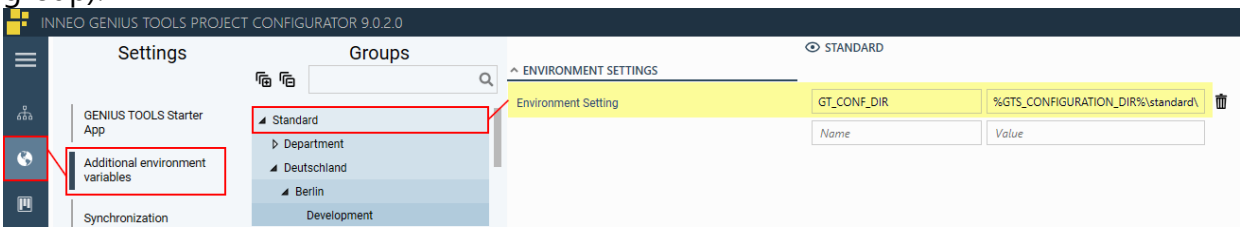

© 2026 INNEO Solutions GmbH

| GTS environment variable | Description |
|---|---|
| | Application-specific reference to the selected project directory. Path: *<Caddepot>\<OperatingEnvironment>\<application>\configuration\projects\<ProjectName>* |
| GTS_ROOT_DIR | Specifies the working environment. Path: *<Cadpool>\<OperatingEnvironment>* |
| GTS_UNIT_DIR_NAME | Contains the folder name of the last unit of a selected organizational structure or is empty if not working with units. |


© 2026 INNEO Solutions GmbH

**Procedure:**

1. **Reset variable for the APP layer**
   In the delivery state of the Startup TOOLS, the startup layer is set to *%GTS_ROOT_DIR%\apps\gtfc\conf*. Reset the startup layer with the environment variable GT_CONF_DIR to:
   `GT_CONF_DIR=%GTS_CONFIGURATION_DIR%\standard`.
   The GTS environment variable GT_CONF_DIR can be created and defined in the *GENIUS TOOLS Project Configurator* as an environment variable in the global settings ("Standard" group).




*Menu point Settings in GENIUS TOOLS Project Configurator*
   Result: As a result, the APP layer of GTFC is the same as the default configuration of GTS, i. e. the GTS configuration layer *Default* (global settings).

2. **Create general CFG file in default directory**
   Copy the *gt_main.cfg* file from the %GTS_ROOT_DIR%*apps**gtfc**conf* folder and place it in the default directory *%GTS_CONFIGURATION_DIR%\standard*.
   The PROJECT layer and the USER layer of *GENIUS TOOLS* for Creo correspond to the configuration layer *Projects* and *User* of GENIUS TOOLS Starter. However, the UNIT layer must be adapted.

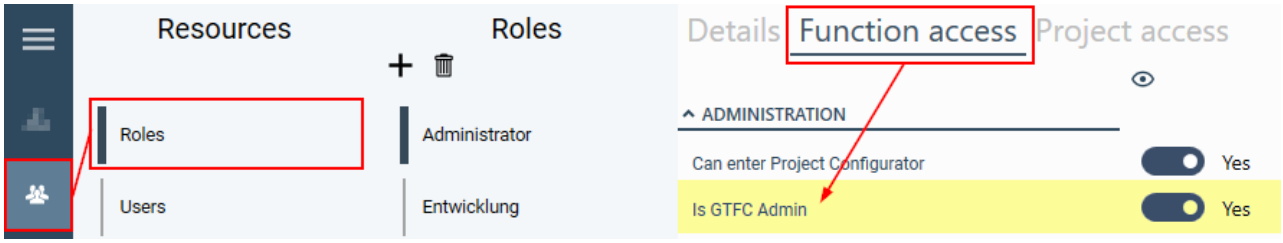
3. **Customize entries for unit directory**
   In the delivery state of the Startup TOOLS, the unit path in the *gt_main.cfg* is specified as follows:
   UNIT path with *Startup TOOLS*: *%GTS_CONFIGURATION_DIR%\standard*.
   Change this to:
   `gt_conf_unit=%GTS_CONFIGURATION_DIR%\units\%GTS_UNIT_DIR_NAME%`.
   Result: As a result, the UNIT layer of *GENIUS TOOLS* for Creo corresponds to the last unit configuration layer of *GENIUS TOOLS Starter*.

**Startup TOOLS variables:**
The following variables are created by using *GENIUS TOOLS Starter*.

The resource folder is defined by the *Startup TOOLS* with
`gt_resource_folder=%GTS_CONFIGURATION_DIR%\gt_resource_folder`.

For access to the editors of the individual modules, *GENIUS TOOLS Starter* provides the option that these can only be opened by the administrator. The variable *%GTFC_ADMIN%* is created for this purpose. The setting is made in the Function Access area of *GENIUS TOOLS Project Configurator*.


© 2026 INNEO Solutions GmbH




The general configuration file *gt_main.cfg* therefore contains several GTS variables.

```
;*** gt_main.cfg ***

gt_conf_project=%GTS_CONFIGURATION_DIR%\projects\%GTS_PROJECT_DIR_NAME%
;gt_conf_project=project
gt_conf_unit=%GTS_CONFIGURATION_DIR%\units\%GTS_UNIT_DIR_NAME%
;gt_conf_unit=%GTS_CONFIGURATION_DIR%\standard
gt_licpath=%GT_LIC_SERVER%
gt_resource_folder=%GTS_CONFIGURATION_DIR%\gt_resource_folder
gt_start_library_editor=%GTFC_ADMIN%
gt_start_material_editor=%GTFC_ADMIN%
gt_start_name_generator_editor=%GTFC_ADMIN%
gt_start_parameter_editor=%GTFC_ADMIN%
gt_start_quick_access_editor=%GTFC_ADMIN%
gt_start_udf_form_editor=%GTFC_ADMIN%

;*** Not in use/Nicht verwendet: ***

;gt_replace_character_if_not_found=
;gt_show_license_dialog=1
;gt_start_assembly=0
;gt_start_assembly_editor=0
;gt_start_assembly_editor=%GTFC_ADMIN%
;gt_start_configuration_utility=0
```

*Example of a gt_main.cfg file with GTS variables*

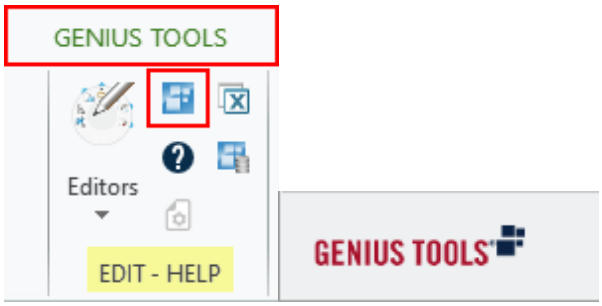
# 25.2 Viewing the configuration

The button *GENIUS TOOLS for Creo Settings* opens an overview of the set configuration options. This view is visible at all levels of the configuration hierarchy[756]. The displayed configuration options can be saved as a CFG file in any location. All configuration options listed here are edited using Configuration Utility[765].

## Starting the program: in all modes

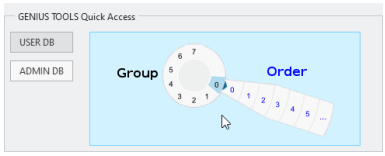
Start *GENIUS TOOLS for Creo Settings* via the *GENIUS TOOLS* ribbon menu in the *EDIT-HELP* section, via the *GENIUS TOOLS* icon in the modules and many utilities or by using *Quick Access Editor*.


© 2026 INNEO Solutions GmbH




*Start via the ribbon menu*

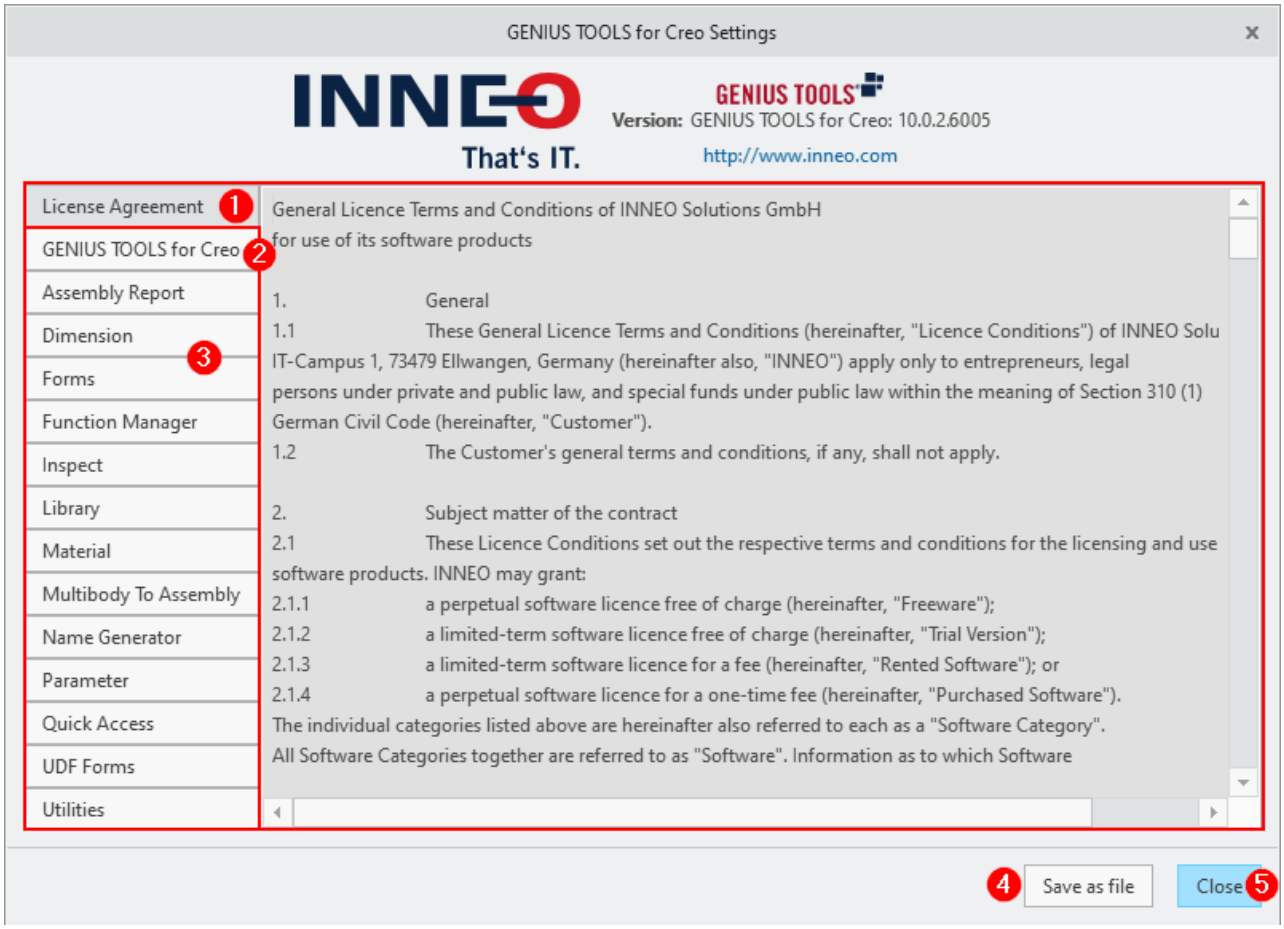
*Start via the GENIUS TOOLS icon, located in the modules and many utilities in the lower left corner of the user interface*




*Start via Quick Access Editor by clicking on the highlighted image*

## Viewing and saving configuration options

The user interface of *GENIUS TOOLS for Creo Settings* consists of the following elements:





© 2026 INNEO Solutions GmbH

1. **License agreement**
2. **Cross-module configuration options**
3. **Configuration options by module**
4. **Save configuration as CFG file**
   By default, the configuration options are saved as a main configuration file (*gt_main.cfg*) and as a file for module settings (*gt_modules.cfg*). In *GENIUS TOOLS for Creo Settings*, you can use this button to save an overview of the set configuration options separately as a CFG file in a location of your choice.
5. **Close**

# 25.3 Telemetry Data Transfer

Telemetry can be collected for troubleshooting, performance optimization, and product improvement and development. Telemetry collection must be explicitly enabled; see *Documentation Telemetry Data* in the Help directory.

# 25.4 Configuration Utility

Available in the ribbon menu GENIUS TOOLS

*Configuration Utility* allows you to maintain all configuration settings for GENIUS TOOLS for Creo centrally and define the desired locations. Doing so replaces manual editing of the configuration files[755] (*gt_main.cfg* and *gt_modules.cfg*), a time consuming task.

With *Configuration Utility* all entries can be made in a graphical user interface. Adjustable value ranges are specified and incorrect settings are displayed.

Editing outside Configuration Utility is not recommended.

A special highlight is the possibility of using variants of values as comments within the graphical user interface. This makes it possible for administrators to very quickly test different configuration variants without having to manually enter the various settings.

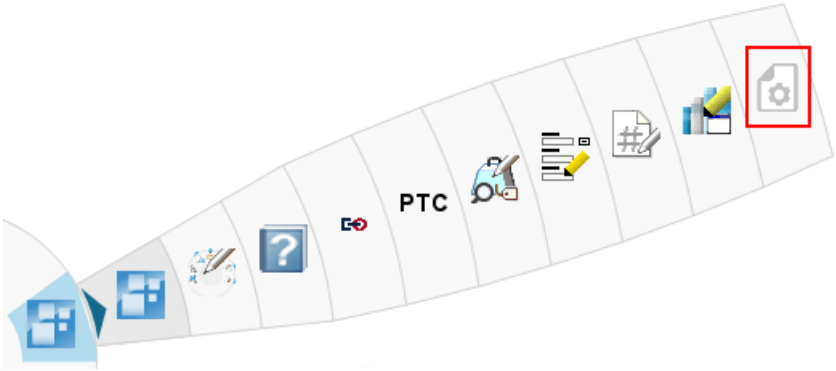
If you want to test the effects of the changes made without restarting the program, click on the function *Reread Configuration* in standby mode, see Directly applying changes[777].

## 25.4.1 Starting the program

*Configuration Utility* can be accessed in various ways via the icon :

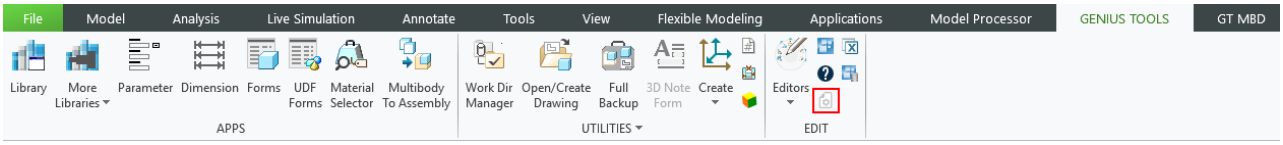
1. In all Creo modes: via *Quick Access*[544]:

 © 2026 INNEO Solutions GmbH 




*Start via Quick Access*

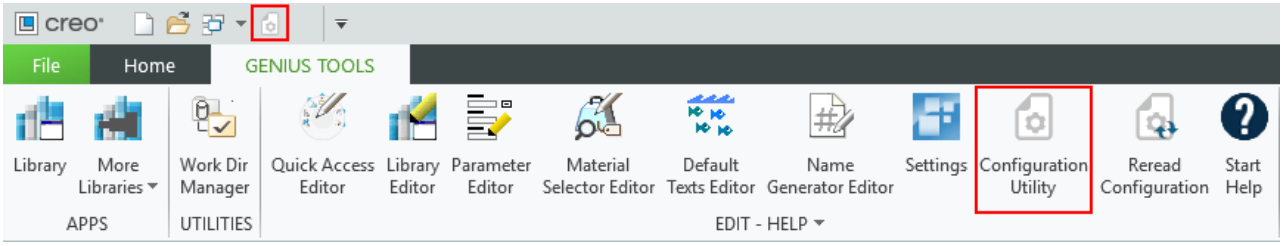
2. In part, assembly and drawing mode: via the *GENIUS TOOLS* ribbon menu in the section *EDIT*.




*Start in part, assembly and drawing mode*

3. In standby mode: via Creo header

4. In standby mode: via the *GENIUS TOOLS* ribbon menu in the group *EDIT-HELP*




*Start in standby mode*

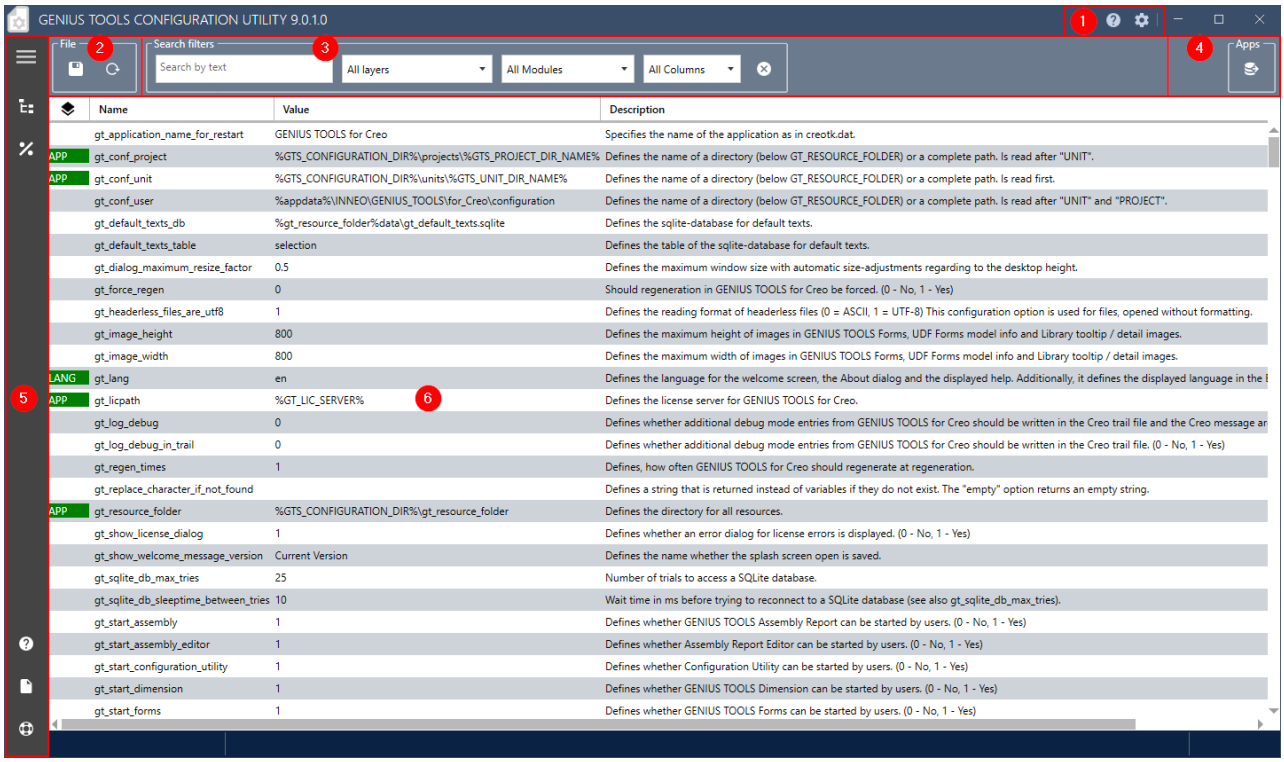
**Please note:** Do not start the *Configuration Utility* outside a Creo environment, otherwise the environment variables may not be read correctly.

Note the special procedure for saving changes and reading in a configuration during a Creo session, see Saving changes 775.

## 25.4.2 User interface

The user interface of *Configuration Utility* consists of the following elements:


© 2026 INNEO Solutions GmbH




*User interface of Configuration Utility*

1. Header: Help and user menu
2. File: Saving changes[775] and Reseting configuration options[778]
3. Search filter[769]
4. Apps: Opens Database Version Control[782] to check the Creo version status
5. Sidebar
6. Configuration overview[768]: Double click on a line opens the editor for a configuration option.

# User menu

The gear icon opens the user menu with the functions:




– Language: German, English. The language can be changed at runtime.

– Theme: blue, light, dark

– Help: Opens this document.

# Sidebar

The hamburger menu is used to expand the functions:

– Path display[780] for the levels App, Unit, Project and User

– Environment variables[781]


© 2026 INNEO Solutions GmbH

– Help: Opens this document.

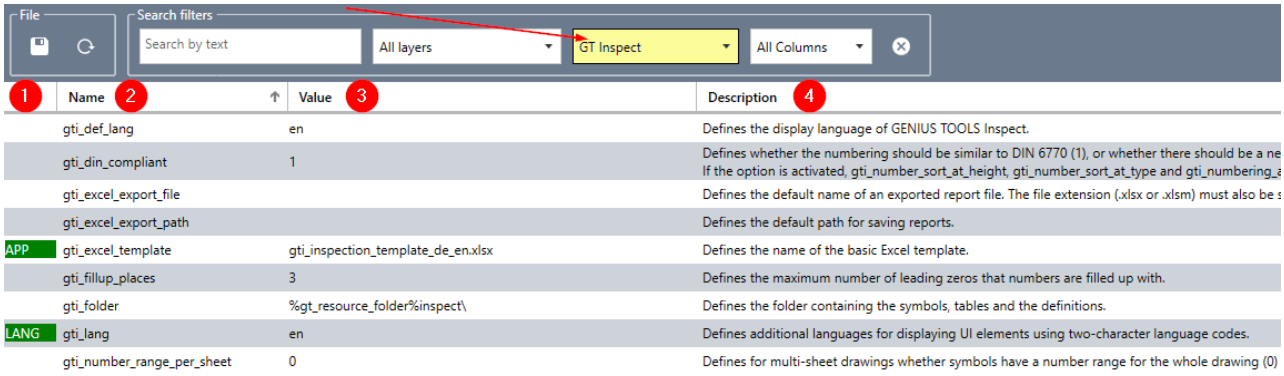
– Log Folder: Opens the folder where the *Configuration Utility* log files can be found.

– Support: Opens the INNEO support web page.

## 25.4.3 Configuration overview

The configuration overview displays all configuration options for the modules of GENIUS TOOLS for Creo.

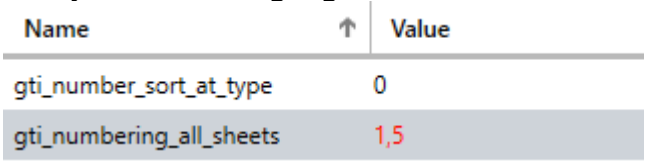
Double-clicking on a line opens the configuration editor[771], in which the desired values are entered for the layers APP to USER. If the preset configuration values have been changed, this can be seen at first glance by the green bar in the first column.




*Overview of configuration options for Inspect*

The overview is divided into the following columns:

1. **Layer** indicates configuration options with modified start values by an entry with green background color. The lowest valid layer in which the value was adjusted is displayed, see also configuration concept[756]. Possible values are:
   – LANG LANG for *message.txt files* from a subfolder under *%GTS_ROOT_DIR%\apps\gtfc\text*
   – ENV ENV for environment variables
   – APP APP / UNIT / PROJECT / USER for the levels
2. **Name** of the configuration option
3. **Value** shows the value that is currently valid for the corresponding configuration option. Faulty values are highlighted with red font.




4. **Description** contains explanatory text for the configuration option. Depending on the language setting, this is available in German or English. In other languages, the English


© 2026 INNEO Solutions GmbH

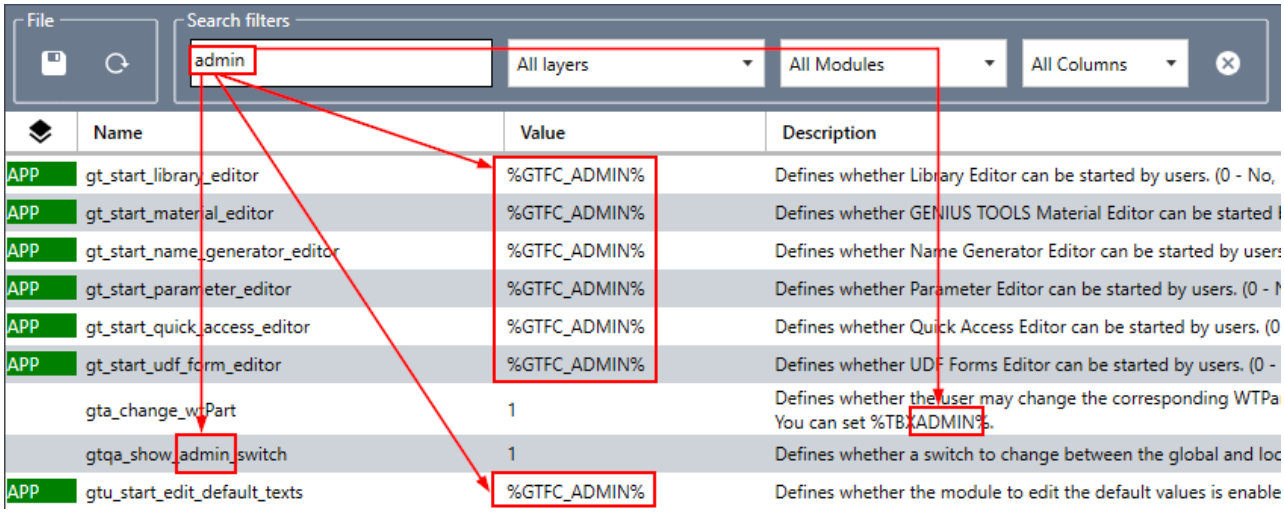
version is used by default. The language can be changed within a session in the user menu[767].

## 25.4.4 Searching configuration options

You can use the search filter to filter configuration options. The filters can be combined in any way. Configuration options that match all filter rules are displayed as search results.

### Searching with text

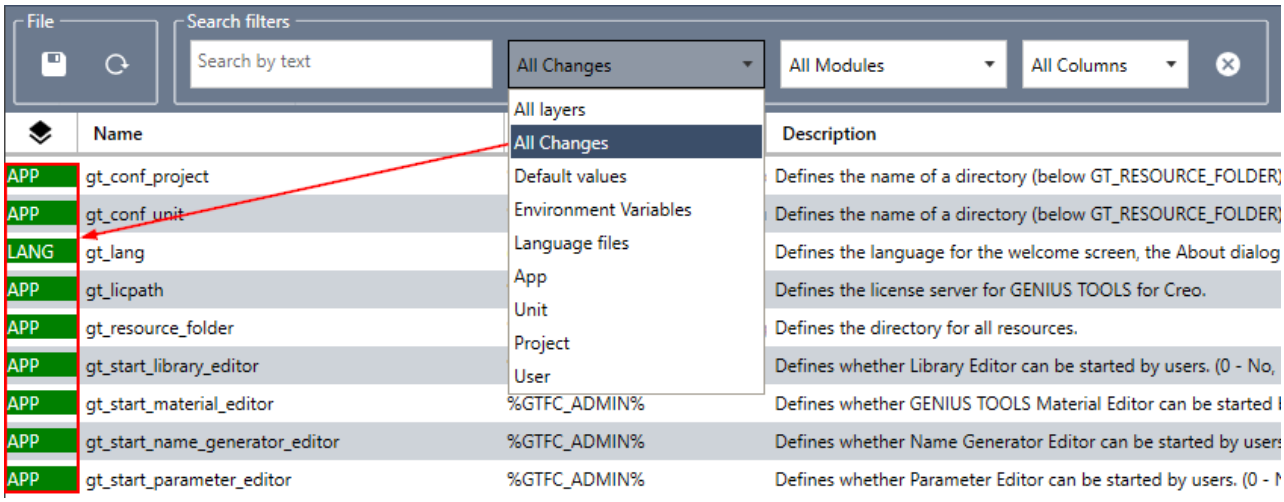
The text search allows searching the table for character combinations, words or word groups.




*Exemplary search for "admin"*

### Searching by layers

The drop-down menu *All layers* refers to the first column of the option table, where changed configuration options are displayed.




*Exemplary search for "All Changes"*


© 2026 INNEO Solutions GmbH

**All layers** does not apply any filter.

**All Changes** shows only the lines where the value of the configuration option was changed (entry highlighted in green in the first column), regardless of the concrete level.

