ALPANA

[code192]

Alpana

2020R1 — Last update: Aug 26, 2020

Code 192

Copyright © 2020 Code 192

Table of Contents

1. About Alpana 13
1.1. Components and Architecture of Alpana 14
1.2. Getting Started 18
1.3. License Portal and Installers Download 20
1.4. Release Notes 21
1.4.1. v3.0.1.x 22
1.4.2. v3.0.2.x 28
1.4.3. v3.1.0 30
1.4.4. v3.1.2 32
1.4.5. v3.2.1 34
1.4.6. v3.2.2 37
1.4.7. v3.2.3 39
1.4.8. 2020 R1 40
1.5. Upgrading 48
1.5.1. License incompatibility for Major upgrades 49
1.5.2. General upgrade procedure for v3 51
1.5.3. Upgrading from an older v2 to v3.0 54
1.5.4. Upgrading from an older v1 version 59

2. Alpana Designer 60
2.1. Installation 61
2.1.1. Pre-Requisites 62
2.1.2. Installing Alpana Designer 64
2.1.3. Anti-virus and Firewall set up 68
2.1.4. Upgrading Alpana Designer 69
2.1.5. Uninstalling Alpana Designer 70
2.2. Troubleshooting 71
2.2.1. Finding error logs 72
2.3. Connecting to Data 74
2.3.1. Alpana data model 75
2.3.1.1. Note on data storage 77
2.3.1.2. Redundant Connections 78
2.3.2. The data Buffer 83
2.3.2.1. Bufferized Connection types : What gets bufferized ? 84
2.3.2.2. Data Flow : How bufferization happens ? 85
2.3.2.3. Buffer database setup 87
2.3.2.4. Buffer Refresh settings 88
2.3.2.5. refreshing file-based connections 90
2.3.3. Creating and Managing Connections 91
2.3.3.1. Create a Connection 92
2.3.3.2. Edit a Connection 93
2.3.3.3. Delete a Connection 94
2.3.4. Connection types 95

2.3.4.1. SQL Server ... 96
2.3.4.2. Excel ... 100
2.3.4.3. CSV ... 103
2.3.4.4. Wonderware Historian ... 105
2.3.4.5. Web ... 106
2.3.4.6. Oracle ... 111
2.3.4.7. Alpana Runtime ... 114
2.4. Wonderware Historian ... 117
2.4.1. Connecting to Wonderware Historian ... 118
2.4.1.1. Version compatibility ... 119
2.4.1.2. Important pre-requisite ... 120
2.4.1.3. Selecting Connection Type ... 121
2.4.1.4. Login to a Wonderware Historian server ... 122
2.4.1.5. Test connection ... 124
2.4.1.6. Data refresh and Buffer ... 126
2.4.2. Creating Queries to Wonderware Historian ... 127
2.4.2.1. Picking Tags ... 128
2.4.2.1.1. Browsing for Tags ... 129
2.4.2.1.2. Selecting Tags ... 131
2.4.2.2. Configuring the Historian Query ... 133
2.4.2.2.1. Going to Query Configuration ... 134
2.4.2.2.2. Query Type ... 135
2.4.2.2.3. Time Settings ... 136
2.4.2.2.3.1. Static Time Settings ... 137
2.4.2.2.3.2. Parameter Based Settings ... 140
2.4.2.2.4. History Values : query parameters ... 142
2.4.2.2.5. History Values: query Format ... 146
2.4.2.2.6. State Statistics : query parameters ... 147
2.4.2.3. Renaming the Query Table ... 149
2.4.3. Wonderware Historian query result ... 150
2.4.3.1. Connection Schema ... 151
2.4.3.2. Important : data limits ... 153
2.5. Aveva Insight ... 154
2.5.1. Connecting to Insight ... 155
2.5.1.1. Selecting Connection Type ... 156
2.5.1.2. Login to Aveva Cloud Operations ... 157
2.5.1.3. Test connection ... 160
2.5.1.4. Data refresh and Buffer ... 161
2.5.2. Querying Insight ... 162
2.5.2.1. Picking Tags ... 163
2.5.2.1.1. Browsing for Tags ... 164
2.5.2.1.2. Selecting Tags ... 166
2.5.2.2. Configuring the query ... 167
2.5.2.2.1. Going to Query Configuration ... 168
2.5.2.2.2. Query Type ... 169

2.5.2.2.3. Time Settings 170
2.5.2.2.3.1. Manual Time Settings 171
2.5.2.2.3.2. Parameter Based Settings 173
2.5.2.2.4. Process Values : query parameters 175
2.5.2.3. Renaming the Query Table 177
2.5.2.4. Configuring the query result 178
2.6. Transforming Data 180
2.6.1. Introduction 181
2.6.1.1. Learning Pre-Requisites 182
2.6.1.2. What are Data Sources 183
2.6.1.3. Content : Data Preparation Activities 184
2.6.2. Creating and Managing Data Sources 185
2.6.2.1. Create a Data Source 186
2.6.2.2. Browsing Data Sources to Edit 187
2.6.2.3. Rename a Data Source 188
2.6.2.4. Delete a Data Source 189
2.6.2.5. Duplicate a Data Source 190
2.6.3. Export / Import Data Sources 191
2.6.3.1. What is a Data Source Export 192
2.6.3.2. Export a Data Source to a file 193
2.6.3.3. Import a Data Source from a file 194
2.6.3.4. Publish a Data Source to Alpana Server 196
2.6.3.5. Import a Data Source from Alpana Server 197
2.6.4. Fetching Data 198
2.6.4.1. using the Data Preview 199
2.6.4.2. getting and browsing the Connection schema 200
2.6.4.3. fetching data from a data Table 203
2.6.4.4. fetching data from a data View 204
2.6.4.5. fetching data from a Stored Procedure 205
2.6.4.6. fetching data from a Table-Valued Function 210
2.6.4.7. Removing a table from the Data Source 214
2.6.5. Custom Query Tables 215
2.6.5.1. Language 216
2.6.5.2. Creating a Custom Query Table 217
2.6.5.3. Updating a Custom Query Table 219
2.6.5.4. Using Parameters 220
2.6.6. Merging Data 221
2.6.6.1. Creating Joins 222
2.6.6.2. Configuring Joins 223
2.6.6.3. Updating Joins 228
2.6.7. Filtering Data 229
2.6.7.1. Opening the Filter Editor 230
2.6.7.2. Managing Filter Conditions 231
2.6.7.3. Configuring Filter Conditions 232
2.6.7.3.1. Filtering string fields 233

2.6.7.3.2. Filtering number fields 235
2.6.7.3.3. Filtering date fields 238
2.6.7.4. Limiting dataset size 242
2.6.8. Calculating Expressions 243
2.6.8.1. Opening Expression Editor 244
2.6.8.2. Managing Expressions 245
2.6.8.3. Expression properties 246
2.6.8.4. Adding a constant 247
2.6.8.5. Adding a field 248
2.6.8.6. Adding a function 251
2.6.8.7. Removing an existing item 254
2.6.8.8. Copying and Pasting an existing item 255
2.6.8.9. Aggregated Expressions 256
2.6.9. Expressions Troubleshooting 259
2.6.9.1. Pitfall : Incorrect result in Expression with division 260
2.6.9.2. Pitfall : Calculating ratios, averages, ... 262
2.6.9.3. Error when mixing aggregate fields and non-aggregate fields 265
2.6.9.4. Error when wrapping aggregate fields inside aggregate functions 266
2.6.10. Value-based Settings 267
2.6.10.1. Value-based Colors 268
2.6.11. Formatting Columns 270
2.6.11.1. Renaming Columns 271
2.6.11.2. Changing Column Type 272
2.6.11.3. Hiding Column 273
2.7. Configuring a Widget 275
2.7.1. Creating and Managing Widgets 276
2.7.1.1. Adding a Widget 277
2.7.1.2. Editting a Widget 281
2.7.1.3. Renaming a Widget 283
2.7.1.4. Deleting a Widget 284
2.7.1.5. Copying/Pasting a Widget 285
2.7.2. Export / Import Widgets 286
2.7.2.1. What is a Widget Export 287
2.7.2.2. Export a Widget to a file 288
2.7.2.3. Import a Widget from a file 289
2.7.2.4. Publish a Widget to Alpana Server 290
2.7.2.5. Import a Widget from Alpana Server 291
2.7.3. Widget Data Binding 293
2.7.3.1. using the Widget Preview 295
2.7.3.2. Troubleshooting Widget data 296
2.7.3.3. Editing Widget data binding 297
2.7.3.4. Selecting a Data Source 299
2.7.3.5. Configuring the aggregation of number fields 300
2.7.3.6. Configure the grouping of category fields 303
2.7.3.7. Binding a data field 305

2.7.3.8. Un-binding a data field ... 307
2.7.3.9. Re-Ordering data fields ... 308
2.7.3.10. Binding to Real-Time Data ... 310
2.7.4. Configure a sorting order ... 312
2.7.5. Widget-level Filtering ... 313
2.7.5.1. Opening the Filter Editor ... 314
2.7.5.2. Managing Filter Conditions ... 315
2.7.5.3. Filtering aggregated values ... 316
2.7.5.4. Configuring Filter Conditions ... 319
2.7.5.4.1. Filtering string fields ... 320
2.7.5.4.2. Filtering number fields ... 322
2.7.5.4.3. Filtering date fields ... 325
2.7.5.5. Limiting dataset size ... 328
2.7.5.6. Getting Distinct Values ... 329
2.7.6. General Widget Properties ... 331
2.7.6.1. Editing Widget Properties ... 332
2.7.6.2. Name, Header and Description ... 333
2.7.6.3. Container Appearance ... 337
2.7.6.4. Widget Menu items ... 339
2.7.6.5. Measure Formatting and Date Formatting ... 341
2.7.6.6. Multi-items Viewport ... 346
2.7.6.7. Animation ... 350
2.7.7. Interactions ... 351
2.7.7.1. Drill-Down ... 352
2.7.7.2. Link To URL ... 353
2.7.7.3. Master Filter ... 357
2.8. Widget Properties Reference ... 358
2.8.1. Cartesian Chart ... 359
2.8.1.1. Binding Data ... 360
2.8.1.2. Chart Types ... 375
2.8.1.3. Switching Chart Types ... 377
2.8.1.4. Combining Chart Types ... 378
2.8.1.5. Series Settings ... 379
2.8.1.5.1. Series Name ... 380
2.8.1.5.2. Axis binding ... 381
2.8.1.5.3. Data Point information ... 382
2.8.1.5.4. Style and Appearance ... 384
2.8.1.6. Axes ... 388
2.8.1.6.1. X Axis ... 389
2.8.1.6.1.1. Strip Line ... 397
2.8.1.6.1.2. Paging / Scrolling ... 402
2.8.1.6.2. Y Axes ... 403
2.8.1.6.2.1. Strip Line ... 408
2.8.1.7. Legend ... 412
2.8.1.8. Managing Series Color ... 415

2.8.1.8.1. Chart value color binding ... 418
2.8.1.9. Trend Lines ... 422
2.8.2. Polar Chart ... 423
2.8.2.1. Binding Data ... 424
2.8.2.2. Chart Types / Draw Types ... 427
2.8.2.3. Switching Chart Types ... 429
2.8.2.4. Series Settings ... 430
2.8.2.4.1. Series Name ... 431
2.8.2.4.2. Style and Appearance ... 432
2.8.2.5. X Axis ... 434
2.8.2.6. Legend ... 435
2.8.3. Map ... 437
2.8.3.1. Maps format : GeoJSON ... 438
2.8.3.2. Choosing a map ... 440
2.8.3.3. Binding Data ... 443
2.8.3.4. Managing Colors ... 447
2.8.3.5. Data Point information ... 450
2.8.3.6. Runtime navigation ... 451
2.8.4. Grid ... 452
2.8.4.1. Binding Data ... 453
2.8.4.2. Layout and appearance ... 458
2.8.4.3. User interaction ... 462
2.8.4.4. Column names ... 467
2.8.4.5. Value display ... 471
2.8.4.6. Conditional Formatting ... 472
2.8.4.7. Grid KPI ... 477
2.8.4.8. Footer ... 481
2.8.4.9. HTML in the Grid ... 483
2.8.5. TreeMap ... 485
2.8.5.1. Binding Data ... 486
2.8.5.2. Managing Colors ... 495
2.8.5.3. Data Point information ... 496
2.8.6. Circular Gauge ... 497
2.8.6.1. Binding Data ... 498
2.8.6.2. Value Display ... 503
2.8.6.3. Style and Appearance ... 507
2.8.7. Linear Gauge ... 512
2.8.7.1. Data Binding ... 513
2.8.7.2. Value display and Limits configuration ... 516
2.8.7.3. Style and Appearance ... 518
2.8.8. Card ... 519
2.8.8.1. Binding Data ... 520
2.8.8.2. Value Display ... 525
2.8.8.3. Card Sparkline ... 533
2.8.9. Sparkline ... 536

2.8.9.1. Binding Data 537
2.8.9.2. Configuring Layout 544
2.8.9.3. Sparkline chart style 553
2.8.9.4. Style and Appearance 557
2.8.10. Banner 559
2.8.10.1. Binding Data 560
2.8.10.2. Style and Appearance 564
2.8.11. Proportion Charts 566
2.8.11.1. Binding Data 567
2.8.11.2. Switching Chart type 581
2.8.11.3. Data Point information 583
2.8.11.4. Legend 586
2.8.11.5. Managing Series Color 590
2.8.11.6. Interaction 594
2.8.11.7. Style and Appearance 595
2.8.12. Gantt 597
2.8.12.1. Binding Data 598
2.8.12.2. Data Point Information 603
2.8.12.3. Events Configuration 604
2.8.12.4. Axis Configuration 609
2.8.12.5. Titles Configuration 613
2.8.13. Sankey 615
2.8.13.1. Binding Data 616
2.8.13.2. Data Point information 620
2.8.14. Energy Rating 621
2.8.14.1. Binding Data 622
2.8.14.2. Grading Calculation and Display 627
2.8.14.3. Titles and Labels 630
2.8.15. Filter Widgets 632
2.8.15.1. ComboBox 633
2.8.15.1.1. Binding Data 634
2.8.15.1.2. Configuring behavior 635
2.8.15.1.3. Style and Appearance 637
2.8.15.2. CheckBox 638
2.8.15.2.1. Binding Data 639
2.8.15.2.2. Configuring behaviour 640
2.8.15.2.3. Style and Appearance 641
2.8.15.3. RadioButton 642
2.8.15.3.1. Binding Data 643
2.8.15.3.2. Configuring behaviour 644
2.8.15.3.3. Style and Appearance 645
2.8.15.4. RangeSlider 646
2.8.15.4.1. Binding Data 647
2.8.15.4.2. Configuring behaviour 649
2.8.15.4.3. Configuring style 650

2.8.15.5. DatePicker ... 652
2.8.15.5.1. Configuring without data ... 653
2.8.15.5.2. Binding Data ... 654
2.8.15.5.3. Configuring behaviour ... 656
2.8.15.5.4. Configuring style ... 659
2.8.15.6. RangeNavigator ... 660
2.8.15.6.1. Binding Data ... 661
2.8.15.6.2. Configuring behaviour ... 662
2.8.15.6.3. Configuring style ... 663
2.8.15.7. TreeView ... 664
2.8.15.7.1. Binding Data ... 665
2.8.15.7.2. Configuring Behaviour ... 667
2.8.16. Static Widgets ... 668
2.8.16.1. Image ... 669
2.8.16.1.1. bind image ... 670
2.8.16.1.2. Image Modes ... 671
2.8.16.2. Label ... 676
2.8.16.2.1. Configuration ... 677
2.8.16.3. Embed ... 678
2.8.16.3.1. Configuration ... 679
2.8.16.3.2. Note : Cross-origin resource sharing ... 681
2.8.16.4. PDF viewer ... 682
2.8.16.4.1. Configuration ... 683
2.8.17. Input Grid ... 686
2.8.17.1. Configuration ... 687
2.9. Using Dashboard Parameters ... 691
2.9.1. What is a Parameter ? ... 692
2.9.2. Managing Parameters ... 693
2.9.2.1. Opening the Parameters Editor ... 694
2.9.2.2. Visualizing Parameter binding ... 697
2.9.2.3. Editing Parameter bindings ... 700
2.9.2.3.1. Necessary connections ... 701
2.9.2.3.2. Removing a connection ... 702
2.9.2.3.3. Adding a connection ... 705
2.9.2.4. Adding/Removing Parameter ... 709
2.9.2.5. Edit Parameter properties ... 710
2.9.2.5.1. Parameter Name ... 711
2.9.2.5.2. Value Type ... 712
2.9.2.5.3. Parameter Default Value ... 713
2.9.2.6. example : Multi-Data Source filtering ... 716
2.9.3. Widget as Master Filter ... 717
2.9.3.1. Configuring a Widget as Master ... 718
2.9.3.2. Setting a Widget Default Selection ... 720
2.9.3.3. Using filters at runtime ... 722
2.9.4. URL Parameters to receive values ... 725

2.9.4.1. Example 726
2.9.4.2. Reserved words 727
2.9.5. Parameter usage 728
2.9.5.1. Label Parameters 729
2.9.5.2. Dynamic calls to Stored Procedures or Table-Valued Functions 730
2.9.5.3. Dynamic calls to Custom Queries 731
2.9.5.4. Dynamic Expression Values 732
2.9.5.5. Dynamic Data Source filters conditions 733
2.9.5.6. Dynamic Widget filters conditions 734
2.9.5.7. Dynamic Historian Queries 735
2.9.5.8. Include Parameter value in a Link to URL 736
2.9.6. Widget-specific binding 737
2.9.6.1. Grid conditional formatting 738
2.9.6.2. Dynamically updating embedded content 739
2.10. Dashboard Layout 740
2.10.1. Managing Layouts 741
2.10.2. Dashboard Theme and Background 743
2.10.3. Layout size and Responsiveness 747
2.10.4. Configuring Widgets on the Layout 749
2.10.5. Layouts for PDF Exports 753
2.10.6. Layouts for Mobile 756
2.11. Dashboard Settings and Publication 758
2.11.1. Dashboard Preview 759
2.11.2. Create and Export / Import Dashboards 760
2.11.2.1. Creating a New Dashboard 761
2.11.2.2. What is a Dashboard file 762
2.11.2.3. Save a Dashboard as file 763
2.11.2.4. Open a Dashboard file 764
2.11.2.5. Publishing a Dashboard to Alpana Server 766
2.11.2.6. Importing a Dashboard from Alpana Server 768
2.11.3. Dashboard-level Settings 770
2.11.3.1. Dashboard Title 771
2.11.3.2. Widget Auto-refresh 772
2.11.3.3. Disabling Comments 774
2.11.3.4. Widget Headers size 775
2.11.4. Designer-level Settings 776
2.11.4.1. Login to Alpana Server 777

3. Alpana Server 779
3.1. About Alpana server 780
3.1.1. Architecture 781
3.1.2. Multitenancy 784
3.2. Installation 786
3.2.1. Pre-Requisites 787
3.2.2. Installing 792
3.2.3. Anti-virus and Firewall set up 801

3.2.4. Post-Installation ... 802
3.2.5. Un-Installing ... 811
3.3. Login and User Profile ... 812
3.3.1. Selecting a Tenant or the Host ... 813
3.3.2. Login options ... 815
3.3.3. User Settings and Logout ... 819
3.3.4. Linked Accounts ... 820
3.3.5. Account details and password ... 822
3.3.6. Account Security and Privacy ... 825
3.3.7. Notifications ... 826
3.3.8. Chat ... 828
3.4. Dashboards ... 832
3.4.1. Categories ... 833
3.4.2. Browsing the Dashboards ... 836
3.4.3. Viewing Dashboards ... 837
3.4.3.1. Opening Dashboards for Viewing ... 838
3.4.3.2. Favorite Dashboards ... 840
3.4.3.3. Refreshing Dashboards ... 841
3.4.3.4. Manual Exports ... 842
3.4.3.5. Dashboard Views ... 845
3.4.3.6. Comments ... 851
3.4.3.7. Pinning Widgets ... 856
3.4.4. Managing / Publishing Dashboards ... 857
3.4.5. Dashboard Version History ... 862
3.4.6. Public / Private Dashboards and Dashboard-level Permissions ... 864
3.4.7. Linking / Embedding Dashboards ... 867
3.5. Widgets ... 869
3.5.1. Viewing Widgets ... 870
3.5.2. Managing / Publishing Widgets ... 872
3.5.3. Widget Version History ... 875
3.5.4. Linking / Embedding Widgets ... 877
3.6. Data Sources ... 879
3.6.1. Managing / Publishing Data Sources ... 880
3.6.2. Data Sources Version History ... 883
3.7. Schedule Management ... 885
3.7.1. Schedules management ... 886
3.7.2. Creating a Schedule ... 888
3.7.2.1. Main ... 889
3.7.2.2. Trigger ... 891
3.7.2.2.1. Timer ... 892
3.7.2.2.2. Data condition ... 897
3.7.2.2.2.1. Managing conditions ... 898
3.7.2.2.2.2. Logical condition ... 900
3.7.2.2.2.3. Field value watcher ... 904
3.7.2.3. Actions ... 908

3.7.2.4. Targets ... 910
3.8. Homepages and Pinning Widgets ... 912
3.8.1. Viewing Homepages ... 913
3.8.2. Creating Homepages ... 915
3.8.3. Pinning Widgets ... 918
3.8.4. Editing the Homepage Layout ... 920
3.9. Runtime Connections ... 922
3.9.1. Folders ... 923
3.9.2. Browsing the Runtime Connections ... 926
3.9.3. Creating a Runtime Connection table schema ... 927
3.9.4. Inputting and Managing data ... 936
3.9.5. Runtime Connections Management ... 938
3.9.6. Managing Permissions of Runtime Connections ... 941
3.10. Administration ... 943
3.10.1. License Management ... 944
3.10.1.1. License Activation ... 945
3.10.1.2. Host-level Features ... 949
3.10.1.3. License Deactivation ... 950
3.10.2. Tenant Management ... 952
3.10.2.1. Browsing the Tenants ... 953
3.10.2.2. Creating a Tenant ... 956
3.10.2.3. Tenant Features ... 958
3.10.3. Users, Roles and Permissions ... 959
3.10.3.1. Permissions descriptions ... 960
3.10.3.2. Roles ... 962
3.10.3.3. Users ... 967
3.10.3.4. User Lockout ... 974
3.10.3.5. User Impersonation ... 978
3.10.3.6. User Settings ... 980
3.10.4. Theme and Appearance ... 984
3.10.4.1. How to Use a Custom CSS ... 987
3.10.5. Host-level Settings ... 992
3.10.5.1. Email (SMTP) ... 993
3.10.5.2. API Origins Whitelist ... 996
3.10.5.3. External Login ... 998
3.10.5.4. Concurrent Login ... 1002
3.10.6. Background Jobs ... 1003
3.11. API ... 1004
3.11.1. General API help and troubleshooting ... 1005
3.11.2. Authentication to the API ... 1007
3.11.3. Remote Actions ... 1009
3.11.3.1. Refreshing / Updating Filters ... 1010
3.11.3.2. Real Time Widgets Data ... 1012
3.11.4. Client-Side Hooks ... 1014
3.12. Maintenance and Migration ... 1015

3.12.1. Start / Stop IIS websites ... 1016
3.12.2. Backup ... 1018
3.12.3. Migration ... 1020
3.13. Troubleshooting ... 1022
3.13.1. Installation Troubleshooting ... 1023
3.13.2. Dashboard error logs ... 1025
3.13.3. Server usage logs ... 1027
3.13.4. Website Troubleshooting ... 1031
3.13.5. License troubleshooting ... 1033
3.13.6. Accessing behind a router ... 1035

4. Alpana Mobile ... 1038
4.1. Installation ... 1039
4.1.1. How to get Alpana Mobile ... 1040
4.1.2. Setup ... 1041
4.2. Usage ... 1042
4.2.1. Login ... 1043
4.2.2. Tenancy ... 1044
4.2.3. Dashboard Page ... 1045

5. Alpana Update File Dependency ... 1046
5.1. Installation ... 1047
5.1.1. Pre-Requisistes ... 1048
5.1.2. Installing ... 1049
5.2. Running the application ... 1052
5.3. Settings ... 1053
5.4. Connecting To Alpana ... 1055
5.5. Jobs ... 1057
5.5.1. Creating a Job ... 1058
5.5.2. Configuring a Job ... 1059
5.5.2.1. Target Dependency ... 1060
5.5.2.2. Source File ... 1062
5.5.2.3. Update Mode ... 1063
5.5.3. Running jobs ... 1065
5.5.4. Deleting a Job ... 1066
5.6. Troubleshooting ... 1067

6. Alpana Real Time Data ... 1068
6.1. Installation ... 1069
6.1.1. Pre-Requisistes ... 1070
6.1.2. Installing ... 1071
6.2. Running the application ... 1074
6.3. Settings ... 1075
6.4. Connecting To Alpana ... 1077
6.5. Jobs ... 1079
6.5.1. Creating a Job ... 1080
6.5.2. Configuring a Job ... 1081

6.5.2.1. Target Widget 1082
6.5.2.2. Data Source 1083
6.5.3. Running jobs 1087
6.5.4. Deleting a Job 1088
6.6. Troubleshooting 1089

7. Alpana OMI App 1090
7.1. Pre-Requisites 1091
7.2. Importing 1092
7.2.1. Importing in WSP 2017 U3 1093
7.2.2. Importing in WSP 2020 1095
7.3. Adding to a Layout 1096
7.4. Configuring 1098
7.4.1. Connecting to Alpana 1099
7.4.2. Selecting Visuals 1101
7.4.3. Configuring interactions 1103

1. About Alpana

Let your Data turn into Insights and put the power into your hands to take Data driven decisions !

Alpana Platform provides real-time access to your data and lets users analyze complex information through graphical representations that are easy to understand.

Alpana is published by [code192]

Last modified: Jun 13, 2019

1.1. Components and Architecture of Alpana

Installable components

Alpana Platform is composed of :

- Alpana Designer : the development tool for creating Data Sources, Widgets, Dashboards
- Alpana Server : the website deployed for sharing content
- Alpana Mobile : the mobile app used for viewing on mobile devices
- Alpana Update File Dependency : the external tool for the live update of file-based connections.

An external tool may also be used for demonstration purposes, but is provided on-demand and with no warranty :

- Alpana Real Time Data : an external tool to demonstrate realtime features of the API

Roles of the Components and Users

Alpana Designer for Creating content

Alpana Designer is a desktop application that allows to connect to data and create the visualizations.

“Developer” users create content using Alpana Designer : Widgets, Dashboards, Data Sources.

> In practice, Alpana Designer is very easy to use, and most *“Business” users* can also be “Developers” with very minimal training.

Alpana Server for Sharing content

Alpana Server is a website application that allows to make use of the content : display and interact with visualizations, export, notify, socialize.

“Business” users connect to content on Alpana Server using either :

- a desktop web browser
- the **Alpana Mobile** app for mobile devices

Each time the content is displayed, it is updated with live queries to your data.

Developer users share/publish content on Alpana Server for Business users to consume.

“Administrator” users manage the content and access on Alpana Server.

> In practice, this is only a matter of permissions, and any person can be both Developer, Administrator and Business user with minimal or no training.

Alpana Update File Dependency for updating file content

Developer users deploy **Alpana Update File Dependency** if they need to display file-based data inside their content, and if they need this data to be updated.

Where to install ?

Alpana Server

To choose where to install **Alpana Server**, remember that it's a website that must :

- be accessible by the Business users from desktop or mobile
- be able to access your data to make fast queries
- be able to access its own local resources appropriately (local database, CPU/disk/RAM, …)

See also the software and hardware pre-requisites.

Alpana Designer

Alpana Designer doesn't have to be installed on the same machine as Alpana Server.
Depending on your license, you can make several installations of Alpana Designer.
Keep in mind that for developing :

- you must have access to your data from the machine where Alpana Designer is installed
- if you have access to Alpana Server from this machine, you can publish directly. If not, you can still publish manually.

See also the pre-requisites.

Alpana Update File Dependency

- doesn't need to be installed if you don't need to publish file-based live data
- must have the required file system access to read the files you need to update
- must have the required network access to reach Alpana Server
- can be installed on several machines, for example if you need to read several files from different machines

Alpana security model

Overview

Alpana Server is offered as an on-premise installation that is administered by the customer (code192 does not offer it as a service).
Alpana Server is a website hosted on Microsoft IIS where users can login with different methods.
The application is developed using the latest web technologies, always maintained up-to-date for security reasons.
The application model, including user login information, is stored inside a Microsoft SQL database, and the customer is responsible for configuring the security of this database.
Information for the Alpana process to login to the SQL database is stored in the file system in the installation folder, and the customer is responsible for securing this folder.
Process data is not hosted by Alpana. Alpana only configures queries to the source data stores.
Configured Alpana objects (Dashboards, Widgets, Data Sources) are stored in the SQL database and secured at the database level.

Transport layer security

Access to the website can be configured through HTTPS in order to ensure security at the transport layer.

Login security

The security of user access to visualizations is managed through the login system and the internal permission system.
Login methods to Alpana Server include : – user accounts fully hosted by Alpana – Active Directory accounts imported through LDAP – Active Directory accounts authenticated through ADFS (SSO) – other SSO methods

Users can access visualizations through multiple entry points, all secured in the same way :

- desktop web browser
- native mobile app
- Alpana Designer
- web API
- native app for OMI
- embedded inside another tool

User administration

Administrators have several tools to help with security :

- They can enforce password complexity rules.
- They can force users to activate their account through e-mail.

- They can impersonate users for administration purposes.

Process data

Access to process data is configured by the developer user inside Alpana Designer.
Credentials to access process data are configured inside the corresponding Connection.
Queries to those connections are stored in Data Sources.
All this information is stored in encrypted form inside the Alpana objects : Data Source objetcs, Widget objects, Dashboard objects.

When published, they are stored in the Alpana SQL database.
When downloaded, they are stored as an encrypted file.

Some process data may be copied temporarily for processing (see an explanation of the Buffer).
Alpana Server can define a dedicated table to host user input (see an explanation of the Runtime Connections)

Access from external applications

In the current OMI app, the below login model has been implemented.
A developer user configures an OMI layout to contain the Alpana app, they provide login information to Alpana Server, select a visualization that they wish to display, and configure it.
When an OMI user will browse to that layout, they will see the visualization thanks to login information configured by the developer.

Making queries to Alpana Server from another website is blocked by default (Cross-origin resource sharing / CORS).
Alpana Server allows to configure exceptions to that rule.

Last modified: Aug 17, 2020

1.2. Getting Started

Getting started with Alpana is easy, and the summary below should help you follow the steps.

Getting Alpana

1. Contact your official Distributor to **get a license**. If you don't know who is your official Distributor, get in touch with us at information@code192.com and we will sort it out !
2. Once your license is created, connect to your License Portal to **download Alpana**.
3. **Read** the *ReadMe* file and the Release Notes. In particular, pay close attention to the Pre-Requisites of Alpana Server, and the upgrade instructions in general.

Installing Alpana Designer

Alpana Designer is used to create the content of Alpana (Widgets, Dashboards, Data Sources).
It may be installed on the same machine as Alpana Server, or remotely. The important part is being able to access the data.

1. Check the Release Notes
2. **Install the Pre-Requisites**
3. **Install Alpana Designer**

Installing Alpana Server

Alpana Server is used to display the content of Alpana (Widgets, Dashboards, Data Sources) and share it with other users.

> ! Important : Alpana Server is a complex application that creates databases, deploys a website, etc.
> It is important to read carefully the Pre-Requisites and installation instructions.

Prepare

1. Check the Release Notes (and Upgrade Instructions if you are upgrading).
2. **Decide** where you will install Alpana Server after reading up on the architecture of the application.
3. **Install the Pre-Requisites**. This step is important and must be done **fully and in the right order**. A machine restart may be required.

Install

Install Alpana Server

Activate a license

1. Make sure you still have **at least 1 Activation** left on your license.
2. **Login** to the Host, see Post-Installation steps.
3. **Activate your license**
4. **Activate the desired license features**
5. **Assign features** to the Tenants as desired.

What's next ?

See Post-Installation steps.

Here are some tips on what to do next :

- Don't forget to change the default passwords !
- Login to a Tenant to configure it. Read up the Administration chapter for more information on configuration.
- Create Categories on the Tenant. This will allow users to publish Dashboards
- Configure e-mail settings. This will allow to invite users by e-mail, let them recover their passwords, and more.
- Create Users and Roles to let users access Alpana Server. Configure the relevant security options.

Last modified: Jun 19, 2020

1.3. License Portal and Installers Download

Web License Portal

The web license portal is located at :
https://secure.softwarekey.com/solo/customers/Default.aspx?AuthorID=5159868

This is where you activate and manage your license with [code192] Alpana software editor.

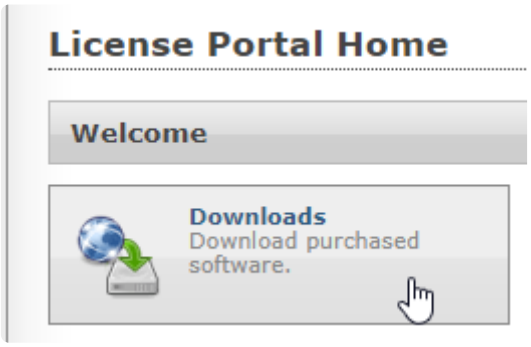
Your login and password was sent to you on license delivery.

Downloading Installers

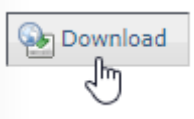
The lastest installers are available in the License Portal.
In order to download them, please follow the below instructions.

Login to the Web License Portal with your Customer ID.

In the home page, click “Downloads” :



Below your license ID, click *Download* :



This downloads `.zip` file containing all installers and documentation allowed by your license.

Last modified: Dec 26, 2019

1.4. Release Notes

The below chapters document the release notes of each version of Alpana.

Last modified: May 12, 2020

1.4.1. v3.0.1.x

Release notes of items completed between Alpana v2.0.8.12 and v3.0.1.1

Upgrading

Upgrading from v2 to v3.0 requires manual steps.
Please read these steps and incompatibilities in the full chapter about Upgrading.

Main Features

Some main features in a few words :

new API

Send and receive context with external applications, live update of data.
ex : Connect OPC, Wonderware sytem Platform and more to Dashboards and get live updates for targeted users.

new Widgets

All Widgets have been refreshed with the latest web technologies, and more configuration options have been exposed.

The following new Widgets Types are available :
Embed, Radar, Polar, Half-Donut / Half Pie, Range Area, Range Column, Step Line, Step Area, Spline Area, Hi/Lo, Hi/Lo Open/Close, Candle, Waterfall.

Charts have huge improvements in configurability :

- any number of chart types can be combined together
- any number of Y axis can be created and assigned to series (for different scaling)
- X axis will not be limited to “Category” type of axis : with a “real date” axis you will better see events in time, with a “real number” axis you will be able to show a real scatter plot, including logarithmic
- a Dimension field can be used to configure a series of separate charts (v2.0 Combo Chart behavior) and separate trends in the same chart (v2.0 other charts behavior), both at the same time

Gauge can display a threshold between good/bad, and can be displayed as a needle with ticks instead of only as a filling gauge

new dashboard layout and themes

Dashboards are not limited vertically to 1 page, and user will be able to scroll down to see more widgets, with no limit from our side.
Dashboard layout is configurable in a finer way : the number of tiles available for alignment is configurable
Theme support has been added : not only dashboard theme can be switched (including a dark theme), but also transparent widgets can be used on top of an image background of your choice

multi-tenant Alpana Server

OEMs will gain in maintenance and deployment costs
Administrators will be able to create a single install hosting dashboards for all their customers in fully separate instances : each will have their own portal and branding, their own dashboards, license features, users, etc… but all will be managed from a central location.

more licensing options

A Freemium tier is available.
Concurrent user licenses, with reserved users.

Ease of Use

Alpana Designer has seen some improvements, mostly focused on usability :

- any number of joins can be created between any 2 tables
- Expression editor is a lot more usable (including also copy-paste of expression parts), and also allows to use another expression inside an expression
- miscellaneous improvements in configuration interfaces

Wonderware Historian improvements

Wide query support allows many new scenarios, since each tagname can be in a separate field : assigning each tag to a different Y axis on a Chart, using a tag as value and another as target, etc …
Parameter-Based queries allows users to update the Historian query at runtime, and enables more data exploration scenarios.

Tasks

Features (61+) that were implemented in or before v3.0.1.x :

- ALP-T805 Filtering : dissociate “value” field and “display” field
- ALP-T744 Widget : difference between HEADER and NAME
- ALP-T747 To be able to Hide a Header of a Widget (Card,Proportion etc..)
- ALP-T763 Make Drill-Up UI more friendly

- ALP-T870 Expressions: Be able to duplicate an expression or a part of expression
- ALP-T762 Grid : implement column filters
- ALP-T907 Gauge : add needle
- ALP-T697 A solution to show quality limits on a Chart
- ALP-T768 Line Chart : allow to configure “show marker” independently for each line
- ALP-T678 Scatter/Bubble : expose chart properties
- ALP-T937 Scatter : implement drill down
- ALP-T945 Chart : dynamic Strip Lines
- ALP-T1235 avoid value label overlap
- ALP-T674 Allow to configure Virtual Scrolling
- ALP-T696 New widget : Step Line
- ALP-T695 Add drill-down to Combochart
- ALP-T671 Implement DateTime axis in all Charts
- ALP-T688 Charts : implement Secondary Axis
- ALP-T700 Combochart : allow to configure “Show Marker” for the line
- ALP-T701 New widget : Step Area
- ALP-T677 Allow Combochart as container to be configured other than full size
- ALP-T708 Widget : implement Spline Area
- ALP-T702 allow to configure separately each series in a Chart
- ALP-T704 implement Forecast Chart
- ALP-T944 logarithmic Chart Axis
- ALP-T947 Chart : better handling of empty points
- ALP-T1331 autoscale Y Axis even when Row is bound
- ALP-T1247 less threatening “no data to display”
- ALP-T690 New widget : Waterfall
- ALP-T692 New widget : radar
- ALP-T813 Link to URL : send Parameter value
- ALP-T785 Gauge: Show Target value in tooltip or next to its bar
- ALP-T717 Allow Dimensions to have measure Formatting
- ALP-T754 Label widget : create links to URL
- ALP-T689 Charts : bind field to color
- ALP-T629 Expression Editor improvements
- ALP-T1002 Designer UI : Widget Edit : show Data and Properties on same view
- ALP-T1004 Server portal : replace 1-level Categories
- ALP-T1047 Color Picker : add Hue selector
- ALP-T1226 Designer : improve configuration UI
- ALP-T1245 implement Combobox incremental search
- ALP-T759 Use Expressions in DataSource-level filters
- ALP-T909 Show the presence of DataSource-level or Widget-level filter
- ALP-T760 Use Expressions in a JOIN condition
- ALP-T838 Data Source Designer : allow to join between column values
- ALP-T749 Expressions : be able to use another Expression as field
- ALP-T715 Need to add Join conditions as many as needed
- ALP-T735 Improve Expression Editor for long expressions
- ALP-T798 implement relative date filter on Data Source

- ALP-T865 Datasource Filter Editor UI : differentiate tables
- ALP-T1375 New widget : Scatter Plot
- ALP-T1403 allow to display value labels inside Proportion Charts
- ALP-T1404 New Proportion Chart : Semi-Pie
- ALP-T1405 New Proportion Chart : Semi-Doughnut
- ALP-T1409 Gauge : show/hide ticks and axis labels
- ALP-T1410 Gauge : improve target visibility
- ALP-T1411 Gauge : show optional green/red threshold
- ALP-T1479 Widget and Data Source filter UI : search box for values
- ALP-T1506 New Widget : Embed
- ALP-T1537 automatic date format localization
- ALP-T1578 New widget : polar
- ALP-T1499 Historian : use widgets to change date range in query

Issues

Bugs (82+) that were fixed in or before v3.0.1.x :

- ALP-I1100 – autorefresh on maximized widget breaks dashboard
- ALP-I1013 – cannot multijoin when tables have same name
- ALP-I1023 – Expressions : missing editor for multiple values
- ALP-I971 – Mobile : DatePicker unusable
- ALP-I974 – mobile: linking widget not working on IOS
- ALP-I1191 – v3 widget Header and description not validated when changing tab
- ALP-I1110 – error when stored procedure schema changed
- ALP-I997 – Designer : Edit > Delete doesn’t work
- ALP-I998 – Designer : widget appears incorrectly selected when closing tab
- ALP-I1001 – data preview content disappears when resizing
- ALP-I1038 – Edit>Delete Widget doesn’t work
- ALP-I750 – Widget: Date Filter: Unhandled Exception
- ALP-I1031 – datetime filter implicit conversion fails on some systems
- ALP-I1037 – wrong buffer table referenced by Custom Query Table
- ALP-I915 – Dev / debug-like / fake-like error message in preview dashboard
- ALP-I941 – this dashboard freezes the browser when filtering
- ALP-I972 – Chart axis custom titles overridden on drill-down
- ALP-I981 – Autoscale starts at 0 when Line Chart has Row
- ALP-I988 – Line Chart can make browser unresponsive
- ALP-I996 – Card “show target value” doesn’t work
- ALP-I1004 – copy-paste a Card loses measure formatting
- ALP-I1012 – Card shows decimals even when measure formatted with 0 decimals
- ALP-I1029 – bad query when joining 2 Custom Queries
- ALP-I1032 – Card shows decimals as 0 when system decimal symbol is not dot
- ALP-I1027 – Drill-Down on MonthYear creates wrong query in this chart
- ALP-I1016 – RadioButton without “All” item displays its state incorrectly

- ALP-I1035 – chart trendline broken when Row field configured
- ALP-I1002 – chart tooltip shows “undefined” when axis has no title
- ALP-I1005 – copy-paste a Chart loses axis labels
- ALP-I937 – Chart value labels overlap
- ALP-I944 – widget maximized on homepage loses legend
- ALP-I991 – Server responsiveness : on small resolutions important function missing
- ALP-I1019 – Designer unhandled NullReferenceException when closing window on error
- ALP-I1006 – Card: Scroll bar is visible in 1 block size widget after resize page
- ALP-I925 – Card shows needless scrollbar
- ALP-I1030 – Inconsistency in MonthYear format
- ALP-I986 – can’t edit Data Source from with SQLite buffer
- ALP-I789 – Designer Preview : IIS Express generates warnings in Windows Events
- ALP-I852 – Widget preview crashes when Widget is resized
- ALP-I872 – widget legend shows scrollbar needlessly
- ALP-I982 – single Gauge as Master fails with JS error
- ALP-I993 – uncaught StackOverflowException when removing Parameter
- ALP-I1021 – in IE, can only drill-up to top
- ALP-I1022 – Chart Hide Legend ignored during drill-down
- ALP-I1028 – Gauge hangs web browser when Value=0
- ALP-I945 – JS Uncaught TypeError when “Go to Dashboard” from Homepage
- ALP-I985 – custom legend lost from this dashboard
- ALP-I1017 – Measure Formatting : must use workaround to configure Auto decimals
- ALP-I1015 – Card truncates suffix
- ALP-I1026 – Web connector can’t handle sub-millisecond precision from WW Online in SQL
- ALP-I1009 – Grid virtual scrolling throws error 500 on Public dashboard
- ALP-I1018 – Published Dashboards can’t be re-opened in Designer after renaming
- ALP-I1020 – Grid uncaught syntax error when using field names with single quote
- ALP-I1036 – Grid KPI Percentage of Target shows all as green
- ALP-I989 – RangeNavigator with single Date displays since January
- ALP-I992 – JS error while loading dashboard on server and clicking on collapse menu
- ALP-I999 – widget CheckBox : label alignement wrong in IE11
- ALP-I1003 – Issues with Measure formatted as Currency
- ALP-I1010 – Grid conditional formatting throws error if Excel cell empty
- ALP-I1025 – Date format wrong in Chart with data converted from Web in SQLite buffer
- ALP-I1092 – Grid with paging breaks header filters
- ALP-I1112 – Map Shape cannot bind to GeoJSON number Property
- ALP-I1111 – Map Shape doesn’t accept number fields
- ALP-I935 – Mobile : Combobox filter not applied
- ALP-I936 – Mobile : Master widget filters not applied
- ALP-I1091 – TreeMap Group By doesn’t accept number fields
- ALP-I959 – : RangeNavigator JS Uncaught TypeError when data has NULL
- ALP-I1113 – Choropleth Map default scale colors wrong
- ALP-I1276 – Opening Widget Filter is slow on big database
- ALP-I1538 – Widget Label : difficult to apply style to some parts
- ALP-I1477 – Some empty widget previews

- ALP-I1613 – Treemap scale wrong order
- ALP-I1039 – filter by datetime in SQLite returns no data
- ALP-I1000 – dashboard file corrupted when closing Designer
- ALP-I938 – Historian Connector Window: Selected tags panel does not appear on small resolution
- ALP-I994 – selection issues in dependent Comboboxes
- ALP-I1014 – Designer shows wrong map name
- ALP-I1007 – show two parameters label values as Widget Title shows: All
- ALP-I967 – no measure formatting in Grid exported as PDF
- ALP-I880 – misc SQL-related errors in Scheduler logs when starting machine
- ALP-I477 – Designer may disappear from taskbar
- ALP-I1646 – Designer doesn’t delete temp file in working folder

Last modified: May 12, 2020

1.4.2. v3.0.2.x

Release notes of items completed between Alpana v3.0.1.1 and v3.0.2.2

Upgrading

Alpana Server install upgrade from v3

To upgrade Alpana Server from v3.0.1.1, first un-install the previous version from Windows Control Panel, as documented here.

License upgrade

> ! An internal change in license naming is causing a breaking change with existing v3 installations that are not “Freemium” licenses.

If you currently have an installation of Alpana Server v3 that is not a “Freemium” license, please follow the below procedure to complete your upgrade :

1. Upgrade Alpana Server as described above
2. Activate your license again as described here (including downloading the license file)

Tasks

No new features have been implemented in this version, as it is mostly a bugfix release.

Issues

Bugs (25) that were fixed between Alpana v3.0.1.1 and v3.0.2.2 :

- ALP-I1718 – Server installer fails with SQL Windows authentication in some cases
- ALP-I1708 – Server installer rolls back when SQL server name contains antislash
- ALP-I1695 – Server : frontend doesn’t start on port 80
- ALP-I1675 – performance issue when loading Alpana Server pages
- ALP-I1662 – IQ_UnableToUpdateBuffer when publishing this dashboard
- ALP-I1656 – Custom X Axis title wrong when changing drill-down fields order
- ALP-I1148 – v3 Polar / Radar doesn’t support Drill-Down
- ALP-I1685 – Designer : The given key was not present in the dictionary. with Excel file.
- ALP-I1605 – RangeNavigator handles reset wrongly when using Clear Filter
- ALP-I1615 – Treemap legend ignoring Measure Formatting
- ALP-I1633 – Chart : Range styles can’t display Trend
- ALP-I1658 – Designer application won’t maximize on second monitor

- ALP-I1681 – renaming of license packs for pricelist v2.9
- ALP-I1683 – Notification Settings can’t be displayed : JS error
- ALP-I1687 – Donut : random JS TypeError occurred
- ALP-I1697 – Server : IC_ParametersInvalid et IC_BufferConnectionFailed
- ALP-I1689 – Combobox JS Type Error when filtering this Chart
- ALP-I1679 – Polar Chart : IQ_ParameterNullOrEmpty when removing field on a widget copy
- ALP-I1684 – Bubble Chart : field configuration reset when editing series
- ALP-I1676 – RangeSlider with date : labels overlap
- ALP-I1688 – issue in this v2 imported dashboard
- ALP-I1690 – Designer UI : remove “Close tab”
- ALP-I1682 – label truncated
- ALP-I1692 – Designer UI : can’t change series order when scrollbar present
- ALP-I1655 – Depending on the dashboard, tooltip can have performance issues.
- ALP-I1686 – Card : number localization issues

Last modified: May 12, 2020

1.4.3. v3.1.0

Release notes of items completed between Alpana and v3.0.2.2 and v3.1.0.7

Upgrading

New Pre-Requisites

Alpana Server and Designer v3.1.0.7 now require .NET Framework 4.8

Alpana Server install upgrade from v3

To upgrade Alpana Server from v3.0.2.2, see Upgrading Alpana from v3.0 to v3.1.0

Main Features

Redundant Connections

A Redundancy mechanism has been implemented on SQL, Historian and Web connectors.

Alpana Update File Dependency

The tool Alpana Update File Dependency has been added and needs to be installed separately for live update of file-based connections.

Tasks

No other new features have been implemented in this version, as it is mostly a technology upgrade and bugfix release.

Issues

Bugs (44) that were fixed between Alpana v3.0.2.2 and v3.1.0.7 :

- ALP-I1694 – Export PDF : issue with Gauges
- ALP-I1693 – Can’t export widgets on this dashboard
- ALP-I1678 – Mobile : DatePicker pop-up doesn’t open
- ALP-I1671 – Server PDF exports: charts overflow at page right
- ALP-I1713 – Chart : random JS error when resizing browser
- ALP-I1717 – Views > Reset Parameter doesn’t clear filter UI
- ALP-I1759 – traduction Française Paramètres SMTP
- ALP-I1763 – all license features assigned to new tenant
- ALP-I1773 – Excel Export not filtered when using custom query and relative dates
- ALP-I1787 – Chart X Axis labels overflow container in some cases

- ALP-I1722 – Designer UI : more explicit message when no permission to publish
- ALP-I1712 – IQ_UnableToRetrieveDataTable happened randomly on this dashboard
- ALP-I1702 – UI Parameter Editor issue when too many Parameters
- ALP-I1706 – Designer allow mark as Public even if impossible
- ALP-I1705 – Designer : copy-paste widget on wrong location
- ALP-I1704 – Grid : clicking header after link to url opens link
- ALP-I1701 – Issue when Expression name contains dots
- ALP-I1700 – Configuring Drill-Down and Master on same widget gives errors
- ALP-I1677 – RangeSlider with datetime : labels shifted
- ALP-I1669 – Cartesian chart javascript error
- ALP-I1659 – Chart : Y Axis auto-scale wrongly applied in mixed cases
- ALP-I1762 – error when selecting in this multi-selection Combobox
- ALP-I1768 – Parameter not set when no Listener present
- ALP-I1802 – Designer crash on copy-paste this Proportion Chart
- ALP-I1788 – Filtering based on WHERE condition on 2 datetime fields
- ALP-I1770 – can’t replace custom Map
- ALP-I1774 – Designer crash when deleting Parameter after selecting condition
- ALP-I1775 – custom Choropleth Map empty
- ALP-I1716 – Reload Dashboard resets filter UI but doesn’t clear filter
- ALP-I1777 – TreeMap tooltips style inconsistent and error-prone
- ALP-I1804 – NullReference while opening this dashboard
- ALP-I1801 – Erreurs sur requête Historian Wide Query avec SQL Français
- ALP-I1791 – Alpana Server setup fails when listing available IP
- ALP-I1731 – Designer: Auto-creation of parameters behaviour
- ALP-I1764 – Excel dependency not updated from file path
- ALP-I1827 – Designer : widgets toolbox : widgets names
- ALP-I1642 – Bubble Map : this custom map doesn’t display shapes
- ALP-I1691 – Bubble Map error : e.map is not a function
- ALP-I1719 – Cartesian chart => Enable scrolling => scrollbar should be hidden
- ALP-I1750 – Tenants not renamed properly
- ALP-I1816 – DatePicker pre-selection marked incorrect when using Highlight Available
- ALP-I1723 – DatePicker : dates unavailable when using Highlight Available and Relative
- ALP-I1769 – Designer : Tab Headers style
- ALP-I1673 – Widgets won’t refresh when resizing application

Last modified: May 12, 2020

1.4.4. v3.1.2

Release notes of items completed between Alpana v3.1.0.7 and v3.1.2.1
Alpana Update File Dependency has been upgraded from v1.0.0.3 to v1.0.0.4

Main Features

Date Time Picker

The Date Picker widget now supports selecting time intervals.

Custom date formats

Custom date formats are now supported.