**Default values** displays only unmodified basic values.

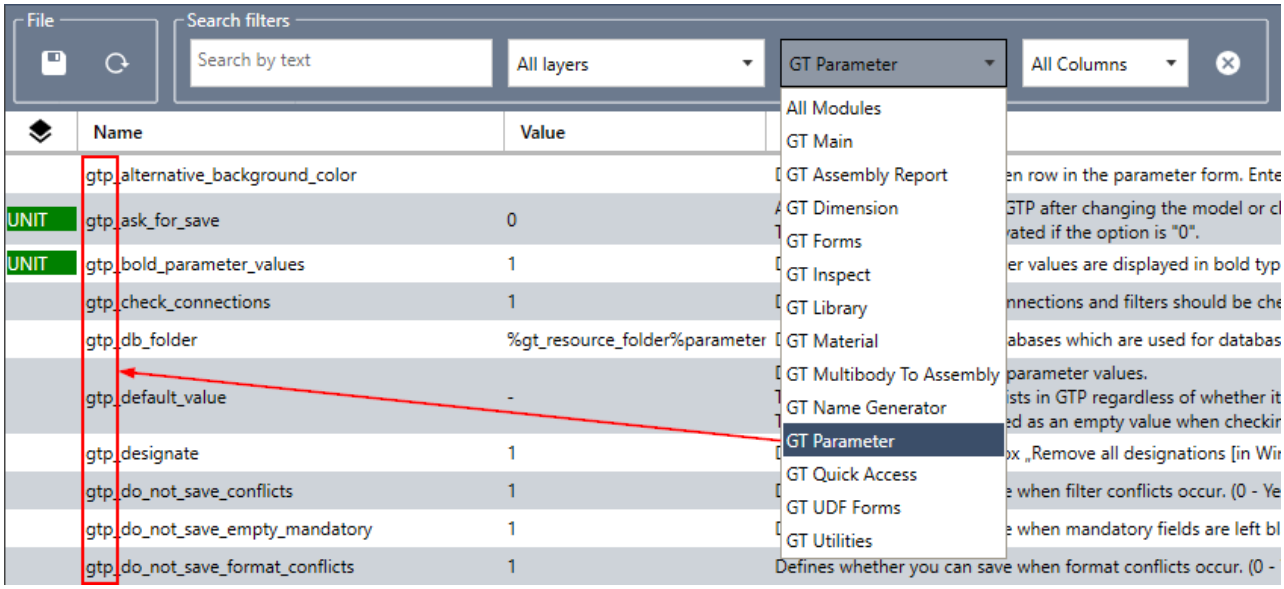
**Environment variables** shows only lines where the value of the configuration option was last set by an environment variable.

**Language files** only displays lines where the value of the configuration option was last set by a *message.txt* in a subfolder of *%GTS_ROOT_DIR%\apps\gtfc\text*.

**App, Unit, Project, User** displays only lines where the value of the configuration option has been set at the selected level.

## Searching for modules

The filter of the third search menu shows only configuration options of the selected modules.




*Exemplary search for "GT Parameter"*

**All Modules** does not apply any filter.

**GT Main** displays only configuration options that make general, module-wide settings. These start with the abbreviation *gt_* and are written to file(s) *gt_main.cfg*.

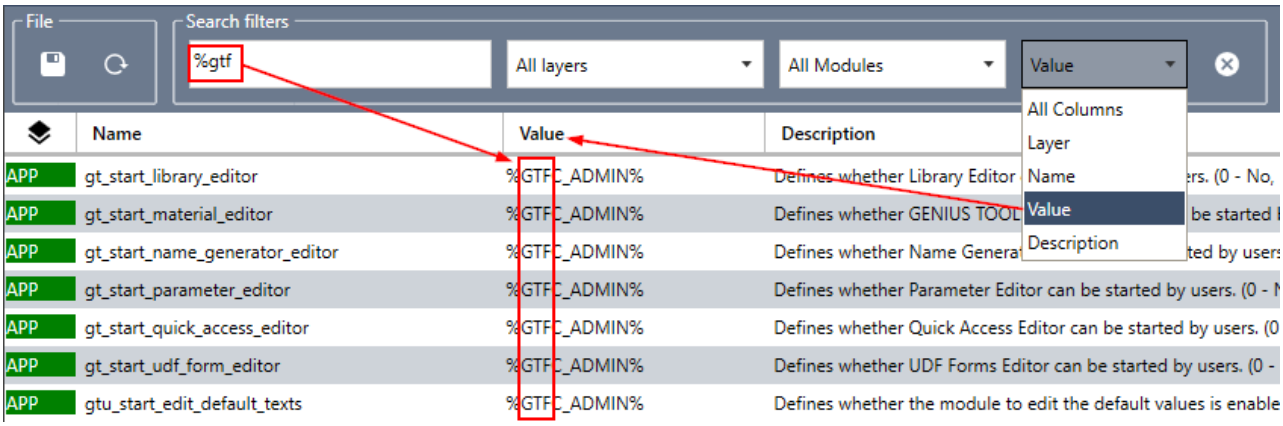
**GT Assembly Report to GT Utilities** displays the configuration options for the corresponding GENIUS TOOLS for Creo module. These can be identified by the respective abbreviation that precedes the name of the configuration option.

Sometimes, configuration options are not retrieved from the actual configuration file due to entry errors or other reasons. *Configuration Utility* can be used to detect such deviating entries and correct them if necessary, see setting values 773.


© 2026 INNEO Solutions GmbH

## Searching for columns

The column filter allows to limit the text search column by column. For example, character combinations can be searched only in the *Value* or *Description* columns - this significantly simplifies the search for frequently occurring characters.




*Exemplary search for "gtf" in the column Value*

## Clearing filters

The *Clear all filters* button resets all fields in the search filter area to their initial value.

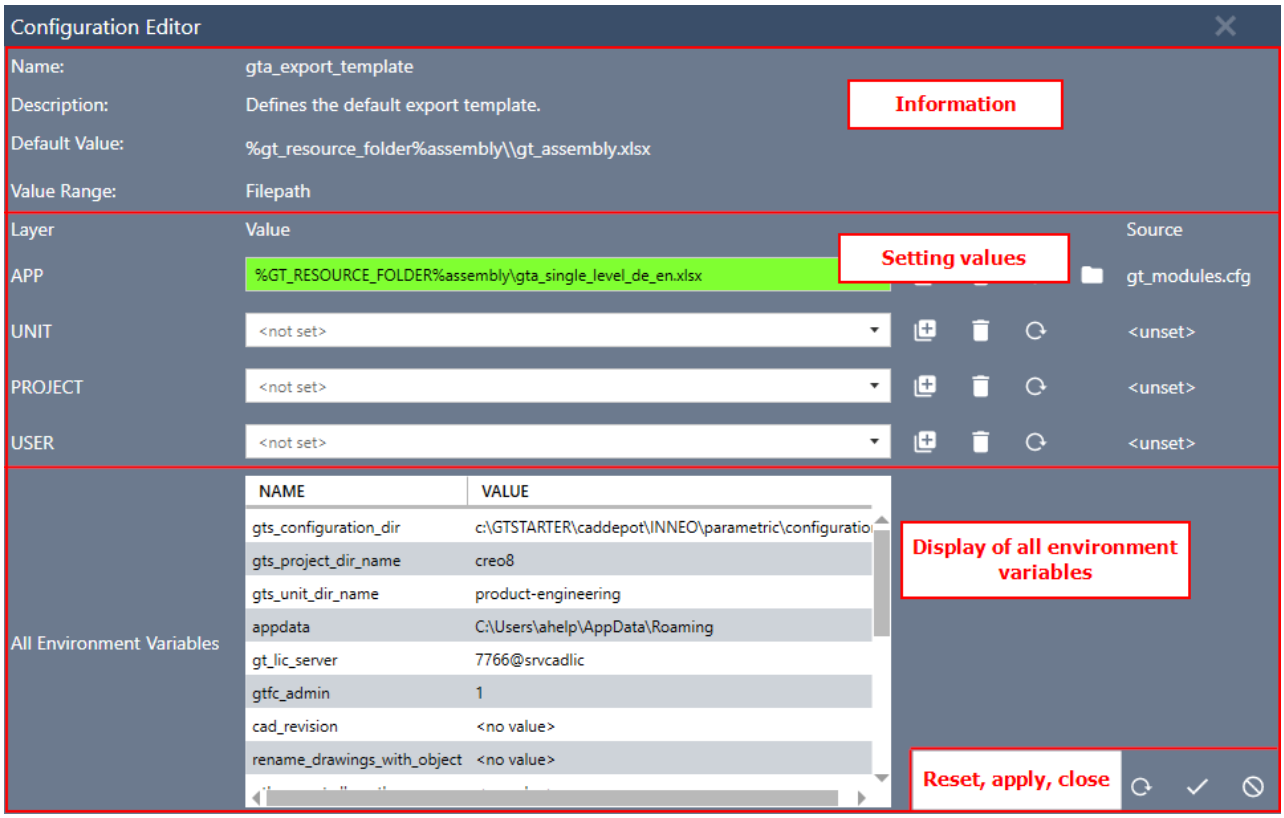
## 25.4.5 Editing configuration options

Double-clicking on a line in the configuration overview opens the configuration editor. Here, individual configuration options for each layer can be viewed, changed, commented and deleted.

All changes made here are later written to the corresponding configuration file during the saving process 775.

**Please note:** Changes must be confirmed with and then reloaded before they are valid, see saving configuration options 775.


© 2026 INNEO Solutions GmbH




*Configuration Editor dialog window*

**Information:** name, description, start value, value range of the selected configuration option.

**Set values:** values can be set, changed or deleted in the different layers.

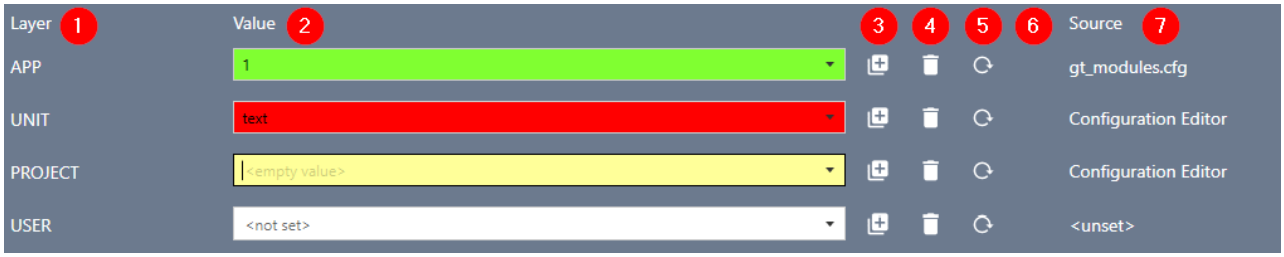
**All environment variables:** The list is for information and does not allow any changes at this point. Changes are made in the window *Environment Variables*[781], accessible via in the sidebar menu.

**Reset, Apply or Close:** Changes must be confirmed with and then reloaded before they are valid, see saving configuration options[775].

## Setting values

In this area, changes can be made to the configuration options on the four layers. (See also Viewing layers[780].)

As soon as values are set in the configuration editor, the configuration editor is displayed as source.





© 2026 INNEO Solutions GmbH

1. **Layer: App, Unit, Project and User**
   Configuration options can be set for each layer individually. The layers are arranged hierarchically in ascending order. A value set in a subsequent line overwrites the value set before it.

2. **Value**
   A writable field in which a new value can be entered. Valid values are highlighted in green, invalid values are highlighted in red. Make sure that you only save configuration options when all lines are highlighted in green or white.
   Existing values can be overwritten. You can also deposit values as comments[778].

| Color | Meaning | Explanation | Impact |
|---|---|---|---|
| green | valid value | The field was filled according to the value range or as environment variable[775]. | Saving without error |
| red | invalid value | The field was not filled in according to the value range. | Invalid values can be saved.<br>**Warning:** Check whether you really want to save red highlighted values. |
| yellow | in process | With the 1st click into a white field, it turns yellow. | As soon as you enter a value, the field turns green (or red in case of error). |
| | empty value | A field can be explicitly set as "empty", but this is not always useful, see note[774]. | Overwrites the start value and all previous layers.<br>**Warning:** Check if you really want to save empty values. |
| white | default | No value set. | Has no effect on other layers. |

All changed values are displayed again before saving the configuration, see Saving changes[775].


© 2026 INNEO Solutions GmbH

**Please note:** It may seem of advantage to set an empty value, e. g. so that the value of the previous layer is overwritten without setting an alternative. As a result, the configuration option is thus written to the corresponding configuration file, as opposed to being deleted. Check whether the affected application can and should process an <empty value> at the corresponding position.

3. **Add comment**
   Button to enter the current value as a comment in the corresponding CFG file during the next saving process, see writing comments[778].

4. **Delete app layer**
   Deletes the current value of the respective layer. Comments are retained.

5. **Reload app layer**
   Restores the original value of the configuration option. Comments are preserved.

6. **Open path**
   If the value range expects a path or folder name, a folder icon  is displayed as soon as an entry has been made on the corresponding layer. A click on the folder symbol opens the entered path.

7. **Source**
   Indicates from which source the current value of the respective layer was obtained. Usually this is a configuration file (*gt_main.cfg* or *gt_modules.cfg*) or the configuration editor, if the corresponding option has already been opened once at runtime of *Configuration Utility*.

**Please note:** If a path is entered that is located in the *gt_resource_folder* directory, the *%gt_resource_folder%* variable is automatically inserted.

## Overwriting layers

By means of the hierarchically structured layers (see configuration concept[756]), you can define values that apply to different areas of your company. The last value set overwrites the layers described before it. To illustrate the principle of operation, here are some examples:

- **Language settings**, e. g. with the configuration option *gt_lang*
  - APP: The corporate language is English. On the layer APP English is entered as language.
  - UNIT: At the location Germany the company language is German, therefore the layer UNIT is additionally defined there with the language German.
  - USER: Within this location, there is a colleague whose native language is Italian and who would therefore like to operate Creo Parametric in Italian. This colleague therefore defines the language Italian for himself at the USER layer.
- **Location libraries,** e. g. with the configuration option *gtl_favorite_button1_liblink*


© 2026 INNEO Solutions GmbH

  - APP: The corporate library is stored on this layer.
  - UNIT: Depending on the subject area that the respective location deals with, there are separate location libraries that are preferred for working with. The corporate library is stored at the second position.
- **Individual working method**, e. g. with the configuration options *gtqa_tooltip_time* and *gtqa_zoom*
  - USER: Zoom of Quick Access / speed of tooltip display adjustable to suit individual working methods.

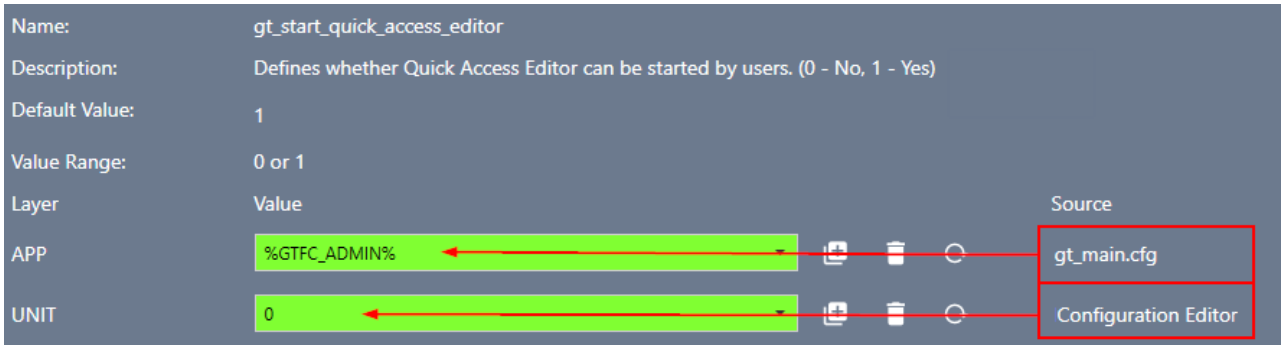
## Using environment variables

Alternatively to an input that corresponds to the value range, you can set environment variables so that the validity of the respective configuration option depends on this variable. You maintain the environment variables without having to explicitly customize the configuration options.

The value of a configuration option was set at the app level by the %GTFC_ADMIN% environment variable and stored in the *gt_main.cfg* source file because it is a common configuration option with the gt_ start abbreviation.

The configuration editor assumes with an even number of percentage characters that it is one or more matching environment variables, then evaluates the entry as correct and stores it in green.

On Unit layer the value was set to 0 with the `configuration editor`.




## 25.4.6 Saving changes

All value changes are saved in three steps.

1. Apply changes in the configuration editor
2. Save changes on the start page
3. Confirm changes in the save dialog

This sets the changes into the configuration files of the App, Unit, Project and User layers.


© 2026 INNEO Solutions GmbH

**Please note:** The set values should be saved globally. Make sure that you open Configuration Utility on the installation computer or that you are in Caddepot.

# 1. Applying changes in the configuration editor

In the lower right area of the configuration editor are three buttons for exiting. These buttons only affect the values. Comments cannot be deleted or restored with these commands. To edit comments, see writing comments 778.

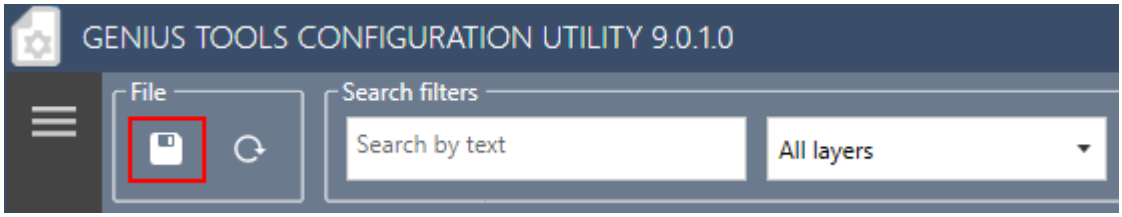



*Button for resetting, applying or closing in the configuration editor*

1. **Reload all (Reset)**
   Resets all values to the previously set values.
   The configuration editor is not closed.
2. **Save and close (Apply)**
   Changed values are written into the configuration overview on the start page and kept for the saving process.
   The configuration editor is closed.
3. **Close without saving**
   Close window and do not apply changed values.
   Another window opens where you need to confirm quitting without saving or you can cancel.
   Comments will still be retained.
   The configuration editor is closed.

# 2. Saving changes on the start page

Click on the Save icon on the start page to save any changes you have made. The save dialog opens.




*Button for saving the previous configurations on the start page of the user interface*

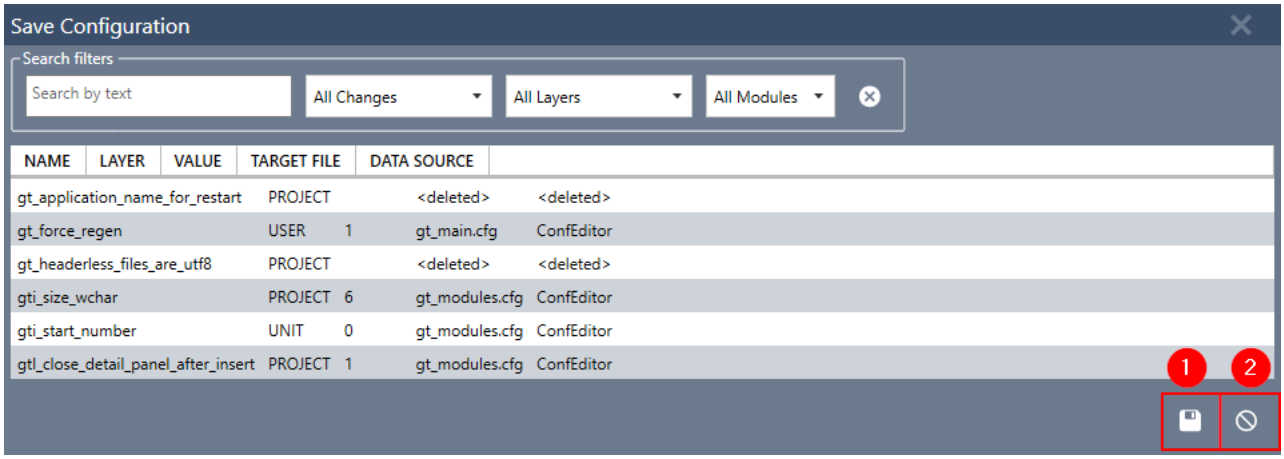
# 3. Confirming changes in the save dialog

The dialog box *Save Configuration* gives you an overview of all the changes you have made. Here you can decide whether you want to finally confirm these changes.


© 2026 INNEO Solutions GmbH

Changes are shown according to layer and value. The data source indicates whether a configuration has been deleted (<*deleted*>), modified (*ConfEditor*) or comes from a specified source (*e. g. gt_main.cfg*).

Configuration options where comments have been modified are also listed without naming the comments. Similar to the configuration overview on the start page, the listing can also be filtered in the save dialog, see Search filter 769.




*Overview of changed configuration options before saving*

If you close the Save dialog without saving (2), the changes will still be kept.

To save the changes, click the file icon (1).

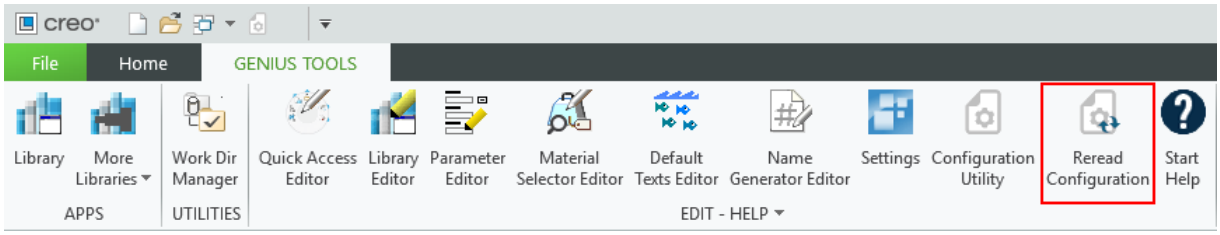
**Result:** The changed values and stored comments are saved in the configuration files of the corresponding layers. General settings are saved in *gt_main.cfg* files and module specific settings are saved in *gt_modules.cfg* files.

With each saving process, the old file is saved as a previous version in a ZIP file. If available, previous versions of the configurations can be viewed via the files stored there, so that any incorrect configurations can be undone at any time.

The changes are automatically applied the next time the Creo program is started. Alternatively, you can apply the changes directly 777.

## 25.4.7 Directly applying changes

If you want to test the effects of the changes made without restarting the program, click on the action *Reread Configuration* in standby mode. This command is only available in standby mode so that all applications are closed for the reconfiguration.



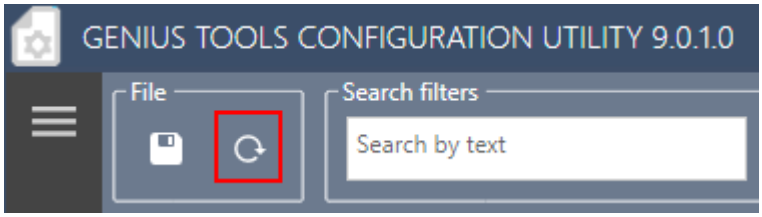

*GENIUS TOOLS ribbon menu in standby mode*


© 2026 INNEO Solutions GmbH

If you have renamed the auxiliary application *GENIUS TOOLS for Creo* in the file *creotk.dat (or: protk_gtfc.dat)*, you must change the *gt_application_name_for_restart* configuration option to this name. This is necessary to enable the action *Reread Configuration*.

## 25.4.8 Reseting configuration options

To restore the previous configuration, click the button *Reload present configuration* on the start page. This resets all changes made in the configuration editor. The starting point is the last saved configuration.




Previously saved configurations cannot be undone with this command. To view previously saved configurations, you must load the saved ZIP file, see Configuration files[780] .

**Please note**: Changes to the environment variables are not reset during this process. These must be reset manually in the window *Environment variables*[781] . This ensures that any path changes to environment variables can be tested effectively.

## 25.4.9 Using comments

By generating comments, you can use variants in the graphical interface and thus quickly test different configuration variants without manually entering the various settings.

Comments

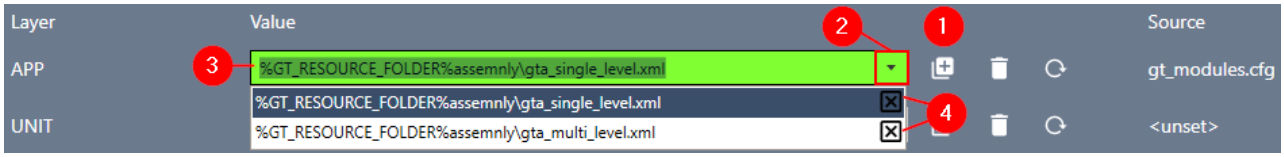
– are commented values in a configuration file,

– have no influence on the GENIUS TOOLS for Creo modules,

– are saved when they are created,

– are separated from the storage process of the values and are available at any time, i. e. they are not deleted by the buttons and in the configuration editor.

You need the following elements in the configuration editor[773] to use comments.




*Set values area in the configuration editor*

1. The button *Add comment* .

2. The drop-down menu behind the text field of the respective layer.
3. The text field of the respective layer in the column *Value*.
4. The *Delete comment* button at the right edge of the individual lines of the respective drop-down menu.

## Adding comments

A comment is created by a new entry in the text field. This overwrites the current value and thus it is lost. To retain the current value, it is advisable to copy it to the comment field beforehand so that it can be selected again afterwards.

**Warning**: If a comment is entered in the text field, the current value is overwritten and thereby lost.

**Procedure if current value is to be retained:**

1. Click on the *Add comment* button (1). The drop-down menu (2) opens automatically. This shows you that the existing entry has been added as a comment.
2. Enter a new entry in the text field of the *Value* column (3).
3. Click the *Add Comment* button (1) to add the new entry as a comment.
4. Repeat the steps for further comments.

If the original value is to be set again, the *Reload app layer* button can be operated.

## Selecting comments

By selecting a comment in the drop-down menu of a layer, the content of the corresponding line can be set as a new value with one click. *Configuration Editor* is then automatically set as the source.

An exception arises, if the new value at the start of Configuration Utility on this layer was already set before by another configuration file and later overwritten by a value from another source. Then the source of the corresponding predecessor is selected.

## Deleting comments

Expand the list of comments in the drop-down menu and click the *Delete Comment* button .

If this is also the current entry, it is automatically removed as well, i. e. the entry is considered an *empty value*.

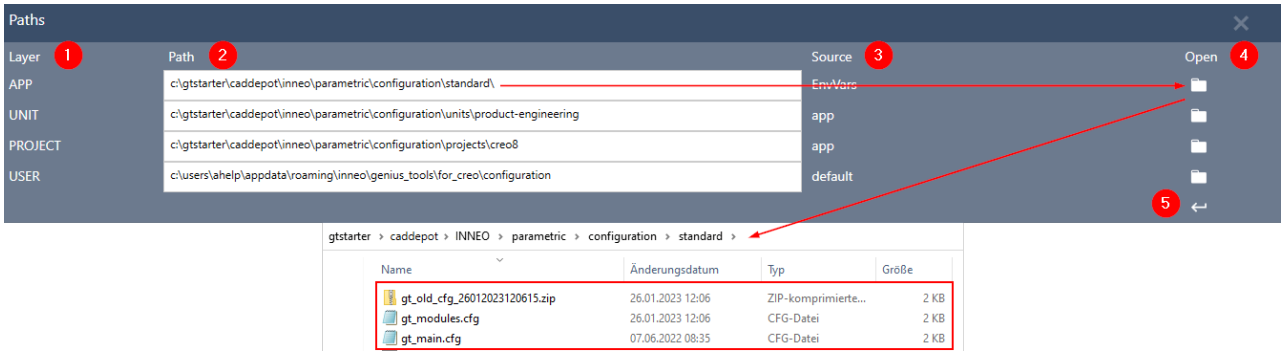
Resetting the configuration 778 in the start page also resets the comments to the last saved state.

**Please note**: Newly created comments are not deleted by closing the configuration editor without saving.


© 2026 INNEO Solutions GmbH

## 25.4.10 Viewing configuration files and layers

The window *Paths* opens with the operation of the button on the sidebar of the start page. It shows the complete paths (2) to the individual layers (1) and allows quick viewing and editing of the configuration files (4), i. e. the general configuration file *gt_main.cfg* and the module file *gt_modules.cfg* on the respective layer.




*Open the current path for the App layer*

1. **Layer:** For information about the layers *App, Unit, Project* and *User*, see configuration and start[755].
2. **Path:** Path specifications cannot be modified at this point.
   The path specifications for the layers *Unit, Project* and *User* can be overwritten in the configuration editor[771] with the configuration options *gt_conf_unit, gt_conf_project* and *gt_conf_user*.
   The path specification for the App layer is determined from the *%GT_CONF_DIR%* environment variable, see configuration concept[756].
3. **Source:** Displays the layer in which the path specification was set. This corresponds to the green entries in the first column of the configuration overview[768].
   We recommend that, as in the above example, both the unit directory specification (defines the Unit layer) and the project directory specification (defines the Project layer) are defined once in the APP layer, i. e. in the *gt_main.cfg* file in the folder defined under APP.