Solved Issues

The following are the main bugs that were fixed between Alpana v3.1.0.7 and v3.1.2.1 :

- ALP-I1861 – Server: dashboard background image : Cannot insert null value to the cache
- ALP-I1867 – Card error when null value present
- ALP-I1865 – JSON Connector error : The method or operation is not implemented
- ALP-I1838 – Historian redundancy not triggered
- ALP-I1868 – Choropleth Map with Calculation broken
- ALP-I1724 – RangeNavigator : wrong selection with Relative Date in this configuration
- ALP-I1834 – X Axis StripLine behaviour incorrect when duration too short
- ALP-I1830 – Bubble Map doesn’t support geojson Point
- ALP-I1871 – Date Picker with range : selection maintained after clearing filter
- ALP-I1729 – Server : notification icon doesn’t display when notifications are present
- ALP-I1841 – Update Dependency : error when trying to update Image
- ALP-I1843 – disable X Axis Stripline with X Axis type Category
- ALP-I1828 – Chart tooltip text overflows background
- ALP-I1829 – Bubble Map tooltip has no background

Known issues

The following are the main bugs that are known and still present at the time of release of v3.1.2.1 :

- ALP-I1879 – Null reference exception when copy-paste Widget in Designer
- ALP-I1877 – Random IOException on Map : can’t access file
- ALP-I1840 – “no such column” when re-opening certain v3.0 file-based dashboards in v3.1
- ALP-I1866 – Custom date format not applied in Chrome
- ALP-I1882 – Date Picker with time : display wrong when filter empty

- ALP-I1881 – Date Picker with time : can’t select time when using relative date
- ALP-I1842 – Update Dependency : error when changing credentials
- ALP-I1736 – Server : pinned Chat disappears

Last modified: Aug 30, 2019

1.4.5. v3.2.1

Release notes of items completed between Alpana v3.1.2.1 and v3.2.1.5
Alpana Update File Dependency has been upgraded from v1.0.0.4 to v1.0.2.1

New Pre-Requisites

IIS Express is no longer necessary as a pre-requisite of Alpana Designer.
It may be uninstalled safely.

New Tools

Alpana Real Time Data

A new tool was introduced for demonstration purposes :
Alpana Real Time Data v1.0.2.0

Main Features

New Widget : Sankey

A new Widget type has been added : the Sankey diagram.
This widget allows beautiful representation of flows that can allow you to easily :

- detect water leaks
- analyze energy flows
- understand logistics flows
  … and more !

New Widget : PDF Reader

A new Widget type has been added : the PDF Reader.
This widget allows to embark PDF files inside a dashboard to display it inside the Dashboard.
This allows to display dynamic dashboard data side-by-side with static PDF data.
For example, the maintenance manual displayed side-by-side with the KPI's related to that maintenance operation.

The PDF file can also be changed live : depending on user action, a different PDF file gets displayed.

New connection : Oracle

Native connection to Oracle 12 and above is now supported.

API features for real time

The Alpana Server web API has been extended to allow to send data to dashboards in real-time from your external application :

- A new Connection called Alpana Runtime allows to push data in a table hosted by Alpana.
- A new Widget configuration allows to configure a Widget without a Data Source, and then push live data into it.

other Widget improvements

The Embed widget now accepts Parameters to dynamically change its content.
The Date Picker can now be configured without a Data Source.
The Grid widget now allows multiple selection as Master Widget.
The Card widget now allows to display the KPI's in any order.

Other features

Administrators now have the possibility to save up concurrent licenses by forcing users to logout after inactivity.

Solved Issues

The following are the main bugs that were fixed between Alpana v3.1.2.1 and v3.2.1.5 :

- ALP-I1879 – Error when copy-paste Widget in Designer
- ALP-I1885 – Designer : dragging widget scrolls layout too fast
- ALP-I1984 – Designer error when cancelling object creation then switching back to tab
- ALP-I1891 – layout shaking when loading and on hover
- ALP-I1848 – Unpredictable behavior when resetting filter if 2 Parameters bound
- ALP-I1648 – Chart: Enable Scrolling with several Series is not based on each Series per Page
- ALP-I1906 – Date Time Picker AM/PM not visible
- ALP-I1882 – Date Picker with time : display wrong when filter empty
- ALP-I1881 – Date Picker with time : can't select time when using relative date
- ALP-I1991 – some Comment text styles not applied
- ALP-I1889 – Background image not displayed in some cases
- ALP-I1840 – "no such column" when re-opening some older CSV dashboards
- ALP-I1877 – Random IOException on Map : can't access file
- ALP-I1988 – Line marker remains highlighted after mouse leaves
- ALP-I1896 – snippet "Embed in code" doesn't escape quotes

Known issues

The following are the main issues that are known and still present at the time of release of v3.2.1.5 :

- The language selector has been removed, and will be replaced by a more complete implementation in an upcoming version.
- ALP-I1998 – Error if opening Dashboard without having opened Designer first. To solve this, open Alpana Designer one time from the desktop shortcut.
- ALP-I1993 – When adding a date dimension to a Chart, it looks like the X Axis is formatted as datetime instead of Year. Please refresh the Widget preview to fix this graphical issue.
- ALP-I1980 – Cannot change profile picture
- ALP-I2005 – new CORS whitelist item overwrites previous item

Last modified: Jun 19, 2020

1.4.6. v3.2.2

Release notes of items completed between Alpana v3.2.1.5 and v3.2.2.10
Alpana Update File Dependency has been upgraded from v1.0.2.1 to v1.0.2.2
Alpana Real Time Data has been upgraded from v1.0.2.0 to v1.0.2.1
Alpana OMI App v1.1 is now available upon request.

Main Features

Alpana v3.2.2.10 implements the following main features :

General improvement : refresh smoothing

To create a more pleasant user experience, the data refresh has been “smoothed” :

- when using filters at runtime, the listener widgets will only display a “loading” animation if the query is slow. Otherwise, the listener widgets will refresh smoothly and just display the new data content.
- when using the real-time API features, the target widget will refresh smoothly and just display the new data content.

Performance improvements

As part of a continuous effort to improve performance, some noticeable progress includes :

- important performance improvements for bufferized connections
- a big performance improvement in Alpana Designer when opening a Connection with many objects in the schema (example : Wonderware Historian “Runtime” database)

Grid widget features

The Grid widget has received miscellaneous improvements based on user requests :

- Footers : you can now add Footers at the bottom of the Grid, to use as summation rows (can also calculate average, count, etc…)
- Nested Columns : you can now group Grid columns and configure a common title on top of the columns titles.
- columns resizing : users can resize Grid columns at runtime, to see content more easily
- alternate background color : to view large tabular data more easily, you can now configure the Grid to display each row with an alternate background color that is configurable

Designer

Some improvements where implemented to make the developer life a bit easier :

- a search text box allows to browse the database schema more conveniently
- file-based Connections now have an easy shortcut to refresh the file content in Alpana Designer
- to allow easier SQL Connection migration, the database name is not reset anymore when changing the host address

API

The API `GetWidgets` method now includes parameters `Filter` and `IsPublic`, to make it easier to search for Widgets.

Solved Issues

The following are the main bugs that were fixed between Alpana v3.2.1.5 and v3.2.2.10 :

- ALP-I1998 – Error if opening Dashboard without having opened Designer first
- ALP-I1993 – When adding a date dimension to a Chart, it looks like the X Axis is formatted as datetime instead of Year.
- ALP-I1980 – Cannot change profile picture
- ALP-I2005 – new CORS whitelist item overwrites previous item
- ALP-I2033 – install with subdomains contains wrong CorsOrigins in appsettings
- ALP-I2019 – Donut selected state doesn't match the filter state
- ALP-I2115 – Widget Export menu invisible if Image Export disabled
- ALP-I2012 – Range Navigator interval includes next period
- ALP-I2015 – Radar Chart item labels are not in front of values
- ALP-I2057 – Mobile : multi-selection Combobox cannot be validated
- ALP-I2013 – Display issues in Grid with paging and several columns with the same name
- ALP-I2021 – Designer "Clear Filter" button in widget edit tab not working
- ALP-I2025 – Gauge's arc disappearing in Designer
- ALP-I2014 – Designer shows empty UI when deleting Widget then switching back to tab
- ALP-I2035 – field type not updated when Custom Query updated

Known issues

The following are the main issues that are known and still present at the time of release of v3.2.2.10 :
(no known issues at this time)

Last modified: Jan 08, 2020

1.4.7. v3.2.3

Release notes of items completed between Alpana v3.2.2.10 and v3.2.3.3
Alpana Update File Dependency has been upgraded from v1.0.2.2 to v1.0.3.1
Alpana Real Time Data has been upgraded from v1.0.2.1 to v1.0.3.1

Main Features

Alpana v3.2.3 is mostly a bugfix release with the following minor additions :

- Parameter default values can now properly handle all the cases : some selected, all selected, none selected, `NULL` selected.
- a new option to help with license troubleshooting
- when opening the link to a dashboard while user is not logged in, there is now a shortcut to login.

Solved Issues

The following are the main bugs that were fixed between Alpana v3.2.2.10 and v3.2.3.3 :

- ALP-I2120 – In some dashboards from previous versions, there was “no data” in the Designer.
- ALP-I2118 – Alpana Update File Dependency internal error when connecting
- ALP-I2122 – “Pin Widget” not working on some widgets
- ALP-I2126 – Wrong display on filter widgets when NULL values are present
- ALP-I2140 – Range Slider sometimes inverts interval and creates wrong query
- ALP-I2139 – Bar Chart : some X Axis settings reset to defaults when scrolling
- ALP-I2137 – Sankey : scroll view not reset on filter

Known issues

The following are the main issues that are known and still present at the time of release of v3.2.3.3 :

- ALP-I2143 – In specific cases, using the text search in multi-selection Combobox makes the page unresponsive

Last modified: Jan 29, 2020

1.4.8. 2020 R1

Release notes of items completed between Alpana v3.2.3.3 and 2020 R1
Alpana Update File Dependency has been upgraded from v1.0.2.1 to 1.1.0.2
Alpana Real Time Data has been upgraded from v1.0.2.1 to 1.1.0.2

Introduction

Version 2020 R1 is the first big release of the year and includes many new features and bugfixes.
As usual, the application is updated to the latest web technologies (Angular 9, .NET Core 3.1), with the latest security features and patches.

> Some of the changes require **special attention when upgrading**, so please read below for details.

New versioning :

After version v3.2.3.3, and since Alpana version 2020 R1, a new versioning scheme is used.
The internal version number for `2020 R1` is `3.3.0.25`.
See upgrading for more information.

Upgrade Instructions

To upgrade your installation, please see the generic upgrading chapter.
In addition, this version has specific instructions listed below.

Alpana Server

New Pre-Requisite

Alpana Server now requires to install a new version of .Net Core v3.1 Runtime & Hosting Bundle.
See Server Pre-Requisites for more information.

> Note : this new .Net Core version can be installed side-by-side with the previous one, which can be useful to minimize downtime during the upgrade.

Previous versions of .Net Core may be uninstalled safely if they are not used by other software.

License upgrade

Important : This is a new major version, so it requires a valid support contract to upgrade, and you must ask for a new license Activation to your support.

Upgrading Schedules

Following a change in the scheduling engine, existing Schedules need to be upgraded manually.

You will have to manually Disable/Enable each Schedule to upgrade it, or to Edit and save it.

In addition :

- “Minutes” will lose the “seconds” details
- “Hourly” will lose the “seconds” details
- “Daily” will lose the “seconds” details
- “Weekly” will lose the “seconds” details
- “Monthly” will lose the “seconds” details
- “Advanced CRON expressions” are replaced by a compatible expression

Enabling Runtime Connections

Following upgrade, to enable the new Runtime Connections feature on a tenant :

- activate the *Runtime Connections* license feature at the Host level
- add the *Runtime Connections* license feature to the desired tenants
- add the *Runtime Connections* permission to the Admin Role of the tenant, and to all desired Users and Roles

Alpana Designer

New Pre-Requisites

Alpana Designer now requires to install .Net Core v3.1 Runtime & Hosting Bundle.
See Designer Pre-Requisites for more information.

Dashboards

Special characters

Following an improvement on the management of international characters, some dashboards need to be upgraded manually.
If your dashboards configuration (headers, descriptions, etc, but not data) contain special characters, such as symbols m³ or non-English characters é or money symbols €, upon upgrade they will be displayed as question mark ?.
To solve this, just open the dashboard in Alpana Designer, and save again.

Main Features

New Features : Paginated PDF Exports, Mobile configuration and Layout

Management

A new concept has been introduced in each Dashboard : Layouts.
You can now display the same Dashboard in many display conditions, and **control** exactly how it will be displayed, for example :

- on a desktop web browser
- on a big screen
- in a PDF report
- on a mobile device
- inside InTouch OMI
- embedded in another website
- …

In all cases, you just re-use the same existing configuration (Widgets, Data Sources, etc) from the same Dashboard, and just change the layout :

- display/hide the Widget on some layouts
- move/resize content to view better
- change the theme to fit the embedding context

In particular, the PDF exports can now have a fully configurable Header and Footer, pagination and paper size.
The Designer has been improved for such tasks, with a new interface for zooming the canvas, viewing PDF page limits, hiding/displaying widgets, …
New PDF export behaviors have also been implemented to handle widgets with hidden data (example : Grid with pagination)

As another consequence, the Mobile layout can now be manually configured to display exactly as desired.

New Feature : Runtime Connections

Alpana now allows to create and host simple data tables.
See *Runtime Connections* on Alpana Server to create and manage those new objects.

Users can use interactive input forms inside dashboards to input data.
See the new Input Grid widget to add this possibility to Dashboards.

Developers can read this data to display it in a Dashboard.
See the new Alpana Runtime Connection type to use it in a Data Source.

New Connection : Insight

Alpana now includes a native Connection to Aveva Insight

It allows to connect to Aveva Cloud Operations to get historical data from Insight.

Connection improvements : Web

The Web connector now allows to connect to more API types, including Ampla.

New Widget : Linear Gauge

KPI's can now be displayed on a scale with the new Linear Gauge Widget type.

The previous "Gauge" Widget was renamed to Circular Gauge

New Widget : Sparkline

You can now display a KPI along with its history on a mini-trend with high/low limits, thanks to the new Sparkline Widget type.

New Widget : Banner

You can now display value cards on a scrolling banner similar to stock exchange websites, thanks to the new Banner Widget type.

New Widget : Gantt

You can now display scheduling and timeline data on the new Gantt Widget type.

New Widget : Energy Rating

You can now display regulatory environmental and energy performance data on the new Energy Rating Widget type.

New Widget : Tree View

You can now select filter values from tree-like hierarchical data thanks to the new Tree View filter Widget type.

Widget improvements

Card re-vamp

The Card has been completely overhauled to allow more configurability of the KPI display.
In addition, the Card can now display a sparkline to add historical insight to the single-value display.
This works particularly well with the new aggregation type called "Last".

Chart improvement

The *Spline Type* option for Spline chart types now includes a "Monotonic" option that doesn't overshoot

(it doesn't go over or under the actual values).

Range Navigator improvement

There is now a new option for the value display of the Range Navigator : *Axis Interval Type*

Designer improvements

New aggregation function : Last

The *Last Value* aggregation function now allows developers to easily display the single last value of a field.
This responds to the common requirement to "only see the value" in a Widget, despite having many values in a Data Source, and the general need for aggregation.

Parameter binding

The Parameter editor has been improved to display all uses of a Parameter, including in Expression, Stored Procedures, Filters, and Widget specific behaviors.
This allows to have a central location for analyzing more complex Parameter behaviors.

The automatic creation and binding of Parameters has also been improved, in order to give a more pleasant developer experience.
Similarly, Parameter deletion has been made easier.

Border configuration for Widgets

Now all Widgets types can have a configurable Border.

Performance

Designer error logs are now paged and will not have performance issues with very large files.

New query configuration : Distinct

All Widgets now have a now query configuration option : distinct.
This is enabled by default as it represents the most common case.
However, if you want to display repeated values from your data, you can now disable this.

Server improvements

LDAP integration

Previously, only integration with ADFS was possible.
Now, administrators can configure their Tenant to accept connection from Active Directory users, by simply configuring an AD server and admin user credentials.
AD users who connect will automatically have their Alpana account created with default Roles.

New Background Jobs management

Long-running tasks are now managed by a Background Jobs manager : manual dashboard exports, manual data import from files, etc
Results are communicated through the Notifications.
This allows a better handling of simultaneous exports

Concurrent login

Administrators can now choose to prevent users from connecting simultaneously from multiple browsers.

Category move

Dashboard Sub-Categories can now be moved to become a Root Category thanks to a dedicated control.

Changes in Guest permissions

The Guest default Server Role now has a new default behavior : by default, it has no access, and will only have access to content if explicitly granted.

Performance

Miscellaneous internal optimizations have lead to general performance improvements in page loading.

Known Issues and Incompatibilities

- The SQLite Connection was only used for internal tasks, and has been dropped from Alpana. If you need this connector, please contact your support.
- Excel formulas are not run in XLSX format
- The Range Navigator in “auto” mode may sometime display incorrect intervals.
- The “Candle” Chart when configured with more than 15 values of “split into same chart” will show the other items as black
- There are some issues with visual feedback when selecting from a chart as master widget
- On an upgraded installation, a message for an “internal error” may be displayed upon connecting, with no functional issue. If this impacts you, please contact support to apply a simple fix.

Solved Issues

The following are the main bugs that were fixed between Alpana v3.2.3.3 and v3.3.0.25 :

- ALP-I2159 – Grid paging broken with Historian query
- ALP-I2195 – Designer crash when removing a field from Grid with KPI
- ALP-I2155 – Scheduling : Data conditions never met
- ALP-I2179 – Parameter placeholder not replaced correctly in published Widget
- ALP-I2123 – Designer doesn’t allow to access changes in parameters of stored procedures
- ALP-I2145 – Expression deleted silently in specific workflow
- ALP-I2147 – Range Navigator may not display all dates

- ALP-I2210 – Server : Installation fails with some password characters
- ALP-I2252 – Grid conditional formatting causes error when bound to Parameter
- ALP-I2178 – Embed link of Widget pinned to Homepage is incorrect
- ALP-I2259 – “no data” in some cases when loading saved View
- ALP-I2175 – Schedule : logical condition’s Filter radiobutton resets automatically
- ALP-I2174 – Schedule : Targets User unbound when Editing
- ALP-I2248 – Schedule configuration : watcher field is not remembered correctly
- ALP-I2265 – error when exporting Host Change Logs to Excel in some cases
- ALP-I2172 – Comments : cannot see the presence of widget comments
- ALP-I2176 – Widget comment thread copied in Main
- ALP-I2180 – Cannot connect specific Excel XLS file : IQ_UnableToUpdateBuffer
- ALP-I2158 – date formatting HH:mm:ss.fff doesn’t show milliseconds
- ALP-I2164 – Designer : cannot see full tagnames when editting connection
- ALP-I2328 – Host admin : incorrect results when filtering Users list by Permission
- ALP-I2144 – Unwanted automatic Parameter links
- ALP-I2157 – Grid selection maintained incorrectly across paging
- ALP-I2109 – Treemap : default selection UI feedback incorrect
- ALP-I2153 – incorrect configuration which shows “No data” in Designer only
- ALP-I2150 – Server login page blocks when login failed
- ALP-I2314 – Euro “€” label rendered as question mark “?” in Chrome
- ALP-I2156 – Layout shaking a lot on certain dashboards with certain window sizes
- ALP-I2161 – Server : dashboard list “blinks” with specific workflow
- ALP-I2167 – Pin Widget disabled on published Widget
- ALP-I2169 – Disabled Link button on pinned dashboard
- ALP-I2154 – Default series name inconsistent
- ALP-I2258 – URL Parameter causes incorrect behavior in Views
- ALP-I2297 – inconsistent Parameter reset with URL Parameter
- ALP-I2319 – Chart : cannot edit legend when multiple measures but no split
- ALP-I2282 – Designer : cannot Publish to overwrite existing Widget
- ALP-I2028 – Column chart X axis labels display issue with long strings
- ALP-I2146 – Range Navigator doesn’t update selection when mouse is released outside
- ALP-I2165 – Grid maintains a visual selection even when not master
- ALP-I2303 – Chart : autoscale slightly wrong in some cases
- ALP-I2151 – Chart allows incorrect Y axis configuration and shows error
- ALP-I2293 – Data Source page : default pagination incorrect
- ALP-I2177 – User gets notified of own comments
- ALP-I2327 – silent error when importing incorrect users from Excel
- ALP-I2312 – Historian : random error “The ConnectionString property has not been initialized.”
- ALP-I2171 – Designer : Logs dialog very slow to open when many logs
- ALP-I2322 – Treemap tooltip shows all decimals
- ALP-I2163 – glitch when cancelling out of Rename Homepage dialog
- ALP-I2162 – Dashboard loses Favourite status when updated
- ALP-I2168 – Designer : setting Autorefresh doesn’t make the project “dirty”
- ALP-I2192 – Designer : state is not dirty when background color changed
- ALP-I2160 – incorrect error message when adding wrong user to chat

- ALP-I2092 – Range Navigator : clicking year has a different display after refresh
- ALP-I2149 – Parameter Editor UI : button overflows
- ALP-I2271 – Designer : Configuring Multi-Items Viewport doesn’t set project to dirty

In addition, many fixes were made to advanced Widget configurations.

Last modified: Jul 06, 2020

1.5. Upgrading

Please read the below chapters to upgrade Alpana.

Last modified: May 12, 2020

1.5.1. License incompatibility for Major upgrades

> ! Upgrading may require to re-activate your license in some cases. Read below for details.
> If this is your case, please **check beforehand that you will have a the ability to activate**, otherwise your Alpana Server installation will stop working while you are reaching Support to solve it.
> In case this happens, you may downgrade again to a version that supports your license, until a new activation is made available to you.
> Check with your Support for more information.

License compatibility

Versions before v3.0

Alpana v1 and v2 licensing is not compatible with current v3 licensing.
When needing to upgrade from v1 or v2, please contact your Distributor to get a v3 license.

Versions since v3.0

A license activation is only valid for the same *major* version.
If the upgrade involves a change in major version, then the license needs to be fully activated again, and you will need to submit again a license file and do a full activation.

> ! Until then, the Alpana Server installation will be unlicensed and will not work.

In order to fix this, please make sure you still are under a support contract, and contact your support to get an Activation for the new version.

Alpana versioning : what is a major version

Initially, Alpana used a version number like `v3.2.3.3`.
For those versions :

- the *major* version corresponds to the first 2 digits of the version number. Example : `v3.2` is the major version for version `v3.2.3.3`
- the *minor* version corresponds to the last 2 digits of the version number. Example : `.0.7` is the minor version for version `v3.1.0.7`

After version v3.2.3.3, and since Alpana version 2020 R1, a new versioning scheme is used :

- the *major* version corresponds to the “year” part of the version number. Example : `2020` is the major version for version `2020 R1`
- the *minor* version corresponds to the “R” part of the version number. Example : `R2 SP1` would be the minor version for version `2020 R2 SP1`

Major	Full
3.0	v3.0.1.1
3.0	v3.0.2.2
3.1	v3.1.0.7
3.1	v3.1.2.1
3.2	v3.2.1.5
3.2	v3.2.2.10
3.2	v3.2.3.3
3.3 (2020)	v3.3.0.25 (2020 R1)

A major upgrade happens when the upgrade involves a change in major version.

Example : from `2020 R1` to `2020 R2 SP1`, the license is still valid because you’re still in version `2020`. You don’t need to activate your license again. It works normally following the upgrade.
Example : from `2020 R2` to `2021 R1`, the license is not valid anymore because you upgrade from version `2020` to version `2021`. **You will need to activate your license again**.
Example : from `v3.2.3.3` to `2020 R1`, the license is not valid anymore because you upgrade from previous version `v3.2` to version `2020`. **You will need to activate your license again**.

Last modified: Jun 19, 2020

1.5.2. General upgrade procedure for v3

Introduction : limits of applicability

This procedure only applies if you are upgrading from Alpana Server v3.0 or newer.
Upgrading from older versions requires manual steps detailed elsewhere.

Some v3 upgrades may still require specific attention or manual steps. Please read the corresponding release notes.

Upgrading Alpana Designer

Upgrading Alpana Designer is straightforward :

1. Un-install any previous version of Alpana Designer.
2. Run the new installer

Upgrading Alpana Server

To upgrade Alpana Server to a newer version, please follow the below procedure.

> ! If your version of Alpana is older than `v3.0`, then this procedure doesn't apply.
> Please contact Support for information on this kind of upgrade.

1. Backup

You must make sure you have all information to install again :

Folder path to the website

Note the current folder path of the installation.

> ! It must be the same during the upgrade, otherwise you will not be able to keep your configuration and dashboards on upgrade !

Network configuration

Note the DNS or IP addresses and ports of Frontend and Backend
If you used HTTPS, gather all required SSL information.

> ✱ If you change those settings during the upgrade, your users won't be able to reach the website on the same address again.

SQL database connection

The upgrade will ask again for credentials to let Alpana access its SQL database.
Check that you have all the required information.

> ! It must be pointing to the same database during the upgrade, otherwise you will not be able to keep your configuration and dashboards on upgrade !

Other IIS configuration

If you tinkered with the IIS configuration manually, you must also back it up, as you will have to re-create it.

(optional) Application backup

As with any maintenance operation, you may wish to do a backup before the upgrade.

2. Read about the upgrade

Read the corresponding Release Notes

Make sure your license will allow the upgrade : read the chapter License incompatibility for Major upgrades.

3. Un-install

Use Windows Control Panel to un-install previous versions of Alpana Server and Designer.

> ✻ Note : If you don't stop the application, Windows will probably require you to reboot.
> **If you don't want to reboot**, you must stop the application before uninstalling.
> To completely stop the application, you may need to stop the IIS server itself.
> ⇒ If you do this, don't forget to start IIS again after uninstallation.

4. Pre-Requisites

Make sure you still meet the new Pre-Requisites (they may change for the newer version).

5. Install

Run the new installers and perform the installation procedures again for Alpana Server and Alpana Designer.

> ✻ When upgrading Alpana Server, please instruct all your users to clear their browser cache or do a hard refresh.

Last modified: May 12, 2020

1.5.3. Upgrading from an older v2 to v3.0

Introduction : limits of applicability

This procedure only applies if you are upgrading from Alpana Server version number starting with `v2`

Dashboards migration

Notice of incompatibility

Due to internal changes for supporting new features, one component from previous v2.0.8.x cannot be migrated natively : Wonderware Historian Connections.
If your v2 Dashboards contain Wonderware Historian Connections, you will have to re-create them manually.

File format difference

Dashboards from Alpana v3.0.1.x use a file format with extension `*.alpd` which is a different format from previous v2.0.8.x versions with extension `*.sydx`
To migrate a dashboard file, please follow the below procedure.

Alpana Server installation migration

Notice of incompatibility

Due to deep internal changes for supporting new features and improving the product, Alpana Server installations from v2.0.8.x cannot be migrated natively.

In order to upgrade Alpana Server installations from v2 (v2.0.8.12 and before) to v3.0 (up to v3.0.1.1 at least), you will need to perform manual steps :

1. upgrade Dashboards to v3.0
2. backup v2.0 installation
3. un-install v2.0
4. install and configure v3.0
5. publish dashboards and re-create users, categories, schedules

Please see below for advice and more details.

New licensing

Upgrading from v2.0 requires to generate a new compatible license key.
Please make sure before upgrading that you will have a compatible license for the new version.