---

**Please note:** If you are working with Startup TOOLS, the App layer should point to the default GENIUS TOOLS Starter folder, see Configuration and start > Working with Startup TOOLS[760].

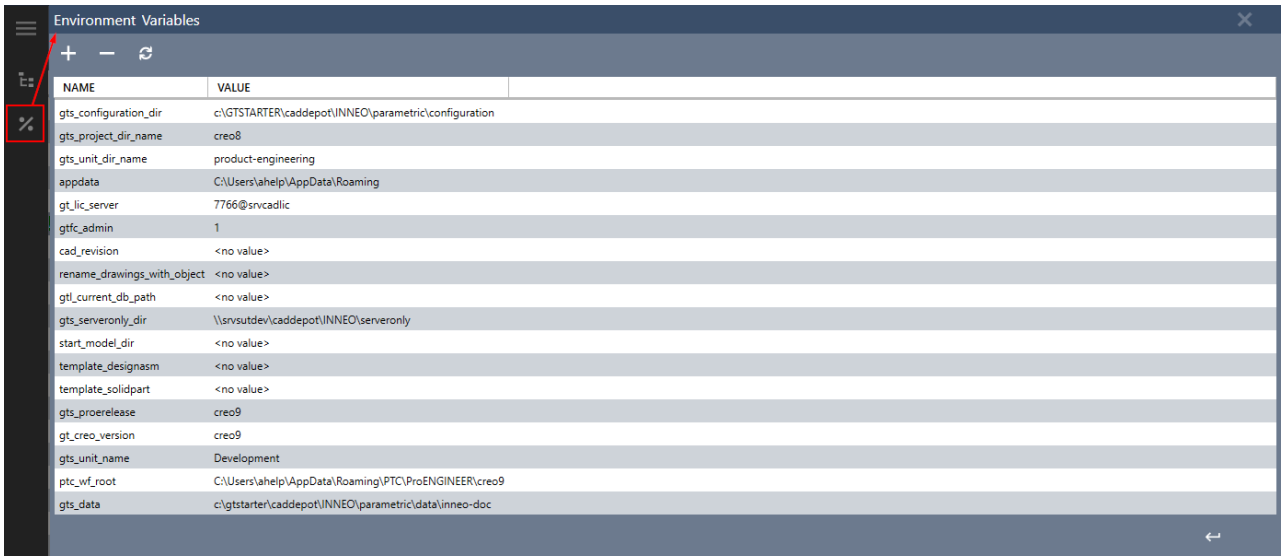
---

4. **Open:** The button opens the folder containing the files *gt_main.cfg* and *gt_modules.cfg* as well as the packed folder *gt_old_cfg_%date%.zip*. after saving.
   The ZIP file contains the previous configuration files of this layer, so that any misconfigurations can be undone at any time.
5. **Button :** Closes the window.


© 2026 INNEO Solutions GmbH

## 25.4.11 Environment variables

The window *Environment Variables* opens with the Percent button on the sidebar of the start page.




Under *Name* all environment variables are listed which are noted as value in a configuration option. Any entry enclosed by percent signs is interpreted as an environment variable.

Environment variables can neither be created nor deleted in *Configuration Utility*.

You can change the values of environment variables to affect paths, for example, which can affect the structure of configuration files and options when you save them. Changed values are applied without a separate save. The changes become valid as soon as the button is operated after closing the menu item Environment variables in the start display.

**Tip:** For test purposes, you can add or delete environment variables using the and and buttons. Environment variables are simulated with these.

## 25.4.12 Use cases

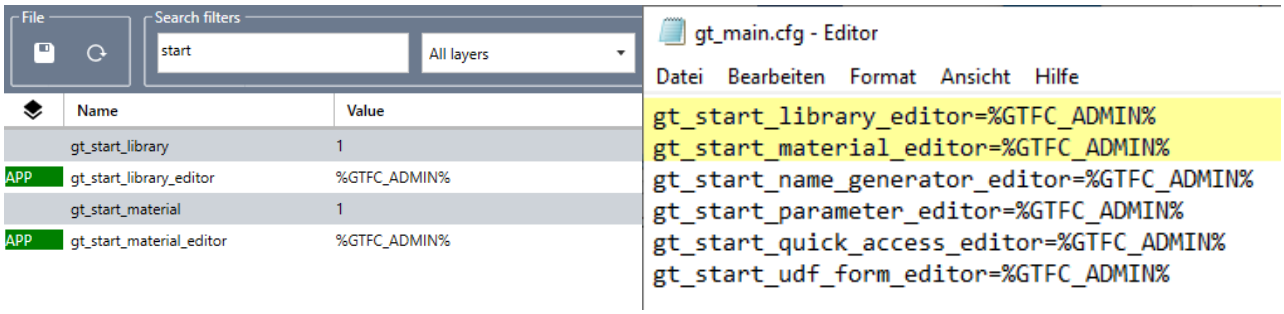
### 25.4.12.1 Set access to programm editors

Access to the editors of the individual modules is controlled by the corresponding start switches. By default, the value is set to "1".

The variable *%GTFC_ADMIN%* is available to set the value depending on the user's authorization.

Example: The settings for *Library* and *Material* are set to %GTFC_ADMIN% at APP level (1) and thus written to the *gt_main.cfg* file located in the *standard* folder (2).


© 2026 INNEO Solutions GmbH




If you work with Startup TOOLS, you can set the variable *%GTFC_ADMIN%* in the *Project Configurator* in the menu item *Resources* > *Roles* > *Tab: Function Access* > *Administration: Is GTfC-Admin*. This sets the variable in the configuration options ("start button") of the following Editors: Library Editor, Material Editor, Name Generator Editor, Parameter Editor, Quick Access Editor, UDF Forms Editor.

# 25.5 Database Version Control

Database Version Control is an administration tool for updating the databases used by GENIUS TOOLS for Creo to the current version that fits the software version in use.

For information on how to use Database Version Control, please refer to Usage[782].

For information on the database versions required by the current GENIUS TOOLS for Creo version, please refer to Database and product versions[783].

## Starting the program

Open Database Version Control from the user interface of the GENIUS TOOLS for Creo Configuration Utility by clicking the icon under *Apps*.

## 25.5.1 Usage

Database Version Control checks all SQLite databases in a directory you select and determines whether they are up-to-date for use with GENIUS TOOLS for Creo.

To update your GENIUS TOOLS for Creo databases, proceed as follows.

1. Under *Database folder path*, select your database directory.
   To refresh the list of databases after switching directories, click *Reload*.
2. Verify the following information in the list of databases:
   a. Path, name and type of each database
   b. Current version and new version. These two columns tell you whether the current version and the new version are identical or whether the database can be updated.


© 2026 INNEO Solutions GmbH

c. Update. This setting determines whether each database will be processed when you start the update function.
Databases for which an update is available are checked by default.

3. Set the *Update* setting for all databases that you want to update.

4. Start the update function by clicking *Update*. 

## 25.5.2 Database and product versions

GENIUS TOOLS for Creo uses SQLite databases whose version has to fit the version of the software that you have in use. The database versions are numbered independently of the product versions. Not every product version requires a database update.

The following table provides an overview of the database versions required by the current GENIUS TOOLS for Creo version.

| GENIUS TOOLS for Creo product version | library | name_generator | material | quick_access | stdTexts |
|---|---|---|---|---|---|
| 6.0.0.0-8.0.1.0 | 1.8 | 2.0 | 1.2 | 1.1 | 6.0.0.0 |
| 8.0.2.0+ | 1.9 | 2.0 | 1.2 | 1.1 | 6.0.0.0 |

# 25.6 Javascript

Available in the ribbon menu GENIUS TOOLS

*JavaScript* is supported in the GENIUS TOOLS components *Forms*[154], *UDF Forms*[619], *Parameter*[494] and *Quick Access*[555]. With the *JavaScript Editor*, you can develop and test JavaScript code for GENIUS TOOLS for Creo.

This allows you, for example, to check parameter and dimension values before transferring them to a model or to automate calculation and modification of such values.

For a practical example of how JavaScript functions can be used, see the application example Using *JavaScript* with *Parameter*[535].


© 2026 INNEO Solutions GmbH

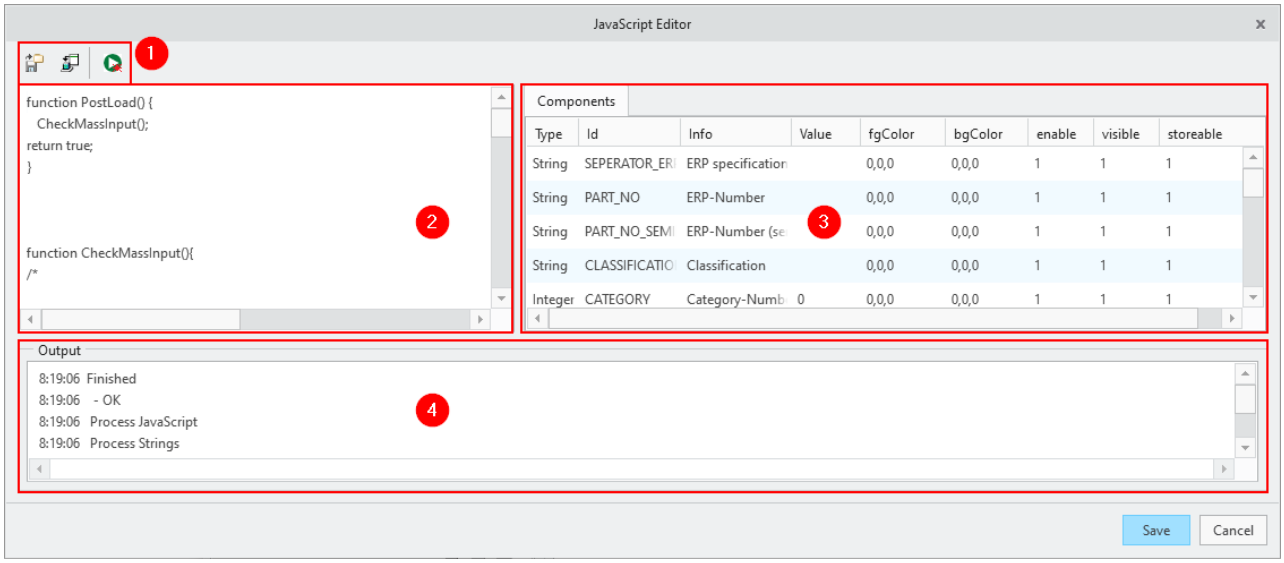
## 25.6.1 JavaScript Editor

Available in the ribbon menu GENIUS TOOLS

The Editor is always started from the respective GENIUS TOOLS for Creo module.

# User interface

The JavaScript Editor user interface consists of the following elements:




1. Command bar
2. Input area
3. Component view
4. Output

# 1. Command bar

The following buttons are included in the command bar:

| Icon | Name | Description |
|---|---|---|
| | Export | Exports the JavaScript as .js file. |
| | Import | Imports JavaScript from a .js file. |
| | Execute | Executes the JavaScript code with the given component values and displays the results in the Output dialog window (4). |

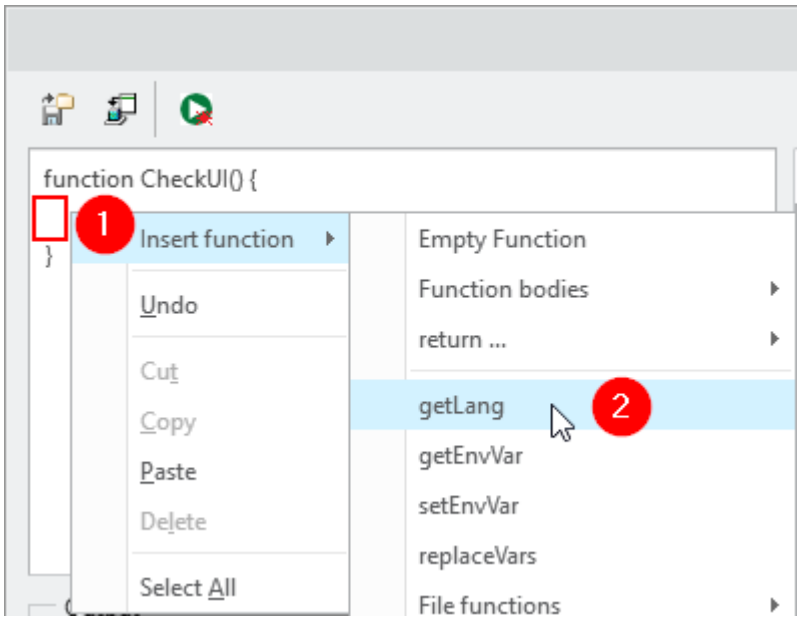

© 2026 INNEO Solutions GmbH

## 2. Input area

JavaScript code is edited in the input area.

Use the context menu to insert code snippets for usage in GENIUS TOOLS modules. Right click on the position at which you want to insert a code snippet (1) and select the snippet from the menu (2).

Press CTRL+S to save your changes and close the editor.




*Inserting a code snippet from the context menu*

Add all code snippets to your program code and export them as .js file. Then open the file in a source code editor to use syntax highlighting and auto complete. Then reimport your source code.

In addition to the implemented functions you have the possibility to insert a function template via *Insert function* > *Empty Function*, which you can assign with your own function definition.

## 3. Component view

The component view displays all components available in the current context with their current properties, for example parameters of the current parameter definition or variables of the current UDF definition. Use the arrow icon ▲ to sort components alphabetically or by their values.

Click into a table cell to edit the values manually.

If JavaScript is executed in the Editor, the modifications are displayed in the component view. Execution in the Editor does not affect the real values in the model.


© 2026 INNEO Solutions GmbH

**Properties overview**
**Type:** Specifies the current type of a component (variable type).

**ID:** The name of a component allowing to address it via JavaScript.

**Info:** The title of a component.

**Value:** The current value of the component.

**fgColor:** Displays the current foreground color of the components as RGB values.

**bgColor:** Displays the current background color of the components as RGB values.

**enable:** Displays the editability of a component (0 - No, 1 - Yes).

**visible:** Displays the visibility of a component (0 - No, 1 - Yes).

**storable:** Displays the storability of a component (0 - No, 1 - Yes).

## 4. Output

In the Command bar, click *Execute* to receive outputs for the current JavaScript.

## 25.6.2 List of Javascript funtions

The following table lists the supported JavaScript functions in the individual GENIUS TOOLS modules.

| Function | Forms | UDF Forms | Parameter | Quick Access |
|---|---|---|---|---|
| alert | X | X | X | X |
| CheckUI | X | X | | |
| confirm | X | X | X | X |
| creoCurrentMdlNameGet | X | X | X | X |
| creoDrwActiveMdlNameGet | X | X | X | X |
| creoListAllMdlsInSession | X | X | X | X |
| creoMapkeyAddToStack | X | X | X | X |
| creoMdlExtensionGet | X | X | X | X |


© 2026 INNEO Solutions GmbH

| Function | Forms | UDF Forms | Parameter | Quick Access |
|---|---|---|---|---|
| creoMdlMassGet | X | X | X | X |
| creoMdlNameGet | X | X | X | X |
| creoMdlOIDGet | X | X | X | X |
| creoParameterValueGet | X | X | X | X |
| creoParameterValueSet | X | X | X | X |
| creoSelectAsmComponentGet | X | X | X | X |
| Empty Function | X | X | X | X |
| excelOpen | X | X | X | X |
| excelCellValueGet | X | X | X | X |
| getEnable | X | X | X | |
| getEnvVar | X | X | X | X |
| getInputValue | X | X | X | |
| getLang | X | X | X | X |
| getStoreable | X | | X | |
| getStringUI | X | X | X | X |
| getVisible | X | X | X | |
| openFileUI | X | X | X | X |
| PostLoad | X | X | X | |
| PostLoadFromFile | X | X | | |
| PostSave | X | X | X | |
| PreSave | X | X | X | |


© 2026 INNEO Solutions GmbH

| Function | Forms | UDF Forms | Parameter | Quick Access |
|---|---|---|---|---|
| print | X | X | X | X |
| printError | X | X | X | X |
| printWarning | X | X | X | X |
| readFile | X | X | X | X |
| replaceVars | X | X | X | X |
| return | X | X | X | X |
| runHttpRequest | X | X | X | X |
| saveFileUI | X | X | X | X |
| setBgColorValue | X | X | X | |
| setEnable | X | X | X | |
| setEnvVar | X | X | X | X |
| setFgColorValue | | | X | |
| setInputValue | X | X | X | |
| setStoreable | | | X | |
| setVisible | X | X | X | |
| writeFile | X | X | X | X |

## 25.6.3 Explanations and examples

| JavaScript function | Example |
|---|---|
| **alert** | |
| Outputs an alert box/dialog box with text. It has to be confirmed with “OK”. | `alert("Please check your input!")` |
| **CheckUI** | |
| Will be executed on clicking the associated button of the current GENIUS TOOLS component. | `function CheckUI() { if(getInputValue("InternalDiameter") >= getInputValue("Diameter")) { return false;` |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
| --- | --- |
| | }<br>return true;<br>} |
| **confirm** | |
| Outputs a dialog box with text. It has to be confirmed with Yes or No.<br>Returns the result of the query dialog (Yes=true/No=false). | var retConfirm = confirm("Do you want to continue?"); |
| **creoCurrentMdlNameGet** | |
| Reads the name of the current Creo model. | |
| **creoDrwActiveMdlNameGet** | |
| Returns the active model of a drawing. | |
| **creoListAllMdlsInSession** | |
| Returns the names of all models in a session. | function test() {<br>array = creoListAllMdlsInSession();<br>for (i = 0; i < array.length; i++)<br>print (array[i]);<br>} |
| **creoMapkeyAddToStack** | |
| Loads a mapkey from JavaScript and executes it.<br>Thus mapkeys can be controlled depending on a value.<br>Note: Use only as PostSave function, because mapkeys usually close windows. | function PostSave() {<br>zd = getInputValue("ZYL_DIAMETER");<br>if (zd==111) {<br>creoMapkeyAddToStack("%cd;");<br>}<br>return true;<br>} |
| **creoMdlExtensionGet** | |
| Reads the file name extension of the current model of the GENIUS TOOLS module. | |
| **creoMdlMassGet** | |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
| --- | --- |
| Reads the mass of the current model. | |
| **creoMdlNameGet** | |
| Reads the name of the active model from the context of the active GENIUS TOOLS module. | |
| **creoMdlOIDGet("ModelName")** | |
| Reads the object ID (OID) of a model whose name (including the file name extension) is passed to the function.<br>For example, the OID of a model can be used in the function **runHttpRequest** in order to access specific objects in a PDM system. | `creoMdlOIDGet("my.prt")`<br>`Usage in an HTTP request:`<br>`var mdl = creoMdlNameGet();`<br>`var oid = creoMdlOIDGet(mdl);`<br>`var request = runHttpRequest("http://pdm/Windchill/servlet/rest/objects/"+ oid + "?%24select=number", "", "1");`<br>`var obj = JSON.parse(request);` |
| **creoParameterValueGet(mdl,"Parameter Name" t/f);** | |
| Reads the value of a parameter of a model that has to be specified.<br>Use t or f (true and false) to determine whether floating point values should be rounded according to the Creo configuration option PARAM_DEC_PLACES (default: 6 digits) (t). If the values are not rounded (f), a maximum of 17 digits (total) will be output . | `creoParameterValueGet(mdl,"NAME" f);` |
| **creoParameterValueGet(ParameterName);** | |
| Reads the value of a parameter of the current model. | `var CurMod = creoParameterValueGet("DESCRIPTION_1_EN");`<br>`alert('Name of the current model: ' + CurMod + '');` |
| **creoParameterValueSet(mdl, ParameterName, Value);** | |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
|---|---|
| Writes a value into a parameter of the current model. The return value is the error code.<br>If the parameter is not in a family table and the function is applied to an instance, the parameter value is written into the generic model. | `var ret = creoParameterValueSet("PART_NO", "Value");`<br>`var ret = creoParameterValueSet(mdl,"PART_NO", "Value");` |
| **creoSelectAsmComponentGet** | |
| Outputs the filename of an assembly component that has to be selected manually in the Creo window. | |
| **Empty Function** | |
| Creates a function template for defining your own functions. `fname` can be selected and replaced by double-clicking. | `function fname() {`<br>`};` |
| **excelOpen** | |
| Opens an Excel spreadsheet specifying an absolute path and allows access to the individual cells from this Excel spreadsheet. | `excelOpen("D:\\gstarter\\caddepot\ \user\\parametric\\data\\user\ \library_dir\\test_file.xlsx")` |
| **excelCellValueGet** | |
| From the Excel spreadsheet opened with **excelOpen**, this function gets the value from an Excel cell specifying sheet name (optional), column and row. | `excelCellValueGet(excel, "1", "C", "3")` |
| **getEnable** | |
| Reads out if a component can be edited. | |
| **getEnvVar** | |
| Reads the value of a Windows environment variable. | `var login = getEnvVar("USERNAME");` |
| **getInputValue** | |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
|---|---|
| Outputs the entered value of a component (e. g. parameter, variable). | `var Offset = getInputValue("z");` |
| **getLang** | |
| Returns the interface language of the current GENIUS TOOLS module. | `getLang();` |
| **getStoreable** | |
| Returns the storability of a component (*Save* option of a parameter). | |
| **getStringUI("request", "default_value" (optional))** | |
| Displays a field for free text entry. Multiple entries are separated with \|. In that case, a drop-down menu will be displayed. | `var text = getStringUI("Screw size","M1.1x.25\|M1.2x.25\|M1.4X.3");` |
| **getVisible** | |
| Reads the visibility of a component. | |
| **openFileUI("*.extension", path (optional), title (optional), preselected_file (optional))** | |
| Opens a window for file selection specifying the file format and file name. | `var filename = openFileUI("*.xlsx", "","Select EXCEL calculation file","");` |
| **PostLoad** | |
| Will be automatically executed after loading the current definition ((UDF) Forms definition). | `function PostLoad() {`<br>`...`<br>`}` |
| **PostLoadFromFile** | |
| Will be automatically executed after loading a value table (CSV/XML-file).<br>Executed after the function **PostLoad**, if it is defined. | `function PostLoadFromFile() {`<br>`...`<br>`}` |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
|---|---|
| **PostSave** | |
| Will automatically be executed in a GENIUS TOOLS component after saving. | `function PostSave() {`<br>`...`<br>`}` |
| **PreSave** | |
| Will automatically be executed in a GENIUS TOOLS component before saving. | `function PreSave() {`<br>`...`<br>`}` |
| **print** | |
| Outputs a status message (green) in the status color code and in the Creo message log. | `print("All changes were saved in the model.")` |
| **printError** | |
| Outputs a error message (red) in the status color code and in the Creo message log. | `printError("Values could not be written into the model!")` |
| **printWarning** | |
| Outputs a warning message (yellow) in the status color code and in the Creo message log. | `printWarning('There are ' + ErrorCount + 'errors in the calculation!')` |
| **readFile** | |
| Reads the contents of a file into a JSON object.<br>var vals = readFile(filename, codex, obj) with:<br>– vals: JSON object of the file<br>– filename: the complete path of the file<br>– codex: the coding (0 - std ASCII, 1: UTF8, 2 UTF16 BE)<br>– obj: an error object for better further processing (obj.code: Error code (0: no error) and obj.text: error text) | `function PostLoad() {`<br>`var vals = readFile("c:\\temp\\test.txt", 0);`<br>`var obj = JSON.parse(vals);`<br>`if (obj.length > 0) {`<br>`setInputValue("DESCRIPTION_1_EN", obj[0]);`<br>`}`<br>`}` |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
| --- | --- |
| can be NULL. | |
| **replaceVars** | |
| The transferred data is replaced with a GENIUS TOOLS replacement function. The transferred values are searched for in the following hierarchical order:<br>1. Creo configuration options (config.pro options)<br>2. Operating system variables<br>3. GENIUS TOOLS for Creo configuration options[799]<br>4. GENIUS TOOLS variables[913], e. g. user entries[913]<br>5. Name Generator[440] | `var t = replaceVars("@mdlpath@ %TBXCONFIG%");` |
| **return** | |
| Returns a Boolean value. Is required for **CheckUI**, **PostLoad** and **PreSave** to return whether the function has been successfully completed. | `return true;`<br>`return false;` |
| **runHttpRequest("URL", "DATA", 1/0, "HeaderOption")** | |
| Executes an HTTP request. GET parameters are passed via the URL: `URL?GET_PARAMETER`. POST parameters are passed via `query`. The function accepts the URL and URL parameters (DATA). An additional parameter USERNAME/PASSWORD takes the value 1 or 0 and defines whether the user should be asked for login information or not. The last entry can be an HTTP header option for further specification of the request. | `var site = "http://ServerName/Folder/db_list_gt.php";`<br>`var query = "DB=pmm_tc.mdb&TABLE=namen&SFIELD=german&FIELDS=german,english&SEARCH=ab*";`<br>`var ret = runHttpRequest(site, query, 0, "Accept-Language: en");` |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
|---|---|
| **saveFileUI("*.extension", path (optional), title (optional), preselected_file (optional))** | |
| Returns the file name. | `saveFileUI("*.extension", "", "OK", "");` |
| **setBgColorValue** | |
| Sets the background color for a GENIUS TOOLS component. Color specification in RGB values.<br>If you do not pass RGB values, but only the component name, the color will be reset to the default row color (white or gray). | `if (x > 1){`<br>`setBgColorValue("z",0,234,50);`<br>`}` |
| **setEnable** | |
| Sets whether a component is editable. | |
| **setEnvVar** | |
| Writes the value in a Windows environment variable. | `setEnvVar("PROJECT_NO", "8004");` |
| **setFgColorValue** | |
| Sets the foreground color of an input field in GENIUS TOOLS Parameter. Color specification in RGB values.<br>If you do not pass RGB values, but only the component name, the color will be reset to the default foreground color. | `setFgColorValue("z",255,0,0);` |
| **setInputValue** | |
| Sets a component (e. g. parameter, variable) to a value. | `setInputValue("z", 30);` |
| **setStoreable** | |
| Sets the storability of a component (*Save* option of a parameter). | |


© 2026 INNEO Solutions GmbH

| JavaScript function | Example |
|---|---|
| **setVisible** | |
| Sets the visibility of a component, for example a separator. | |
| **writeFile** | |
| Writes a JSON object to a file.<br>var ret = writeFile(filename, jData, codex, obj)<br>With:<br>– ret: "ok" if no error occurred<br>– filename: the complete path of the file<br>– jData: Json object string<br>– codex: the coding (0 - std ASCII, 1: UTF8, 2 UTF16 BE)<br>– obj: an error object for better further processing (obj.code: Error code (0: no error) and obj.text: error text) can be NULL. | `function PostLoad() {`<br>`var obj = {};`<br>`var line0= getInputValue(`<br>`"DESCRIPTION_1_EN");`<br>`var data = [];`<br>`data[0] = line0;`<br>`var jData = JSON.stringify(data);`<br>`var ret =writeFile("c:\\temp\\test.txt", jData, 0);`<br>`}` |

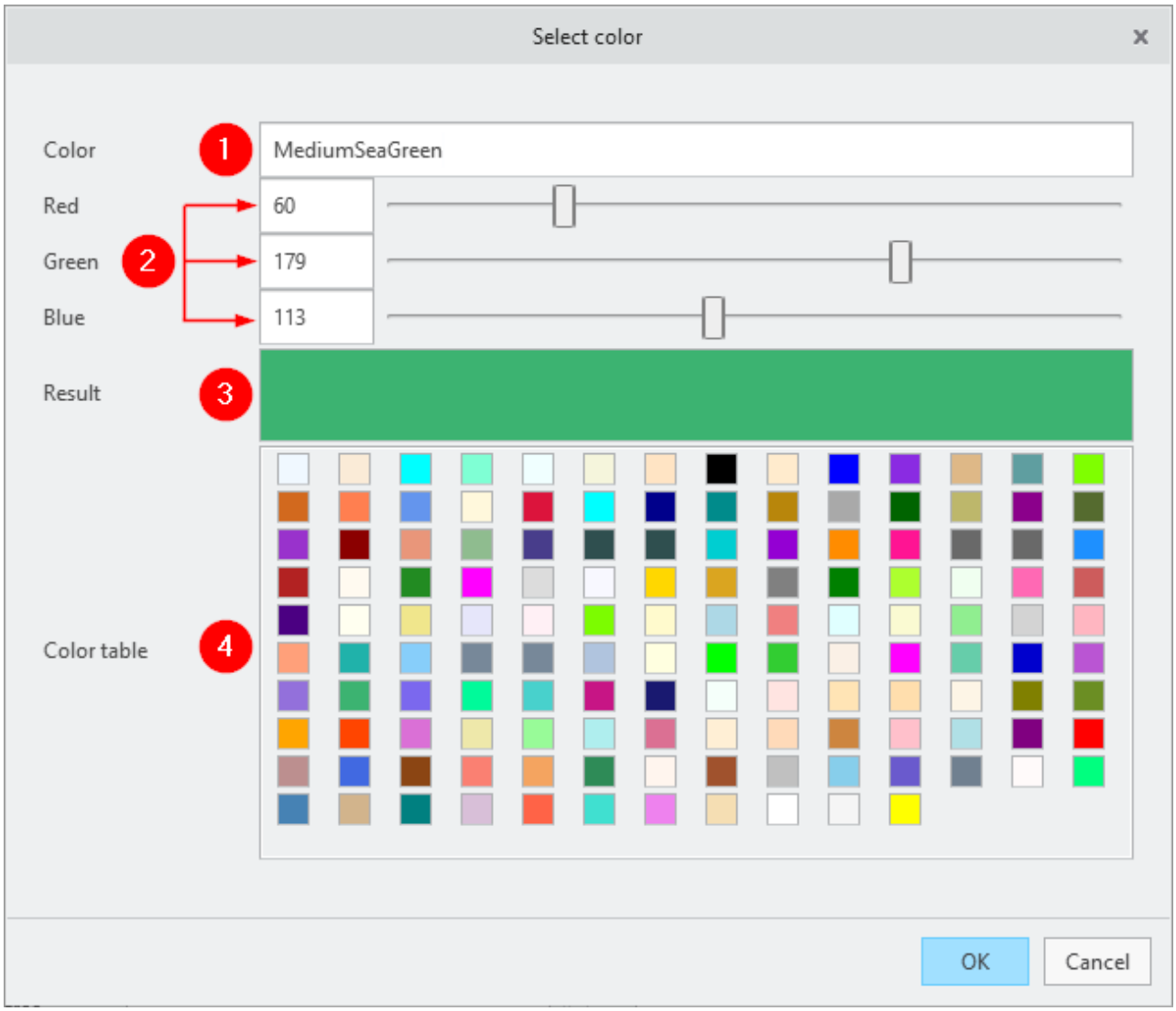
## 25.7 Color selector

The color selection dialog is used for the simple determination of background colors in the different *GENIUS TOOLS* modules.