General advice

You should keep your v2.0 installation running as long as possible in order to be able to view the old configuration at any time. You should wait to have re-created your dashboards before deploying v3.0 fully, in order to ensure continuity of service.
Dashboards can be migrated using Alpana Designer only, by following the above procedure.

1 – Upgrade Dashboards

To upgrade a file `dashboard_v2.sydx` :

1. rename the file with extension `*.alpd`, for example `dashboard_v2.alpd` (*)
2. open the file in Alpana Designer
3. check that migration happened correctly, make any fix necessary, or take advantage of new features
4. save the dashboard with a new name : for example `dashboard_v3.alpd`

> ! (*) The `dashboard_v2.alpd` file is just a renaming of the file extension, but the file is not in the correct format for v3. This file will cause confusion, so please delete it when you are finished migrating.

Other files upgrade procedure

Other files are migrated with the same procedure as above, just replace the correct file extension names :

- Dashboards : `*.sydx` becomes `*.alpd`
- Widgets : `*.sydw` becomes `*.alpw`
- Data Sources : `*.syds` becomes `*.alps`

2 – Backup v2.0 installation

backup

Perform a complete backup of your v2.0 installation :

- archive the complete folder `C:\code192\Alpana Server\Server.Web\App_Data\`
- archive the Alpana Server database you configured when installing v2. You can find its name by using `C:\code192\Alpana Server\Utilities\ConnectionStringEditor\ConnectionStringEditor.exe`

prepare user import

If you have many users to re-create, you can export the list as CSV from v2 : In the Alpana Server database, export the content of the [ALP_User] table as CSV.

prepare website binding

In IIS (inetmgr), take a note of the current bindings for your Alpana website : port number, name, …

prepare configuration

In Alpana Server web portal, go to the administration pages (at “http://yoursite/en-us/administration”) and take a note of all configuration to re-create :

- site URL
- branding
- e-mail settings
- AD settings

List other items

If there are other items to re-create from Server, list them :

- Categories
- Schedules
- published Dashboards/Widgets/DataSources

3 – Uninstall v2.0

The best workflow is to leave v2.0 installed as long as possible and deploy v3.0 on a different machine. This way, you can ensure that the migrated platform will behave as expected. For this workflow, you can delay the un-install of v2.0 to a later step.

However, if you need to install Alpana v3.0 on the same machine as the current v2.0, you will need to uninstall v2.0 first.

uninstall Dashboard Designer

From Windows Control Panel, uninstall “code192 Dashboard Designer”

uninstall Dashboard Server

> ! WARNING : At this step, you will lose your v2.0 dashboards and configuration if you didn’t make backups.

1. Make sure that you have backups

2. From Windows Control Panel, uninstall “code192 Dashboard Server”
3. During uninstallation, select to uninstall license, and select to “Remove resources and drop the database”. WARNING : At this step, you will lose your v2.0 dashboards and configuration if you didn’t make backups.
4. in Windows services.msc , make sure that the Service “Code192 Dashboard Server Scheduler” is well un-installed

Clean up

Services

If a service didn’t uninstall properly, you can follow the following Windows procedure :

1. get the Service name from services.msc, for example `Code192 Dashboard Server Scheduler`
2. Open a command prompt with administrative rights
3. stop the service using the command `SC STOP` for example : `SC STOP Code192 Dashboard Server Scheduler`
4. delete the service using the command `SC DELETE` for example : `SC DELETE "Code192 Dashboard Server Scheduler"`

Folders

Unless you have other [code192] products installed, make sure that there are no more code192 folders inside :
`C:\Program Files (x86)\`
`C:\ProgramData\`
`C:\` root directory

4 – Install and configure v3.0

Installation

- Check the new pre-requisites for Alpana v3.0. In particular, there is new software pre-requisites and a more recent version of .NET Framework required.
- Follow installation instructions to install Alpana v3.0 Server and Designer.
- When installing Alpana Server, install with the appropriate port

configure Alpana Server

In Alpana Server web portal, go to the administration pages and restore all the configuration needed :

- site URL
- branding
- e-mail settings
- AD settings

5 – Publish dashboards and re-create users, categories, schedules

In your new fresh install of Alpana Server v3.0, you will need to re-create items as backed-up previously :

create Categories

- Create the Categories needed. You can make use of v3.0 new feature of Categories contained under other Categories

Publish items

- Publish the validated Dashboards/Widgets/DataSources

Update Links

- Dashboards URL have a different format, so Links To URL between Dashboards need to be updated.

give access to users

- re-create Roles for users
- re-create Users. If you have many users, you can make use of the [ALP_User] table you exported as CSV from v2, and use *New User > Import from CSV*.
- assign permissions for Roles and Users to Dashboards and other items

Last modified: May 12, 2020

1.5.4. Upgrading from an older v1 version

Introduction : limits of applicability

This procedure only applies if you are upgrading from Alpana Server version number starting with `v1`

Notice of incompatibility

After v1 (Alpana `v1.1.1.5` and earlier), major changes were made in the core of Alpana.
Due to those breaking changes, v1 installation and dashboards will not be compatible with later versions.

Procedure

If you need to upgrade from v1, Support can help you find a solution. Please contact your Distributor.

Last modified: May 12, 2020

2. Alpana Designer

Alpana Designer is used to create the content of Alpana (Widgets, Dashboards, Data Sources).
Drag and drop widgets into the interface and manipulate your data right away. The desktop application allows you full autonomy. Once the Dashboard ready, share with your colleagues through Alpana Server and start to collaborate.

Read below the documentation of Alpana Designer to learn more.

Last modified: Dec 26, 2019

2.1. Installation

Installation of Alpana Designer is light on resource and pretty straightforward.
Proceed to the next chapters to complete the installation.

Last modified: Apr 16, 2019

2.1.1. Pre-Requisites

This section explains the system requirements to run Alpana Designer.

Software Requirements

The following minimum software requirements are necessary to run Alpana Designer :

Operating System

Windows OS (x64) :

- Windows 7+
- Windows Server 2012+

Microsoft .NET Framework

Microsoft .NET Framework 4.8 must be installed.

An offline installer exists at the following direct link :
https://dotnet.microsoft.com/download/dotnet-framework/thank-you/net48-offline-installer

(Alternately, go to the download page then download the “Runtime”.)

Installing this component requires to restart the machine.

AspNet Core

“ASP.NET Core 3.1 Runtime – Windows Hosting Bundle” must be installed.

Installers for latest v3.1.3 can be found at the direct link here :
https://download.visualstudio.microsoft.com/download/pr/ff658e5a-c017-4a63-9ffe-e53865963848/15875eef1f0b8e25974846e4a4518135/dotnet-hosting-3.1.3-win.exe

(Alternately, go to the .NET Core 3.1 download page and select version ASP.NET Core Runtime “Hosting Bundle” for Windows.)

Supported Browsers

Recent versions of Google Chrome, Firefox, Edge (only the recent “Chromium” versions of Microsoft Edge).

Hardware Requirements

The following minimum hardware requirements are necessary to run Alpana Designer, but more is better for bigger projects :

- Processor : Dual Core 32-bit CPU (x86)
- Hard disk : 1 GB, preferably SSD
- RAM : 2 GB, preferably more
- Accelerated graphics

> Note : Graphic acceleration
> Alpana Designer is a WPF application that requires accelerated graphics for best performance.
> On a physical host this is enabled, but on virtual machines this is a dedicated setting (in VmWare, under “Settings>Display>Accelerate 3D graphics”)

Last modified: Apr 30, 2020

2.1.2. Installing Alpana Designer

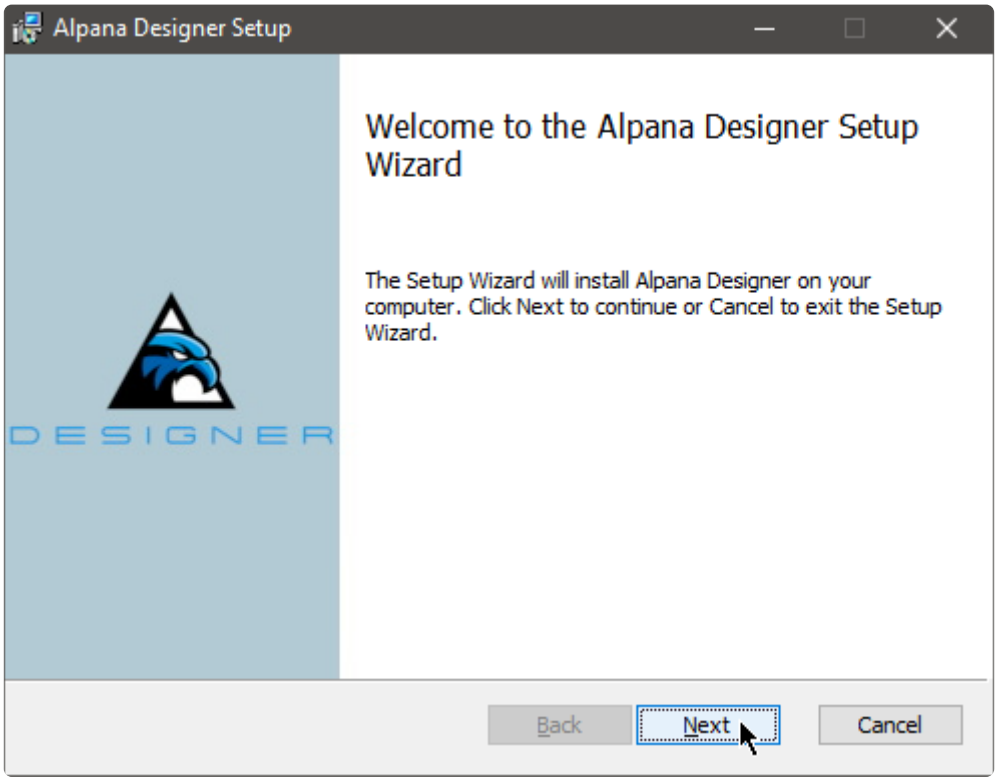
Download installer

See here to download the installer

Run installer

Double-click the installer to start installation :

On the welcome screen, click *Next* to continue :

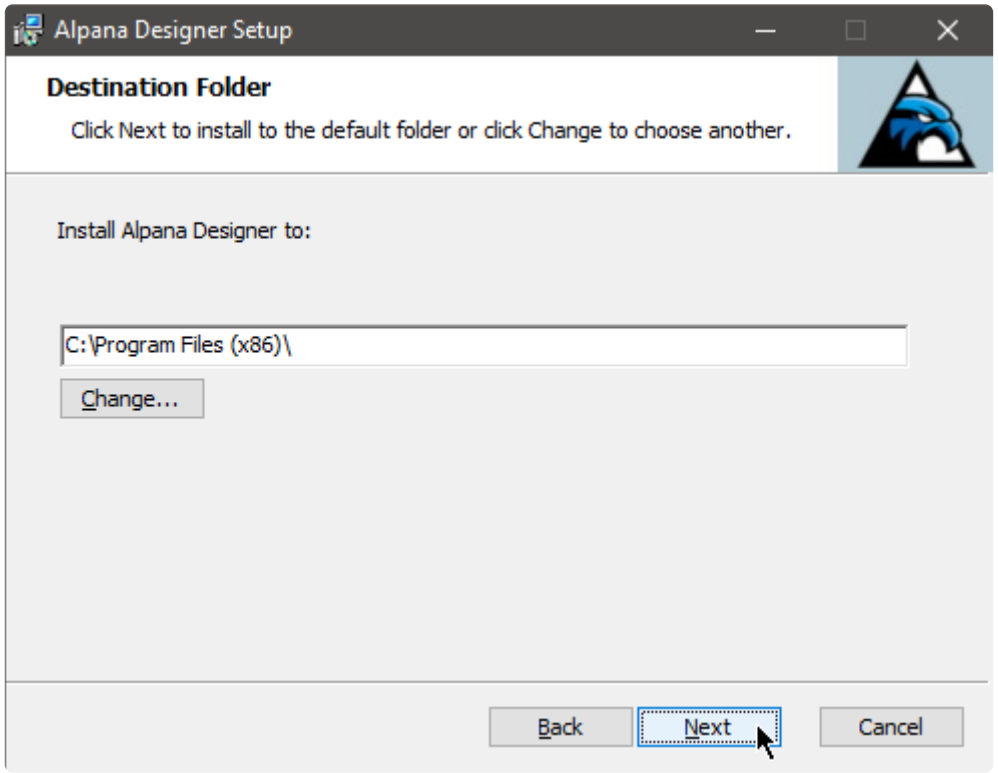


Accept EULA

Read and accept the license agreement, and click *Next* to continue :

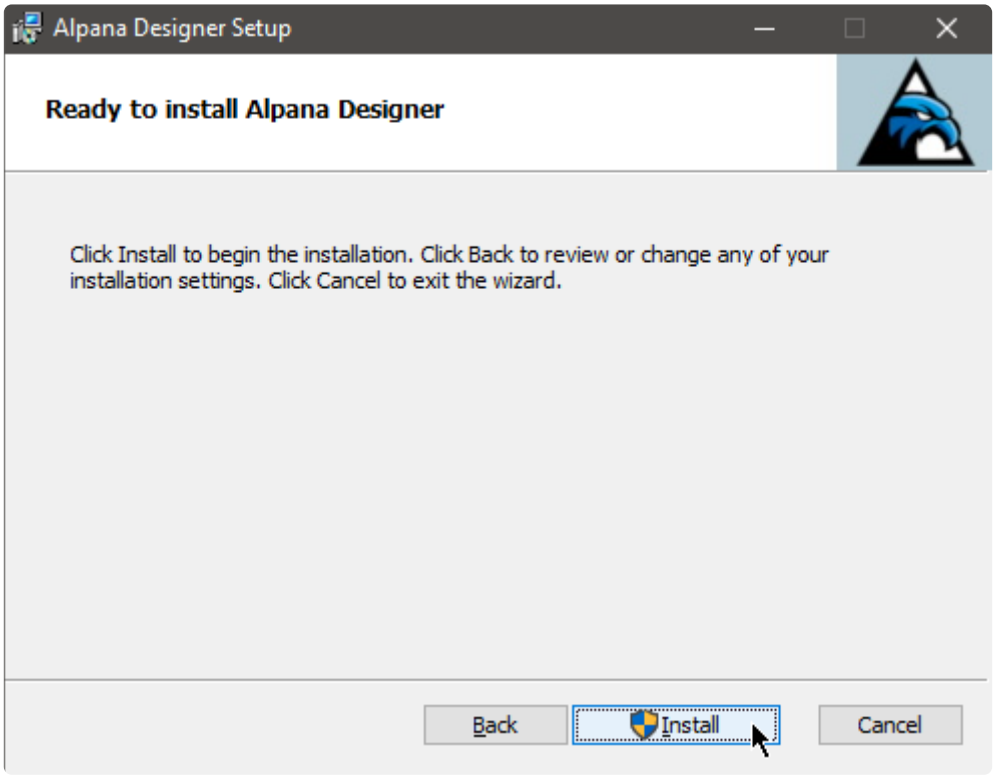
Install location

Validate the install location and click *Next* to continue :



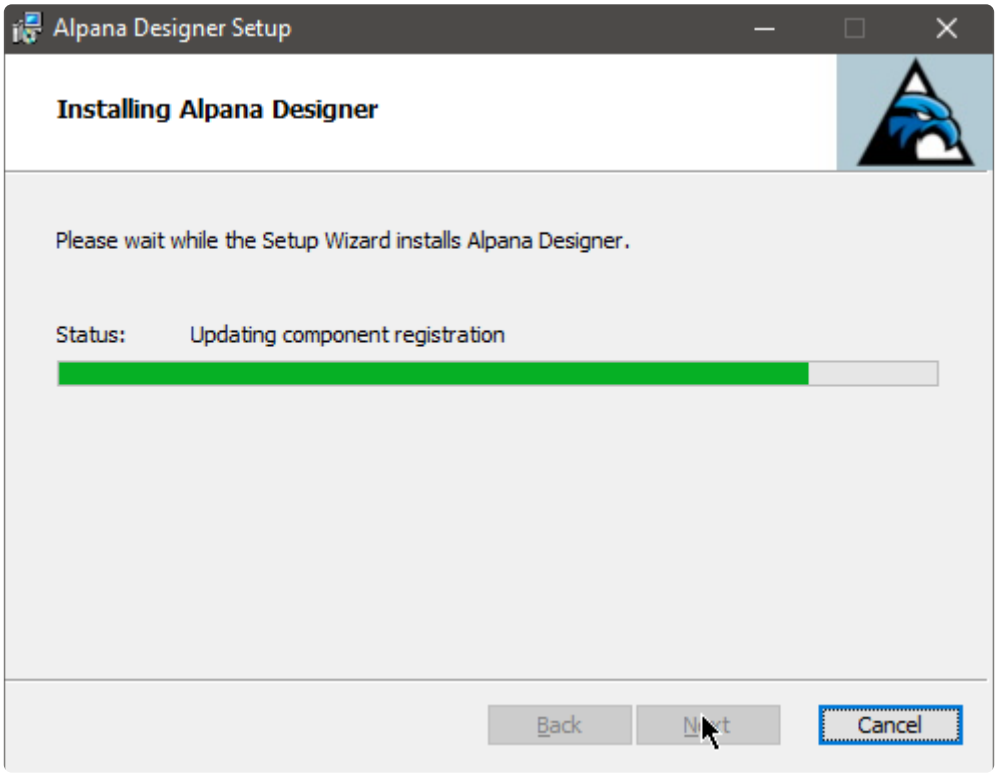
Start installation

Confirm the installation by clicking *Install*. You may be required to enter administrator credentials :



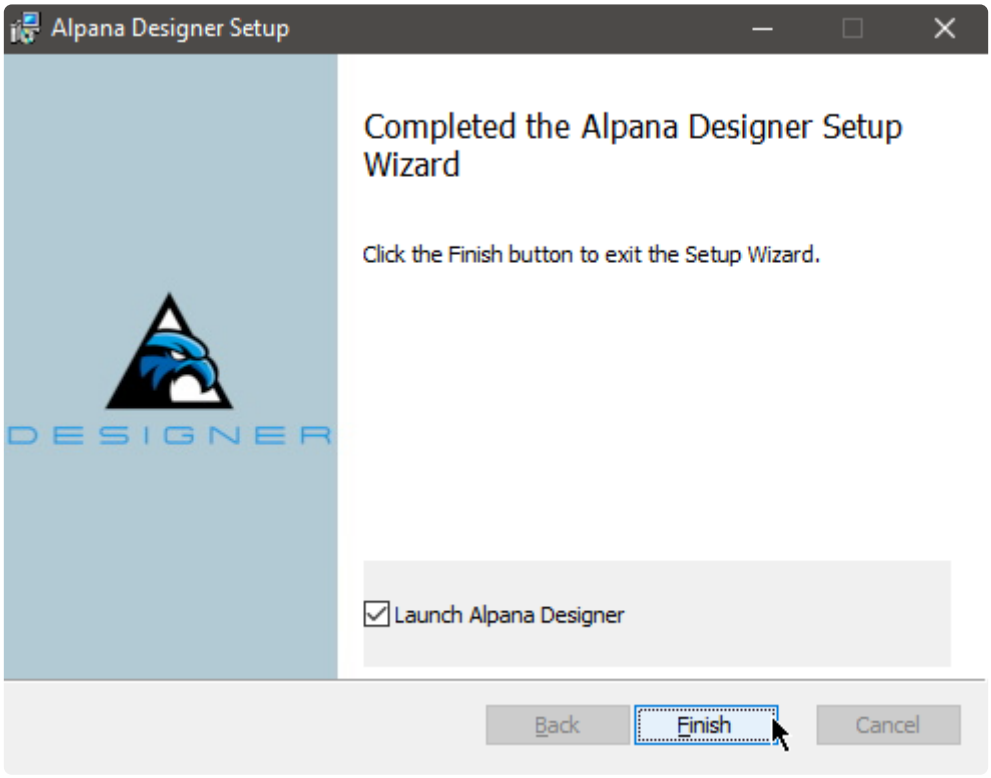
Wait

Wait for the installation to complete :



Complete

When the installation has completed, you can click *Finish* to exit the installer :



Last modified: Apr 30, 2020

2.1.3. Anti-virus and Firewall set up

Ports

Port `5001` must be available for Alpana Designer.
This is where the previews are running.

Anti-virus

Alpana Designer may make queries to external IP addresses (for fetching data from data sources, for example on port 1433 for SQL).
This may be detected as suspicious behavior by some security software.
For normal operation, you should let your security software allow this access to Alpana Designer.

Two processes may need to access external IP addresses :
`code192.Alpana.Designer.WPF.exe` : for the Data Source preview
`code192.Alpana.Designer.Preview.exe` : for the Widget previews

Last modified: Aug 26, 2020

2.1.4. Upgrading Alpana Designer

Upgrading Alpana Designer is straightforward :

1. Un-install any previous version of Alpana Designer.
2. Run the new installer

Last modified: May 12, 2020

2.1.5. Uninstalling Alpana Designer

To uninstall Alpana Designer, use *Windows Control Panel* > *Uninstall a Program* and select *Uninstall*.

Last modified: May 28, 2019

2.2. Troubleshooting

Troubleshooting Alpana Designer involves reading error messages from error logs.
To learn how to find those logs, see next.

Last modified: Apr 16, 2019

2.2.1. Finding error logs

What are error logs

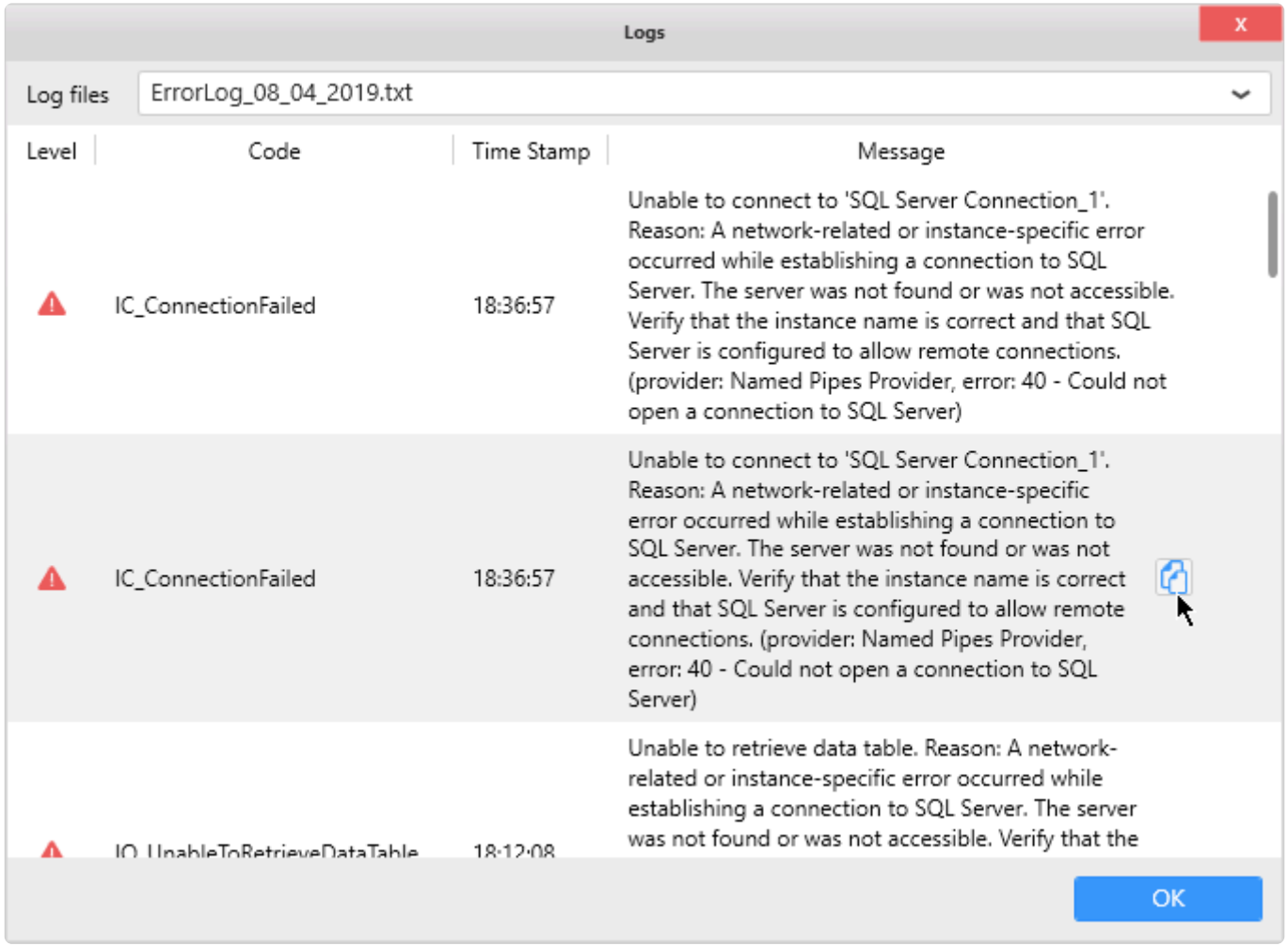
Alpana Designer can encounter errors while configuring Dashboards. For example if the desired query cannot be run at that moment.

For similar reasons, during a Dashboard Preview, a “Data Retrieval Error” can be displayed.
Typically, this will show a *Data retrieval error* message in place of the Widget’s visuals :

Browsing error logs easily

Since Alpana Designer v3.0, you can use a convenient browser for your error logs.
Click on the following icon at the top of the Designer window :

Select an error log file to open (by default, the latest is open), and browse the list of errors for that day.
You can also copy an error log to paste it into a ticket :



Where are error logs

In order to investigate and solve these errors, you can find the error log files at the following location on the machine where Alpana Designer is installed :
`%localappdata%\code192\Alpana Platform\Alpana Designer\(version number)\ErrorLogs\ErrorLog_*.txt`

> Tip :
> To access the Local App Data folder, you can write the following in the location bar of Windows Explorer :
> `%localappdata%`
> Press enter and you will browse the desired location.
> Usually, this folder is located at an address like `C:\Users\(your user name)\AppData\Local\`

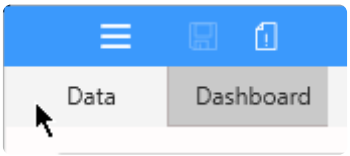
There is one `ErrorLog_*.txt` per day.

Last modified: Oct 31, 2019

2.3. Connecting to Data

Alpana Designer allows to connect to your data using *Connections*.
The current chapter describes this process.

Since Alpana Designer v3.0, all data preparation is made inside the *Data* activity tab.