Predefined HTML standard colors as well as own colors can be used.

The color selection dialog is always called by an editor of one of the *GENIUS TOOLS* modules.


© 2026 INNEO Solutions GmbH




*Farbauswahldialog*

| | | |
|---|---|---|
| 1 | Color | When the dialog is started, the color value of the calling GENIUS TOOLS module is written into this field.<br>The following are supported:<br>– predefined color names (e.g. Red)<br>– comma-separated RGB values (e.g. 100,255,16)<br>– colors in hexadecimal notation (e.g. #ffff10)<br>When the dialog is loaded, the transferred color is analyzed and displayed.<br>Changes in this field do not update the color display in the dialog.<br>Confirming the dialog transfers the value in this field to the calling editor. |
| 2 | Red | Determination of the color via RGB. |


© 2026 INNEO Solutions GmbH

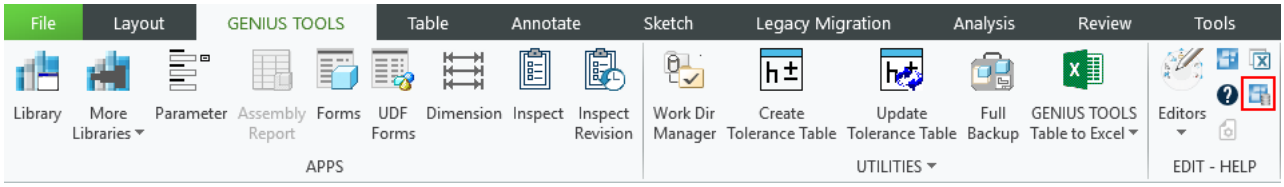
| | | |
|---|---|---|
| | Green<br>Blue | In the input field, the individual color components in the range [0,255] can be defined.<br>The sliders also control the individual color components.<br>These changes are immediately visible in the surface. |
| 3 | Result | This field displays the resulting color. |
| 4 | Color table | Table with predefined colors that can be selected with a mouse click. |

# 25.8 Display Embedded Data

Embedded Data are information on a model that have been generated by GENIUS TOOLS components *Assembly Report*, *Forms*, *UDF Forms* and *Inspect Revision*. This data can be displayed and deleted in the current model.

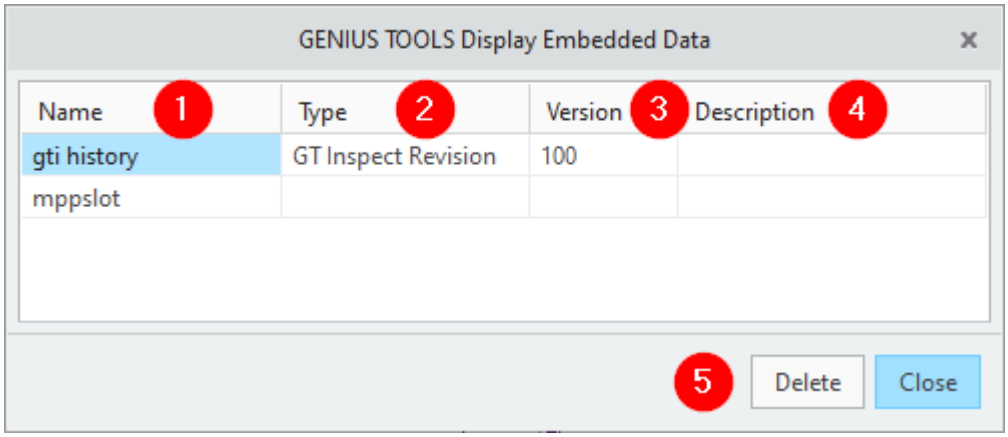
## Starting the program: in part, assembly and drawing mode

The button  for the function is located in the segment *EDIT-HELP* in the *GENIUS TOOLS* ribbon menu.




## 25.8.1 User interface

The user interface of the *Display Embedded Data* dialog has four columns.




*Overview of embedded data*

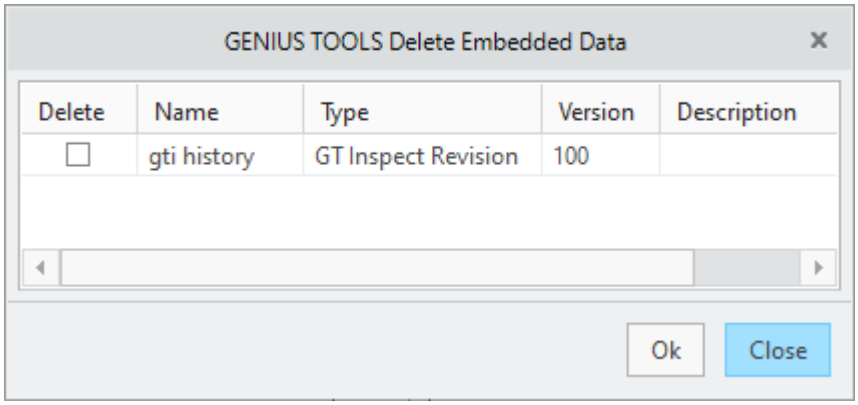
1. File name (e. g. XML file)


© 2026 INNEO Solutions GmbH

2. Data type: specifies whether the data comes from one of the following GENIUS TOOLS for Creo components:
   - Assembly Report
   - Inspect Revision
   - Forms
   - UDF Forms
3. Version (if assigned)
4. Description text
5. Button to open the *Delete* dialog.

## Delete dialog

In the first dialog, select a line for a GENIUS TOOLS for Creo-module by highlighting it and click *Delete (5)*. This opens the *Delete* dialog containing the selected line:




*Delete embedded data*

Tick the checkbox and confirm by pressing the *Ok* button.

# 25.9 Configuration Options

This is an overview of configuration options in the GENIUS TOOLS for Creo modules

This overview is created by an automated process.

**Please note**: Do not use environment variables such as `GT_RESOURCE_FOLDER` or `GT_CONF_USER` in Creo configuration files! These variables are only available after the start of the GENIUS TOOLS for Creo.

## General Configuration Options

In the configuration file *gt_main.cfg* the settings for GENIUS TOOLS for Creo are made.


© 2026 INNEO Solutions GmbH

***gt_application_name_for_restart***

**Options** Any text

**Default** GENIUS TOOLS for Creo

Specifies the name of the application as in creotk.dat.

***gt_auto_resolve_with_continue***

**Options** 0 or 1

**Default** 0

Defines whether conflicts should be automatically resolved when checking modifiability. The following configurations are set in this case:dm_auto_conflict_resolution = yesdm_enable_onfly_checkout = yesdm_checkout_on_the_fly = continue

***gt_conf_project***

**Options** Path or foldername

**Default** default

Defines the name of a directory (below GT_RESOURCE_FOLDER) or a complete path. Is read after "UNIT".

***gt_conf_unit***

**Options** Path or foldername

**Default** default

Defines the name of a directory (below GT_RESOURCE_FOLDER) or a complete path. Is read first.

***gt_conf_user***

**Options** Path or foldername

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\configuration

Defines the name of a directory (below GT_RESOURCE_FOLDER) or a complete path. Is read after "UNIT" and "PROJECT".


© 2026 INNEO Solutions GmbH

***gt_default_texts_db***

**Options** Path or foldername

**Default** %gt_resource_folder%data\gt_default_texts.sqlite

Defines the sqlite-database for default texts.

***gt_default_texts_table***

**Options** Table name

**Default** selection

Defines the table of the sqlite-database for default texts.

***gt_dialog_maximum_resize_factor***

**Options** Number between 0 and 1

**Default** 0.5

Defines the maximum window size with automatic size-adjustments regarding to the desktop height.

***gt_force_regen***

**Options** 0 or 1

**Default** 0

Should regeneration in GENIUS TOOLS for Creo be forced. (0 - No, 1 - Yes)

***gt_headerless_files_are_utf8***

**Options** 0 or 1

**Default** 1

Defines the reading format of headerless files (0 = ASCII, 1 = UTF-8) This configuration option is used for files, opened without formatting.

***gt_image_height***

**Options** Any number to up to 1200

**Default** 800

Defines the maximum height of images in GENIUS TOOLS Forms, UDF Forms model info and Library tooltip / detail images.


© 2026 INNEO Solutions GmbH

### *gt_image_width*

**Options** Any number to up to 1200

**Default** 800

Defines the maximum width of images in GENIUS TOOLS Forms, UDF Forms model info and Library tooltip / detail images.

### *gt_lang*

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines the language for the welcome screen, the About dialog and the displayed help. Additionally, it defines the displayed language in the External Model Data Viewer.

### *gt_licpath*

**Options** Port@IP

**Default** 7766@localhost

Defines the license server for GENIUS TOOLS for Creo.

### *gt_log_debug*

**Options** 0 or 1

**Default** 0

Defines whether additional debug mode entries from GENIUS TOOLS for Creo should be written in the Creo trail file and the Creo message area. (0 - No, 1 - Yes)

### *gt_log_debug_in_trail*

**Options** 0 or 1

**Default** 0

Defines whether additional debug mode entries from GENIUS TOOLS for Creo should be written in the Creo trail file. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_mbd_allow_active_combstate_activation***

**Options** 0 or 1

**Default** 0

Defines the behavior of the current orientation (camera position) of the model when an already active combined view is activated again.0 - Orientation will be maintained.1 - The orientation will be reset as defined in the combined view.This applies to all GENIUS TOOLS MBD for ISO GPS tools, that can activate combined views.

***gt_regen_times***

**Options** 1, 2 or 3

**Default** 1

Defines how often GENIUS TOOLS for Creo should regenerate at regeneration.

***gt_replace_character_if_not_found***

**Options** Any text

**Default**

Defines a string that is returned instead of variables if they do not exist. The "empty" option returns an empty string.

***gt_resource_folder***

**Options** Path or foldername

**Default** %TEMP%

Defines the directory for all resources.

***gt_rest_auth_provider***

**Options** Path

**Default**

Defines a part of a path between Windchill and the servlet. It have to include the seperators.Example:/protocolAuthforhttps://windchill.ptcmscloud.com/Windchill/protocolAuth/servlet/rest/structure/drawings


© 2026 INNEO Solutions GmbH

***gt_rest_sso_connector***

**Options** Any Text

**Default** "GT_SSO_Proxy.exe"

Defines the login program to be used for the single sign-on login.

***gt_send_telemetry_data***

**Options** 0 or 1

**Default** %GT_TELEMETRY%

Defines whether anonymized telemetry data is sent to INNEO Solutions GmbH.

***gt_show_license_dialog***

**Options** 0 or 1

**Default** 1

Defines whether an error dialog for license errors is displayed. (0 - No, 1 - Yes)

***gt_show_welcome_message_version***

**Options** Any Text or empty

**Default** Current Version

Defines the name whether the splash screen open is saved.

***gt_sqlite_db_max_tries***

**Options** Any number

**Default** 25

Number of trials to access a SQLite database.

***gt_sqlite_db_sleeptime_between_tries***

**Options** Any number

**Default** 10

Wait time in ms before trying to reconnect to a SQLite database (see also gt_sqlite_db_max_tries).


© 2026 INNEO Solutions GmbH

***gt_ssl_verification***

**Options** 0 or 1

**Default** 0

***gt_start_assembly***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Assembly Report can be started by users. (0 - No, 1 - Yes)

***gt_start_assembly_editor***

**Options** 0 or 1

**Default** 1

Defines whether Assembly Report Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_configuration_utility***

**Options** 0 or 1

**Default** 1

Defines whether Configuration Utility can be started by users. (0 - No, 1 - Yes)

***gt_start_dimension***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Dimension can be started by users. (0 - No, 1 - Yes)

***gt_start_dissolve_table***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Dissolve Table can be started by users. (0 - No, 1 - Yes).


© 2026 INNEO Solutions GmbH

***gt_start_export_tdp***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Export TDP can be started by users. (0 - No, 1 - Yes)

***gt_start_forms***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Forms can be started by users. (0 - No, 1 - Yes)

***gt_start_forms_editor***

**Options** 0 or 1

**Default** 1

Defines whether the editor of the forms may be started.

***gt_start_function_manager***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Function Manager can be started by users. (0 - No, 1 - Yes)

***gt_start_inspect***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect can be started by users. (0 - No, 1 - Yes)

***gt_start_inspect_3d***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect 3D can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_start_inspect_3d_revision***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect 3D Revision can be started by users. (0 - No, 1 - Yes)

***gt_start_inspect_editor***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_inspect_revision***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect Revision can be started by users. (0 - No, 1 - Yes)

***gt_start_library***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Library can be started by users. (0 - No, 1 - Yes)

***gt_start_library_editor***

**Options** 0 or 1

**Default** 1

Defines whether Library Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_material***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Material can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_start_material_editor***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Material Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_mbd_management_tools***

**Options** 0 or 1

**Default** 0

Defines whether the menu ribbon GENIUS TOOLS MBD can be started.

***gt_start_mbd_tables***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS MBD Tables can be started by users. (0 - No, 1 - Yes)

***gt_start_mbdinfo***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS MBD Info can be started by users. (0 - No, 1 - Yes).

***gt_start_multibody_to_assembly***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Multibody To Assembly can be started by users. (0 - No, 1 - Yes)

***gt_start_name_generator***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Name Generator can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_start_name_generator_editor***

**Options** 0 or 1

**Default** 1

Defines whether Name Generator Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_orientation_manager***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Orientation Manager can be started by users. (0 - No, 1 - Yes)

***gt_start_parameter***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Parameter can be started by users. (0 - No, 1 - Yes)

***gt_start_parameter_editor***

**Options** 0 or 1

**Default** 1

Defines whether Parameter Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_quick_access***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Quick Access can be started by users. (0 - No, 1 - Yes)

***gt_start_quick_access_editor***

**Options** 0 or 1

**Default** 1

Defines whether Quick Access Editor can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_start_stack_note***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Stack Note can be started by users. (0 - No, 1 - Yes).

***gt_start_stack_note_3d***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Stack Note 3D can be started by users. (0 - No, 1 - Yes).

***gt_start_stack_note_editor***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Stack Note Editor can be started by users. (0 - No, 1 - Yes).

***gt_start_udf_form***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS UDF Forms can be started by users. (0 - No, 1 - Yes)

***gt_start_udf_form_editor***

**Options** 0 or 1

**Default** 1

Defines whether UDF Forms Editor can be started by users. (0 - No, 1 - Yes)

***gt_start_utilities***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Utilities can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gt_start_value_transfer***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Value Transfer can be started by users. (0 - No, 1 - Yes)

***gt_version_string***

**Options** Any text

**Default**

Shows current version of GENIUS TOOLS for Creo.

***gt_visible_details_rows_set***

**Options** Any number >0

**Default** 9

Defines the height of the detail area as row count

***gt_window_size_position_save***

**Options** 0 or 1

**Default** 1

Defines whether the position and size of GENIUS TOOLS dialogs are saved and used again the next time. Also saves the status of the parameter model list.

## GENIUS TOOLS Assembly Report

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Assembly Report are made.

***gta_autoload_folder***

**Options** Path

**Default**

Defines the folder in which the currently selected definition is searched for in order to update it.


© 2026 INNEO Solutions GmbH

***gta_change_wtPart***

**Options** 0 or 1

**Default** 1

Defines whether the user may change the corresponding WTPart name if there is no match in the windchill database.You can set %TBXADMIN%.

***gta_clear_pos_parameter***

**Options** 0 or 1

**Default** 0

Defines whether all POS parameters created up to that point are reset to -1 or "-1" before the POS parameter is written.

***gta_dec_places***

**Options** 0 to 6

**Default** 2

Defines the number of decimal places displayed in tables in GENIUS TOOLS Assembly Record.

***gta_default_file***

**Options** Filepath

**Default** %gt_resource_folder%assembly\gt_assembly.xml

Defines the file name of the default report for Assembly Report.

***gta_empty_export_rows***

**Options** 0 or 1

**Default** 0

***gta_export_creo_position***

**Options** 0 or 1

**Default** 1

Defines whether the position of a component should be exported to CSV or Excel in addition to the columns defined in the editor.


© 2026 INNEO Solutions GmbH

***gta_export_file***

**Options** Filename

**Default**

Defines the default name of an exported report file. You can use the GENIUS TOOLS for Creo variables. (For example: "=%PART_NO%_%CAD_REVISION%_%CAD_CREATED_ON%") Please note: Do not use file extensions if you use Excel and CSV exports.

***gta_export_path***

**Options** Path or foldername

**Default**

Defines the default path for saving reports.

***gta_export_position_value***

**Options** 0 or 1

**Default** 1

Defines whether the position value of a component should be exported to CSV or Excel in addition to the columns defined in the editor.

***gta_export_rownumber***

**Options** 0 or 1

**Default** 1

Defines whether the row index should be exported to CSV or Excel in addition to the columns defined in the editor.

***gta_export_template***

**Options** Filepath

**Default** %gt_resource_folder%assembly\\gt_assembly.xlsx

Defines the default export template.


© 2026 INNEO Solutions GmbH

***gta_export_type***

**Options** 0 or 1

**Default** 1

Defines whether the component type should be exported to CSV or Excel in addition to the columns defined in the editor.

***gta_fill_empty_with_default***

**Options** 0 or 1

**Default** 1

Defines whether the component parameter for the index should be filled with its default value if the parameter does not exist in the model.

***gta_lang***

**Options** Language code (en, de, es, ...)

**Default** de

Defines additional languages to support multi-language titles in GENIUS TOOLS Assembly Report using two-character language codes.

***gta_level_row_colors***

**Options** Any text

**Default**

Defines the colors of table rows according to their level in the bill of materials. Separated by comma and given as hex values.

***gta_open_export_csv***

**Options** 0 or 1

**Default** 1

Defines whether the exported CSV is opened or not.


© 2026 INNEO Solutions GmbH

***gta_save_xml_in_mdl***

**Options** 0 or 1

**Default** 1

Defines whether the GENIUS TOOLS Assembly Report XML template is written into the model (0 - XML template is saved externally in a file, 1 - XML template is written into the model).

***gta_show_mdl_list***

**Options** 0 or 1

**Default** 1

Defines whether the list of assemblies of the first level is visible or not.

***gta_struct_insert_space***

**Options** 0 to 10

**Default** 3

Defines the number of blanks that are prefixed per level in each entry.

***gta_update_component_parameter_type***

**Options** 0 or 1

**Default** 1

Defines whether component parameters are updated to the data type of the current definition.

***gta_wt_part_attribute_name***

**Options** Any Text

**Default** Number

Defines an attribute to find the WTPart that contains the product structure.

***gta_wt_part_attribute_value***

**Options** Any Text

**Default** %PTC_WM_PART_NUMBER%

Defines an attribute value to find the WTPart that contains the product structure.


© 2026 INNEO Solutions GmbH

***gta_wt_part_by_attribute***

**Options** 0 or 1

**Default** 0

Defines whether the associated WTPart is searched for using attribute (1) or by owner (0).

***gta_wt_part_pos_attribute_name***

**Options** Any Text

**Default** LineNumber

Defines an attribute of a WTPart that represents the position.

## GENIUS TOOLS Dimension

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Dimension are made.

***gtd_can_change_family_table_values***

**Options** 0 or 1

**Default** 1

Defines whether users can change entries in family tables in GENIUS TOOLS Dimension.

***gtd_dec_places***

**Options** 0 to 6

**Default** 2

Defines the number of decimal places displayed in GENIUS TOOLS Dimension.

***gtd_filter***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Dimension is started with active name filter (named dimensions). (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtd_filter_additive***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Dimension filters working additive or logical grouped. (0 - logical grouped, 1 - additive)

***gtd_filter_value***

**Options** Floating-point number

**Default** 0.001

Defines the start value for the Value filter.

***gtd_filter_value2***

**Options** Floating-point number

**Default** 5.000

Defines the start value for the second Value filter.

***gtd_label_col_size***

**Options** Any number

**Default** 10

Defines the width of the dimension names column in characters. A value less than 5 will be ignored.

***gtd_regenerate_only_dimensions_model***

**Options** 0 or 1

**Default** 0

Defines whether the root model (assembly) and all its sub models are regenerated (0) or only the currently selected model in GENIUS TOOLS Dimension with its sub models (1).

***gtd_value_col_size***

**Options** Any number

**Default** 5

Defines the width of the dimension values column in characters. A value less than 5 will be ignored.


© 2026 INNEO Solutions GmbH

# GENIUS TOOLS Export TDP

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Export TDP are made.

***gttdp_can_change_workdir***

**Options** 0 or 1

**Default** 1

Defines whether the working directory may be changed in order to find the required models.

***gttdp_chapter_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%export_tdp\chapters\

Defines the folder path for the chapter templates.

***gttdp_component_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%export_tdp\components\

Defines the folder path for the TDP components (f.e. drawing formats).

***gttdp_delete_temp_folder***

**Options** 0 or 1

**Default** 1

Defines whether temporary folder, which are created during each export, should be deleted after the export.

***gttdp_file_name***

**Options** Any Text

**Default** @mdln@

Defines the default name for the PDF file to be exported.


© 2026 INNEO Solutions GmbH

***gttdp_imagemagick***

**Options** Path or foldername

**Default** @unzip

Defines the installation location of an existing ImageMagick installation.@unzip unzips the included ImageMagick installation into the directory defined in gttdp_tools_unzip_folder.

***gttdp_lang***

**Options** Path or foldername

**Default** en

Defines the display language of GENIUS TOOLS Export TDP.

***gttdp_miktex***

**Options** Path or foldername

**Default** @unzip

Defines the installation location of an existing MikTex installation.@unzip - Unpacks the included MikTex installation into the gttdp_tools_unzip_folder directory.

***gttdp_open_after_export***

**Options** 0 or 1

**Default** 0

Defines whether the checkbox for opening after export should be checked (1) or unchecked (0) by default.

***gttdp_show_cmd_commands***

**Options** 0 or 1

**Default** 1

Defines whether executed commands should be displayed for reporting purposes (1) or not (0).

***gttdp_template_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%export_tdp\templates\

Defines the folder path for the TDP templates.


© 2026 INNEO Solutions GmbH

***gttdp_tools_unzip_folder***

**Options** Path or foldername

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\

Defines the folder in which tools should be unpacked with @unzip.

## GENIUS TOOLS Forms

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Forms are made.

***gtf_ask_before_changing_fam_table***

**Options** 0 or 1

**Default** 0

Defines whether users will be asked before changes to entries in family tables are applied in GENIUS TOOLS Forms.

***gtf_auto_replace_comma_for_float_values***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Forms also accepts numerical inputs with comma (,) as decimal separator (1) or not (0).

***gtf_autoload_folder***

**Options** Path

**Default**

Defines the folder that would be searched for the xml based on the value of gtf_autoload_parameter.

***gtf_autoload_overwrite***

**Options** 0 or 1

**Default** 0

Defines whether it should be looked for an external form only if no internal exists (0) or always (1).


© 2026 INNEO Solutions GmbH

***gtf_autoload_parameter***

**Options** Any text

**Default** WEBCODE

Defines the parameter which value should be used to post load external saved forms.

***gtf_def_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en,de,fr

Defines the languages of element descriptions that are displayed in the Forms Editor.

***gtf_default_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%forms

Defines the default folder for imports and exports of definitions and values.

***gtf_descriptionwidth***

**Options** Any number

**Default** 15

Defines the width of the name column of GENIUS TOOLS Forms in characters.

***gtf_editor_multiselect***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Froms Editor "Add multiple items until cancel" is selected. (0 - is unselected, 1 - is selected)

***gtf_external_data_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%forms

Defines the folder that is used for automatic CSV imports.


© 2026 INNEO Solutions GmbH

***gtf_import_xml_use_name_if_no_id_defined***

**Options** 0 or 1

**Default** 1

Defines if at import of values from xml the element name should be used if the id is missing.

***gtf_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines the standard language of element descriptions which are displayed in Forms and the Forms Editor (standard is the current Creo language).

***gtf_midwidth***

**Options** Any number

**Default** 15

Defines the width of the middle column in Forms in characters.

***gtf_namewidth***

**Options** Any number

**Default** 10

Defines the width of the name column in Forms in characters.

***gtf_open_with_selected_model***

**Options** 0 or 1

**Default** 1

Defines whether the current model (0) or the selected model (1) should be used at the startup of GENIUS TOOLS Forms.

***gtf_separator***

**Options** Any character

**Default** ;

Defines the separator for reading CSV files.


© 2026 INNEO Solutions GmbH

***gtf_show_regen_btn***

**Options** 0 or 1

**Default** 0

Defines whether the regeneration from forms save process is deactivate able.

## GENIUS TOOLS Function Manager

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Function Manager are made.

***gtfm_add_cs_to_existing_function***

**Options** Path or foldername

**Default** %gt_resource_folder%function_manager\AddCS.xml

Defines the template used for adding combined views later.

***gtfm_add_info_to_existing_function***

**Options** Path or foldername

**Default** %gt_resource_folder%function_manager\AddInfo.xml

Defines the template used for adding info objects later.

***gtfm_appTable_description_visible***

**Options** 0 or 1

**Default** 1

Defines, whether the appearance tables inside GENIUS TOOLS Function Manager show the descriptions of the entries in a seperate column (1), or inside of tooltips (0).

***gtfm_color_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%function_manager\color_templates\

Defines the path to the color tables.


© 2026 INNEO Solutions GmbH

***gtfm_default_text_database***

**Options** Any Text

**Default** %gt_resource_folder%function_manager\gt_function_db.sqlite

Defines the folder path to the stored standard text database.

***gtfm_default_text_for_name***

**Options** Any Text

**Default** key

Defines the column used when filling in function names with default texts in Function Manager.

***gtfm_default_text_table***

**Options** Any Text

**Default** selection

Defines the database table from which the standard texts are used. Refering to the database stored under “gtfm_default_text_database”.

***gtfm_init_templatenode_selection***

**Options** Path or foldername

**Default** Functions\B01-Master.xml

Defines the template that will be selected when the "Add" dialog is opened.

***gtfm_iso_separator***

**Options** One letter

**Default** -

Defines the seperator for ISO-names.

***gtfm_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines the display language of GENIUS TOOLS Function Manager.


© 2026 INNEO Solutions GmbH

***gtfm_mil_separator***

**Options** One letter

**Default** _

Defines the seperator for military-names.

***gtfm_naming_scheme***

**Options** 1 or 2

**Default** 1

Defines the preset naming scheme (1=Functional Design, 2=Military Standard).

***gtfm_template_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%function_manager\function_templates\

Defines the path to the templates for functions.

## GENIUS TOOLS Inspect

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Inspect are made.

***gti_activate_simple_linked_auto_placement***

**Options** 0 or 1

**Default** 1

Defines whether the symbol will be automatically placed during "Simple linked insertion". (0 - No, 1 - Yes)

***gti_adopt_rotation_from_target***

**Options** 0 or 1

**Default** 1

Defines whether inspection symbols adopt the rotation of their target. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

### *gti_decimal_marker_follow_dtl*

**Options** 0 or 1

**Default** 1

Defines whether the DTL setting "decimal_marker" is read, thereby starting the replacement of "." with "," in Inspect report tables (0 - No, 1 - Yes).

### *gti_def_lang*

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines the display language of GENIUS TOOLS Inspect.

### *gti_din_compliant*

**Options** 0 or 1

**Default** 1

Defines whether the numbering should be similar to DIN 6770 (1), or whether there should be a new number for each symbol according to the defined rules (0).If the option is activated, gti_number_sort_at_height, gti_number_sort_at_type and gti_numbering_all_sheets get ignored.

### *gti_excel_export_file*

**Options** Filename

**Default**

Defines the default name of an exported report file. The file extension (.xlsx or .xlsm) must also be specified. You can use the GENIUS TOOLS for Creo variables. (For example: "=% PART_NO%_%CAD_REVISION%_%CAD_CREATED_ON%")

### *gti_excel_export_path*

**Options** Path

**Default**

Defines the default path for saving reports.


© 2026 INNEO Solutions GmbH

***gti_excel_template***

**Options** Filename

**Default** gti_template_de.xlsx

Defines the name of the basic Excel template.

***gti_fillup_places***

**Options** 0 to 5

**Default** 3

Defines the maximum number of leading zeros that numbers are filled up with.

***gti_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%inspect\

Defines the folder containing the symbols, tables and the definitions.

***gti_lang***

**Options** Language code (en, de, es, ...)

**Default** de

Defines additional languages for displaying UI elements using two-character language codes.

***gti_number_range_per_sheet***

**Options** 0 or 1

**Default** 0

Defines for multi-sheet drawings whether symbols have a number range for the whole drawing (0) or per sheet (1).

***gti_number_sort_at_height***

**Options** -1, 0 or 1

**Default** 0

Defines whether symbols are numbered by height in ascending (1), descending (-1) or creation order (0).


© 2026 INNEO Solutions GmbH

***gti_number_sort_at_type***

**Options** -1, 0 or 1

**Default** 0

Defines whether symbols are numbered by type asc (1), desc (-1) or creation order (0).

***gti_numbering_all_sheets***

**Options** 0 or 1

**Default** 0

Defines whether symbols should be numbered across all sheets.

***gti_placement_exception_rule***

**Options** Any Text (e.g. RegExp)

**Default**

Defines a rule, in the form of a regular expression, to ignore target elements when placing.

***gti_revision_excel_coloring***

**Options** 0 or 1

**Default** 1

Defines whether the symbols should be colored in the Excelexport.

***gti_revision_excel_export_file***

**Options** Filename

**Default**

Defines the default name of an exported report file. The file extension (.xlsx or .xlsm) must also be specified. You can use the GENIUS TOOLS for Creo variables. (For example: "=%PART_NO%_%CAD_REVISION%_%CAD_CREATED_ON%")

***gti_revision_excel_export_path***

**Options** Path

**Default**

Defines the default path for saving reports.


© 2026 INNEO Solutions GmbH

***gti_revision_excel_template***

**Options** Filename

**Default** gti_revision_template_de.xlsx

Defines the name of the basic Excel template.

***gti_revision_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%inspect\

Defines the folder for the search for the base templates for Excel.

***gti_revision_parameter***

**Options** Parametername

**Default** %CAD_REVISION%

Defines the parameter which is read out for the revision display.

***gti_shape_and_position_as_note***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Inspect interprets notes with contained shape and position tolerance symbols as main type "shape and position". ( 0 - note, 1 - shape- and position)

***gti_show_old_symbols***

**Options** 0 or 1

**Default** 1

Defines whether currently existing symbols from old revisions are also displayed.

***gti_start_file***

**Options** Filename

**Default** gti_definition.xml

Defines the name of the basic definition file.


© 2026 INNEO Solutions GmbH

***gti_start_number***

**Options** 0 or 1

**Default** 1

Defines the starting number for incrementing inspection symbols.

## GENIUS TOOLS Library

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Library are made.

***gtl_action_copy_set_file_as_common_name***

**Options** 0 or 1

**Default** 1

If active (1), a copied file is renamed with the Common Name.

***gtl_action_insert_with_mapkey***

**Options** 0 or 1

**Default** 1

Defines whether inserting a part from GENIUS TOOLS Library is done by mapkey or a programatic function (different user experience).

***gtl_action_mfg_insert_as_fixture***

**Options** 0 or 1

**Default** 1

Defines whether insert as fixture / copy insert as fixture is aviable (1) or not (0).

***gtl_action_open_defined_simp_rep***

**Options** Name of the simplified representation

**Default** DEFAULT ENVELOPE REP

Defines the simplified representation that get called by "Open / Insert simplified graphics"

***gtl_blue_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 0000FF

Specifies the color value for blue in GENIUS TOOLS Library.


© 2026 INNEO Solutions GmbH

***gtl_close_detail_panel_after_copy***

**Options** 0 or 1

**Default** 1

Defines whether the detail dialog is closed (1) or remains open (0) after a file is copied.

***gtl_close_detail_panel_after_insert***

**Options** 0 or 1

**Default** 0

Defines whether the detail dialog is closed (1) or remains open (0) after a file is inserted.

***gtl_close_detail_panel_after_open***

**Options** 0 or 1

**Default** 1

Defines whether the detail dialog is closed after a file is opened or inserted (1) or remains open (0).

***gtl_copy_check_existence***

**Options** 0 or 1

**Default** 1

Defines whether the new names are checked for existence before the start of a copy process (could take a long time).

***gtl_copy_drawings_with_same_name***

**Options** none, part, assem or both

**Default** %rename_drawings_with_object%

Defines whether drawings of the same name are copied as well. Default is the value of the Creo option "rename_drawings_with_object". Possible options: none - drawings are not copied. assem - Only for assemblies. part - Only for parts. both - both.


© 2026 INNEO Solutions GmbH

***gtl_copy_remove_wt_templates_from_ws***

**Options** 0 or 1

**Default** 0

Defines whether the template for copying should be removed from the workspace (only Windchill) after the copy process (1) or not (0).

***gtl_db_path***

**Options** Folderpath

**Default** %gt_resource_folder%library\

Defines the path to the library databases.

***gtl_def_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines an alternative language code for the case that the translation (defined in gtl_lang) is not found in the library database.

***gtl_detail_image_folder***

**Options** Path or foldername

**Default** %gtl_current_db_path%

Defines the folder for images displayed in the detail window:%gtl_current_db_path% - uses /library//img_detail/Otherwise, the specified path is used.

***gtl_detail_window_autoincrease_size***

**Options** 0 or 1

**Default** 1

Defines whether the configuration options gtl_detail_window_select_height, gtl_detail_window_select_width, gtl_detail_window_udf_forms_height and gtl_detail_window_udf_forms_width are used (1) or not (0).


© 2026 INNEO Solutions GmbH

***gtl_detail_window_detail_image_height***

**Options** Any number

**Default** 0

Defines the height of a detail image.

***gtl_detail_window_detail_image_show_title***

**Options** 0 or 1

**Default** 1

If a detail image for an object is found, its title will be displayed next to it (1) or not (0).

***gtl_detail_window_detail_image_width***

**Options** Any number

**Default** 0

Defines the width of the detail image.

***gtl_detail_window_height***

**Options** Any number

**Default** 450

Defines the initial height of the detail dialog in pixel.

***gtl_detail_window_hide_details***

**Options** 0 or 1

**Default** 0

Defines whether object details are always shown (0) or hidden initially (1).

***gtl_detail_window_move_by_tree_width***

**Options** 0 or 1

**Default** 1

Defines whether the detail dialog is displayed next to the model tree. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtl_detail_window_preselected_tab***

**Options** 0 to 2

**Default** 2

Defines the default tab of the Detail window in GENIUS TOOLS Library (0: Details, 1: Selection 2: Form).

***gtl_detail_window_preselected_tab_gph***

**Options** -1 to 2

**Default** -1

Defines the default tab of the Detail window in GENIUS TOOLS Library (-1: Inherit from gtl_detail_window_preselected_tab, 0: Details, 1: Selection 2: Form).

***gtl_detail_window_select_height***

**Options** Any number

**Default** 600

Defines the minimal height of the Detail window in pixel after changing to the "Selection" tab.

***gtl_detail_window_select_width***

**Options** Any number

**Default** 300

Defines the minimal width of the Detail dialog in pixel after changing to the "Selection" tab.

***gtl_detail_window_selection_information_height***

**Options** Any number

**Default** 4

Defines the height of selection information in rows.

***gtl_detail_window_show_both_languages***

**Options** 0 or 1

**Default** 0

Defines whether both languages (gtl_lang and gtl_def_lang) are displayed in the Detail window (1) or gtl_lang only (0).


© 2026 INNEO Solutions GmbH

***gtl_detail_window_show_status_in_head_area***

**Options** 0 or 1

**Default** 0

Defines whether the status should be displayed inside the head area of the detail dialog (0 - No, 1 - Yes).

***gtl_detail_window_show_variant_attribute_type***

**Options** 0 or 1

**Default** 0

Shows attribute types (D:, P:) in the Selection tab of the Detail window of GENIUS TOOLS Library (1).

***gtl_detail_window_udf_forms_height***

**Options** Any number

**Default** 700

Defines the minimal height of the Detail dialog in pixel after changing to the Forms tab (UDF Forms).

***gtl_detail_window_udf_forms_width***

**Options** Any number

**Default** 350

Defines the minimal width of the Detail dialog in pixel after changing the Forms tab (UDF Forms).

***gtl_detail_window_width***

**Options** Any number

**Default** 300

Defines the initial width of the Detail dialog in pixel.

***gtl_dnd_enabled***

**Options** 0 or 1

**Default** 1

Activates (1) or deactivates (0) Drag and Drop for GENIUS TOOLS Library in Creo 3.0.


© 2026 INNEO Solutions GmbH

***gtl_downsync***

**Options** 0 or 1

**Default** 1

Defines whether libraries are cached locally on initial call (1) of if they are always read from GT_RESOURCE_FOLDER (0) This configuration option does not affect the editor.

***gtl_downsync_path***

**Options** Folderpath

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\library\

Defines the path for locally caching libraries. Depends on gtl_downsync.

***gtl_editor_create_db_security_copy_once_a_day***

**Options** 0 or 1

**Default** 1

Defines whether a backup of a database is created once a day after opening it with the Library Editor (1) or not (0).

***gtl_editor_file_import_action_fallback***

**Options** 0 to 262143

**Default** 16383

Defines one or more actions which are added to library objects during import or object creation via addition (31=1+2+4+8+16). 1 - Open, 2 - Insert, 4 - Copy, 8 - Copy insert, 16 - Insert (with reference), 16384 - Merge, 32768 - Die, 65536 - Punch, 131072 - Insert as Copy Geom

***gtl_editor_file_import_check_selection_existance***

**Options** 0 or 1

**Default** 0

Defines whether the existence check during the import is also applied to instances (1). PLEASE NOTE: This affects import times significantly.


© 2026 INNEO Solutions GmbH

***gtl_editor_find_double_objects_by_path_and_name***

**Options** 0 or 1

**Default** 0

Defines how duplicate objects are checked (0 - by object name, 1 - by object path and object name).

***gtl_editor_link_double_objects***

**Options** 0 or 1

**Default** 1

Defines how doubled objects are treated by GENIUS TOOLS Library (0 - every doubled object is created during the import, 1 - Doubled objects are linked and only created once during the import).

***gtl_editor_mnu_creator_automatic_selection***

**Options** 0 or 1

**Default** 1

Defines whether tree view selections are inherited between subnodes of the tree in the MNU export dialog. (Yes - 1, No - 0)

***gtl_editor_selections_inherit_instances***

**Options** 0 or 1

**Default** 1

Defines whether images are handed down from a generic to its instances. (0 - No, 1 - Yes)

***gtl_editor_sqlite_allow_unsecure_write_operations***

**Options** 0 or 1

**Default** 1

Defines whether unsecure but faster write operations are used for database access. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtl_editor_use_black_on_white_for_screenshots***

**Options** 0 or 1

**Default** 1

Defines, if images inside of a batch rework get created with the system colors "Black on White" (1) or the current settings (0).

***gtl_favorite_button1_liblink***

**Options** Any text

**Default** company

Defines the file name of the first favorite library (this is sepearted of the others).

***gtl_favorite_button2_liblink***

**Options** Any text

**Default** designtools

Defines the file name of the second favorite library.

***gtl_favorite_button3_liblink***

**Options** Any text

**Default** planttools

Defines the file name of the third favorite library.

***gtl_favorite_button4_liblink***

**Options** Any text

**Default** user

Defines the file name of the fourth favorite library.

***gtl_favorite_button_show***

**Options** 0 or 1

**Default** 1

Defines whether the buttons for favorites are displayed in GENIUS TOOLS Library. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtl_favorite_path***

**Options** Path or foldername

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\library\

Defines the path to save GENIUS TOOLS Library favorites.

***gtl_filter_blue***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Blue"

***gtl_filter_blue_text***

**Options** Any text

**Default**

Defines the description of the status "Blue", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_green***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Green"

***gtl_filter_green_text***

**Options** Any text

**Default**

Defines the description of the status "Green", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_lilac***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Lilac"


© 2026 INNEO Solutions GmbH

***gtl_filter_lilac_text***

**Options** Any text

**Default**

Defines the description of the status "Lilac", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_normal***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Normal"

***gtl_filter_normal_text***

**Options** Any text

**Default**

Defines the description of the status "Normal", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_red***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Red"

***gtl_filter_red_text***

**Options** Any text

**Default**

Defines the description of the status "Red", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_teal***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Teal"


© 2026 INNEO Solutions GmbH

***gtl_filter_teal_text***

**Options** Any text

**Default**

Defines the description of the status "Teal", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_filter_yellow***

**Options** 0 or 1

**Default** 1

Default filter setting for the filter "Yellow"

***gtl_filter_yellow_text***

**Options** Any text

**Default**

Defines the description of the status "Yellow", displayed in the Detail dialog. If the option is empty, the language-dependend default is used.

***gtl_green_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 008800

Specifies the color value for green in GENIUS TOOLS Library.

***gtl_gtf_save_forms_in_model***

**Options** 0 or 1

**Default** 1

Defines whether Forms are copied into a model (1) or not (0) during the creation of a new copy of a model.

***gtl_gtng_gtf_show_name_dialog_everytime***

**Options** 0 or 1

**Default** 0

Defines whether the name dialog is displayed always (1), or only if it has not been opened manually before (0).


© 2026 INNEO Solutions GmbH

***gtl_gtng_new_name_rule***

**Options** Any Text

**Default** @number@@oldname@

Defines the rule for generating names for objects that should be copied.@number@ = generated name@oldname@ = name of the object

***gtl_gtng_overwrite_std_number_definition***

**Options** 0 or 1

**Default** 1

Defines whether the Name Generator definition @number@@oldname@ is replaced by @number:%gtl_gtng_standard_db_filter_for_file_copy%@@oldname@ also if the definition has been set explicitly.

***gtl_gtng_standard_db_filter_for_file_copy***

**Options** Any text

**Default**

Limits the displayed name configurations from GENIUS TOOLS Name Generator. If only one result remains, this name configuration is automatically used.

***gtl_gtp_ignore_start_gtp_after_model_creation***

**Options** Any Text

**Default**

Defines the submodel types, separated by comma, that should not start GENIUS TOOLS Parameter after model creation.submodel types:- part_solid, part_composite, part_sheetmetal, part_concept_model- asm_design, asm_interchange, asm_intchg_subst, asm_intchg_func, asm_class_cav, asm_verify, asm_procplan, asm_ncmodel, asm_class_scan_set, asm_figure, asm_ext_simp_rep, asm_configure_asm- mfg_ncasm, mfg_ncpart, mfg_expmach, mfg_cmm, mfg_sheetmetal, mfg_cast, mfg_mold, mfg_dieface, mfg_harness, mfg_procplan, old_reg_mfg- bulk, regen_backup, manikin_top, manikin_seg, ce_solid, creo_intrch, creo_cnf_factory, ecad_asm, ecad_board_part


© 2026 INNEO Solutions GmbH

***gtl_gtp_start_gtp_after_model_creation***

**Options** Any Text

**Default**

Defines the model types (prt,asm,drw), seperated by comma, that should start GENIUS TOOLS Parameter after model creation.

***gtl_home_db***

**Options** Filename

**Default**

When containing a path to a database the path bar of GTL will display an icon showing a house. Clicking this icon will open this database.

***gtl_img_create_detail_size***

**Options** Any Number > 0

**Default** 200

Defines the size for image creation for details.

***gtl_img_create_tooltip_size***

**Options** Any Number > 0

**Default** 200

Defines the size for image creation for tooltips.

***gtl_img_detail_size***

**Options** Number >= 20

**Default** 100

Defines the rendering size for detail images in pixel.

***gtl_img_size***

**Options** 100 or 40

**Default** 100

Defines the displayed icon size in pixel.


© 2026 INNEO Solutions GmbH

***gtl_img_switch_size***

**Options** Number >= 20

**Default** 40

Defines the rendering size for switch images in pixel.

***gtl_info_folder***

**Options** Folderpath

**Default** %gtl_current_db_path%

Defines a folder that will be searched for information documents for library objects during a batch processing run.

***gtl_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines a language which is used to display library objects as a language code.

***gtl_lilac_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** FF00FF

Specifies the color value for lilac in GENIUS TOOLS Library.

***gtl_list_use_with_double_click***

**Options** 0 or 1

**Default** 0

Defines whether the walk through the catalogue structure goes by single click (0) or double click (1).

***gtl_mark_category_images***

**Options** 0 or 1

**Default** 1

Defines whether images of library categories get a triangle symbol for better differentiation.


© 2026 INNEO Solutions GmbH

***gtl_mdl3d_search_by_bounding_box***

**Options** 0 or 1

**Default** 1

Defines the 3D search behavior together with gtl_mdl3d_search_by_voxel.

***gtl_mdl3d_search_by_voxel***

**Options** 0 or 1

**Default** 1

Defines the 3D search behavior together with gtl_mdl3d_search_by_bounding_box.

***gtl_mdl3d_search_factor_bb_bounding_box***

**Options** 0 to 1

**Default** 0.5

Defines the billing factor. The values entered for the three configuration options gtl_mdl3d_search_factor_bb_bounding_box, gtl_mdl3d_search_factor_bb_bounding_box_unnorm und gtl_mdl3d_search_factor_bb_mass_center must add up to 1 (= 100%). They determine the weighting of the individual search options.

***gtl_mdl3d_search_factor_bb_bounding_box_unnorm***

**Options** 0 to 1

**Default** 0

Defines the billing factor. The values entered for the three configuration options gtl_mdl3d_search_factor_bb_bounding_box, gtl_mdl3d_search_factor_bb_bounding_box_unnorm und gtl_mdl3d_search_factor_bb_mass_center must add up to 1 (= 100%). They determine the weighting of the individual search options.


© 2026 INNEO Solutions GmbH

***gtl_mdl3d_search_factor_bb_mass_center***

**Options** 0 to 1

**Default** 0.5

Defines the billing factor. The values entered for the three configuration options gtl_mdl3d_search_factor_bb_bounding_box, gtl_mdl3d_search_factor_bb_bounding_box_unnorm und gtl_mdl3d_search_factor_bb_mass_center must add up to 1 (= 100%). They determine the weighting of the individual search options.

***gtl_mdl3d_search_factor_bounding_box***

**Options** 0 to 1

**Default** 0.5

Defines the billing factor. The values entered for the two configuration options gtl_mdl3d_search_factor_bounding_box and gtl_mdl3d_search_factor_voxel must add up to 1 (= 100%). They determine the weighting of the individual search options.

***gtl_mdl3d_search_factor_voxel***

**Options** 0 to 1

**Default** 0.5

Defines the billing factor. The values entered for the two configuration options gtl_mdl3d_search_factor_bounding_box and gtl_mdl3d_search_factor_voxel must add up to 1 (= 100%). They determine the weighting of the individual search options.

***gtl_mdl3d_search_voxel_per_axe***

**Options** Any number

**Default** 10

Defines the number of voxels for 3D resolution. Please note: This value must be consistent in the database and in the program.

***gtl_parameter_multiple_value_separator***

**Options** One letter

**Default**

Specifies the character that separates multi-value parameter values.


© 2026 INNEO Solutions GmbH

***gtl_red_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 880000

Specifies the color value for red in GENIUS TOOLS Library.

***gtl_retrieve_run_batch***

**Options** Any text

**Default**

Defines a batch file that is called in preparation for a file load to session.@name@ will be replaced by object name.@path@ will be replaced by object path.

***gtl_run_mode***

**Options** 0 or 1

**Default** 1

Defines the display mode of GENIUS TOOLS Library. (0 - as external dialog, 1 - inside the Creo navigator pane)

***gtl_search_btn_search***

**Options** 0 or 1

**Default** 0

Defines whether a search is performed after a click on the extended search button. (0 - No, 1 - Yes)

***gtl_search_no_type_no_folder***

**Options** 0 or 1

**Default** 0

Defines whether folders will be searched (1) or not (0) when the type flag is not set in the search options.


© 2026 INNEO Solutions GmbH

***gtl_search_tree_dependent***

**Options** 0 or 1

**Default** 1

Defines whether the complete database or only categories/subcategories get searched. The search in categories doen't show objects from selections. (0 - No, 1 - Yes).

***gtl_search_values_per_page***

**Options** Any number

**Default** 25

Number of rows per page. Values <1 and >999 are read as 25.

***gtl_show_both_languages***

**Options** 0 or 1

**Default** 0

Defines whether both languages (gtl_lang and gtl_def_lang) are displayed in the library browser (1) or only gtl_lang with the fallback gtl_def_lang (0).

***gtl_show_category_names***

**Options** 0 or 1

**Default** 0

Defines whether category names are displayed.

***gtl_show_font_size***

**Options** Any number

**Default** 5

Defines the font size in the library browser.

***gtl_show_object_names***

**Options** 0 or 1

**Default** 1

Defines whether object names are displayed in GENIUS TOOLS Library. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtl_show_path***

**Options** 0 or 1

**Default** 1

Defines whether the path of underlying categories will be shown (1) or not (0).

***gtl_show_tooltip_image***

**Options** 0 or 1

**Default** 1

Defines whether tooltip images are displayed.

***gtl_start_db***

**Options** Filename

**Default**

Defines the library loaded on startup. If empty, the alpha-numerical first is used.

***gtl_teal_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 008888

Specifies the color value for teal in GENIUS TOOLS Library.

***gtl_tooltip_image_folder***

**Options** Path or foldername

**Default** %gtl_current_db_path%

Defines the folder for tooltip images:%gtl_current_db_path% -
uses /library//img_tooltip/Otherwise it uses the path given in this value

***gtl_yellow_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** FFA500

Specifies the color value for yellow in GENIUS TOOLS Library.

## GENIUS TOOLS Material

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Material are made.


© 2026 INNEO Solutions GmbH

***gtm_allow_edit_favorites***

**Options** 0 or 1

**Default** 1

Defines whether the user is allowed to edit the favorites.

***gtm_can_set_to_system_default***

**Options** 0 or 1

**Default** 1

Defines whether system default is setable or not.

***gtm_close_at_set_material***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Material should be restartet after material declaration.

***gtm_command_file***

**Options** Filepath

**Default** %gt_resource_folder%material\material.db

Defines the database used by GENIUS TOOLS Material and GENIUS TOOLS Material Editor.

***gtm_db_def_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines a language code for a language to be used if the translation is not found in gtm_db_lang.

***gtm_db_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines the displayed language of material attributes as a language code (standard is the Creo language).


© 2026 INNEO Solutions GmbH

***gtm_delete_not_current_materials***

**Options** 0 or 1

**Default** 0

This configuration deletes all materials that do not match the current material when a new material is set.This could cause troubles while using:- Relations- Pro/Program- Family tables- Multi body- ...Setting is explicitly not recommended.

***gtm_editor_material_browser_path***

**Options** Path

**Default** %GTS_SERVERONLY_DIR%
\tools\freeware_gt_material_browser\GT_Material_Browser.exe

Defines the path where GENIUS TOOLS Material Browser get searched. If it is found it could be started from GT Material Editor. At Version 1.0.4.0 also path transfer ist supported (Temppath if WT connection is set).

***gtm_exclude_material_from_update_all***

**Options** Any Text

**Default** 0

Defines a material that is excluded from the update of all materials (without extension).

***gtm_favorite_file***

**Options** Filepath

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\material\favorites.txt

Defines the file whether the favorite materials get saved.

***gtm_infoDoc_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%material\info\

Defines the directory containing information documents for GENIUS TOOLS Material, if no path is specified.


© 2026 INNEO Solutions GmbH

***gtm_remember_selected_filter***

**Options** Any number

**Default** 1

The number of layers that should be kept.

***gtm_replace_materials_with_same_name***

**Options** 0 or 1

**Default** 1

Defines the behavior of GENIUS TOOLS Material if an assigned material has the same name as the new assigned material (0 - the old material will not be replaced, 1 - the old material will be replaced).

***gtm_show_body_selection***

**Options** 0 or 1

**Default** 1

Defines whether the body selector would be shown or not.

***gtm_show_infoDoc_in_list***

**Options** 0 or 1

**Default** 1

Defines whether the info symbol should already be shown in the shortlist or only after picking the material.

***gtm_show_properties_button***

**Options** 0 or 1

**Default** 1

Defines whether the "Open material properties" control is shown in GENIUS TOOLS Material (1) or not (0).

***gtm_visible_details_rows_set***

**Options** 8-21

**Default** 9

Defines the height of the material list in rows.


© 2026 INNEO Solutions GmbH

# GENIUS TOOLS <%GTMTAB%>

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS <%GTMTAB%> are made.

***gtmtab_color_checked***

**Options** 0 or 1

**Default** 1

Defines whether colorization is activated by default for MBD Tables. (0 - No, 1 - Yes)

***gtmtab_display_all_size_dimensions***

**Options** 0 or 1

**Default** 1

Defines whether the Size Table includes all toleranced dimensions, or just the fit dimensions.0 - Fit dimensions1 - All dimensions

***gtmtab_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines the default languages for GENIUS TOOLS MBD Tables as a list of language abbreviations.

***gtmtab_sizeTab_color_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%mbdTables\sizeTable\

Defines the path to the color codings for the Size Table.

***gtmtab_sort_size_table_by***

**Options** 0 to 2

**Default** 1

Defines the sorting of the Size Table.0 - None1 - Dimension value2 - Tolerance range


© 2026 INNEO Solutions GmbH

***gtmtab_sort_surffin_table_by***

**Options** 0 or 1

**Default** 1

Defines the sorting of the Surface Finish Table.0 - None1 - Roughness

***gtmtab_surfTab_color_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%mbdTables\surfFinTable\

Defines the path to the color codings for the Surface Finish Table.

***gtmtab_transfer_checked***

**Options** 0 or 1

**Default** 1

Defines whether annotation transfer is enabled by default for MBD Tables. (0 - No, 1 - Yes)

## GENIUS TOOLS Name Generator

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Name Generator are made.

***gtng_copy_template_if_filter_db_not_found***

**Options** Any Text

**Default**

Defines whether and which template would be used for the creation of undefined filter databases. If the definition is empty no db would be created.

***gtng_db_name_filter***

**Options** Any text

**Default**

All file names of configuration definitions are searched for the value defined here. Only definitions containing this value are displayed in Name Generator.


© 2026 INNEO Solutions GmbH

***gtng_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%name_generator\

Defines a directory containing the global name configurations for Name Generator.

***gtng_local_folder***

**Options** Path or foldername

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\name_generator\

Defines a directory containing the local name configurations for Name Generator.