Last modified: Jun 17, 2019

2.3.1. Alpana data model

Alpana Designer allows to access your data by providing a variety of **Connection** types. Once the appropriate credentials are validated, you will be able to navigate the data structure and retrieve the raw data.

Now, in order to feed your *Dashboard* with data, you will need to choose what data table from your *Connection* will become a **Data Source**.

In a basic workflow, this is enough. However, *Alpana Designer* lets you compose more complex preparation on your data when needed. For example, you can join tables together, including from multiple *Connections*, filter data, create Expressions, …

Connections

What is a Connection

The **Connection** represents how you access the data :

- the Connection *type*, e.g. : Wonderware Historian, SQL Server, Excel, …
- for server-based *Connections* : the *server address*
- for file-based *Connections* : the *file address*
- the necessary *credentials* : authentication mode, user name, password.

How Connections are managed in Alpana Designer

In *Alpana Designer*, you can :

- Create a Connection
- Edit a Connection to modify one of its parameters
- Delete a Connection to remove it from the current Dashboard

Data Sources

What is a Data Source

The **Data Source** is a representation of your data that you prepared from your created *Connection(s)*. It's what your *Widgets* (and your *Dashboard*) see.

A *Data Source* can be created by adding one or more element from one or more *Connection(s)* and joining them together :

- a data Table

- a data View
- a Stored Procedure
- a Custom Query

On top of that, further data preparation can be made :

- hiding or renaming columns to simplify the Data Source usage
- filtering the *Data Source* to limit the data
- creating Expressions in order to make more complex calculations

How Data Sources are managed in Alpana

In *Alpana Designer*, you can fully manage and export Data sources.

Last modified: Apr 30, 2019

2.3.1.1. Note on data storage

Alpana generally doesn't replicate data.

Alpana Dashboards, Data Sources and Connections generally don't contain the data itself. They only contain information on how to query the data sources and prepare the data on the fly.
Each Dashboard contains the definition of the Data Sources and Connections that were created within.

There are some exceptions listed below :

Exceptions

Buffer

A slight exception is the Buffer : for bufferized Connections, the buffer temporarily holds the data from the latest query, but it is flushed on the next refresh.

For details, see the chapter on The data buffer

File-based Connections

For file-based Connections (Excel, CSV, …), Alpana stores a copy of the source file.
This is to allow to continue displaying data even when the original file is not accessible.

When saving a Dashboard.alpd file, the file will contain a copy of all files from the dependent file-based Connections.
So if a dashboard is based only on file data, it can be re-opened in a different environment even without the original files.

When getting data for a file-based Connection, the workflow is :

1. if file exists at location defined in the Connection, query the file
2. else open file embarked withing the Dashboard.alpd file

Last modified: Mar 14, 2019

2.3.1.2. Redundant Connections

What are redundant Connections

Since v3.1.0, Alpana implements a redundancy mechanism on Connections that allows to store multiple connection configurations inside a single Connection.
Each connection configuration is defined by a set of parameters (database name, user name, etc…), called a *Parameter Set*.

This can be used in basically 3 types of scenarios :

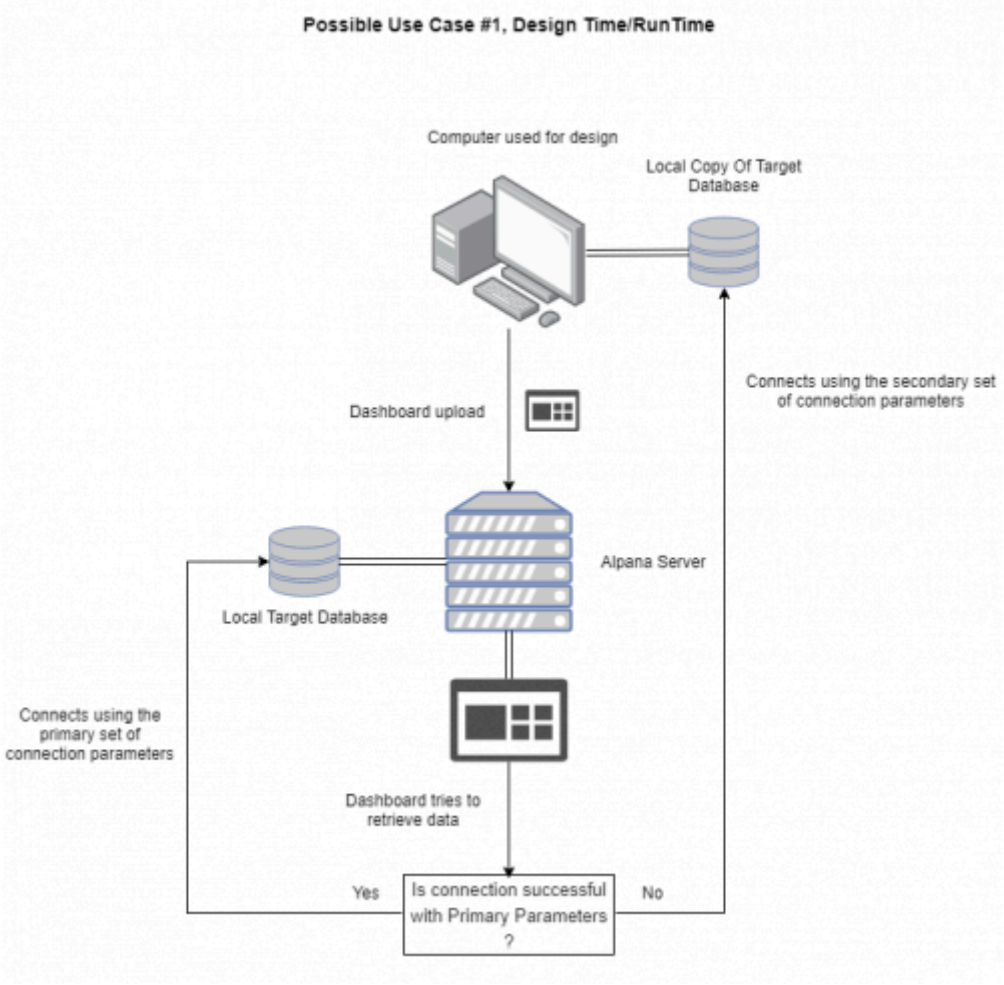
1. Using different databases for design time and run time
2. Implementing a fail-over mechanism to keep Dashboards working even when the primary database is down
3. a combination of scenario 1 and 2

Scenario 1 : Design Time / Run Time

You can select a different Parameter Set for Alpana Designer and Alpana Server.

This is useful for example if the developer doesn't have access to the production database from the development machine, maybe because they are on a different network.
In this case, you can configure 2 different Parameter Sets :

1. one to work in the Designer by pointing on the local development database
2. one for the published Dashboards by pointing to the production database

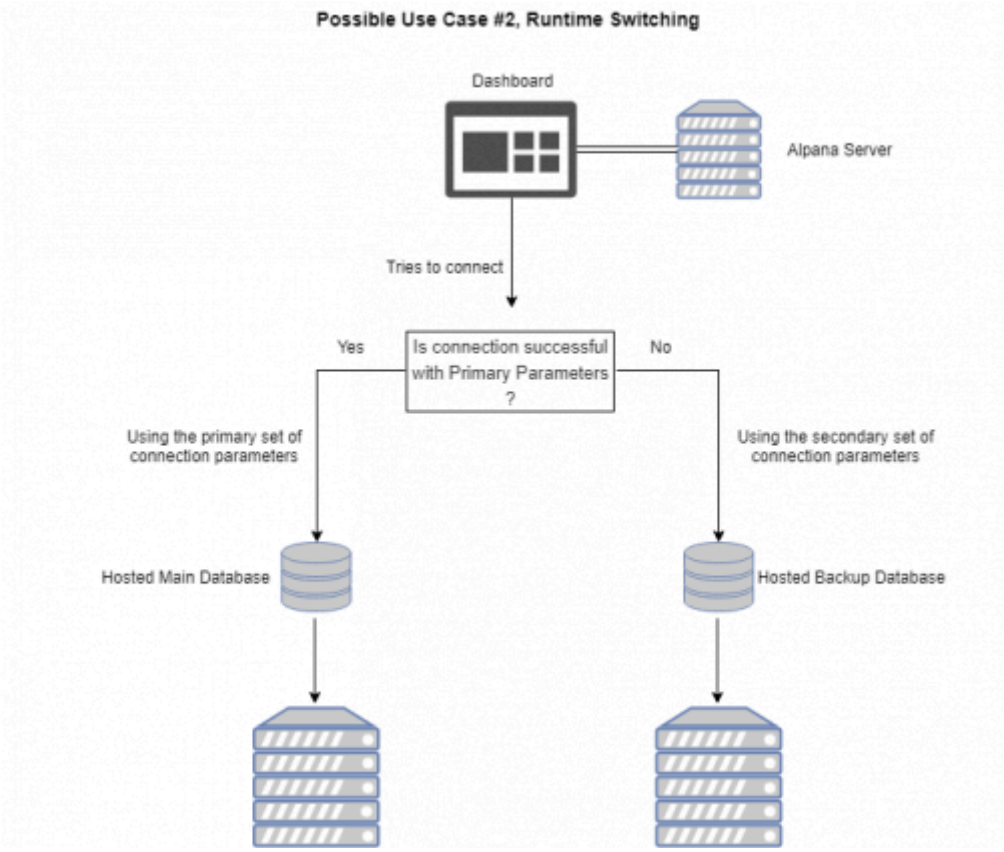


Scenario 2 : Database1 / Database2

The **Primary** *Parameter Set* is where Alpana will try to connect first.
If the connection fails, Alpana will use the **Secondary** *Parameter Set*.

This is useful for example to ensure that Dashboards can always get data, even when the main database is unavailable.
In this case, you can configure 2 different Parameter Sets :

1. one pointing to the main database
2. one pointing to the backup database



Scenario 3 : combine

You can create 3 parameter sets for handling both scenarios above :

1. Parameter Set 1 : primary development
2. Parameter Set 2 : primary production
3. Parameter Set 3 : backup production

Fallback Logic

When a primary connection fails, Alpana will check the next one in order until it succeeds, or until all fail.

> ! Sometimes the connection failure is slow to detect, for example on some network timeouts.
> If the primary connection fails with a 30s timeout, it means the fallback mechanism won't kick in until then.
> This delay will be repeated each time the dashboard is loaded, and until the primary connection is back online.

As a mitigation, all widgets that use the same connection will inherit the fallback.
This means that during a single page load, the timeout will happen only once per Connection.

> Note : there is a caching mechanism, so you may not notice the redundancy immediately.
> When redundancy happens, you may need to refresh the page.
> When the connection becomes good again, you may need to wait a few minutes for the dashboards to apply the primary connection again.

Alpana Designer

In Alpana Designer, the order is :

1. try to connect to the *Primary parameters at design time*
2. if it fails, try to connect to the *Primary parameters after publication*
3. if it fails, try to connect to other parameters

Alpana Server

In Alpana Server, the order is :

1. try to connect to the *Primary parameters after publication*
2. if it fails, try to connect to the *Primary parameters at design time*
3. if it fails, try to connect to other parameters

Connection Types

Redundant connections are supported by the following Connection types :

- SQL
- Historian
- Web

Managing *Parameter Sets*

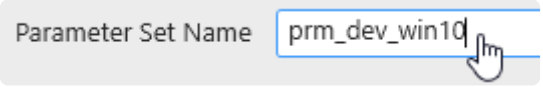
A *Parameter Set* is the set of all the parameters required to connect to the data.

Default behavior

By default, only one Parameter Set is created for the current connection, and is used for both Designer and Server.

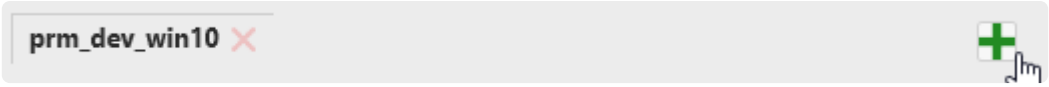
Renaming *Parameter Sets*

You can rename each Parameter Set :



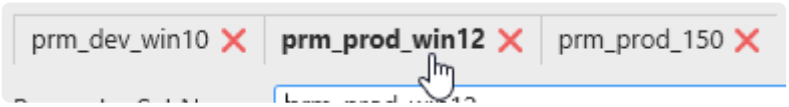
Creating *Parameter Sets*

Click the + icon to create a new Parameter Set :



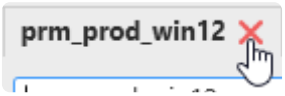
Editing other *Parameter Sets*

To edit a Parameter Set, click on the corresponding tab :



Deleting *Parameter Sets*

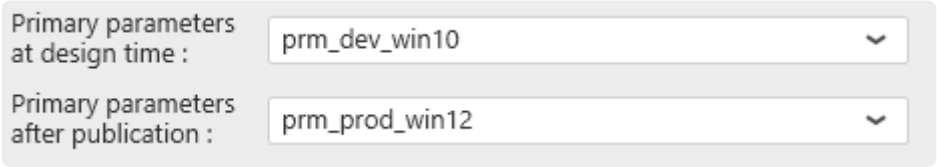
Click on the red cross to delete the corresponding Parameter Set :



Designer ⇔ Server switching

To choose which Parameter Set to use for Designer or Server, select the desired Parameter Set in each list :

1. *Primary parameters at design time* : primary connection to use while in Alpana Designer
2. *Primary parameters after publication* : primary connection to use while published in Alpana Server



Last modified: Aug 23, 2019

2.3.2. The data Buffer

Introduction : What is the Buffer ?

The primary source for data in Alpana is a **database** (SQL Server, Oracle).

When a *Dashboard* requires data from a *Data Source* that is based only on **database *Connections***, the query is composed by *Alpana* and executed directly on the target database.

> ✻ except in multi-source : when the database Connection is used together with another Connection, it also becomes **bufferized** as multi-source (see next page)

On the other hand, Data that comes from other types of *Connections*, need to be **bufferized** : data is first fetched from the source and copied inside a *Buffer database* internal to *Alpana*, then *Dashboards* will query this *Buffer*.

Last modified: May 11, 2020

2.3.2.1. Bufferized Connection types : What gets bufferized ?

All Connection types that are not **databases** are bufferized :

- RESTful API (web)
- Insight
- Alpana Runtime Connections
- file-based Connections : Excel, CSV

Multi-Source case

Multi-Source is a special case : in a Multi-Source context, even database-based Connections are bufferized.

Example :
SQL Connection1 points to your “MESDB” database, and SQL Connection2 points to your “OTHER_DB” database.

You create DataSource1 by making a join between Table1 from Connection1 and Table2 from Connection2.

⇒ Your DataSource1 is a **Multi-Source** Data Source. It is bufferized.

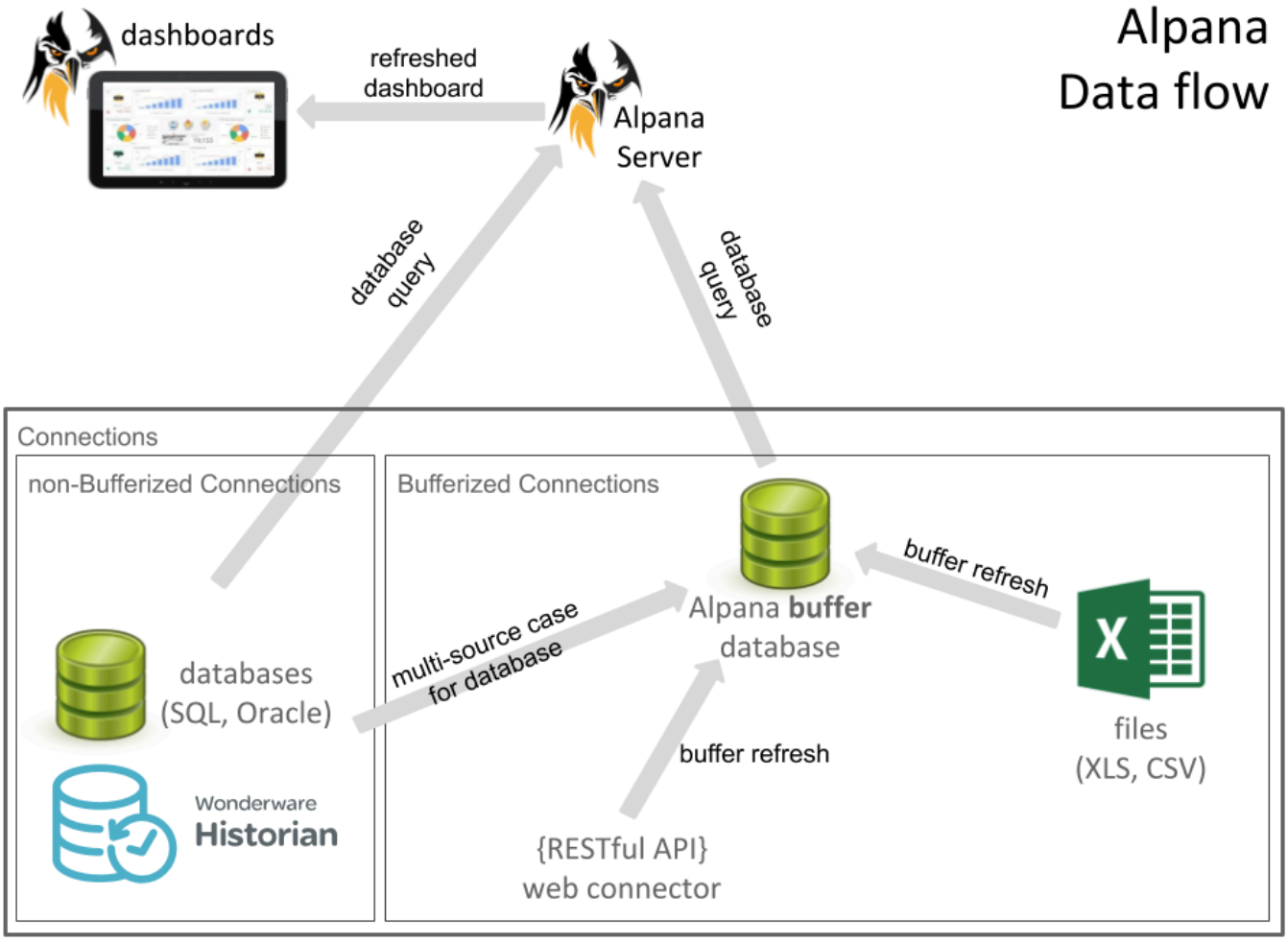
Last modified: Jun 16, 2020

2.3.2.2. Data Flow : How bufferization happens ?

Data flow

When data is requested from a bufferized *Connection*, the following happens :

- If **Buffer Refresh** is not enabled : data is retrieved from the buffer.
- If **Buffer Refresh** is enabled :
  - If the buffer is valid ⇒ data is retrieved from the buffer.
  - If the buffer is out of date ⇒ the buffer is refreshed by data from the Connection (the Web connection, the CSV file, etc…).



Content of the Alpana buffer database

Alpana Server creates, deletes and updates tables in the buffer database.

This is a purely internal mechanism and should not be relied on by external tools.

> ! You shouldn’t use the buffer database for other purposes, otherwise you might lose your data.

Last modified: Mar 12, 2019

2.3.2.3. Buffer database setup

In Alpana v3.0.0, the Buffer database is setup automatically :

- for the Alpana Server : this setting applies to published Dashboards. A Microsoft SQL database is created by Alpana Server to host the buffer tables.
- for each Alpana Designer installation : this setting applies when editing Dashboards on that host. A SQLite database is created in a local folder.

Last modified: May 28, 2019

2.3.2.4. Buffer Refresh settings

What is Buffer Refresh Settings

This setting represents : How long do you want to query the old data in the buffer instead of getting new data from the source ?

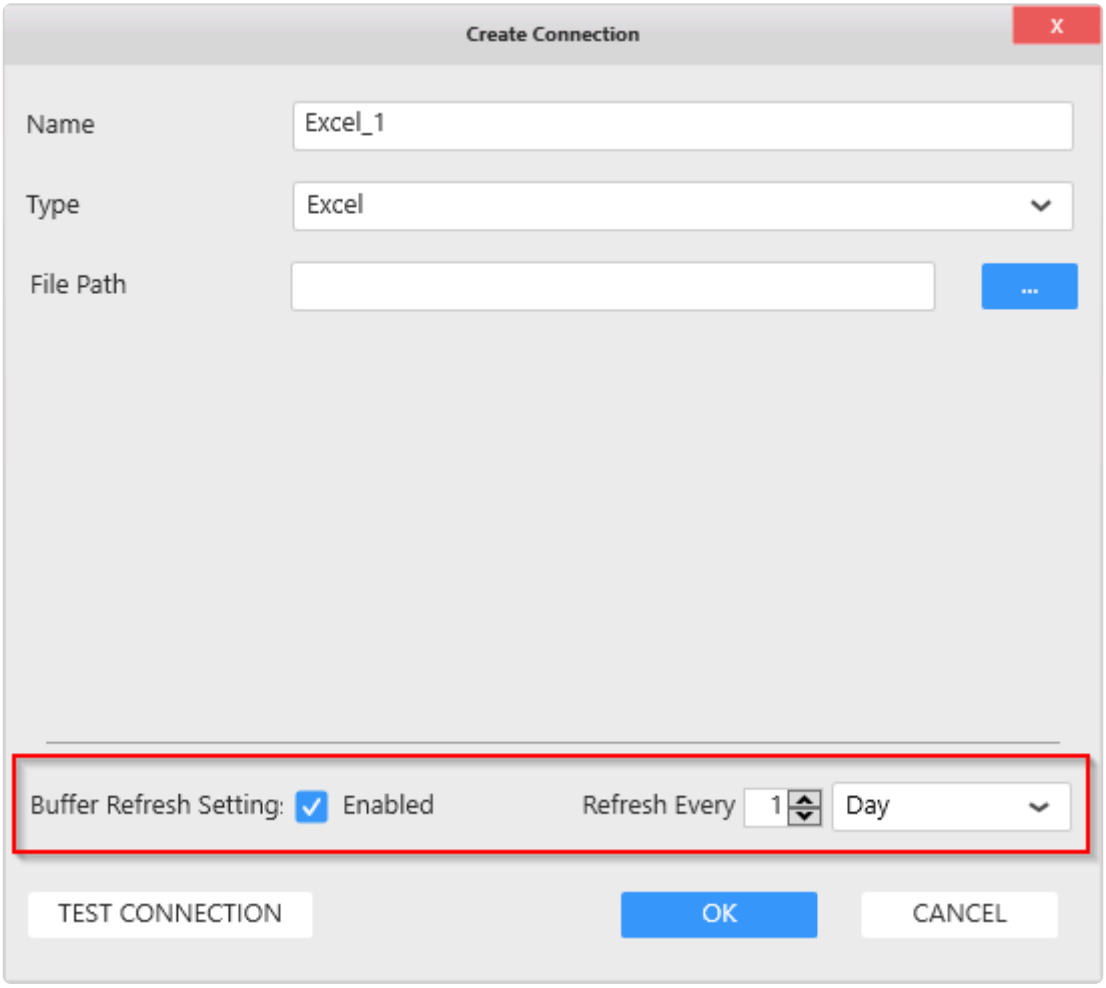
For example

You connect to an Excel file.
The data is first copied in the buffer.
If you don’t set up Buffer Refresh Settings, it will never be updated.
If you set up “Refresh Every 1 Week”, it will not be update for 1 Week.
Then, the next time someone makes a query to this Connection, new data is copied in the Buffer.

Where to set up

Each *Connection* has its own refresh settings.

You can change the refresh duration by default at the Connection creation (or edition).
Just edit the Connection, and set up “Buffer Refresh Settings” as Enabled, and a duration to “Refresh Every”.



Note : don't forget to set Buffer Refresh settings also for a SQL Connection if you intend to use it in a multi-Connection Data Source !

Last modified: Mar 12, 2019

2.3.2.5. refreshing file-based connections

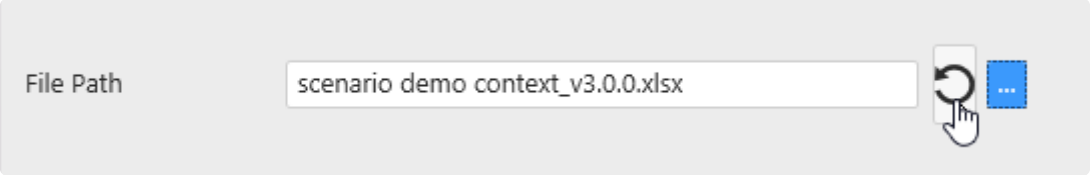
Published Dashboards

> Once published, the file-based connections **don't refresh** based on the file location.

If you want to refresh data for a dashboard based on a file (CSV, Excel, JSON), use the external tool Alpana Update File Dependency

Refreshing in the Designer

Inside the Designer, you can manually force the refresh by Editing the Connection and clicking the refresh button next to the file name :



This will import the file changes inside the Designer, regardless of the Buffer Refresh settings.

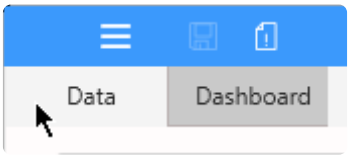
> Note : this only works if you configured the file name during the current Designer session, because the file name is not stored

Last modified: May 26, 2020

2.3.3. Creating and Managing Connections

Alpana Designer allows to create as many Connections as required.

Work with Connections is all done inside the *Data* activity tab of Alpana Designer :

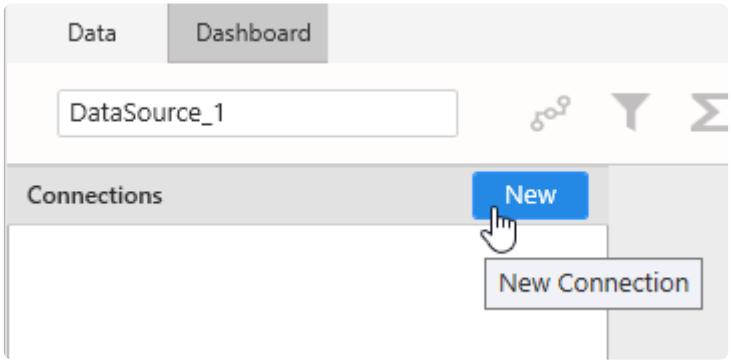


To learn how to create and manage Connections, see the current chapter.

Last modified: Apr 19, 2019

2.3.3.1. Create a Connection

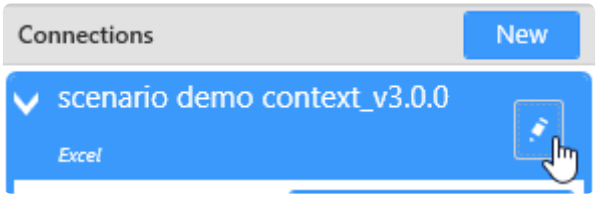
To create a new Connection, click *New* :



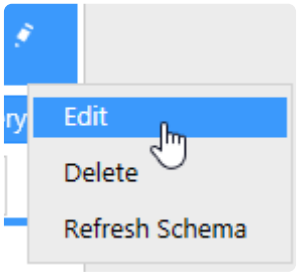
Last modified: Apr 16, 2019

2.3.3.2. Edit a Connection

To open a Connection to start to Edit it, click the pen icon next to the Connection name :



From the pop-up, select *Edit* :

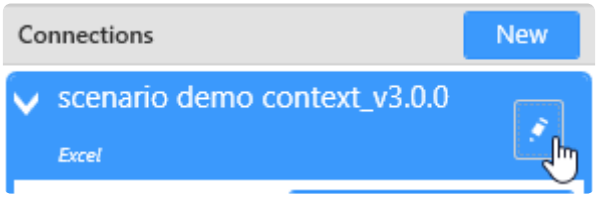


Then the corresponding “Edit Connection” dialog opens for edition (see below for each Connection type).

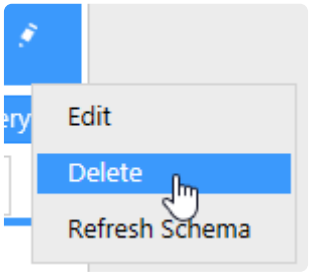
Last modified: Apr 16, 2019

2.3.3.3. Delete a Connection

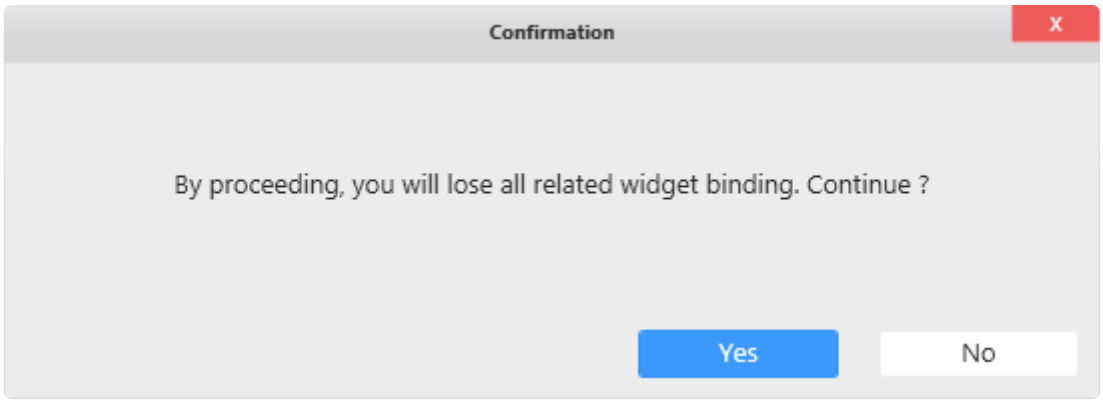
To Delete an existing Connection, click the pen icon next to the Connection name :



From the pop-up, select *Delete* :



If the Connection is used by any Widget or any Data Source, the following message will appear :



If you choose “Yes”, the Connection is deleted and all dependent Data Sources and Widgets lose their data binding.

Last modified: Apr 16, 2019

2.3.4. Connection types

All connection types are listed in the next chapters, with all their configuration properties.

Last modified: May 15, 2019

2.3.4.1. SQL Server

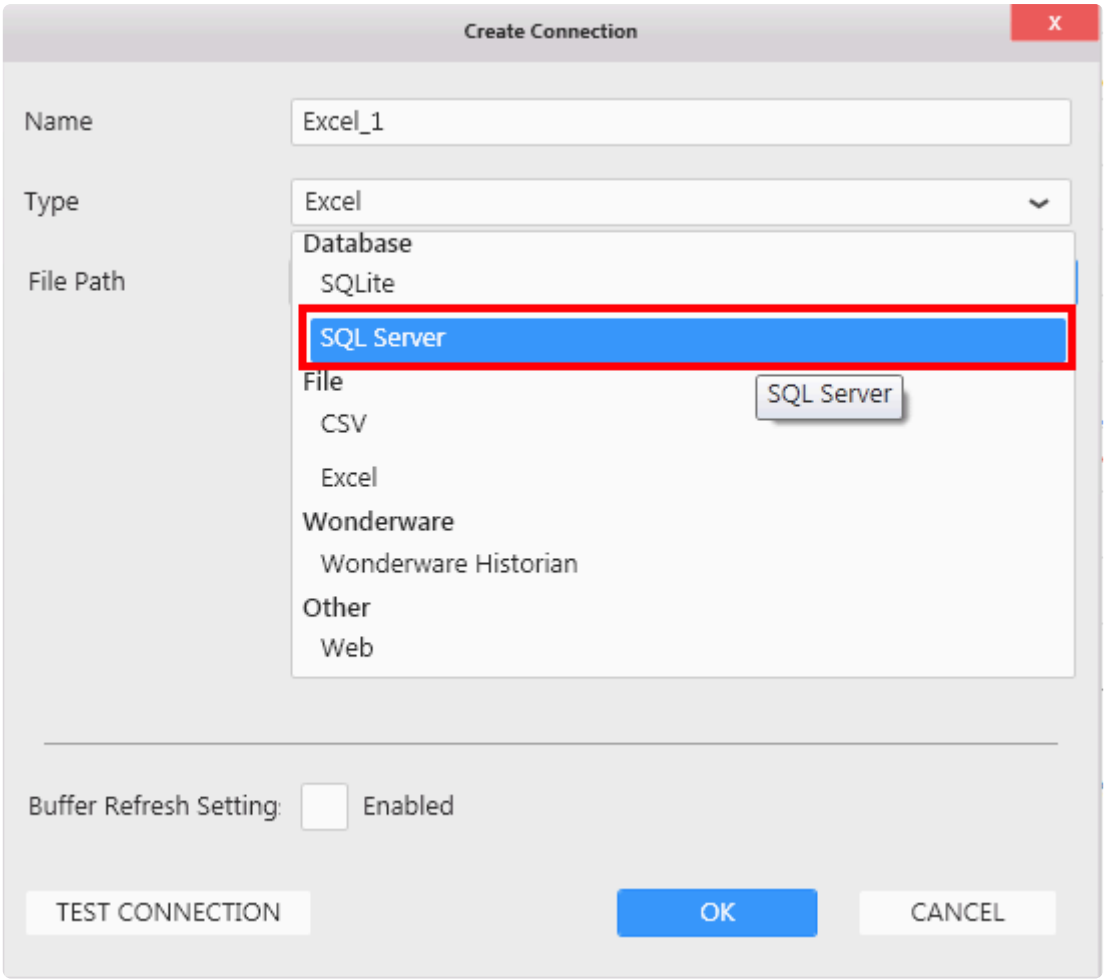
Versions

This connector can be used to get data from local or remote Microsoft SQL Server installations with the following versions :

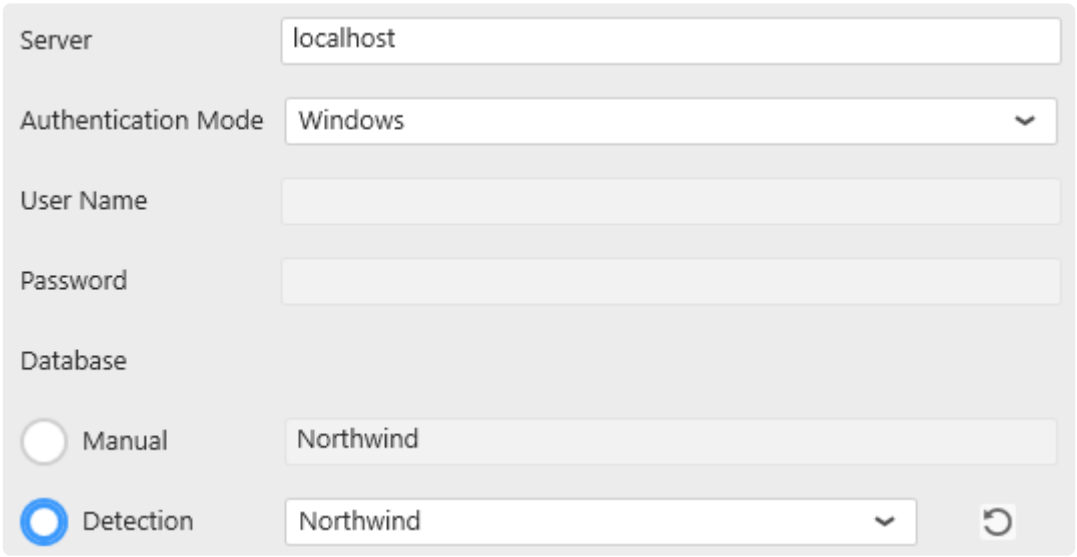
- Microsoft SQL Server 2012 and later
- Microsoft SQL Server Express 2012 and later

Selecting Connection Type

In the “Create Connection” dialog change the Connection “Type” by selecting “*SQL Server*” :



Connection Parameters



Server

This is the address of the SQL Server instance you need to access. Possible values are :

- `(local)` or `localhost` : local server
- `<NetBIOSName>` : server named with a NetBIOS name
- `<IPv4Address>` : server accessible through a IP v4 address
- `<IPv6Address>` : server accessible through a IP v6 address
- `<server address>\<instance name>` : connect to a specific instance. For SQL Express, it may be necessary to use `<server address>\SQLEXPRESS`
- `<server address>,1433` : 1433 is the default port for SQL Server, but can be changed like this Be aware that when searching for a server instance, Alpana waits for a response. If the server instance doesn't respond, there is a rather long timeout.

Authentication Mode

"SQL Server" : will use SQL Server authentication with "User name" and "Password".

"Windows" : will use Windows Authentication from the Windows user that generates the query.

Important note on Windows Authentication

> ! Important :
> *Windows Authentication* will use a **different Windows user** when in Designer and when published in Server.
> That Windows user may not have the required access to SQL Server.
> This will result in **Data Retrieval Error** messages.

When published in Alpana Server, the Windows user will be that configured in the IIS site : `IIS APPPOOL\Alpana3App` (see the Server Architecture chapter).

This is because there are 2 different threads owned by 2 different Windows users :

- the Designer is in a thread owned by your current Windows user
- the Server is in a thread is owned by the IIS application pool identity : `IIS APPPOOL\Alpana3App`

In order to allow Windows authentication on published Dashboards to a database X on SQL Server Y, you need to :

- make sure the `IIS APPPOOL\Alpana3App` SQL user exists on SQL Server Y, and that the corresponding Windows user is bound. If you installed Alpana Server with SQL authentication, or if SQL Server Y is on another SQL Server, then you will have to manually create this user in SQL.
- make sure the `IIS APPPOOL\Alpana3App` SQL user has the required permissions on database X. For example, this user will need `db_owner` permissions to run stored procedures or functions.

Database

The name of the database to connect to.

Detection

This lets you select the database name from the list of databases.

> Tip : After changing credentials, you can fetch again a new list of database names, by clicking the refresh icon.

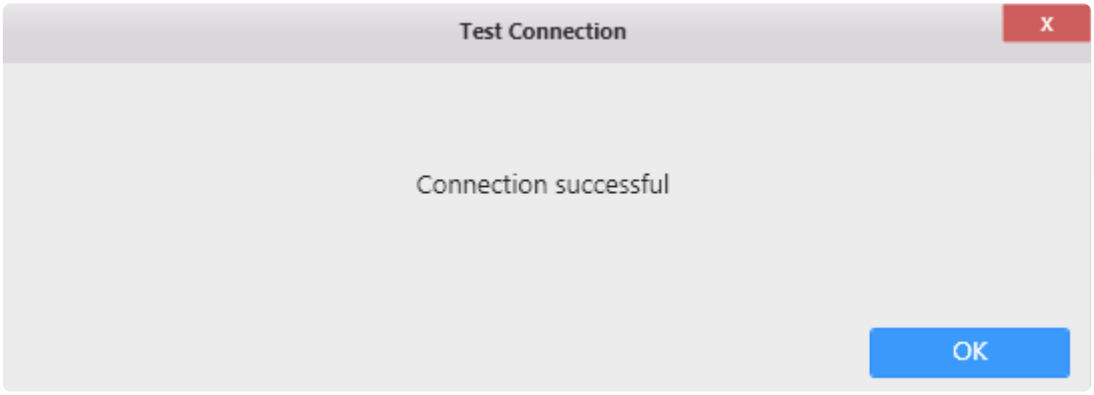
Manual

This lets you write the database name as free text.

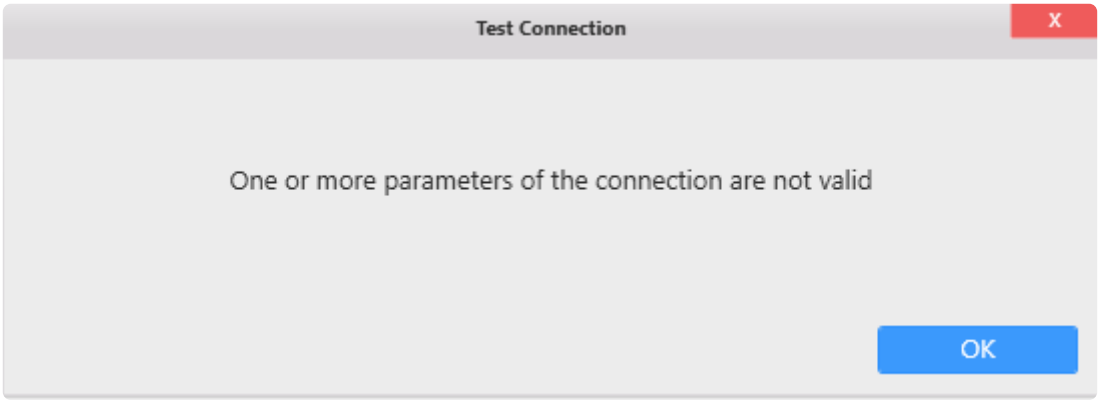
This is useful for creating a connection to a database which you cannot access while in the Designer. For example when the database name is different in the production environment.

Test Connection

To test the connection for validity, click “Test Connection” button. The following confirmation message will confirm its validity.



If the connection is invalid, the following message appears :



Then you will need to fix the Connection parameters (credentials, server name, …) or check network connectivity (is the SQL Server accessible from here ?)

Connection Schema

The SQL Server Connector fetches the following object types in the database schema :

- Tables
- Views
- Stored Procedures
- Table-Valued Functions

Data refresh and Buffer

The SQL Connector is not bufferized by default.

However, when data is joined with another Alpana Connection (including another SQL Connection), data is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: May 13, 2020

2.3.4.2. Excel

Formats

You can connect to Microsoft Excel workbooks in format `*.xlsx` or `*.xls` containing “database-like” tabular data.

The Excel workbook should have column names at first row of sheets followed by data rows.

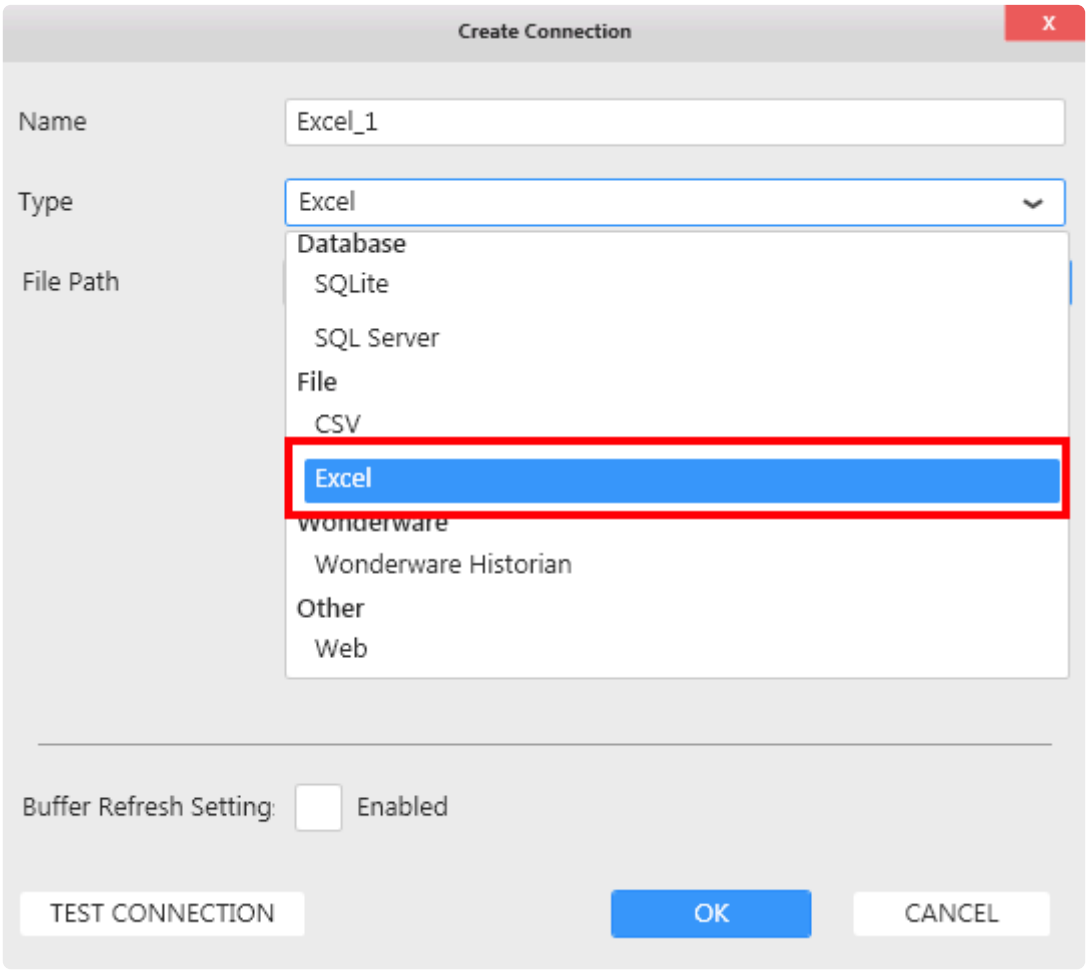
The data type of all values in a column should all be the same. Trying to parse an Excel file where a column has a number as first value and string as second value will result in a parsing error in SQL buffer.

Other complex sheet designs may not be supported.

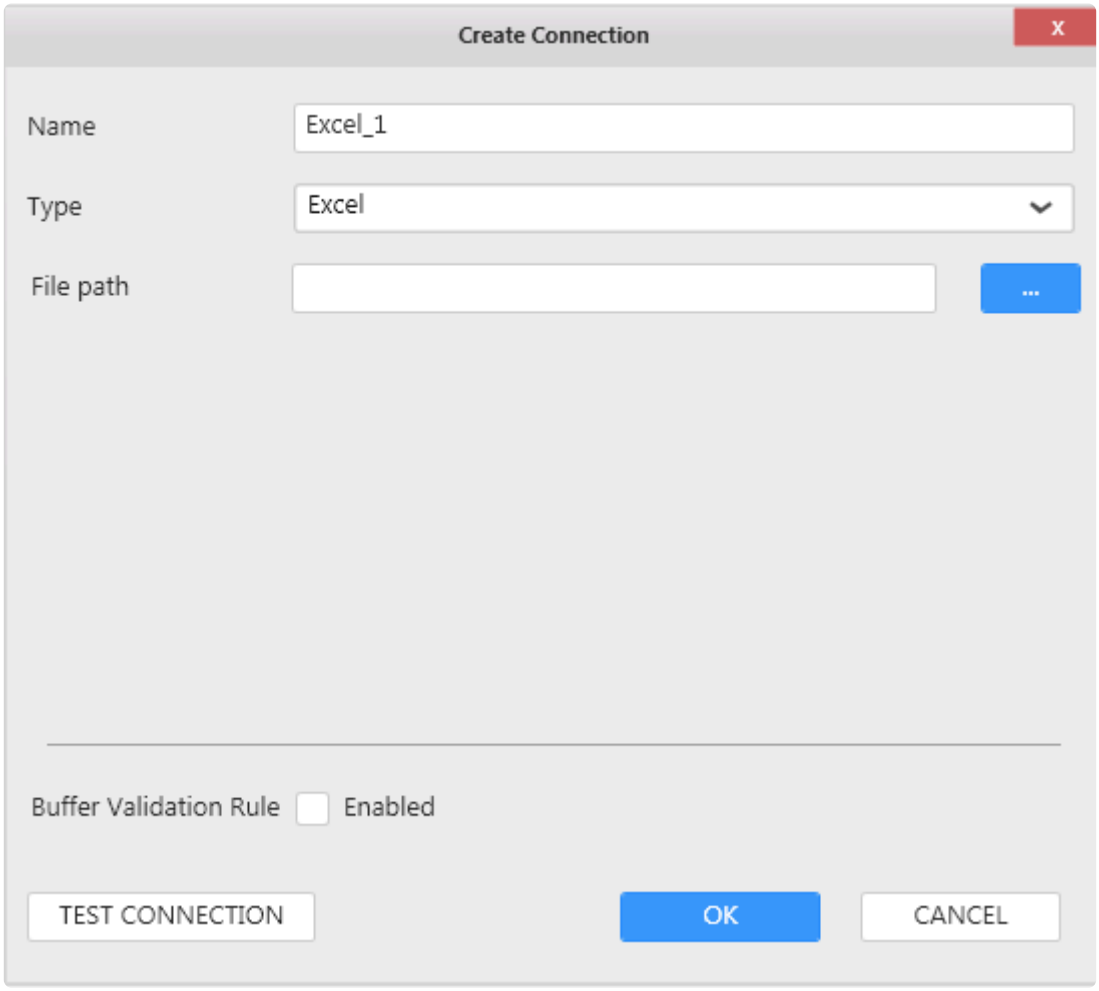
Formulas are executed every time the file is opened.

Selecting Connection Type

In the “Create Connection” dialog change the Connection “Type” by selecting “*Excel*” (should be selected by default) :



Connection Parameters



File Path

Click to browse for an Excel file.

> **!** Important :
> The path configured here may not be accessible when published in Alpana Server for different reasons (maybe it's a different machine and the Excel file is not there at the same path, maybe the Windows user doesn't have access to the path).
> When published in Server, the Windows user will be that configured in the IIS site, by default "LocalSystem", which can correspond to "NT AUTHORITY\SYSTEM".
> When the file is not found at the configured path, Alpana uses the copy of the Excel file that is embarked within the Dashboard. If so, then you will not see your data update.

Connection Schema

The Excel Connection presents Excel Sheets as tables to be used in a Data Source.

Data refresh and Buffer

The Excel Connector is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: Mar 12, 2019

2.3.4.3. CSV

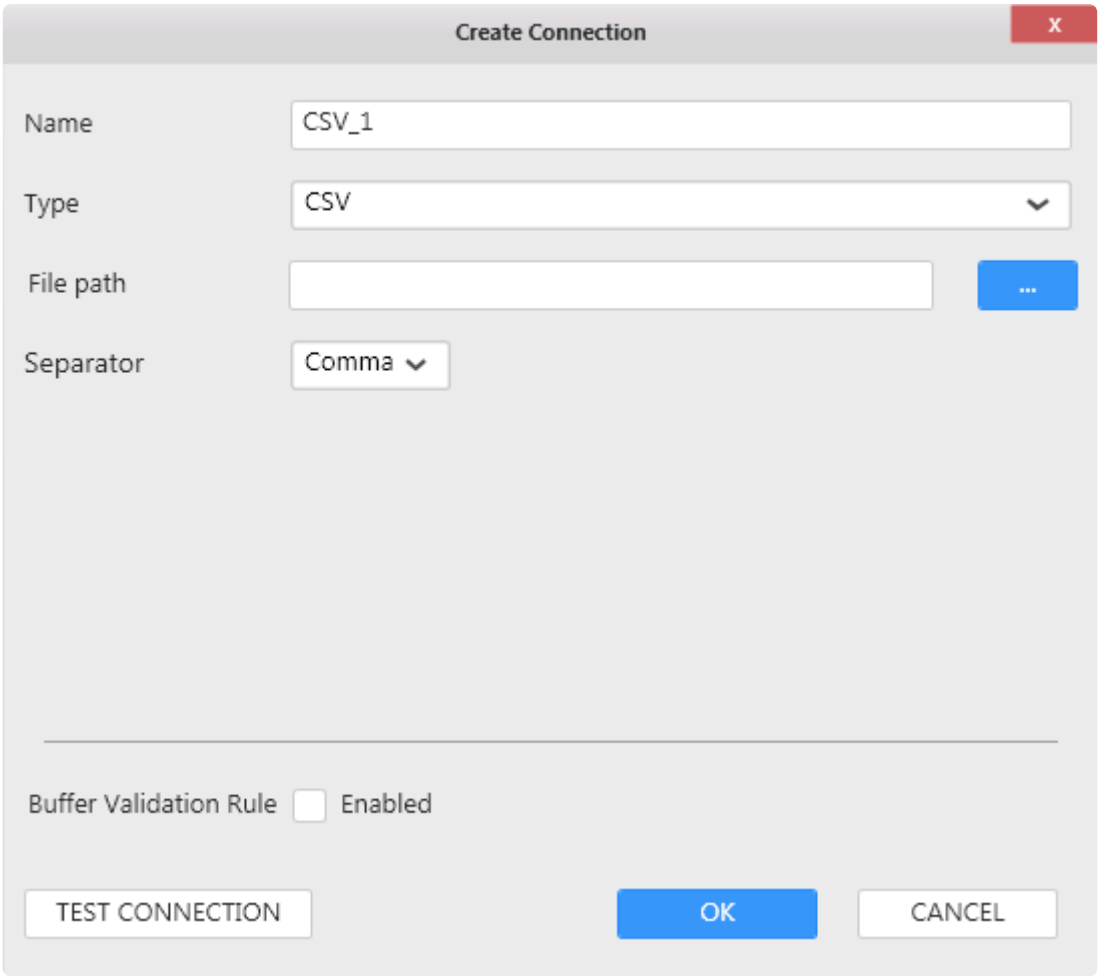
Formats

Although “CSV” means “Comma-Separated Values”, Alpana supports other separators than comma :

- Comma : “ , “
- Semicolon : “ ; “
- Space : “ “
- Tab : “ “

Selecting Connection Type

In the “Create Connection” dialog change the Connection “Type” by selecting “CSV” :



Connection Parameters

File Path

Click to browse for a CSV file.