***gtng_use_windchill_credentials_for_server_request***

**Options** 0 or 1

**Default** 1

If a Windchill server is recognized, this option defines whether the login data is requested (1) or an URL without login data is retrived (0).

## GENIUS TOOLS <%GTOM%>

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS <%GTOM%> are made.

***gtom_projection_menu_type***

**Options** 0 or 1

**Default** 1

Defines which menu type is used for projections. (0 - Options Menu, 1 - Radio Group)

***gtom_projection_type_init***

**Options** 0 or 1 or 2

**Default** 1

Defines the projection type, which is selected by default.0 - Rectangle Projection1 - Isometric2 - Trimetric

## GENIUS TOOLS Parameter

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Parameter are made.


© 2026 INNEO Solutions GmbH

***gtp_alternative_background_color***

**Options** Any text

**Default**

Defines the color of every even row in the parameter form. Enter the colour as RGB values (e.g.: 122,45,89). The default color (grey) is used if the config option is empty. If the value is "n" or "no", no background color will be used.

***gtp_ask_for_save***

**Options** 0 or 1

**Default** 1

A window will pop up in the GTP after changing the model or closing the dialog, if there are parameters with changed values.This behaviour will be deactivated if the option is "0".

***gtp_bold_parameter_values***

**Options** 0 or 1

**Default** 0

Defines whether the parameter values are displayed in bold typeface. (0 - No, 1 - Yes)

***gtp_check_connections***

**Options** 0 or 1

**Default** 1

Defines whether database connections and filters should be checked after importing the parameter values and missing values be set automatically. (0 - No, 1 - Yes)

***gtp_db_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%parameter\database\

Defines the folder for the databases which are used for database nodes.


© 2026 INNEO Solutions GmbH

***gtp_default_value***

**Options** Any text

**Default** -

Defines the default value for parameter values.This value is added to value lists in GTP regardless of whether it is valid for a parameter. If an empty value is set as the default value, it is not added to value lists.The default value is considered as an empty value when checking for empty values, e.g. in mandatory fields.

***gtp_designate***

**Options** 0 or 1

**Default** 1

Defines whether the checkbox „Remove all designations [in Windchill mode]“ in the tools menu of GT Parameter is displayed. (0 - no, 1 - yes)

***gtp_do_not_save_conflicts***

**Options** 0 or 1

**Default** 1

Defines whether you can save when filter conflicts occur. (0 - Yes, 1 - No)

***gtp_do_not_save_empty_mandatory***

**Options** 0 or 1

**Default** 1

Defines whether you can save when mandatory fields are left blank. (0 - Yes, 1 - No)

***gtp_do_not_save_format_conflicts***

**Options** 0 or 1

**Default** 1

Defines whether you can save when format conflicts occur. (0 - Yes, 1 - No)


© 2026 INNEO Solutions GmbH

***gtp_dock_dialog_to_mdl***

**Options** 0 or 1

**Default** 1

Defines the behavior of the GTP dialog. If the value is "1" the dialog belongs to the specific model window. If the value is "0" the dialog can be open and the user can interact with the model window.

***gtp_dropdown_height_max***

**Options** Any number

**Default** 10

Defines the maximum number of elements in a dropdown list displayed when lists from databases, CSV or text files are used in GT Parameter.

***gtp_editor_new_def_without_sys_params***

**Options** 0 or 1

**Default** 0

Defines whether existing system parameters of Creo/Windchill should be included when a new parameter definition is created from a model. (0 - System parameters are not included, 1 - System parameters are included into the definition)

***gtp_editor_open_csv_program***

**Options** Filename

**Default** notepad.exe

Defines the standard program to edit CSV and text files.

***gtp_file***

**Options** Filepath

**Default** %GT_RESOURCE_FOLDER%parameter\gtp_int_de\gtp_int_de.xml

Defines the path to the parameter definition file for GT Parameter (fallback).


© 2026 INNEO Solutions GmbH

***gtp_file_param***

**Options** Any text

**Default** MC_CHECKTYPE

Defines a parameter of the type string, which is used to automatically select a parameter definition file in conjunction with "gt_lst".

***gtp_fill_empty_description***

**Options** 0 or 1

**Default** 0

Defines whether an empty description of a Creo parameter should be overwritten with the description from the current parameter definition. (0 - No, 1 - Yes)

***gtp_filter_auto_fill_back***

**Options** 0 or 1

**Default** 1

Defines whether the filtering is set automatically if only one selection is possible. (0 - No, 1 - Yes)

***gtp_filter_auto_single_fill***

**Options** 0 or 1

**Default** 1

Defines whether the last available hit of a filtering is set automatically into a parameter value field (1) or not (0).

***gtp_gtr_rules***

**Options** Path

**Default** %gt_resource_folder%parameter\

Defines the GENIUS TOOLS Value Transfer config load path, except it is not explicit defined.


© 2026 INNEO Solutions GmbH

***gtp_ignore_start_gtp_after_model_creation***

**Options** Any Text

**Default**

Defines the submodel types, separated by comma, that should not start GENIUS TOOLS Parameter after model creation.submodel types:- part_solid, part_composite, part_sheetmetal, part_concept_model- asm_design, asm_interchange, asm_intchg_subst, asm_intchg_func, asm_class_cav, asm_verify, asm_procplan, asm_ncmodel, asm_class_scan_set, asm_figure, asm_ext_simp_rep, asm_configure_asm- mfg_ncasm, mfg_ncpart, mfg_expmach, mfg_cmm, mfg_sheetmetal, mfg_cast, mfg_mold, mfg_dieface, mfg_harness, mfg_procplan, old_reg_mfg- bulk, regen_backup, manikin_top, manikin_seg, ce_solid, creo_intrch, creo_cnf_factory, ecad_asm, ecad_board_part

***gtp_import_folder***

**Options** Path or foldername

**Default** @curworkdir@

Defines the folder for the import list that can be used for parameter import.

***gtp_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines language codes for additional languages to support language-dependent parameter titles in the parameter definition of GT Parameter.

***gtp_lock_change_generic***

**Options** 0 or 1

**Default** 1

Defines whether parameters of instances that are not in the family table may be changed in GTP. (1 - Yes, 0 - No)For the setting 1, if there are no other restrictions for the parameter of an instance, a symbol is shown to mark that the parameter is not in the family table.

***gtp_lock_rel_locked_params***

**Options** 0 or 1

**Default** 1

Defines whether parameters locked by relations may be changed in GTP.


© 2026 INNEO Solutions GmbH

***gtp_lst***

**Options** Filepath

**Default** %GT_RESOURCE_FOLDER%parameter\gtp.lst

Defines the path to the list file containing the overview of all available parameter definitions.

***gtp_model_tree_column_width***

**Options** Any Number > 0

**Default** 10

Defines the width of additionally displayed parameter columns, measured in the standard width for characters.

***gtp_model_tree_columns***

**Options** Any text

**Default**

Defines the parameters which are displayed as an extra column in the model list.

***gtp_overwrite_description***

**Options** 0 or 1

**Default** 0

Defines whether a description (not empty) of a Creo parameter can be overwritten by GENIUS TOOLS Parameter. (0 - No, 1 - Yes)

***gtp_regen***

**Options** 1, 0, 2, or -1

**Default** 1

Defines whether models will be automatically regenerated after saving and whether the checkbox „Regenerate after saving" in the tools menu of GT Parameter is displayed. (0: display unchecked box, 1: display checked box, 2: no automatic regeneration and hide checkbox, -1: automatic regeneration and hide checkbox)


© 2026 INNEO Solutions GmbH

***gtp_save_hidden***

**Options** 0 or 1

**Default** 0

Defines whether hidden parameters are transfered into the model on Save. (0 - No, 1 - Yes)

***gtp_save_model_at_save_press***

**Options** 0 or 1

**Default** 0

Defines whether the model should saved at GT Parameter save 1 or not 0*.

***gtp_show_duplicate_warning***

**Options** 0 or 1

**Default** 1

Defines whether a warning message box is shown if duplicated parameters occur (1) or only an information message in the message area of Creo (0).

***gtp_show_hidden_params***

**Options** 0 or 1

**Default** 0

Defines whether the hidden parameter tab is displayed in GT Parameter (1) or not (0).

***gtp_show_mdl_list***

**Options** 0 to 2

**Default** 2

Defines whether the model list is displayed: 0 - list is hidden, 1 - list is always displayed, 2 - list is model-dependent.

***gtp_start_drw***

**Options** 0 or 1

**Default** 1

Defines whether the parameters of the active model (0) or the drawing parameters (1) are displayed first by GT Parameter in drawing mode.


© 2026 INNEO Solutions GmbH

***gtp_start_gtp_after_model_creation***

**Options** Any Text

**Default**

Defines the model types (prt,asm,drw), seperated by comma, that should start GENIUS TOOLS Parameter after model creation.

***gtp_use_type_insensitive_dbs***

**Options** 0 or 1

**Default** 1

Defines how to handle integer and double values in databases.1: The values are saved as strings in the database and will be handled as strings.0: The double values are stored as rounded values in the database and it should be queried for ranges, not for values.

***gtp_web_server_url***

**Options** Path or foldername

**Default**

Defines the path to the web server for database queries.

## GENIUS TOOLS Quick Access

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Quick Access are made.

***gtqa_always_option***

**Options** 0 or 1

**Default** 0

Defines whether the modus option "Always" is displayed and usable in the Quick Access Editor.

***gtqa_background_circle***

**Options** 0 or 1

**Default** 1

Defines whether the inner circle of Quick Access is visible.


© 2026 INNEO Solutions GmbH

### *gtqa_background_picture*

**Options** Path and/or file name with extension .png

**Default** gtqa_quick_access.png

Defines the path or file name of the image that is used as logo in the ring menu of GENIUS TOOLS Quick Access. The image has to be saved as a PNG file in one of these folders: \quick_access\img_background\\text\resource\

### *gtqa_change_closed_groups*

**Options** 0 or 1

**Default** 0

Defines whether the locked groups (8 and 9) are shown and editable in the QuickAccess editor.

### *gtqa_command_file*

**Options** Filepath

**Default** %gt_resource_folder%quick_access\quick_access_%GT_CREO_VERSION%.db

Defines the path to the database used by Quick Access and Quick Access Editor.

### *gtqa_command_icon_folder*

**Options** Path or foldername

**Default** %gt_resource_folder%quick_access\img_w20\

Defines the directory for the GENIUS TOOLS Quick Access command icons.

### *gtqa_db_def_lang*

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines a language code for a language to be used for GENIUS TOOLS Quick Access and its Editor if the translation is not found in gtqa_db_lang.


© 2026 INNEO Solutions GmbH

***gtqa_db_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines a language code for a language to be used for the tooltips (standard is the Creo language).

***gtqa_editor_create_db_security_copy_once_a_day***

**Options** 0 or 1

**Default** 1

Defines whether a backup of the database is created once a day after opening it with the Library Editor (1) or not (0).

***gtqa_group_bow***

**Options** 0 or 1

**Default** 0

Defines the group display style. (0 = straight, 1 = curved)

***gtqa_local_command_file***

**Options** Filename

**Default**

Defines a second, local database for Quick Access and Quick Access Editor. e.g. %appdata %/inneo/genius_tools/quick_access/quick_access.db

***gtqa_local_command_group_split***

**Options** 0 to 9 or "all"

**Default** all

Defines which groups of the Quick Access are read from the local database (specify the highest group) and can be configured by user with the Quick Access Editor (0-9) or all (all).


© 2026 INNEO Solutions GmbH

***gtqa_local_command_icon_folder***

**Options** Path or foldername

**Default**

Defines the directory for the icons used by Quick Access.e.g. %appdata %/inneo/genius_tools/quick_access/img_w20/

***gtqa_no_tooltips***

**Options** 0 or 1

**Default** 0

Defines whether tooltips are displayed in Quick Access. (0 - tooltips are displayed, 1 - tooltips are not displayed)

***gtqa_show_admin_switch***

**Options** 0 or 1

**Default** 1

Defines whether a switch to change between the global and local database is shown.

***gtqa_start_mapkey***

**Options** Any text

**Default** <

Defines the mapkey for starting Quick Access.

***gtqa_tooltip_background_alpha_channel***

**Options** 0 to 254

**Default** 180

Defines the opacity of the background from 0 - transparent to 254 - opaque

***gtqa_tooltip_time***

**Options** Any number

**Default** 1.0

Defines the time until tooltips in GENIUS TOOLS Quick Access are displayed in seconds (Default: 1.0).


© 2026 INNEO Solutions GmbH

***gtqa_undock_groups_with_middle_mouse_button***

**Options** 0 or 1

**Default** 1

Undock the open group as a separate window by use of the middle mouse button.

***gtqa_undock_groups_with_right_mouse_button***

**Options** 0 or 1

**Default** 0

Undock the open group as a separate window by use of the right mouse button.

***gtqa_zoom***

**Options** 1.0 to 2.0

**Default** 1.6

Defines the zoom factor of GENIUS TOOLS Quick Access. Zoom >= 1.0 < 1.4 -> 20x20 px IconsZoom >= 1.4 < 1.9 -> 30x30 px IconsZoom >= 1.9 -> 40x40 px Icons

## GENIUS TOOLS <%GTSN%>

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS <%GTSN%> are made.

***gtsn_can_change_workdir***

**Options** 0 or 1

**Default** 1

Defines whether the working directory may be redefined. (0 - No, 1 - Yes)

***gtsn_excel_export_file***

**Options** Any Text

**Default** export.xlsx

Defines the target file for the Excel export.


© 2026 INNEO Solutions GmbH

***gtsn_excel_export_path***

**Options** Any Text

**Default**

Defines the target path for the Excel export.

***gtsn_excel_template***

**Options** Any Text

**Default** template.xlsx

Defines the name of the basic Excel template.

***gtsn_folder***

**Options** Path or folder

**Default** %gt_resource_folder%stack_note\

Defines the folder from which the definitions for GENIUS TOOLS Stack Note can be read.

***gtsn_note_database***

**Options** Path

**Default** stack_note.db

Defines the file path to the database containing the note templates.

***gtsn_note_database_external***

**Options** Filename

**Default**

Defines the file path to the database that contains the additional note elements.

***gtsn_placement_db***

**Options** File name

**Default** stack_note_placement.db

Defines the file name for the placement database.


© 2026 INNEO Solutions GmbH

***gtsn_show_import_old_data***

**Options** 0 or 1

**Default** 0

Defines whether the button for importing legacy data is displayed in GENIUS TOOLS Stack Note. (0 - No, 1 - Yes).

***gtsn_standard_languages***

**Options** Any text

**Default** de,en

Defines the default languages for GENIUS TOOLS Stack Note as a list of language abbreviations.

***gtsn_symbol_database***

**Options** Path

**Default** stack_note_symbol.db

Defines the file path to the database containing the symbol definitions.

## GENIUS TOOLS <%GTSN3D%>

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS <%GTSN3D%> are made.

***gtsn3d_can_change_workdir***

**Options** 0 or 1

**Default** 1

Defines whether the working directory may be redefined. (0 - No, 1 - Yes).

***gtsn3d_excel_export_file***

**Options** Any Text

**Default** export.xlsx

Defines the target file for the Excel export.


© 2026 INNEO Solutions GmbH

***gtsn3d_excel_export_path***

**Options** Any Text

**Default**

Defines the target path for the Excel export.

***gtsn3d_excel_template***

**Options** Any Text

**Default** template.xlsx

Defines the name of the basic Excel template.

***gtsn3d_folder***

**Options** Path or folder

**Default** %gt_resource_folder%stack_note3D\

Defines the folder from which the definitions for GENIUS TOOLS Stack Note 3D can be read.

***gtsn3d_note_database***

**Options** Filename

**Default** stack_note3D.db

Defines the file path to the database containing the note elements.

***gtsn3d_note_database_external***

**Options** Filename

**Default**

Defines the file path to the database that contains the additional note elements.

***gtsn3d_standard_languages***

**Options** Any text

**Default** de,en

Defines the default languages for GENIUS TOOLS Stack Note 3D as a list of language abbreviations.


© 2026 INNEO Solutions GmbH

***gtsn3d_symbol_database***

**Options** Filename

**Default** stack_note3D_symbol.db

Defines the file path to the database containing the symbol definitions.

## GENIUS TOOLS UDF Forms

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS UDF Forms are made.

***gtuf_create_layer***

**Options** 0 or 1

**Default** 1

Defines whether a layer should be created with the name of the UDF and whether the help features should be placed on it.

***gtuf_data_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%udf_forms\gt_udf

Defines the path where the UDF models are located.

***gtuf_dec_places***

**Options** 0 to 6

**Default** 2

Defines the number of decimal places displayed in tables in GENIUS TOOLS UDF Forms.

***gtuf_definition_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%udf_forms\gt_definition

Defines the path where the UDF definitons are located.


© 2026 INNEO Solutions GmbH

***gtuf_external_data_folder***

**Options** Path

**Default** %gt_resource_folder%udf_forms\gt_data

Defines which path should used for external data that are defined without a specific path.

***gtuf_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines language codes for additional languages in which UI elements should be displayed.

***gtuf_save_xml_in_mdl***

**Options** 0 or 1

**Default** 1

Defines whether the XML definitions of GENIUS TOOLS UDF Forms are written to the model. (0 - XML definitions are saved in external files, 1- XML definitions are written to the model)

***gtuf_separator***

**Options** Any character

**Default** ;

Defines the separator for reading CSV files.

***gtuf_show_create_by_creo***

**Options** 0 or 1

**Default** 0

Defines whether the function "Create new UDF group by Creo std. Dialog" in GENIUS TOOLS UDF Forms is disposal.

***gtuf_show_create_without_references***

**Options** 0 or 1

**Default** 1

Defines whether the button to create a UDF without predefined references is shown (1) or not (0).


© 2026 INNEO Solutions GmbH

***gtuf_show_status***

**Options** 0 or 1

**Default** 1

Defines whether the status area below the input fileds should be displayed or not.

# GENIUS TOOLS Utilities

In the configuration file *gt_modules.cfg* the settings for GENIUS TOOLS Utilities are made.

# General Utilities Configuration Options

***gtu_autostart_mapkey_inline***

**Options** 0 or 1

**Default** 0

Defines whether the mapkey should be startet synchronously or asynchronously.

***gtu_autostart_mapkey_mapkey***

**Options** %%Any text

**Default**

Defines the Mapkey. WARNING: The mapkey must be written in one line. Mapkey from the config.pro file can be used by: %%MapkeyName;

***gtu_comp_file_name***

**Options** Path or foldername

**Default** gtu_component_parameters.xml

File name of configuration xml.

***gtu_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** en

Defines the display language of GENIUS TOOLS Utilities.


© 2026 INNEO Solutions GmbH

***gtu_start_autostart_mapkey***

**Options** 0 or 1

**Default** 1

Defines whether a defined mapkey "gtu_autostart_mapkey_mapkey" should be started. (0 - No, 1 - Yes)

***gtu_start_close_all_windows***

**Options** 0 or 1

**Default** 1

Defines whether the button for "Close all other windows" is displayed. (0 - No, 1 - Yes)

***gtu_start_copyCParamToSubsCParam***

**Options** 0 or 1

**Default** 1

Defines whether the utility "GENIUS TOOLS Copy Component Parameter To Substitution Component Parameter" is available. (0 - No, 1 - Yes)

***gtu_start_extendedDimensionFunctions***

**Options** 0 or 1

**Default** 0

Defines whether the module for changing dimensions is displayed (0 - No, 1 - Yes).

***gtu_start_loadSaveConverter***

**Options** 0 or 1

**Default** 1

Defines whether the utility "GENIUS TOOLS Load Save Convertert" is avaiable. (0 - No, 1 - Yes)

***gtu_start_openGeomOrigin***

**Options** 0 or 1

**Default** 1

Defines whether the button for "GT Open Base Model" is displayed. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtu_start_pdm_commands***

**Options** 0 or 1

**Default** 1

Defines whether the buttons for "Open workspace" and "Open commonspace" are displayed. (0 - No, 1 - Yes)

***gtu_start_selectSurfacesByColor***

**Options** 0 or 1

**Default** 1

Defines whether the utility "GENIUS TOOLS Select Surfaces by Color" is available. (0 - No, 1 - Yes)

***gtu_start_toggleSymbolGroups***

**Options** 0 or 1

**Default** 1

Defines whether the "GtuToggleSymbolGroups" button is provided (0 - No, 1 - Yes).

***gtu_start_ui_change***

**Options** 0 or 1

**Default** 1

Defines whether the the module UI Change gets started. (0 - No, 1 - Yes)

***gtu_start_usage_logger***

**Options** 0 or 1

**Default** 0

Defines whether the the usage logger is active. (0 - No, 1 - Yes)

***gtu_ui_change_check_material_check_material_has_no_param***

**Options** 0 or 1

**Default** 1

Defines whether a material without a revision parameter generates a warning in the material check (1) or not (0).


© 2026 INNEO Solutions GmbH

***gtu_ui_change_check_material_check_material_not_in_db***

**Options** 0 or 1

**Default** 1

Defines whether a material which is not in the database generates a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_multiple_materials***

**Options** 0 or 1

**Default** 1

Defines whether multiple material files with the same name generate a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_no_material***

**Options** 0 or 1

**Default** 1

Defines whether no material generates a warning in the material check (1) or not (0). This error message appears up to Creo version 6.

***gtu_ui_change_check_material_check_old_material***

**Options** 0 or 1

**Default** 1

Defines whether a different revision parameter of a material generates a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_only_current_material***

**Options** 0 or 1

**Default** 0

Defines whether only the active material is checked.


© 2026 INNEO Solutions GmbH

***gtu_ui_change_check_material_system_material_is_wrong***

**Options** 0 or 1

**Default** 1

Defines whether the Creo std. material ptc_system_mtrl_props as current would be displayed as an failure case (1) or not (0).

***gtu_ui_change_check_material_version***

**Options** 0 or 1

**Default** 0

Defines whether the material version check should be done. (0 = No, 1 = Yes)

***gtu_ui_change_check_material_version_parameter***

**Options** Any text

**Default** REVISION

Defines the parameter that is filled with the material version.

***gtu_ui_change_filter_annotation_planes***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS should filter the annotation planes based on the orientation set in the active combined view (1) or not (0). Default setting: 1

***gtu_ui_change_hole_optionmenu***

**Options** 0 or 1

**Default** 0

Defines whether the HoleMenu should be expanded or if it keeps its standard height. This should not be used with Creo 5.0 or higher, because PTC changed the custom settings in newer versions.

***gtu_ui_change_hole_optionmenu_length***

**Options** Any number >0

**Default** 20

Defines the number of rows shown in the HoleMenu.


© 2026 INNEO Solutions GmbH

***gtu_ui_change_show_forms_icon***

**Options** 0 or 1

**Default** 1

Defines whether the Forms icon is shown if a model is displayed with Forms (1) or not (0).

***gtu_ui_change_show_info***

**Options** 0 or 1

**Default** 1

Defines whether the text from gtu_ui_change_show_info_text will be displayed in the Creo Window if there is no model loaded (1) or not (0).

***gtu_ui_change_show_info_background***

**Options** Hexadecimal 000000 - FFFFFF

**Default**

Specifies the color value for the background in GENIUS TOOLS Utilities Show Info.

***gtu_ui_change_show_info_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 000000

Specifies the color value for the text in GENIUS TOOLS Utilities Show Info.

***gtu_ui_change_show_info_text***

**Options** Any text

**Default** %GT_VERSION_STRING%

%PROE_START%

INNEO Solution GmbH

Text output that will be displayed if gtu_ui_change_show_info = 1.


© 2026 INNEO Solutions GmbH

***gtu_usage_logger_file***

**Options** Filname with extension .db

**Default** usage.db

Defines the file name for the usage logger database.

***gtu_usage_logger_path***

**Options** Path or foldername

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\usage_logger\

Defines the path in which the usage logger database (gtu_usage_logger_file) is written.

## 3D Note Form

***gtu_3d_note_form_filter_hole_notes***

**Options** 0 or 1

**Default** 1

Defines whether notes on holes are hidden when the dialog box is opened (1, filter is enabled) or not (0).

***gtu_3d_note_form_filter_with_input_panels***

**Options** 0 or 1

**Default** 1

Defines whether notes without input panels are hidden when the dialog box is opened (1, filter is enabled) or not (0).

***gtu_3d_note_form_force_regneration***

**Options** 0 or 1

**Default** 1

Defines the regeneration behavior in GENIUS TOOLS 3-D note editor (0 - std. regeneration, 1 - enforced regeneration)


© 2026 INNEO Solutions GmbH

***gtu_3d_note_form_pagesize***

**Options** Any number >0

**Default** 15

Define the number of shown lines per page

***gtu_start_3d_note_form***

**Options** 0 or 1

**Default** 1

Defines whether the button "GENIUS TOOLS 3D Note Form" is displayed. (0 - No, 1 - Yes)

## Close All Other Windows

***gtu_close_all_windows_save_changed_models***

**Options** 0 or 1

**Default** 0

Defines if changed models get saved at close window.

***gtu_start_close_all_windows***

**Options** 0 or 1

**Default** 1

Defines whether the button for "Close all other windows" is displayed. (0 - No, 1 - Yes)

## Annotation Info

***gtu_annotation_info_excel_template***

**Options** Filename

**Default** gtu_annotation_info_template_de_en.xlsx

Defines the name of the basic Excel template.


© 2026 INNEO Solutions GmbH

***gtu_annotation_info_mandatory_ref_number_for_linear_dimensions***

**Options** Any number >0

**Default** 2

Defines the minimum number of semantic references a linear dimension must have in order to be marked as semantically correct in the GENIUS TOOLS Annotation Info. This applies to:Driving dimensionsDriven dimensionsReference dimensions(Default value: 2)

***gtu_annotation_info_show_hole_notes***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Annotations Info should list automatically generated hole notes. (0 - No, 1 - Yes)

***gtu_start_annotation_info***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Annotation Info can be started by users. (0 - No, 1 - Yes).

## Annotation Transfer

***gtu_start_annotation_transfer***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Annotation Transfer can be started by users. (0 - No, 1 - Yes)

## Open Base Model

***gtu_start_openGeomOrigin***

**Options** 0 or 1

**Default** 1

Defines whether the button for "GT Open Base Model" is displayed. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

# Extend Relations

***gtu_relationextension_calculateBoundingBox***

**Options** 0 or 1

**Default** 1

Defines whether the commands GT_CalculateBoundingBox and GT_CalculateBodyBoundingBox are available in the relations.

***gtu_relationextension_doubleToString***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_DoubleToString is available in the relations.

***gtu_relationextension_round***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_Round is available in the relations.

***gtu_relationextension_unitToString***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_UnitToString is available in the relations.

***gtu_relationextension_updateBodyParamMass***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_UpdateBodyParamMass is available in the relations.

***gtu_relationextension_updateBodyParamMaterial***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_UpdateBodyParamMaterial is available in the relations.


© 2026 INNEO Solutions GmbH

***gtu_relationextension_updateBodyParamVolume***

**Options** 0 or 1

**Default** 1

Defines whether the command GT_UpdateBodyParamVolume is available in the relations.

***gtu_start_relationExtension***

**Options** 0 or 1

**Default** 1

Defines whether the utility with relation extensions is available. (0 - No, 1 - Yes)

## CS Assembler

***gtu_csassembler_component_cs_name***

**Options** Component name

**Default** PLACEMENT_CS

The name of the inserted coordinate system from the component that should be assembled.

***gtu_csassembler_load_simplified***

**Options** 0 or 1

**Default** 0

Defines whether models are loaded with a temporary simplified representation (faster) or fully (more accurate; important for fitting components together) for assembly.

***gtu_csassembler_maximal_cs_count***

**Options** Any number >= 0

**Default** 5000

Defines the maximal used number of coordinate systems.


© 2026 INNEO Solutions GmbH

***gtu_csassembler_multi_level***

**Options** 0 or 1

**Default** 1

Defines whether coordinate systems from assembled components should be used as target coordinate systems inside the same assemble task.

***gtu_csassembler_use_std_ics_as_fallback***

**Options** 0 or 1

**Default** 0

Defines, if a fallback coordinate system should be used. This is defined by the XML node std_ics_name. This must be clearly defined in the template assembly.(0 - No, 1 - Yes). Default: 0

***gtu_csassembler_xml_path***

**Options** Path or foldername

**Default**

Start path to choose a XML-file.

***gtu_start_csassembler***

**Options** 0 or 1

**Default** 1

Defines whether the the module CS Assembler gets started. (0 - No, 1 - Yes)

## Select Surfaces by Color

***gtu_start_selectSurfacesByColor***

**Options** 0 or 1

**Default** 1

Defines whether the utility "GENIUS TOOLS Select Surfaces by Color" is available. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

# Show Thread Size

***gtu_show_thread_size_check_param***

**Options** Any text

**Default** SCHRAUBEN_GROESSE

Defines a language dependent feature parameter. If defined, it will be checked if the parameter exists. If the parameter does not exist, the function is unavailable.