> **!** Important :
> The path configured here may not be accessible when published in Alpana Server for different reasons (maybe it's a different machine and the CSV file is not there at the same path, maybe the Windows user doesn't have access to the path).
> When published in Server, the Windows user will be that configured in the IIS site, by default "LocalSystem", which can correspond to "NT AUTHORITY\SYSTEM".
> When the file is not found at the configured path, Alpana uses the copy of the CSV file that is embarked within the Dashboard. If so, then you will not see your data update.

Connection Schema

The CSV Connection presents a single object in the schema : a table with the same name as the file.

Data fields type

Since CSV doesn't allow to create explicit data types, Alpana has to detect the data types of fields.
If this automatic detection is incorrect, you can always change the data type in the Data Source : Changing Column Type

> **!** Integers have a size limit. If this causes an issue, you can switch to the "Number" data type.

Data refresh and Buffer

The CSV Connector is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: Mar 21, 2019

2.3.4.4. Wonderware Historian

Specific documentation to connect to Wonderware Historian is available here : Wonderware Historian

Last modified: Mar 13, 2019

2.3.4.5. Web

Formats

The Web Connector allows to get data from RESTful Web Services that serve JSON content.

Supported response content include the three most commons standards: JSON:API, ODATA, JSEND

JSON:API format example:

```
{
  "links": {
    "self": "http://example.com/articles"
  },
  "data": [
    {
      "type": "articles",
      "id": "1",
      "attributes": "att1"
    },
    {
      "type": "articles",
      "id": "2",
      "attributes": "att2"
    }
  ]
}
```

With this format, the connector will create a one table schema. This table will be named “data”.

ODATA Example:

```
{
  "@context": "http://host/service/$metadata",
  "value": [
    {
      "name": "Orders",
      "kind": "EntitySet",
      "url": "Orders"
    },
    {
      "name": "OrderItems",
      "title": "Order Details",
      "url": "OrderItems"
    },
    {
      "name": "TopProducts",
      "title": "Best-Selling Products",
      "kind": "FunctionImport",
      "url": "TopProducts"
    }
  ]
}
```

Similarly to the JSON:API format, the connector will create a one table schema. This table will be named “value”.

JSEND Example:

```
{
  "status": "success",
  "message": "Data found",
  "data": {
    "requests": [
      {
        "id": 10,
        "request_code": "8B2BD1",
        "provider_name": "Al Marasim Ladies Salon",
        "provider_name_ar": "Al Marasim Ladies Salon",
        "provider_name_en": "Al Marasim Ladies Salon",
        "service_name": "Nails & Feet",
        "start_datetime": "2018-09-04 18:03:07",
        "status_label": "Pending",
        "status": 1
      },
      {
        "id": 8,
        "request_code": "8713ED",
        "provider_name": "Provider",
        "provider_name_ar": "Provider",
        "provider_name_en": "Provider",
        "service_name": "Nails & Feet",
        "start_datetime": "2018-09-04 18:03:07",
        "status_label": "Pending",
        "status": 1
      }
    ]
  }
}
```

For this exact format, the Web connector would create a one table schema with one table named “request”. However, depending on how the JSEND API was programmed, the sub-object under data could be named differently e.g “payload”, “post” or basically any name the original developers decided to use. In this cases, the connector will name the table accordingly to the keyword used.

In addition to these three standard formats, the Web connector also support multiple non-standard json responses.

Simple Array:

```
[
  {
    "Id": 8569,
    "Module": "Production",
    "Shift": "Noon",
    "Staff": "Captain Marvel",
    "Kiln_Feed": 89.6533838600779
  },
```

```
  {
    "Id": 8563,
    "Module": "Production",
    "Shift": "Noon",
    "Staff": "Captain America",
    "Kiln_Feed": 89.3475083300634
  },
  {
    "Id": 8556,
    "Module": "Production",
    "Shift": "Noon",
    "Staff": "Captain America",
    "Kiln_Feed": 88.3035192802069
  }
]
```

With this type of response the connector will create a one table schema with one table named “data array”.

Multiple Arrays

```
{
  "TableA": [
    {  // data record 1 for TableA
      "FieldA1": "ValueA1_1",
      "FieldA2": "ValueA2_1",
      ...
    },
    {  // data record 2 for TableA
      "FieldA1": "ValueA1_2",
      "FieldA2": "ValueA2_2",
      ...
    },
    {  // data record 3 for TableA
      "FieldA1": "ValueA1_3",
      "FieldA2": "ValueA2_3",
      ...
    },
    ...
  ],
  "TableB": [
    {  // data record 1 for TableB
      "FieldB1": "ValueB1_1",
      "FieldB2": "ValueB2_1",
      ...
    },
    ...
  ],
  ...
}
```

Here a multi-table schema will be created with one table for each object array present in the response.

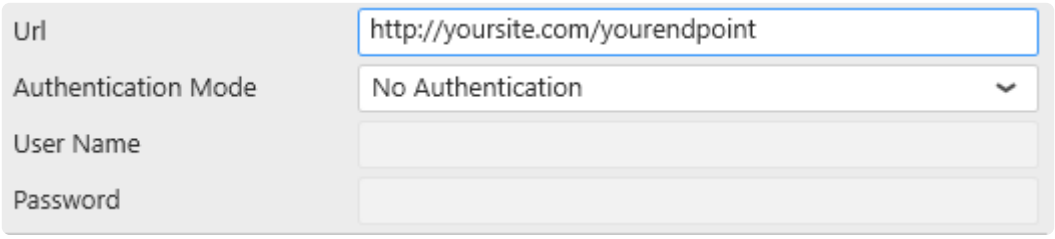
Data Property

```
{
  "cityid": "101010100",
  "date": "201805",
  "cityEn": "beijing",
  "avgbWendu": "28",
  "avgyWendu": "16",
  "maxAqi": "186",
  "minAqi": "99",
  "maxAqiLevel": "4",
  "minAqiLevel": "1",
  "data": [
    {
      "ymd": "2018-05-01",
      "aqi": "121",
      "aqiLevel": "3"
    },
    {
      "ymd": "2018-05-02",
      "aqi": "44",
      "aqiLevel": "1"
    }
  ]
}
```

If no common standard is recognized, the connector will look for a “data” property. In this response exemple, the connector would create a single-table schema with a table named “data”.

Selecting Connection Type

In the “Create Connection” dialog change the Connection “Type” by selecting “*Web*” :



Connection Parameters

URL

Enter the API URL which must be a valid REST API as seen above.

Authentication Type

Select the authentication to use :

- “No Authentication”

- “Basic HTTP Authentication” : with User name and Password

Data refresh and Buffer

The Web Connector is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: Jun 18, 2020

2.3.4.6. Oracle

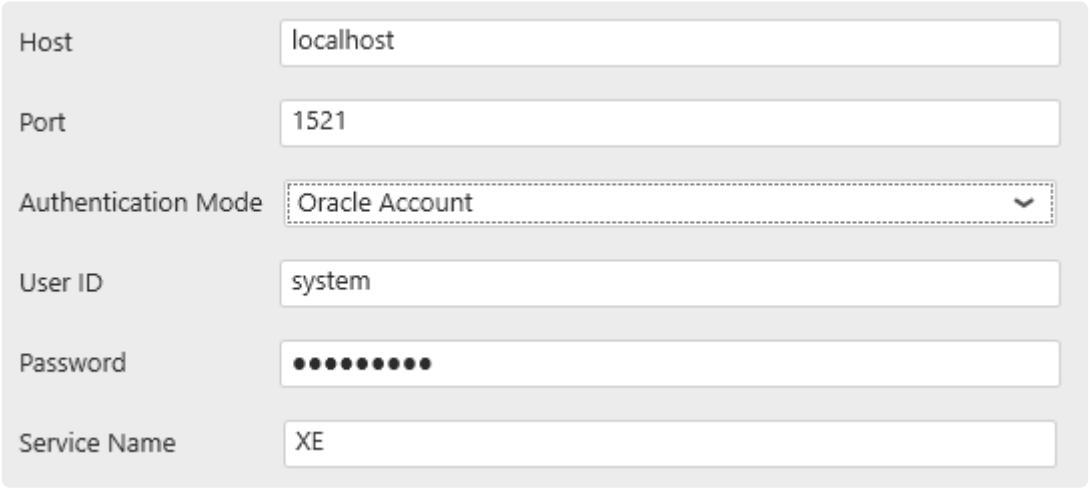
Versions

This connector can be used to get data from local or remote Oracle Database installations with version 12 or above.

Selecting Connection Type

In the “Create Connection” dialog change the Connection “Type” by selecting “*Oracle*” :

Connection Parameters



Host

This is the address of the Oracle instance you need to access. Possible values are :

- `localhost` : local server
- `<IPv4Address>` : server accessible through a IP v4 address

Be aware that when searching for a server instance, Alpana waits for a response. If the server instance doesn’t respond, there is a rather long timeout.

Authentication Mode

“Oracle Account” : will use Oracle authentication with “User ID” and “Password”.

“Windows” : will use Windows Authentication from the Windows user that generates the query.

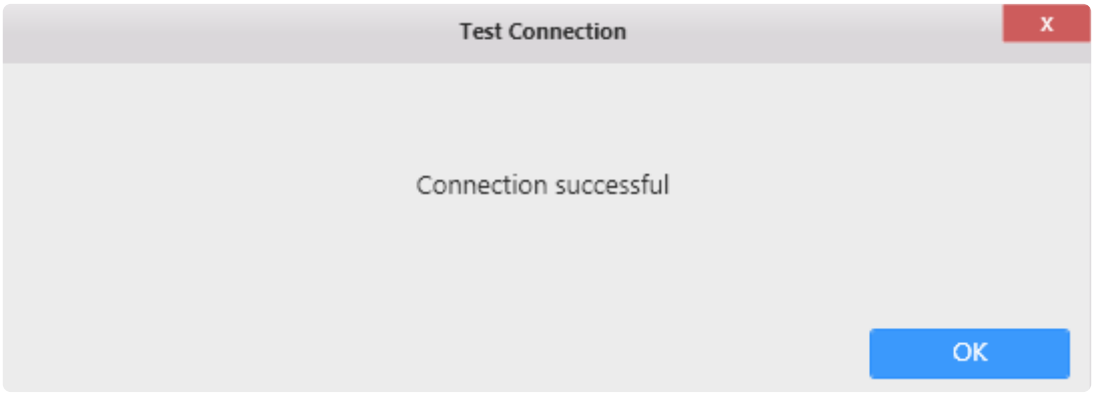
> ! Important :
> *Windows Authentication* will use a **different Windows user** when in Designer and when published in Alpana Server. That Windows user may not have the required access to Oracle.
> When published in Alpana Server, the Windows user will be that configured in the IIS site `IIS APPPOOL\Alpana3App`.

Service Name

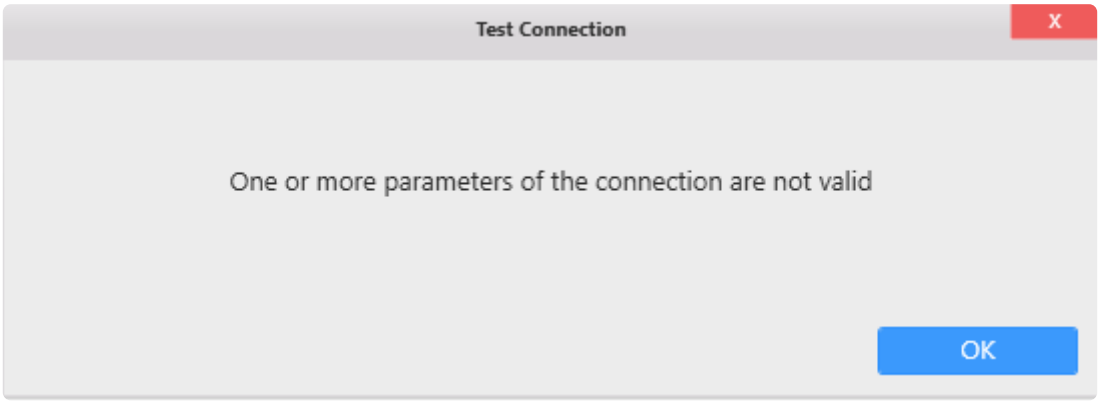
The name of the Service to connect to.

Test Connection

To test the connection for validity, click “Test Connection” button. The following confirmation message will confirm its validity.



If the connection is invalid, the following message appears :



Then you will need to fix the Connection parameters (credentials, server name, …) or check network connectivity (is the Oracle accessible from here ?)

Connection Schema

The Oracle Connector fetches the following object types in the database schema :

- Tables
- Views
- Stored Procedures

Data refresh and Buffer

The Oracle Connector is not bufferized by default.

However, when data is joined with another Alpana Connection (including another Oracle Connection), data is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: Oct 15, 2019

2.3.4.7. Alpana Runtime

Introduction

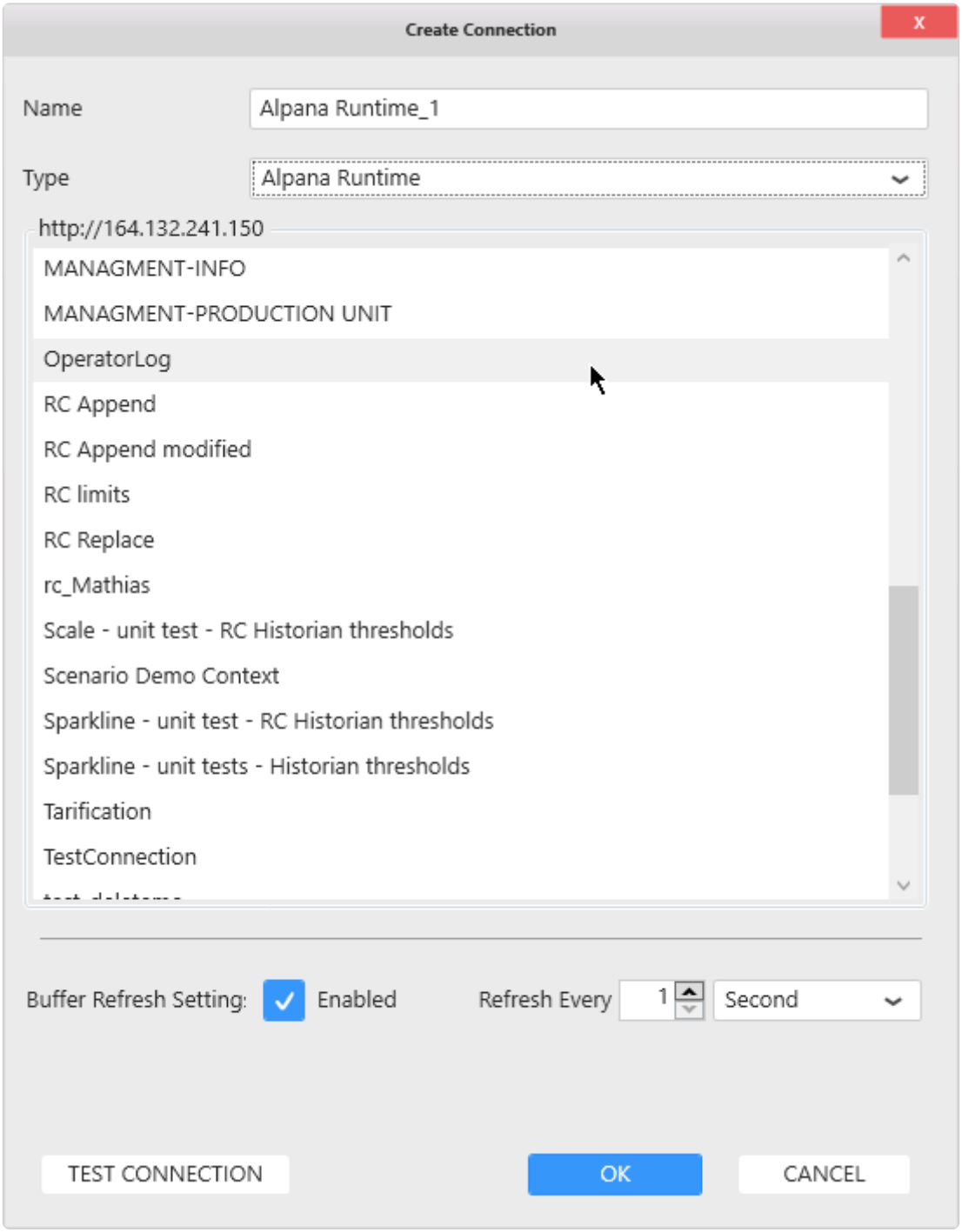
The *Alpana Runtime* connection allows to connect to Runtime Connections you created in Alpana Server, see Runtime Connections.
This allows you query this data in a Data Source, and then display it in Widgets like any other data.

Pre-requisite : login

In order to create a Connection from the Designer, you need to be logged-in to the Alpana Server, see Login to Alpana Server

Selecting a Runtime Connection

Once you are connected to Alpana Server, selecting the *Alpana Runtime* connection type presents the list of all Runtime Connections published on the Server.



Select the desired Runtime Connection from the list.

Connection Schema

The Connection presents a single table called `Data`, to be used in a Data Source.
This is the table defined in Alpana Server.

Data refresh and Buffer

The Alpana Runtime Connector is bufferized.

If you need to get new data, see Chapter The data Buffer

Last modified: Jun 16, 2020

2.4. Wonderware Historian

The Alpana connector for Wonderware Historian allows to get data from the most common query types of Wonderware Historian in a familiar interface.

Getting data is done in two steps :

1. creating the *Connection* : logging into the Wonderware Historian database
2. using the Connection to create *Tables* : create queries to Historian and fetch data into tables

Last modified: May 15, 2019

2.4.1. Connecting to Wonderware Historian

As with other *Connection* types, connecting to Wonderware Historian means defining how Alpana will find the data : what is the server address and what is the login.
See the following chapters for detailed information.

Last modified: May 15, 2019

2.4.1.1. Version compatibility

Alpana allows to connect natively to the following Wonderware Historian versions :

- 2012 (10.0)
- 2012 R2 (11.0)
- 2014 (11.5)
- 2014 R2 (11.6)
- 2014 R2 SP1 (11.6)
- 2017 Update 3 (17.3)
- 2020 (20.0)

Last modified: Jul 30, 2020

2.4.1.2. Important pre-requisite

> **Important**
> If you connect to a fresh install of *Wonderware Historian* Server, tags may not be available for browsing.

Starting Historian clients once

In order to make tags available for browsing, you will have to **startup Wonderware Historian Client *Query* or Wonderware Historian Client *Trend***.

This needs to be done only once. Then the tags will be made available for Alpana.

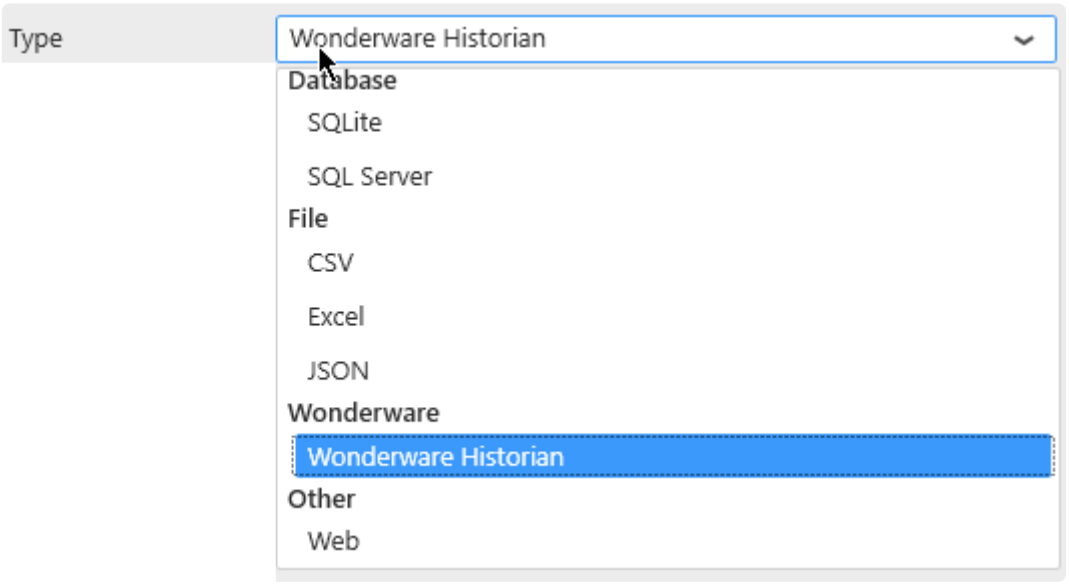
Enabling SQL login to Historian

Alpana requires SQL authentication to login to Historian.
In some versions, those are disabled by default and need to be enabled or created.
See your Wonderware Historian documentation for more details

Last modified: Jul 19, 2019

2.4.1.3. Selecting Connection Type

In the “Create Connection” dialog change the Connection Type by selecting “*Wonderware Historian*” :



Last modified: May 15, 2019

2.4.1.4. Login to a Wonderware Historian server

Connection Parameters

To login to a Wonderware Historian server, fill-in the following fields :

If in the current session you have already connected to a Historian server, you can select it in the combobox below to auto-fill the Host Input.

> ! Important :
> Because the Historian Connector can connect to any server regardless if it is a local server or not, you need to specify a **SQL Server account** type to connect to the `Runtime` Database.

Host

This is the address of the SQL Server instance you need to access for the Wonderware Historian “Runtime” database. Possible values are :

- `(local)` or `localhost` : local server
- `<NetBIOSName>` : server named with a NetBIOS name
- `<IPv4Address>` : server accessible through a IP v4 address
- `<IPv6Address>` : server accessible through a IP v6 address
- `<server address>\<instance name>` : connect to a specific instance. For SQL Express, it may be necessary to use `<server address>\SQLEXPRESS`
- `<server address>,1433` : 1433 is the default port for SQL Server, but can be changed like this

> ✱ Be aware that when searching for a server instance, Alpana waits for a response. If the server instance doesn’t respond, there is a rather long timeout.

Username and Password

The Username and Password must be valid SQL credentials configured with Wonderware Historian.

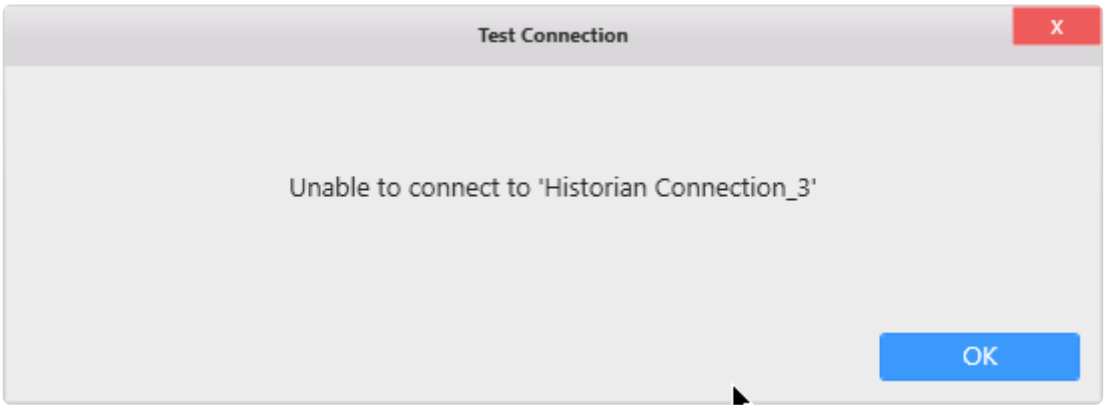
At least, the SQL user must have read permissions to the Historian " `Runtime` " database.

Connecting

When all fields are filled, click OK to create the Connection.

Connection errors

If a connection error happens, a pop-up window appears.



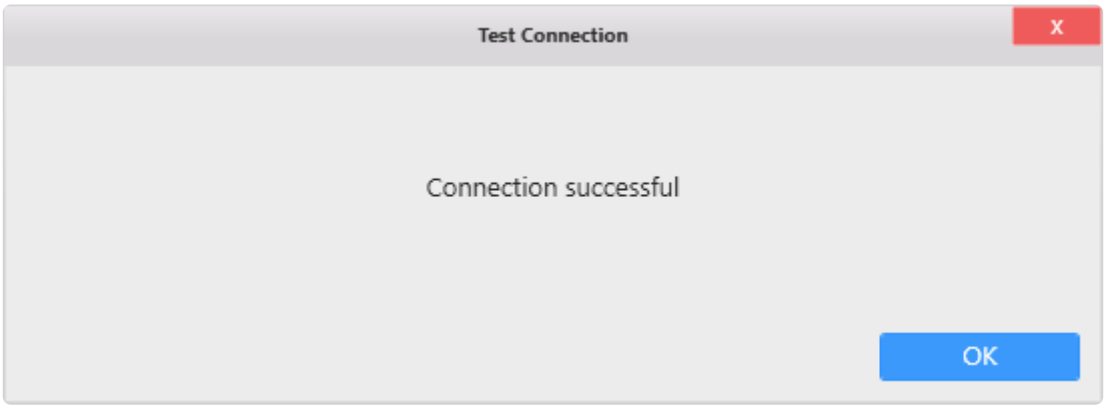
Editing Connection

As with other Connection types, the Historian Connection can be modified later, see the corresponding chapter.

Last modified: Jul 26, 2019

2.4.1.5. Test connection

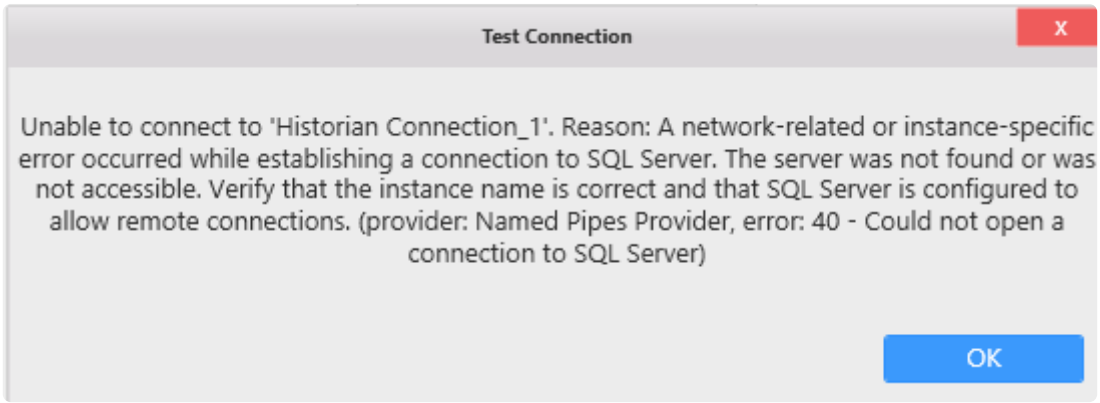
To test the connection for validity, click *Test Connection* button. The following message will confirm its validity.