***gtu_show_thread_size_check_param_fallback***

**Options** Any text

**Default** SCREW_SIZE

Defines a language dependent feature parameter. If defined, it will be checked if the parameter exists. If the parameter does not exist, the function is unavailable.

***gtu_show_thread_size_text_definition***

**Options** Any text

**Default** &SCHRAUBEN_GROESSE:FID_@feat_thread_id@ (□n□ @D)

This value has to be customized for "Show Thread Size" to work with the set Creo language.

***gtu_show_thread_size_text_definition_fallback***

**Options** Any text

**Default** &SCREW_SIZE:FID_@feat_thread_id@ (□n□ @D)

This configuration option must be customized for the used Creo language in order for GENIUS TOOLS Show Thread Size to work. The fallback is only used if gtu_show_thread_size_check_param is not found.

***gtu_start_show_thread_size***

**Options** 0 or 1

**Default** 1

Defines whether the button "Show Thread Size" is displayed. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

## GTOL Text and Editor

***gtu_gtol_text_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%utilities\gtol_text\

Defines the path where the GTol Text definitions are located.

***gtu_gtol_text_info_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%utilities\gtol_text\info\

Defines the path where the GTol Text info files are located.

***gtu_gtol_text_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** %gt_lang%

Defines the display language of GENIUS TOOLS GTol Text.

***gtu_gtol_text_table_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%utilities\gtol_text\tables\

Defines the path where the GTol Text tolerance table files are located.

***gtu_start_gtol_text***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS GTol Text can be started by users. (0 - No, 1 - Yes)

***gtu_start_gtol_text_editor***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS GTol Text Editor can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

## Show Information

***gtu_start_ui_change***

**Options** 0 or 1

**Default** 1

Defines whether the the module UI Change gets started. (0 - No, 1 - Yes)

***gtu_ui_change_check_material_check_material_has_no_param***

**Options** 0 or 1

**Default** 1

Defines whether a material without a revision parameter generates a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_material_not_in_db***

**Options** 0 or 1

**Default** 1

Defines whether a material which is not in the database generates a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_multiple_materials***

**Options** 0 or 1

**Default** 1

Defines whether multiple material files with the same name generate a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_no_material***

**Options** 0 or 1

**Default** 1

Defines whether no material generates a warning in the material check (1) or not (0). This error message appears up to Creo version 6.


© 2026 INNEO Solutions GmbH

***gtu_ui_change_check_material_check_old_material***

**Options** 0 or 1

**Default** 1

Defines whether a different revision parameter of a material generates a warning in the material check (1) or not (0).

***gtu_ui_change_check_material_check_only_current_material***

**Options** 0 or 1

**Default** 0

Defines whether only the active material is checked.

***gtu_ui_change_check_material_system_material_is_wrong***

**Options** 0 or 1

**Default** 1

Defines whether the Creo std. material ptc_system_mtrl_props as current would be displayed as an failure case (1) or not (0).

***gtu_ui_change_check_material_version***

**Options** 0 or 1

**Default** 0

Defines whether the material version check should be done. (0 = No, 1 = Yes)

***gtu_ui_change_check_material_version_parameter***

**Options** Any text

**Default** REVISION

Defines the parameter that is filled with the material version.

***gtu_ui_change_filter_annotation_planes***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS should filter the annotation planes based on the orientation set in the active combined view (1) or not (0). Default setting: 1


© 2026 INNEO Solutions GmbH

***gtu_ui_change_hole_optionmenu***

**Options** 0 or 1

**Default** 0

Defines whether the HoleMenu should be expanded or if it keeps its standard height. This should not be used with Creo 5.0 or higher, because PTC changed the custom settings in newer versions.

***gtu_ui_change_hole_optionmenu_length***

**Options** Any number >0

**Default** 20

Defines the number of rows shown in the HoleMenu.

***gtu_ui_change_show_forms_icon***

**Options** 0 or 1

**Default** 1

Defines whether the Forms icon is shown if a model is displayed with Forms (1) or not (0).

***gtu_ui_change_show_info***

**Options** 0 or 1

**Default** 1

Defines whether the text from gtu_ui_change_show_info_text will be displayed in the Creo Window if there is no model loaded (1) or not (0).

***gtu_ui_change_show_info_background***

**Options** Hexadecimal 000000 - FFFFFF

**Default**

Specifies the color value for the background in GENIUS TOOLS Utilities Show Info.

***gtu_ui_change_show_info_color***

**Options** Hexadecimal 000000 - FFFFFF

**Default** 000000

Specifies the color value for the text in GENIUS TOOLS Utilities Show Info.


© 2026 INNEO Solutions GmbH

***gtu_ui_change_show_info_text***

**Options** Any text

**Default** %GT_VERSION_STRING%

%PROE_START%

INNEO Solution GmbH

Text output that will be displayed if gtu_ui_change_show_info = 1.

## Component Parameters

***gtu_comp_file_name***

**Options** Path or foldername

**Default** gtu_component_parameters.xml

File name of configuration xml.

***gtu_start_component_params***

**Options** 0 or 1

**Default** 1

Defines whether the button "Component params" is displayed. (0 - No, 1 - Yes)

## Find Contact Surfaces

***gtu_start_find_contact_surfaces***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Find Contact Surfaces can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

## Select Contact Surfaces

***gtu_start_select_contact_surfaces***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Select contact surfaces can be started by users. (0 - No, 1 - Yes)

## Load Save Converter

***gtu_csassembler_load_simplified***

**Options** 0 or 1

**Default** 0

Defines whether models are loaded with a temporary simplified representation (faster) or fully (more accurate; important for fitting components together) for assembly.

***gtu_start_loadSaveConverter***

**Options** 0 or 1

**Default** 1

Defines whether the utility "GENIUS TOOLS Load Save Convertert" is avaiable. (0 - No, 1 - Yes)

## Convert Materials

***gtu_start_convert_materials***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Convert Materials can be started.


© 2026 INNEO Solutions GmbH

# Points

***gtu_points_autoselect_points_with_rule***

**Options** Any regular expression

**Default**

Defines a rule to preselect Points at start up.

***gtu_points_curve_chain_precision***

**Options** Any real number between 0 and 1

**Default** 0.1

Defines the standard value for curve displays and results with gtu_points_curve_output_type in a definition for displaying curves.

***gtu_points_curve_output_type***

**Options** 0 to 3

**Default** 0

Defines the preselection for displaying curve points0 - References1 - Catenary Curve2 - Absolute Length3 - Relative Length

***gtu_points_precision***

**Options** C++ Float printf definition (e.g. .6 for 6 decimals)

**Default** 15.6

Defines the number of decimals.

***gtu_points_remove_double_selected_points***

**Options** 0 or 1

**Default** 1

Defines whether points that are defined twice should be removed from the saved file.

***gtu_points_write_filename***

**Options** Any text

**Default**

Defines the export file name. You can use variables.


© 2026 INNEO Solutions GmbH

***gtu_points_write_ijk***

**Options** 0 or 1

**Default** 0

Defines whether the normal to next surface should be written to the file.

***gtu_points_write_template_extension***

**Options** Any text

**Default**

Defines the file extension at the usage of templated export.

***gtu_points_write_template_filenames_fill_up_length***

**Options** Any Number > 0

**Default** 20

Defines the fill up length at usage from modelnames in templates.

***gtu_points_write_template_footer***

**Options** Any Text

**Default**

Defines a footer for the .dat file.

***gtu_points_write_template_header***

**Options** Any Text

**Default**

Defines a header for the .dat file.

***gtu_points_write_template_line_left***

**Options** Any Text

**Default**

Defines a line start for the .dat file.


© 2026 INNEO Solutions GmbH

***gtu_points_write_template_line_right***

**Options** Any Text

**Default**

Defines a line end for the .dat file.

***gtu_points_write_template_modelnames***

**Options** 0 or 1

**Default** 0

Defines if model names should be written into the template.

***gtu_points_write_template_names***

**Options** 0 or 1

**Default** 0

Defines whether names should be written to the output file.

***gtu_points_write_template_names_fill_up_length***

**Options** Any number >= 0

**Default** 20

Fill up the name up to # characters.

***gtu_points_write_template_names_split***

**Options** Any Text

**Default** _

Replace the defined string throug a space character.

***gtu_points_write_template_seperator***

**Options** Any Letter

**Default**

Defines the seperator for outputs with template.


© 2026 INNEO Solutions GmbH

***gtu_points_write_use_template***

**Options** 0 or 1

**Default** 0

Defines whether the gtu_points_write_template_* definitions should be used (can also be selected in the options tab at runtime).

***gtu_start_points***

**Options** 0 or 1

**Default** 1

Defines whether the button "GTU Points" is displayed. (0 - No, 1 - Yes)

## Create search.pro

***gtu_create_search_pro_exclude_current_path***

**Options** 0 or 1

**Default** 1

Defines whether the current working directory is included in the search.pro file (0) or not (1).

***gtu_create_search_pro_exclude_file***

**Options** Filename

**Default**

Defines a file containing search paths which are not included in the new search.pro file.

***gtu_create_search_pro_line_start***

**Options** Any character

**Default**

Defines characters which are inserted at the beginning of each line.

***gtu_create_search_pro_path_end_with_slash***

**Options** 0 or 1

**Default** 1

Defines whether a slash (/) (or backslash \) is added to the end of a file path (1) or not (0).


© 2026 INNEO Solutions GmbH

***gtu_create_search_pro_standard_save_folder***

**Options** 0 to 2

**Default** 0

Defines the standard save location. 0 - Creo standard directory, 1 - directory of the current object, 2 - current working directory.

***gtu_create_search_pro_standard_save_name***

**Options** Filename

**Default** search.pro

Defines the suggested file name of the search.pro file in the save dialog.

***gtu_start_create_search_pro***

**Options** 0 or 1

**Default** 1

Defines whether the button for "Create Search.pro" is displayed. (0 - No, 1 - Yes)

## Sort Combined Views

## Edit Standard Texts

***gtu_start_edit_default_texts***

**Options** 0 or 1

**Default** 1

Defines whether the module to edit the default values is enabled. (0 - No, 1 - Yes)

## Show Pitch

***gtu_show_pitch_check_param***

**Options** Any text

**Default** STEIGUNG

Defines a language-dependend feature parameter. If defined, it is checked if the parameter is existing. If the parameter does not exist, the function is unavailable.


© 2026 INNEO Solutions GmbH

***gtu_show_pitch_check_param_fallback***

**Options** Any text

**Default** PITCH

Defines a language-dependend feature parameter. If defined, it is checked if the parameter is existing. If the parameter does not exist, the function is unavailable.

***gtu_show_pitch_text_definition***

**Options** Any text

**Default** M@D x &STEIGUNG:FID_@feat_thread_id@[.2]

This value has to be customized for "Show Pitch" to work with the set Creo language.

***gtu_show_pitch_text_definition_fallback***

**Options** Any text

**Default** M@D x &PITCH:FID_@feat_thread_id@[.2]

This value has to be customized for "Show Pitch" to work with the set Creo language. The fallback is used if gtu_show_pitch_check_param can not be found.

***gtu_start_show_pitch***

**Options** 0 or 1

**Default** 1

Defines whether the button "Show Pitch" is displayed. (0 - No, 1 - Yes)

## Toggle Symbol Variants

***gtu_start_toggleSymbolGroups***

**Options** 0 or 1

**Default** 1

Defines whether the "GtuToggleSymbolGroups" button is provided (0 - No, 1 - Yes).


© 2026 INNEO Solutions GmbH

# Table to CSV

***gtu_start_table_to_csv***

**Options** 0 or 1

**Default** 1

Defines whether the the module Table to CSV gets started. (0 - No, 1 - Yes)

***gtu_table_to_csv_export_file***

**Options** Filename

**Default**

Defines the file name when exported. May contain variables. If this value is empty, the regular file name will be used.

***gtu_table_to_csv_export_folder***

**Options** Path or foldername

**Default**

Defines the destination folder of the CSV file

***gtu_table_to_csv_export_sep***

**Options** Any character

**Default** ;

Defines the separator for writing CSV files.

***gtu_table_to_csv_table_template***

**Options** Filename

**Default**

Template for Table to CSV.

***gtu_table_to_csv_write_file_as_utf8***

**Options** 0 or 1

**Default** 1

Defines whether a CSV is written as ASCII (0) or UTF8 (1).


© 2026 INNEO Solutions GmbH

***gtu_table_to_csv_write_file_as_utf8_with_bom***

**Options** 0 or 1

**Default** 0

Defines whether the UTF8 file is written with (1) or without (0) Byte Order Mark (BOM).

## Table to Excel and Copy Table 1:1 to Excel

***gtu_start_table_to_excel***

**Options** 0 or 1

**Default** 1

Defines whether the button "Copy Table to Excel" is displayed. (0 - No, 1 - Yes)

***gtu_table_to_excel_autosave***

**Options** 0 or 1

**Default** 0

Defines whether the Excel table to be created is written to the defined report folder without a save dialog in the one-to-one export. See configuration "gtu_table_to_excel_report_folder".

***gtu_table_to_excel_copy_borders***

**Options** 0 or 1

**Default** 1

Defines whether the table boundaries will be copied to the Excel table. (0 - No, 1 - Yes)

***gtu_table_to_excel_erase_replaced_comments***

**Options** 0 or 1

**Default** 0

Defines whether the comments in the Excel sheets will be deleted after filling (1) or not (0).

***gtu_table_to_excel_export_rule_sheet***

**Options** 0 or 1

**Default** 1

Defines whether to report the rule worksheet as well. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

***gtu_table_to_excel_name_by_template***

**Options** Any text

**Default**

With the parameters and variables defined here, a name for the output file of "Table to Excel" is generated automatically.

***gtu_table_to_excel_open_export***

**Options** 0 or 1

**Default** 1

Defines whether a report should be opened after export.

***gtu_table_to_excel_report_folder***

**Options** Path or foldername

**Default**

Defines the standard directory for saving Excel reports.

***gtu_table_to_excel_rule_sheet_name***

**Options** Any Text

**Default** Rules

Defines the name of the rule worksheet.

***gtu_table_to_excel_run_check***

**Options** 0 or 1

**Default** 1

Specifies whether to check whether an Excel session is open.

***gtu_table_to_excel_std_template_name***

**Options** Any text

**Default**

Defines a name of an Excel template, selected at the start of the dialog.


© 2026 INNEO Solutions GmbH

***gtu_table_to_excel_template_folder***

**Options** Path or foldername

**Default** %gt_resource_folder%utilities\table_to_excel\

Defines the path to the template directory for "Table to Excel" (.xlsx-files).

***gtu_table_to_excel_use_model_instead_of_drawing_name***

**Options** 0 or 1

**Default** 0

Defines whether the drawing name (0) or the model name (1) are used for naming the Excel table.

## Create Tolerance Table

***gtu_start_tolerance_table***

**Options** 0 or 1

**Default** 1

Defines whether the button "Create Tolerance Table" is displayed. (0 - No, 1 - Yes)

***gtu_tol_table_creo_insert***

**Options** 0 or 1

**Default** 1

Defines the insertion of the tolerance table (0 - insertion without preview, 1 - insertion with preview on the mouse cursor).

***gtu_tol_table_decimal_marker_follow_dtl***

**Options** 0 or 1

**Default** 1

Reads the DTL file (1*) or not (0). If the DTL option decimal_marker = COMMA is set, the representation of numerical values will be changed from period to comma as decimal separator.


© 2026 INNEO Solutions GmbH

### *gtu_tol_table_fitsize_color*

**Options** 0,1,2,3,4,5,6,8,9,10 oder 12

**Default** 9

Defines the color of the fitsizes in the tolerance table. The following colors can be used:0 Red (PRO_COLOR_LETTER)1 Green (PRO_COLOR_HIGHLITE)2 White (PRO_COLOR_DRAWING)3 Background (PRO_COLOR_BACKGROUND)4 Grey (PRO_COLOR_HALF_TONE)5 Blue (PRO_COLOR_EDGE_HIGHLIGHT)6 Grey (PRO_COLOR_DIMMED)8 Magenta (PRO_COLOR_ERROR)9 Cyan (PRO_COLOR_WARNING)10 Green (PRO_COLOR_SHEETMETAL)12 Brown (PRO_COLOR_CURVE)

### *gtu_tol_table_fitsize_height*

**Options** Positive real number

**Default** 2.5

Defines the text height of the tolerance table column "Fitsize".

### *gtu_tol_table_fitsize_text*

**Options** Any text

**Default**

Defines the column name "Fitsize". If nothing is specified, the standard of GENIUS TOOLS is used, based on the current Creo language. You can split multi-line headings by using /n.

### *gtu_tol_table_fitsize_width*

**Options** 1-999

**Default** 19

Defines the width of the fitsize column.

### *gtu_tol_table_font*

**Options** Any text

**Default**

Defines the font of the tolerance table. The Creo standard font is used, if no value is set.


© 2026 INNEO Solutions GmbH

***gtu_tol_table_form***

**Options** A or B

**Default** A

Defines the type of the tolerance table. The available table configurations can be found in the online help.

***gtu_tol_table_hidden_font***

**Options** Any text

**Default** isofont

Defines the font for the hidden table header. If no font is defined the header would get the drawing font.

***gtu_tol_table_lower_text***

**Options** Any text

**Default**

Defines the column name "Minimum". If nothing is specified, the standard of GENIUS TOOLS is used, based on the current Creo language. You can split multi-line headings by using /n.

***gtu_tol_table_minimal_pending_zeros***

**Options** 0 to 999

**Default** 3

Defines the minimum number of characters of a tolerance value which are displayed.

***gtu_tol_table_only_dims_from_current_sheet***

**Options** 0 or 1

**Default** 1

Defines whether all dimensions (0) or only the dimensions of the current sheet (1) are displayed in the tolerance table.


© 2026 INNEO Solutions GmbH

***gtu_tol_table_origin_at_bottom_right***

**Options** 0 or 1

**Default** 0

Defines the origin of the table and its growth direction:0 - The origin of the table is at the top left and its growth direction to the bottom right.1 - The origin of the table is at the bottom right and its growth direction to the top left.These configuration did affect the table placement only with gtu_tol_table_creo_insert = 0.

***gtu_tol_table_show_diameter***

**Options** 0 or 1

**Default** 1

Defines whether a diameter sign is displayed on diameter tolerances. (0 - No, 1 - Yes)

***gtu_tol_table_show_plus_at_positive***

**Options** 0 or 1

**Default** 0

Defines whether a plus sign is displayed on positive numbers (1) or not (0).

***gtu_tol_table_show_prefix***

**Options** 0 or 1

**Default** 1

If the prefix sign should be shown or not

***gtu_tol_table_show_sign_before_zero***

**Options** Any text

**Default**

If a table cell contains only a zero, the text defined here is displayed before this zero.

***gtu_tol_table_sort_order***

**Options** ASC or DESC

**Default** ASC

Defines whether the table is sorted in ascending or descending order.


© 2026 INNEO Solutions GmbH

***gtu_tol_table_std_height_plus***

**Options** Positive real number

**Default** 2.5

Defines the cell height with the inner text height (text height + standard height = cell height).

***gtu_tol_table_text_color***

**Options** 0,1,2,3,4,5,6,8,9,10 or 12

**Default** 0

Defines the text color of the tolerance table. The following colors can be used:0 Red (PRO_COLOR_LETTER)1 Green (PRO_COLOR_HIGHLITE)2 White (PRO_COLOR_DRAWING) 3 Background (PRO_COLOR_BACKGROUND)4 Grey (PRO_COLOR_HALF_TONE)5 Blue (PRO_COLOR_EDGE_HIGHLIGHT)6 Grey (PRO_COLOR_DIMMED)8 Magenta (PRO_COLOR_ERROR)9 Cyan (PRO_COLOR_WARNING)10 Green (PRO_COLOR_SHEETMETAL)12 Brown (PRO_COLOR_CURVE)

***gtu_tol_table_text_height***

**Options** Positive real number

**Default** 2.5

Defines the text height of the tolerance table column tolerance.

***gtu_tol_table_tolerance_color***

**Options** 0,1,2,3,4,5,6,8,9,10 or 12

**Default** 0

Defines the color of the tolerances in the tolerance table. The following colors can be used:0 Red (PRO_COLOR_LETTER)1 Green (PRO_COLOR_HIGHLITE)2 White (PRO_COLOR_DRAWING)3 Background (PRO_COLOR_BACKGROUND)4 Grey (PRO_COLOR_HALF_TONE)5 Blue (PRO_COLOR_EDGE_HIGHLIGHT)6 Grey (PRO_COLOR_DIMMED)8 Magenta (PRO_COLOR_ERROR)9 Cyan (PRO_COLOR_WARNING)10 Green (PRO_COLOR_SHEETMETAL)12 Brown (PRO_COLOR_CURVE)


© 2026 INNEO Solutions GmbH

***gtu_tol_table_tolerance_height***

**Options** Positive real number

**Default** 2.5

Defines the text height of the tolerance table column header.

***gtu_tol_table_tolerance_minimal_pending_zeros***

**Options** -1 till 9

**Default** 2

Defines the number of decimal places for the values in the column Tolerance. 0: No decimal places are displayed.1...9: Number of decimal places. The last digit will be rounded if necessary. -1: Decimal places are shown without alteration, i. e. no zeros are added.

***gtu_tol_table_tolerance_text***

**Options** Any text

**Default**

Defines the column name "Tolerance". If nothing is specified, the standard of GENIUS TOOLS is used, based on the current Creo language. You can split multi-line headings by using /n.

***gtu_tol_table_tolerance_width***

**Options** 1-999

**Default** 19

Defines the width of the tolerance column.

***gtu_tol_table_upper_text***

**Options** Any text

**Default**

Defines the column name "Maximum" is defined. If nothing is specified, the standard of GENIUS TOOLS based on the current Creo language is used. You can split multi-line headings by using /n.


© 2026 INNEO Solutions GmbH

## Open/Create Drawing

***gtu_ord_copy_common_name_on_drw_create***

**Options** 0 or 1

**Default** 1

Defines whether the PTC_COMMON_NAME remains the standard name of the drawing file (0) or the PTC_COMMON_NAME is copied from the model (1).

***gtu_ord_createdrw***

**Options** Any text

**Default**

Defines a mapkey which is started after the create drawing dialog is opened. If a submapkey should be used please write it like %%mapkey_name;.

***gtu_ord_drw_name***

**Options** Any Text or Empty

**Default** @mdlname@

Defines the name of the drawing that should be created or opened. The use of variables is possible. If empty, the WT number generator is used.Definition e.g. : $$repl$MO-$DW-$@mdlname@$$

***gtu_ord_ignore_name_for_common_name***

**Options** 0 or 1

**Default** 1

Defines whether the common name is set as expected (0) or whether the common name is not used if it is identical to the model name (1)

***gtu_ord_pdm_auto_open_one_drawing***

**Options** 0 or 1

**Default** 0

Defines whether a single associated drawing found in the PDM system should be opened by default (1)


© 2026 INNEO Solutions GmbH

***gtu_ord_pdm_close_dialog_after_show_url***

**Options** 0 or 1

**Default** 1

Defines whether PDM dialog should be closed after displaying the detail page (1)

***gtu_ord_pdm_file_name_attribute***

**Options** Any text

**Default** objCadModelName

Defines the Windchill attribute in which the file name will be found. It may be differ depending on the specific Windchill installation and version. (e.g. name, number, objCadModelName)

***gtu_ord_pdm_look_for_used_parts***

**Options** 0 or 1

**Default** 0

Defines whether drawings should also be searched that reference the current part as a subpart (REST API support required).

***gtu_ord_pdm_odata_file_name_attribute***

**Options** Any text

**Default** FileName

Defines the Windchill attribute in which the file name will be found. It maybe differ depending on the specific Windchill installation and version e.g. name, number, objCadModelName.

***gtu_ord_pdm_show_Thumbnails***

**Options** 0 or 1

**Default** 1

Defines whether thumbnails from Windchill (when using REST API) should be downloaded (1) or not (0).


© 2026 INNEO Solutions GmbH

***gtu_ord_pdm_shown_attributes***

**Options** Any text

**Default**

Type in additional attributes to be shown. These must be deposited in Windchill. This may be parameters of other object attributes (e.g. status, version,...) Type in the ids of the attributes as comma separated values (e.g. "REVISION,VERSION,STATUS" or "IBA|REVISION,IBA|VERSION,IBA|STATUS")

***gtu_ord_pdm_sort_out_doubles***

**Options** 0 or 1

**Default** 1

Defines whether duplicate drawings from the PDM (resource, drawing model) should be filtered from the resource list (1)

***gtu_ord_post_drw_name***

**Options** Any text

**Default**

Defines a suffix for the file name of drawings. = PREFIX + + SUFFIX + .DRW

***gtu_ord_pre_drw_name***

**Options** Any text

**Default**

Defines a prefix for the file name of drawings. = PREFIX + + SUFFIX + .DRW

***gtu_ord_try_to_use_selected_part_if_inside_asm***

**Options** 0 or 1

**Default** 1

Defines whether the select model request is shown every time (0) or only if no model is already selected (1).


© 2026 INNEO Solutions GmbH

***gtu_ord_use_pdm_server***

**Options** 0 or 1

**Default** 1

Defines whether the Windchill plugin should be searched automatically as soon as Windchill is connected

***gtu_start_open_create_drawing***

**Options** 0 or 1

**Default** 1

Defines whether the button for "Open/Create Drawing" is displayed. (0 - No, 1 - Yes)

***gtu_tol_table_sort_order***

**Options** ASC or DESC

**Default** ASC

Defines whether the table is sorted in ascending or descending order.

## Full Backup

***gtu_start_fullbackup***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Full Backup can be started by users. (0 - No, 1 - Yes)

## Command Control

***gtu_command_control_configuration***

**Options** NAME_1:MODE_1|NAME_2:MODE_2|NAME_3:MODE_3

**Default** ProCmdMdlTreeWfChkInExp:UNAVAILABLE

Defines the commands and the defined modes:NAME_1:MODE_1|NAME_2:MODE_2|NAME_3:MODE_3Mode:UNAVAILABLEINVISIBLEDISALLOW


© 2026 INNEO Solutions GmbH

***gtu_start_command_control***

**Options** 0 or 1

**Default** 0

Defines whether GENIUS TOOLS Command Control would be startet. (0 - No, 1 - Yes)

## Work Dir Manager

***gtu_start_work_dir_manager***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Work Dir Manager can be started by users. (0 - No, 1 - Yes)

***gtu_work_dir_manager_always_at_front***

**Options** 0 or 1

**Default** 0

Definies whether the dialog stays on top, or falls behind others.

***gtu_work_dir_manager_autodelete_after_days***

**Options** Any Number > 0

**Default** 1

Defines the number of days until paths get deleted from list.

***gtu_work_dir_manager_first_chars***

**Options** Any Number

**Default** 10

Defines the number of characters from left side to shorten the path for display.

***gtu_work_dir_manager_last_chars***

**Options** Any Number

**Default** 25

Defines the number of characters from right side to shorten the path for display.


© 2026 INNEO Solutions GmbH

***gtu_work_dir_manager_save_path***

**Options** Path

**Default** %appdata%\INNEO\GENIUS_TOOLS\for_Creo\work_dir_manager\

Defines the path for system files.

## Combined View Gallery

***gtu_combined_view_gallery_creation_rule***

**Options** Any Text (e.g. RegExp)

**Default**

Defines a rule, in the form of a regular expression, to find combined views whose screenshot is displayed when the program is started.

***gtu_combined_view_gallery_folder***

**Options** Path or folder

**Default** %gt_resource_folder%utilities\combined_view_gallery\

Defines the path from which additional data for combined view gallery is read.

***gtu_combined_view_gallery_lang***

**Options** Language abbreviation (en, de, es, ...)

**Default** de

Defines additional languages for displaying UI elements using two-character language codes.

***gtu_start_combined_view_gallery***

**Options** 0 or 1

**Default** 1

Defines whether GENIUS TOOLS Combined View Gallery can be started by users. (0 - No, 1 - Yes)


© 2026 INNEO Solutions GmbH

# 26 Variables

The following variables can be used in various GENIUS TOOLS for Creo applications and can be combined with regular text.

Example: The specification `@date@_Project5_@mdln@` resolves to *2017-16-02_Project5_prt0001*.

If a variable cannot return a value, the `gt_replace_character_if_not_found` configuration option determines the returned value. If the configuration option is not set, the variable name will be returned as entered.

## User input

Instead of using a variable which is resolved automatically, you can also use text input by a user.

To cause a prompt (text box) to be displayed to the users, use the format `==LabelText==`. The text enclosed by double equal signs is displayed in the input prompt. This text, including the equal signs, is then replaced by the text entered by the user.

Example for generating a file name:

```
@date@_==Please enter descriptive name==_@mdln@
```

## Date and time information

| Text operation | Description | Example |
|---|---|---|
| **@date@** | Returns the current date in format yyyy-mm-dd for easy sorting. | 2017-13-02 |
| **@datede@** | Returns the current date in German format dd.mm.yyyy for easy sorting. | 13.02.2017 |
| **@dateen@** | Returns the current date in English format dd-mm-yyyy. | 13-02-2017 |
| **@dateus@** | Returns the current date in US-American format mm-dd-yyyy. | 02-13-2017 |
| **@dateusshort@** | Returns the current US-American format, short notation mm-dd-yy. | 02-13-17 |


© 2026 INNEO Solutions GmbH

| Text operation | Description | Example |
|---|---|---|
| **@time@** | Returns the current time in a notation that can also be used for filenames, short notation HH-MM (0-23)-(0-59). | 14-15 |
| **@timede@** | Returns the current time in German format, short notation HH:MM (0-23): (0-59). | 14:15 |
| **@timeen@** | Returns the current time in US-American format, short notation hh:MM (1-12):(0-59) AM/PM. | 02:15 AM |

## Object information

| Text operation | Description | Example |
|---|---|---|
| **@curworkdir@** | Returns the current working directory. | c:\temp |
| **@clipboard@** | Returns the current buffer memory. | |
| **@common@** | Returns the common name. | |
| **@count_form@** | Returns the number of internal forms of a model. | |
| **@mdln@** | Returns the current model name. | |
| **@mdlpath@** | Returns the current directory of the current model. | |
| **@mdlpathr@** | Returns the current directory of the current model with a slash instead of a backslash (“/” instead of “\”). | |
| **@genname@** | Returns the generic name of the current instance. | |
| **@feat_id@** | Returns the feature ID of the currently selected feature | 7400 |
| **@filename@** | Returns the file name. | |


© 2026 INNEO Solutions GmbH

| Text operation | Description | Example |
|---|---|---|
| **@filepath@** | Returns the current directory of the model. | |
| **@fileversion@** | Returns the current version of the file **with** prefixed "." (number following the file suffix e.g.: test.prt.2). | .1 |
| **@mdlno@** | Returns the current version of the file **without** prefixed "." (number following the file suffix e.g.: test.prt.2). | 1 |
| **@mdltype@** | Returns the type of a model. (part, assembly, drawing) | PRT |
| **@selmdln@** | Returns the name of the currently selected model. | |
| **@selmdlpath@** | Returns the path of the currently selected model. | |

## Variables with limited usage

Some variables can only be used in certain GENIUS TOOLS components.

| Text operation | Description | GENIUS TOOLS Module | Example |
|---|---|---|---|
| **@counter@** | Generates a continuous number. | Quick Access, Library, Name Generator | |
| **@number@** | | Quick Access, Library, Name Generator, Multibody to Assembly | |
| **@is_embedded@** | Returns "1" for embedded models, otherwise "0". | Value Transfer | |


© 2026 INNEO Solutions GmbH

| Text operation | Description | GENIUS TOOLS Module | Example |
|---|---|---|---|
| **@oldname@** | | Quick Access, Library, Name Generator, Multibody to Assembly | |

## Drawing information

| Text operation | Description | Example |
|---|---|---|
| **@curdrwmdln@** | Returns the name of the current model of a drawing. | js:alert(replaceVars("@curdrwmdln@.@curdrwmdltype@")); |
| **@curdrwmdltype@** | Returns the type (e. g. PRT) of the current model of a drawing. | js:alert(replaceVars("@curdrwmdln@.@curdrwmdltype@")); |
| **@pageno@** or **@sheetno@** | Output of the current page on the drawing. | |
| **@sheetimagestyleclass="Default" scale@** | Output of the current page base imagestyleclass="Default" scale of the drawing. | 2:1 |
| **@sheetformat@** | Returns the (paper) format of the current drawing (A0-A4 in compliance with DIN 476 or A-F in compliance with ANSI A). | A3 or B |
| **@maxpage@** | Output of the total number of pages in the current drawing. | |

## Parameter information

| Text operation | Description | Example |
|---|---|---|
| **%PARAM%** | Returns the value of a parameter. The input between the percent signs may vary. | %DESIGNATION% |


© 2026 INNEO Solutions GmbH

| Text operation | Description | Example |
|---|---|---|
| **%curmod:PARAM%** | Returns a parameter value of the current model of a drawing into a drawing. The replacement of PARAM may vary.<br>Outside of drawings this notation does not make sense. | % curmod:DRAWINGNUMBER% |
| **%dubase:PARAM%** | Returns a double Value in e^ notation instead of the regular output rounded to 6 decimal places. This can also be used in combination with *curmod* and *curmat*. The order % curmat:dubase:PARAMETER% must be observed. | % dubase:DOUBLE_PARAMETER% |
| **§env-var§** | Outputs an environment variable of Model Processor/from Windows. | §USERNAME§<br>§COMPUTERNAME§<br>§HOMEDRIVE§<br>§LOGONSERVER§<br>§USERDOMAIN§ |

## Text operations for variables

A regular expression[487] cannot contain string operations.

The first character of the count always has position 0.

| Text operation | Description | Example |
|---|---|---|
| **$$sub$x$y$TEXT$$** | Cuts an input out of a text (TEXT) starting with x and ending with y; x and y must be numeric values. Make sure the text is of appropriate length. | $$sub$2$5$@para@$$ |
| **$$pre$x$TEXT$$** | Sub-string from start (character at position 0) till character x. | |
| **$$pree$x$TEXT$$** | Sub-string from start (character at position 0) till character x before the end of the string. | |


© 2026 INNEO Solutions GmbH

| Text operation | Description | Example |
|---|---|---|
| **$$poste$x$TEXT$$** | Sub-string with length x, measured from the end of the string. Is equivalent to the last x characters of the string. | |
| **$$posts$x$TEXT$$** | Sub-string till the end, starting at character x. Thus is equivalent to variable number of output characters. | |
| **$$repl$x$y$TEXT$$** | Replace x with y in the text. TEXT can be any text or another variable. | |

## Comment texts for Inspect

To export tables from *GENIUS TOOLS Inspect* to Excel, the report parameters must be saved as comments in Excel.

For report parameters, the text in the comment must consist of the component abbreviation *gti:* and a keyword:

| Comment text | Column name |
|---|---|
| gti:<columnName> | Output of additional user-defined parameters |
| gti:gtol_bottom_text | Bottom text |
| gti:val_tol | Boundaries |
| gti:tpe_sym | Creo symbol |
| gti:gtol_datum_references | Datum references |
| gti:descr | Description |
| gti:dim_value_text | Dimension text |
| gti:grd | Grid |
| gti:ipc_dim | Inspection dimension |
| gti:gtol_left_text | Left text |
| gti:low_tol | Lower boundary |
| gti:tpe_main | Main type |
| gti:max_dim | Maximal dimension |


© 2026 INNEO Solutions GmbH

| Comment text | Column name |
|---|---|
| gti:min_dim | Minimal dimension |
| gti:nme_sym | Name of symbol |
| gti:bse_dim | Nominal dimension |
| gti:gti_note | Note |
| gti:gti_param | Parameter |
| gti:gtol_right_text | Right Text |
| gti:num_sheet | Sheet |
| gti:src | Source |
| gti:tpe_sub | Subtype |
| gti:num_sym | Symbol number |
| gti:tpd_dim | Theoretically precise dimension |
| gti:cls_tol | Tolerance class |
| gti:tpe_tol | Tolerance standard |
| gti:mod | Tolerance table |
| gti:gtol_value | Tolerance value |
| gti:gtol_top_text | Top text |
| gti:upp_tol | Upper boundary |


© 2026 INNEO Solutions GmbH

# 27 Regular Expressions

Use Regular Expressions in *GENIUS TOOLS* for Creo to check value inputs or allow only rule-compliant inputs to be saved.

| Character | Description |
|---|---|
| \ | Indicates the following character as a special or verbatim character. For example, "n" corresponds to the character "n". "\n" corresponds to a line-break character. The sequence "\\" corresponds to "\", "\(" corresponds to "(". |
| ^ | Corresponds to the beginning of the input. |
| $ | Corresponds to the end of the input. |
| * | Corresponds to the preceding character zero or multiple times. "zo*" matches e.g. "z" or "zoo". |
| + | Corresponds to the proceding character one or multiple times. "zo+" for example matches "zoo", but does not match "z". |
| ? | Corresponds to the proceeding character zero or one time. For example "a?ve?" matches the "ve" in "never". |
| . | Corresponds to all single characters except for a line-break character. |
| (Pattern) | Matches Pattern and saves the equivalent. The compared substring can be retrieved from the resulting matches listing using the elements [0]... [n]. To compare characters in parentheses ( ) use "\(" or "\)". |
| x\|y | Corresponds to either x or y. For example matches "l\|red" either "l" or "red". "(l\|r)ed" matches "led" or "red". |
| {n} | n is a positive integer. Corresponds to exactly n times. "o{2}" for example does not match the "o" in "Robert" but the first two "o"s in "Booooat". |


© 2026 INNEO Solutions GmbH

| Character | Description |
|---|---|
| {n,} | n is a positive integer. Corresponds to at least n times. “o{2}” for example does not match the “o” in “Robert” but all “o”s in “Booooat”. “o{1,}” is equivalent to “o+”. “o{0,}” is equivalent to “o*”. |
| {n,m} | m and n are positive integers. Corresponds to at least n and at most m times. For example “o{1,3}” matches the first three “o”s in “Booooooat”. “o{0,1,}” is equivalent to “o?”. |
| [xyz] | A group of characters. Corresponds to any of the included characters. “[abc]” for example matches the “a” in “falling”. |
| [^xyz] | A group of excluded characters. Corresponds to any character not included. “[^abc]” for example matches the “f” in “falling”. |
| [a-z] | A character range. Corresponds to any character in the specified range. For example, “[a-z]” matches any lowercase alphabetic character in the range from “a” to “z”. |
| [^m-z] | An excluded range of characters. Corresponds to any character not included in the specified range. “[m-z]” for example matches all characters not included in the range from “m” to “z”. |

## Examples

| Regular expression | Description | Example |
|---|---|---|
| [a-zA-Z]*_[a-zA-Z]*-+, | Any alphabetic string with an underscore | user_tbx |
| [0-9]{5} | Five random numbers | 12345 |
| ^.{7}$ | 7 random characters | t_p.prt |
| ^[A-Z]{1}[a-z]{2,10} | A capital letter at the beginning followed by two to ten lower case letters | Tuser |


© 2026 INNEO Solutions GmbH

| Regular expression | Description | Example |
|---|---|---|
| [0-9]{2}\.[0-9]{2}\.[0-9]{4} | German date format | 01.07.1975 |
| [0-9]{4}-[0-9]{2}-[0-9]{2} | Date format ISO 8601 | 1975-07-01 |
| [0-9]{2}/[0-9]{2}/[0-9]{4} | Date format US | 01/07/1975 |
| [0-9]{2}-[a-zA-Z]{3}-[0-9]{4} | English date format | 01-Jul-1975 |
| [a-zA-Z]{3,9}\s[0-9]{1,2},\s[0-9]{4} | English date format (long form) | July 01, 1975 |
| ^(?!ST\|B00\|DEF).* | combstatefilter="^(?!ST\|B00\|DEF).*"<br>All combined states that do not begin with ST, B00, or DEF are used. This filters out the Creo Parametric standard state “Default all”. | Combined states are filtered with:<br>ST*<br>B00*<br>DEF* |
| ^ABC | Starts with ABC | ABCDEF |
| ABC$ | Ends with ABC | UVWXYZ |
| ^. {0,3}$ | String with a maximum of 3 characters | AB |
| ^\d+\.\d+\.\d+$ | Version number | 1.0.3 |
| ^[A-Z] {3}-\d{4}$ | Format | ABC-1234 |
| \b[A-Z] {2}\d{6}\b | Serial number | AB123456 |
| (?<![a-zA-Z0-9])CAD(?![a-zA-Z0-9]) | Exact term (not part of a word) | CAD |
| (?<=\s)Rev\.\s?[A-Z] | “Rev A” or “Rev B” after a space | Version Rev A |


© 2026 INNEO Solutions GmbH

| Regular expression | Description | Example |
|---|---|---|
| \s{2,} | More than one space (for format check) | Scale: 1:100 |
| ^DWG\d{6}$ | Drawing number | DWG123456 |
| ^\d+(.\d+)?\s?(mm | Unit of measurement | cm |
| ^±\d+(\.\d+)?$ | Tolerance specification | ±0.2 |
| ^\d{1,4}x\d{1,4}x\d{1,4}$ | Dimension chain: width x height x depth | 120x50x80 |
| ^PART_[0-9]{3}$ | part | PART_001 |


© 2026 INNEO Solutions GmbH

# 28 Available Arithmetic Operations

You can use arithmetic operations and mathematical functions in different modules of GENIUS TOOLS for Creo.

These operations differ in notation from the default Creo calculations. To use an arithmetic operation in an input field, start with the equal sign, then enter the mathematical function.

**Examples**

```
=12-d4
```

```
=d42/23
```

**Please note**: Units such as degree (in angular dimensions) or length units are ignored. Only the values are used for calculations.

| Mathematical Function | Description | Example |
|---|---|---|
| + | Addition | =d27+5 |
| - | Subtraction | =12-d4 |
| / | Division | =d42/23 |
| * | Multiplication | =d31*3 |
| Math.sqrt(x) | Square root | =Math.sqrt(9) |
| Math.pow(x,y) | Exponentiation (x to the power of y) | =Math.pow(d2,3) |
| Math.abs(x) | Absolute value | =Math.abs(-5) |
| Math.round(x) | Round | =Math.round(2.565) |
| Math.ceil(x) | Round up to the next integer | =Math.ceil(3.6) |
| Math.floor(x) | Round down to the previous integer | =Math.floor(3.4) |


© 2026 INNEO Solutions GmbH

# 29 FAQ

This page contains an overview of frequently asked questions about GENIUS TOOLS for Creo and possible solutions.

GENIUS TOOLS for Creo is the name given to all the extension modules contained in the products GENIUS TOOLS Parameter, GENIUS TOOLS Library, Startup TOOLS or GENIUS TOOLS MBD for ISO GPS.

## General Questions About GENIUS TOOLS for Creo

**Why are not all GENIUS TOOLS for Creo modules displayed in the ribbon menus GENIUS TOOLS / GT MBD?**

- The extension modules are mode-dependent. Check whether you are in the correct Creo mode (e. g. part mode, assembly mode).
- Depending on the GENIUS TOOLS product you have purchased, only the modules contained in it will be displayed. Check whether you have a product license for this module. See product overview.[15]
- Make sure that all required Creo licenses have been loaded. Even if you can open Creo models, the menu ribbons GENIUS TOOLS / GT MBD will only be displayed if the corresponding Creo license is available.
- Some modules in the GENIUS TOOLS ribbon menu require a subscription license. You can find an overview of these modules under License-dependent functions[26]. Check whether you have started Creo with a subscription license for the corresponding GENIUS TOOLS product.

**Why are not all GENIUS TOOLS for Creo Editors displayed in the ribbon menus GENIUS TOOLS / GT MBD?**

Access to the Editors of GENIUS TOOLS modules can be restricted via the configuration settings. Please contact your administrator.

**What happens if the license for GENIUS TOOLS for Creo gets lost?**
If the floating license (e.g. due to Windows hibernation mode) is lost, all GENIUS TOOLS applications and their functions are deactivated. To avoid loss of data all user interfaces remain open. As soon as the license is regained you can continue working as usual. Regaining a license from the network may require some time.

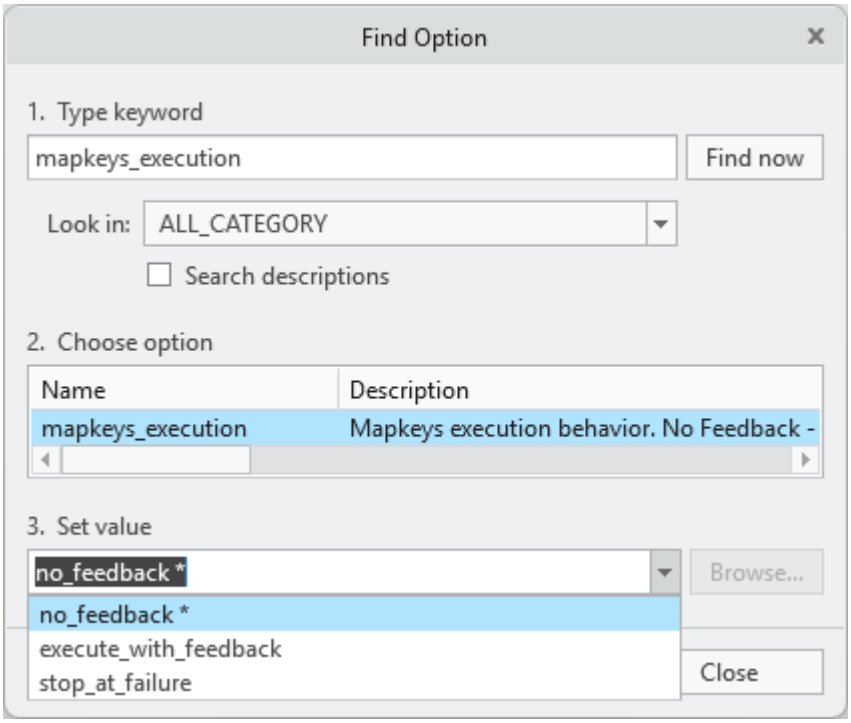
**Excel connection does not work. What can I do?**
Make sure Microsoft Office 2016 or newer is installed. In many cases the Excel connection also works with older Office versions (minimum 2007) but some GENIUS TOOLS functions require Office 2016; therefore proper operation of GENIUS TOOLS for Creo with older Office versions cannot be guaranteed.


© 2026 INNEO Solutions GmbH

If the error message *Failed to initialize Microsoft Excel* is displayed in the Creo log, this can be due to a corrupted Office installation. Try running the Office repair option.

## Questions about error messages

**A function is not being executed. A Mapkey error message appears.**
If a Mapkey error is displayed, open the Configuration Editor and click *Find...* to open the *Find Option* window. Enter *mapkeys_execution* as the keyword and set the value to *no_feedback **.




## Questions about individual GTfC modules

**GENIUS TOOLS Library:**

**Umlauts in my library object status are not displayed. Is there a solution?**
Check the *gt_modules.cfg* and the `gt_headerless_files_are_utf8` configuration options in *gt_main.cfg*. As the status names are specified via configuration options it is possible that the CFG file was not saved correctly and that the umlauts are misinterpreted.

**GENIUS TOOLS Name Generator:**

**My Name Generator database is not used automatically in my Quick Access mapkey. I always have to select the correct name configuration in the selection dialog. What can I do?**
Do you have multiple name configurations with similar file names? The filter rules do not search for uniquely matching names. For instance, if you have two name configurations (*FileName.db* and *FileName_User.db*), and a mapkey with `number:FileName`, the selection dialog will always appear. Ensure that the name configurations are uniquely named. Rename


© 2026 INNEO Solutions GmbH

the name configurations (e.g. *FileName_Global.db* and *FileName_User.db*) and specify the uniquely identifiable name configuration in the mapkey.

This behavior also occurs in environments with local and global name configurations.

**GENIUS TOOLS Parameter:**

**I receive a save / regenerate error; what solutions are available?**
Have you activated the units relation in Creo? Check the Creo configuration option `relations_num_const_units`. Try to specify values with units (10[kg] instead of 10).

Check the info icons in the parameter form and in the tabs and check the status signal light. Pay attention to mandatory parameters!

**Open / create drawing:**

**I receive an error message that a drawing already exists even though it does not open via Open / create drawing.**
This error may occur if drawing models or frames for a drawing are missing. Unfortunately, the repair dialog cannot be opened due to an API error. You have to open the drawing manually.


© 2026 INNEO Solutions GmbH

# 30 Copyright

**Copyright 2026 by:**

INNEO Solutions GmbH

IT-Campus 1

73479 Ellwangen

Germany

This documentation is protected by copyright. All rights reserved.

Without prior written consent of an authorized representative of INNEO Solutions GmbH it must not be copied, photocopied, reproduced, translated, communicated or converted to electronic or machine-readable form in whole or in part.

The unauthorized use of the documentation can lead to a claim for liquidated damages or legal prosecution. INNEO Solutions GmbH does not accept liability for possible faulty information in this documentation and the consequences resulting from it.

**Note on registered trademarks:**
Most of the software, hardware and trade names mentioned in this documentation are also registered trademarks of the respective software manufacturers.

**Registered trademarks and trade names of INNEO Solutions GmbH:**
GENIUS TOOLS, Startup TOOLS, INNEO

# - 3 -

3D Model 104
3D Model compare 306

# - A -

Action 293
Annotation elements
  display 642
  display unused 642
Annotations
  restore original state 649
  transfer 649
Annotations on shape and position tolerances
  edit 719
  templates 719
ASCII file 679
Assembly 303, 673
  complete assembly 303
  standard part with selection table 299
Assembly Report
  configure position attributes 73
  configure table view 71
  define head parameters 71
  define number ranges 74
  define table attributes 72
Auto Placement Filter 192

# - B -

Basic Frame
  edit 717
Batch mode 356

# - C -

Check revision status 535
Close
  all other windows 652
Color selection
  select surfaces by color 737
Color selector 796
Coloring 167
Combined views (=combined states)
  add 170
  display redundancies 642
  edit 178
  sort alphabetically 747
Comment text 202, 918
Common name 42, 466
Component
  assemble automatically 673
Component parameter 656
  create 660
Components
  replace 133
  suppress 133
Configuration
  view 763
Configuration concept 756
Configuration layers 756
  overwrite 774
Configuration Options 799
  environment variables 781
  paths 780
  save 775
  save as CFG file 763
  setting values 772
  view 763
Configuration overview 768
Contact surfaces
  find 709
  select 734
Creo commands in the user interface
  hide/disable 655
Creo object types
  search 304
Creo special characters 194, 248

# - D -

Databases
  default texts 676
  multilingual 320, 676
  update 782
Dimension 99, 707, 742
DIN 6770 187
Directory 751
Drawing
  create 732
  open 732
Drawing symbols
  group 215, 267

# - E -

Editor not displayed 925
Embedded data 798
Environment variables 760, 775, 781


© 2026 INNEO Solutions GmbH

ERP system 320
Excel 41, 149, 187, 625, 642, 686, 689
  export 202
Excel connection does not work 925
Export
  3D Model 104
Export templates 108

## - F -

Family table 601
  add dimensions 99
  import 374
Family table variants
  name assignment 440
Form 124, 131
Forms
  use 315
Function Manager
  setting properties to be editable 180
Functional Design 179
Functions
  add 171
  color 175
  delete 169
  edit 169
  integrate PDFs / web links 170
  manage 167, 169
  rename 170

## - G -

General Configuration Options
  Assembly Report 811
  Dimension 816
  Export TDP 818
  Forms 820
  Function Manager 823
  Inspect 825
  Library 830
  Material 849
  MBD Tables 853
  Name Generator 854
  Orientation Manager 855
  Parameter 855
  Quick Access 863
  Stack Note 867
  Stack Note 3D 869
  UDF Forms 871
  Utilities 873
GENIUS TOOLS Inspect Revision 231
GENIUS TOOLS Library 15
GENIUS TOOLS Model Processor 455
GENIUS TOOLS Parameter 15
GENIUS TOOLS Startup TOOLS 780
  environment variables 760
  working directory 760
Global name configuration 447
Glossary
  GENIUS TOOLS Assembly Report 42
  GENIUS TOOLS Forms 124
  GENIUS TOOLS Inspect 188, 243
  GENIUS TOOLS Inspect 3D Revision 281
  GENIUS TOOLS Inspect Revision 231
  GENIUS TOOLS Library 294
  GENIUS TOOLS Parameter 466
  GENIUS TOOLS UDF Forms 601
  Inspect 188
  Inspect 3D 243
GT MBD
  activate 30

## - I -

Information overview 741
Inspection symbols 187, 188, 243
  filter 198, 251
  hide/unhide 201, 253
  insert, automatic 192, 247
  insert, free 192
  link 192, 198, 247, 251
  number 196, 250
  place 192, 247
  position 192, 247
  sort 198, 251

## - J -

JavaScript 63, 154, 535, 619, 783

## - L -

LaTeX 108
Library 293
  create 367
  optimize search 379
  read in additional information 379
Library category
  create 368
Library objects
  create 369


© 2026 INNEO Solutions GmbH

Library objects
update 377
License 26
License lost 925
Local name configuration 447

## - M -

Mapkey 123, 330, 556
intelligent 440, 454, 455, 544
Mass
assign 483
calculate 483
Material
Material version check 408
Materials
assign 394
check 408
check in a model 399
convert 663
delete 398
exclude material from update 399
filter material properties 392
material directory 403
material properties 392
search 393
update 399
value 392
version 408
MBD Tables
edit 433
sizes 422
surface finishes 429
Military Standard 179
MNU export 359
Model
Variants 131
Model Based Definition 167
Model properties
change 131
Module not displayed 925
MTL files 410
MUI files
import 386

## - N -

Name configuration
global 447
local 447
Name Generator
Fallback 450
Integration in Quick Access 454
Mapkey 454
Windchill 452
Naming 440
Notes
edit multiple notes 636
with calculations 638
Number ranges 41, 65, 66

## - O -

Objects
back up 712
Object collector 360
Open / create drawing
Exisiting drawing is not opened 925
Overwrite
configuration layers 774

## - P -

Parameter 41, 535
add 498
create table parameters 58
parameter definitions 494
parameter management concept 467
reading parameter values from text files 476
use of variable parameters 601
Parameter values
copy in symbols 224, 276
link in symbols 224, 276
Parts list
bulk bill of materials 41
multi-level bill of materials 41
removing single lines 83
single-level bill of materials 41
PDM system 320
Pitch
show 742
Place
exceptions 194, 248
PLM system 320
Points
export 679
Position number
position number assignment 41
Position PreSave rule 65
Position values from Windchill 80
programm editors
access 781


© 2026 INNEO Solutions GmbH

PTS file 679

## - Q -

QAEX file 560
Quick Access
mode dependency 547
selection dependency 548
Quick Access Editor
mapkey definition 556

## - R -

Regular Expressions 194, 248, 487, 920
Relations
extend 702
Report definition 42
automatic update 79
create 69
external 43
internal 43
number 69
number with different number ranges 75
Report table 52
Reports 42
create 690
CSV 690
Excel 690
export 41
PDF 690
REST API 507
Revision 231, 281
Revision parameter 238, 288
Ribbon menu GT MBD
activate 30
Rounding according to IEEE 754 702

## - S -

Save
configuration options 775
objects from older Creo versions in currently used version 730
Search.pro 665
Setting position numbers
Windchill 52, 65
Shortcut 544
Snapshot
create 235, 285
Start mechanism 758
start switches 781
Startup TOOLS 15
environment variables 760
working directory 760
Status dialog 476
Status indicator 476
Subscription license 26
SYM file 749
Symbol definition attributes 224, 276
Symbol variants
toggle 749
Symbols
switch between all variants of a grouped symbol 749

## - T -

Tables
export 202, 254, 642
export a placed drawing table 687
export a report table 689
export One-to-One 686
export without template 686
fill and export via template 687
place 192
tolerance table 669
update 439
Tables in 3D 422
Tables MBD
edit 433
sizes 422
surface finishes 429
TDP 104
Technical Data Package 104
Telemetry 765
Template 41, 231, 303, 304, 684, 701
create template for export 693
Tolerance table
create 669
Tooltip 545

## - U -

UDF
as layer 617
place dependently 607
UDF Forms
use 315
UDF groups
create 601
edit subsequently 601


© 2026 INNEO Solutions GmbH

Umlauts (Library) 925

# - V -

Value Transfer 594
Value-dependent unit specifications on drawings 535
Variables 913
date and time information 913
drawing information 916
object information 914
parameter information 916
text operations 917
with limited usage 915

# - W -

Windchill 60, 238, 288, 293, 320, 410, 448, 466, 507, 732
Using position values from Windchill 52, 65, 80
Working directory 751, 760

# - Z -

Zoom 563


© 2026 INNEO Solutions GmbH