If the connection is invalid, an appropriate error message will be displayed instead.

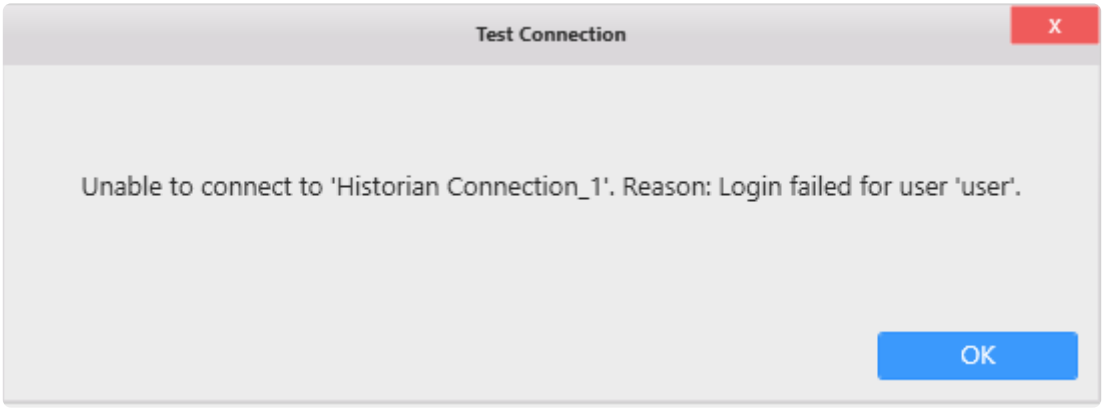
Then you will need to fix the Connection parameters (credentials, server name, …) or check network connectivity (is the SQL Server accessible from here ?)

The server was not found



Usually after a long timeout, if the error message says “Unable to connect […] The server was not found or was not accessible”, check that the server is accessible from this machine on that port.

Login failed



If the message says “login failed for user”, then Alpana successfully connected to the server, but the credentials were invalid.

Check if that SQL user exist and the password is correct.

Last modified: May 15, 2019

2.4.1.6. Data refresh and Buffer

Since Alpana v3.0, the Wonderware Historian Connector is **not bufferized**.

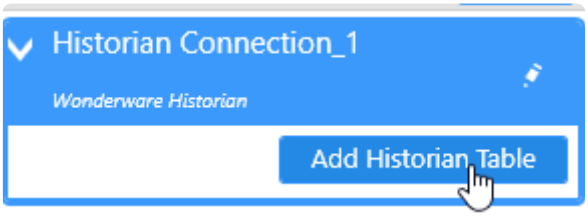
Of course, like other data Connectors, it may become bufferized when used in Multi-Connection Data Sources.
See the corresponding chapter.

Last modified: Jun 17, 2019

2.4.2. Creating Queries to Wonderware Historian

Once the Connection to Wonderware Historian is created, you can re-use it to make several different queries.

Select the desired Historian Connection, and click on *Add Historian Table* :



This will pop-up the *configure Historian Query* dialog.
When the query configuration is complete, a query Table is added to the current Data Source.

Last modified: May 15, 2019

2.4.2.1. Picking Tags

The first part of the configuration is selecting Historian tags.
This is documented in the next chapters.

Last modified: May 15, 2019

2.4.2.1.1. Browsing for Tags

Start browsing for Tags

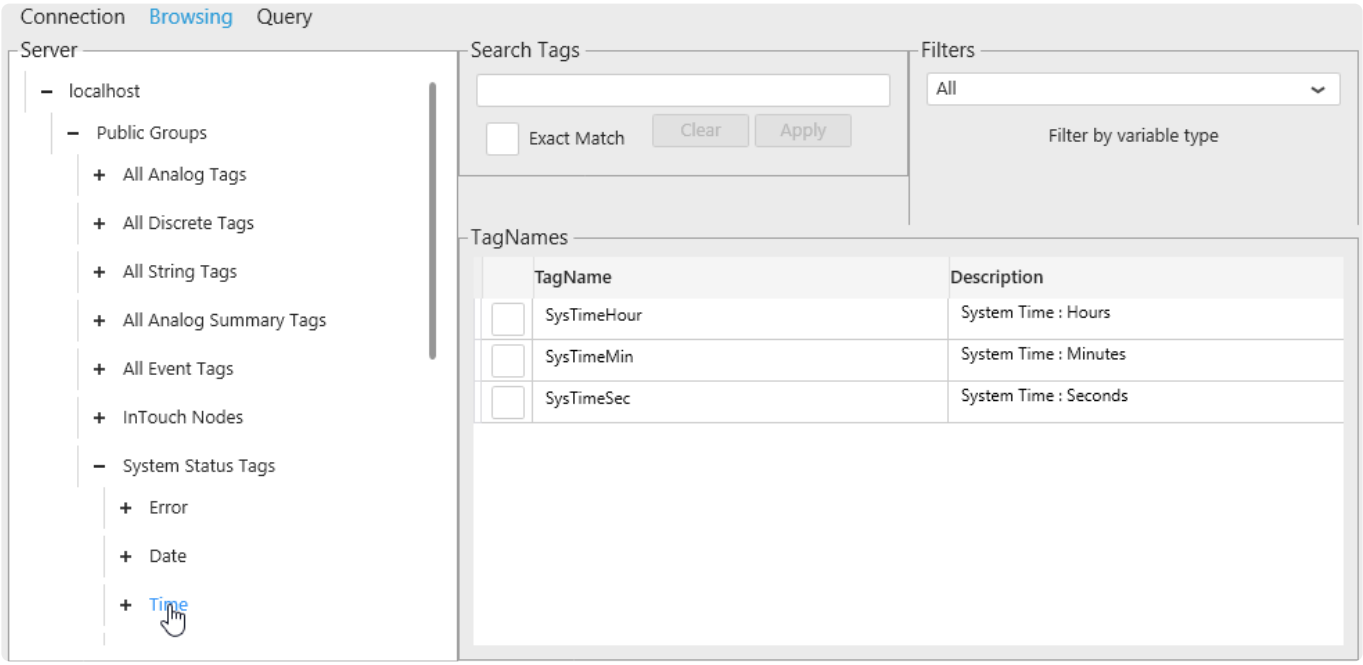
Click on the “Browsing” tab (open by default) to start browsing for tags to pick :

Navigating the Tag hierarchy

The Historian Tag hierarchy is presented in a tree on the left, with Tag Groups as nodes.

Click on a +/- in the tree to unfold/fold a Tag Group.

Click on Group name to view the list of Tags under that Group.



The list of Tags in the currently selected Group is presented in a list on the right.

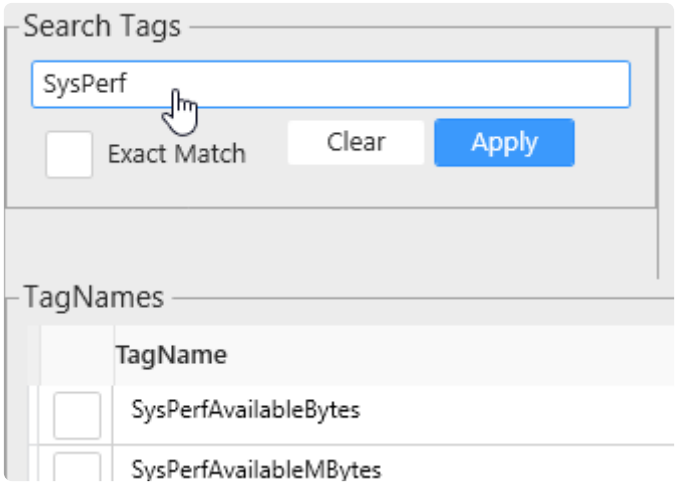
Filtering the Tag list

> Search and Filter only apply to the currently selected Tag Group. If no Tag Group is selected, then the result is always empty.

Filter by Tag name and Description

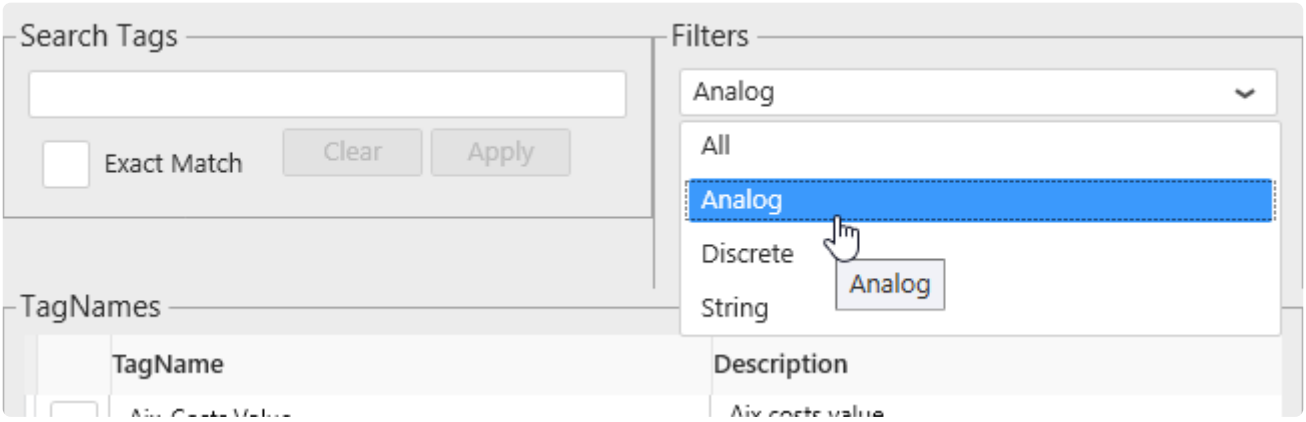
In “Search tags”, enter part of a Tag name or Tag description and click “Apply” to filter Tags whose name contain that text.

You can also check for an “Exact Match” only.



Filter by data type

In “Filters”, you can filter the tag list by data type :



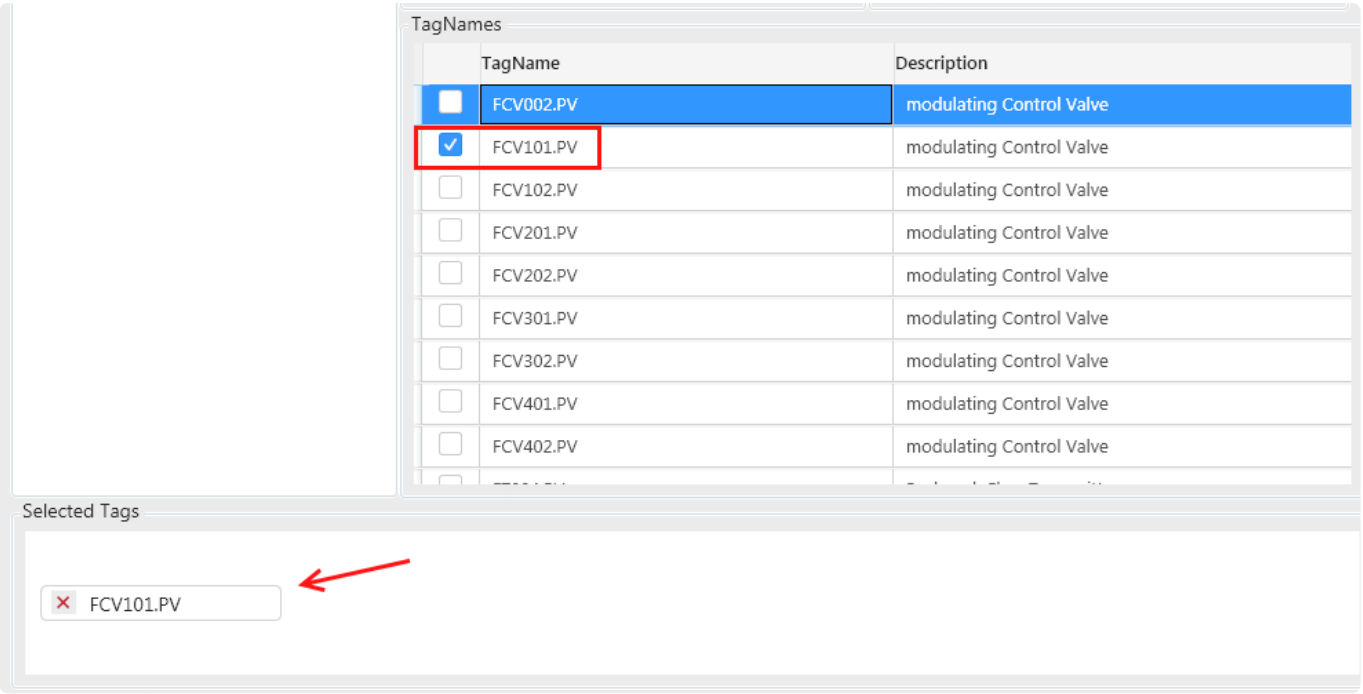
Last modified: May 15, 2019

2.4.2.1.2. Selecting Tags

> **Important**
> To make a correct Wonderware Historian Query you have to select one tag at least

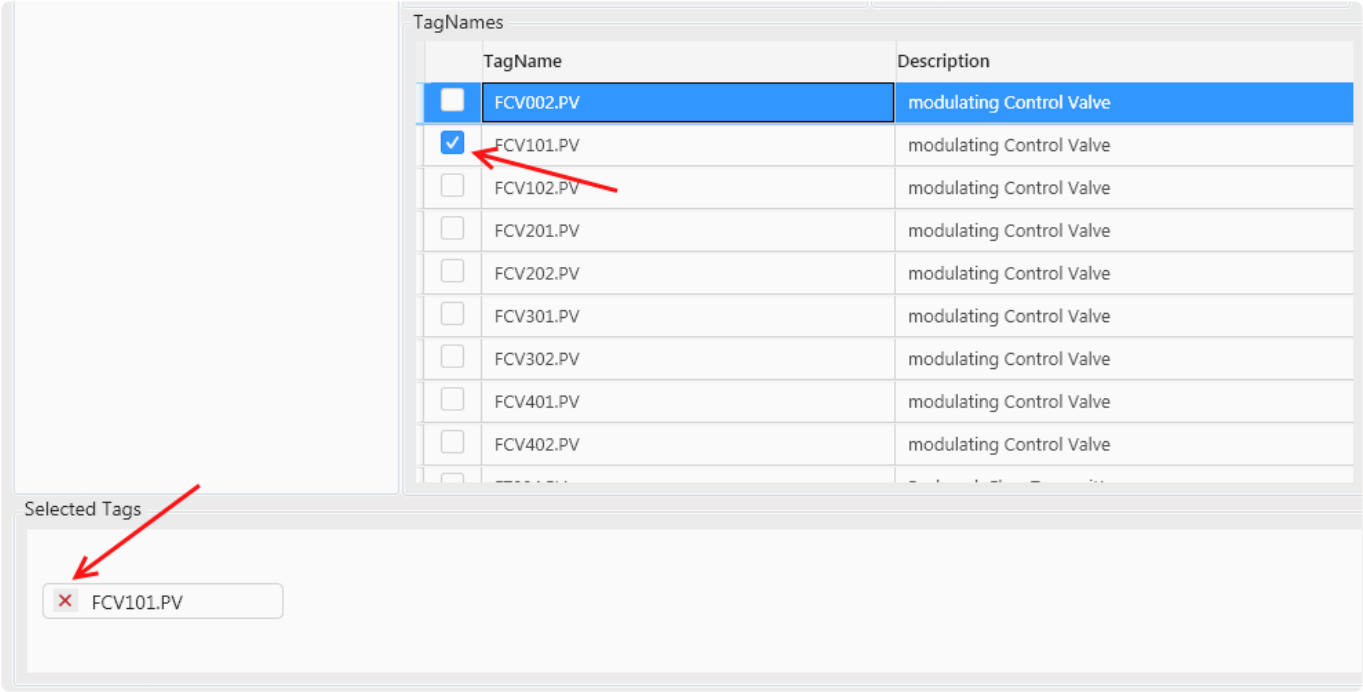
Select Tags

To select a Tag, you can click on its checkbox to select it. Once a tag is selected, it appears in the Selected Tags panel like below:



Unselect Tags

You can unselect a Tag by clicking again on its Selected check box or you can click on the cross button in the Selected Tags panel like below:



Last modified: May 15, 2019

2.4.2.2. Configuring the Historian Query

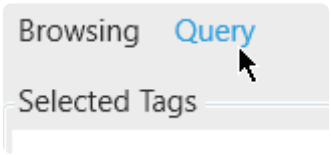
Wonderware Historian allows many ways to query the data.
To make users feel at home, Alpana replicates these configuration options in the query configuration.

See the next chapters for details.

Last modified: May 15, 2019

2.4.2.2.1. Going to Query Configuration

To set all the main Wonderware Historian query parameters, click on the Query Tab shown below



Note that the list of currently selected tags appear at the top of the panel.

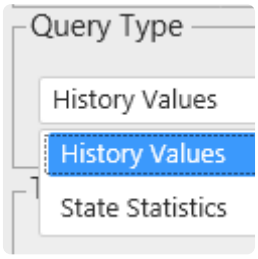
Last modified: May 15, 2019

2.4.2.2.2. Query Type

In Alpana, two Query Types are possible for Wonderware Historian :

- History Values
- State Statistics

You can select the Query Type from the listbox at the top of the Query panel :



History Values

This is the classic Historian Query for showing trends.
A history of Tag values is returned.

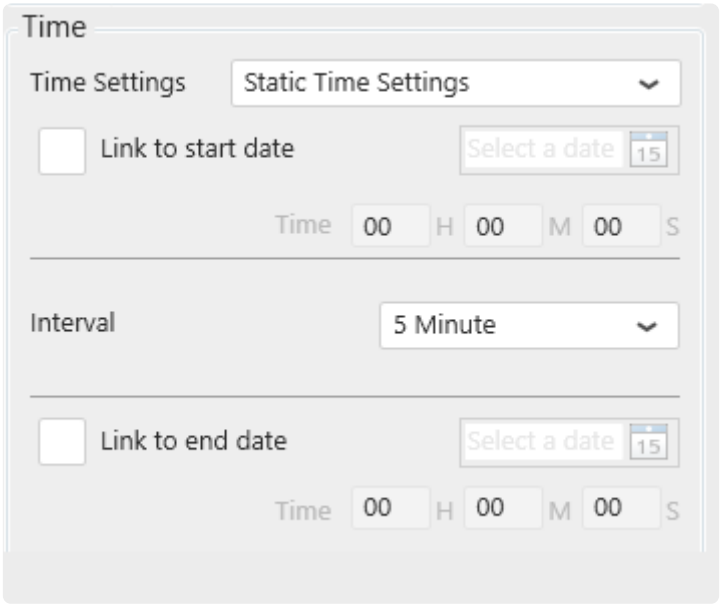
State Statistics

This query (sometimes called “State Summary Values”) allows to calculate how much time a Tag had value.
This allows to calculate for example, how long an equipment was in each of its states, and calculate performance indicators.

Last modified: May 15, 2019

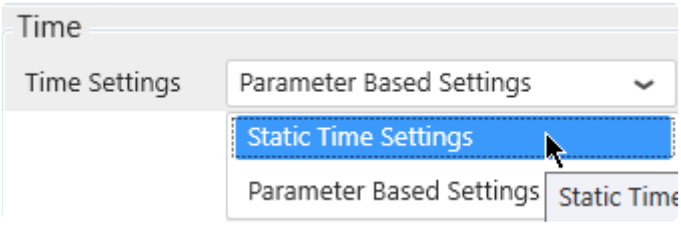
2.4.2.2.3. Time Settings

In the panel on the left, you can configure the Time settings for the query



The time interval of the query can be :

- Static : User interaction at runtime cannot change the time interval
  - a frozen time range
  - a moving time window always ending **now**
- Parameter-based : User interaction at runtime can dynamically change the time interval of the query



Last modified: Jun 18, 2019

2.4.2.2.3.1. Static Time Settings

What are Static Time settings

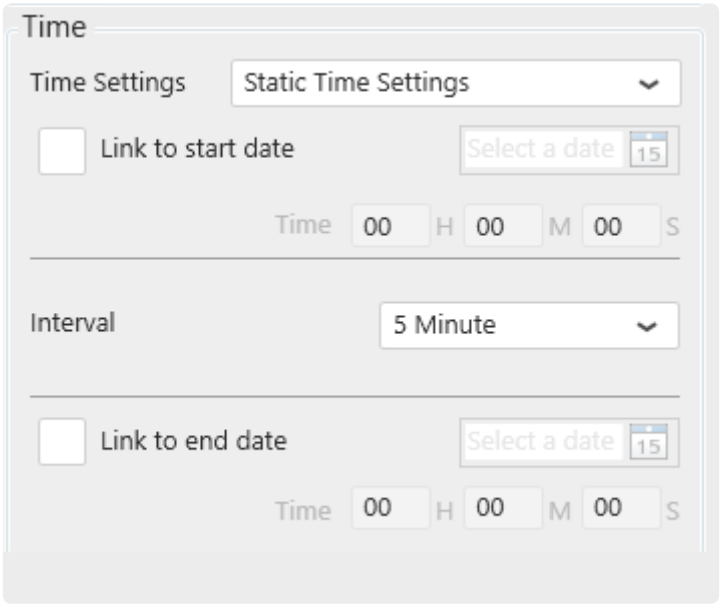
When using *Static Time Settings*, the start date and end date of the Historian query cannot be changed by user interaction at runtime.
Note that you can still create a moving time frame (see below).

This allows to make a pre-defined query to show users precisely the data you want to show.
You can still let users filter this data by time, but they cannot explore outside the pre-defined query.

> If you want to let users explore any time frame at runtime, use Parameter-Based Time Settings instead.

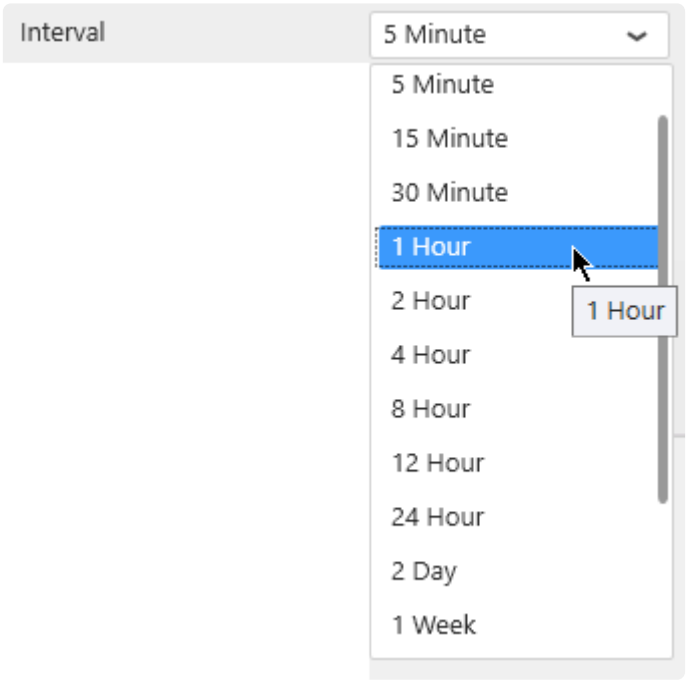
How to configure

Use the panel on the left to configure the Time settings for the query :



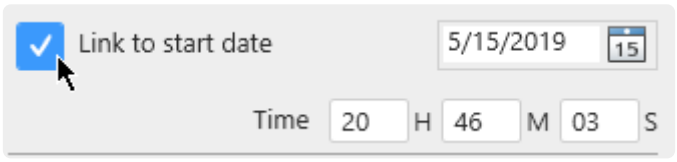
Moving time window

By default, the time range is a moving time window that ends **now** and starts some time ago depending on the *Interval*.
You can set the time Interval by using the corresponding listbox :

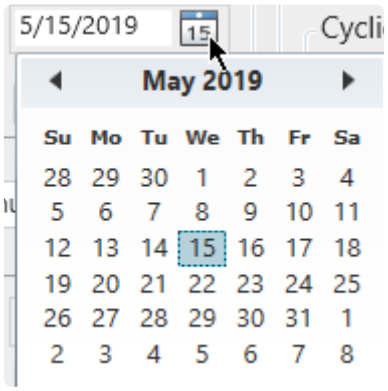


Frozen time range

Instead, you can choose to freeze a time range : the data returned will always be between a pre-defined *start date* and *end date*.
You have the possibility to freeze the *start date* or the *end date* here by checking it like below the desired checkbox :



Then you can configure the Datetime from the text box or by clicking on the calendar :



You can set the time by typing it in the following input :

Note :
If you select both, the interval option will be disabled

Last modified: Jun 18, 2019

2.4.2.2.3.2. Parameter Based Settings

What are Parameter-Based Time Settings

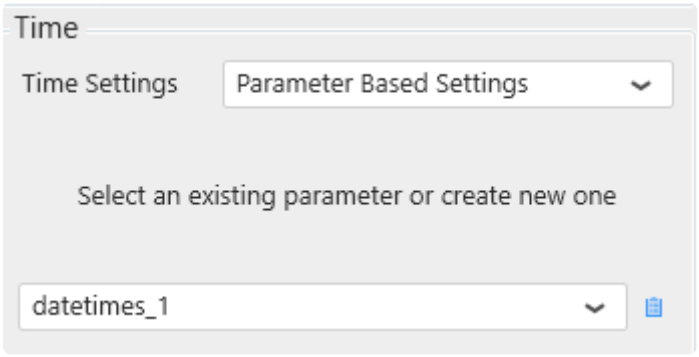
With *Parameter-Based Time Settings*, the start date and end date of the Historian query are contained in a Alpana Parameter.
This means that the values can be changed at runtime using User interaction (filters, URL parameters).

> This is useful for letting users explore any time range in the Historian data.

How to configure

> To configure a parameter-based Historian query, the Alpana Parameter containing the values must already exist and have multiple values (exactly 2 values) : the start date and end date.

Use the panel on the left to select an existing Parameter from the dropdown list, or create a new Parameter using the shortcut button :



Hint

A recommended way to proceed is to :

1. have a separate DataSource with field containing a list of possible dates/times for the user to select from (this is easy to do in Excel, CSV, SQL)
2. create a Master Widget for the user to select the date/time, and configure it with the above DataSource
3. make a default selection in the Master Widget, possibly using relative time default selection (if the Widget type allows it), so that the Parameter always has a value
4. configure the Historian query with Parameter-Based Time Settings, using the Parameter that was automatically generated by the Master Widget

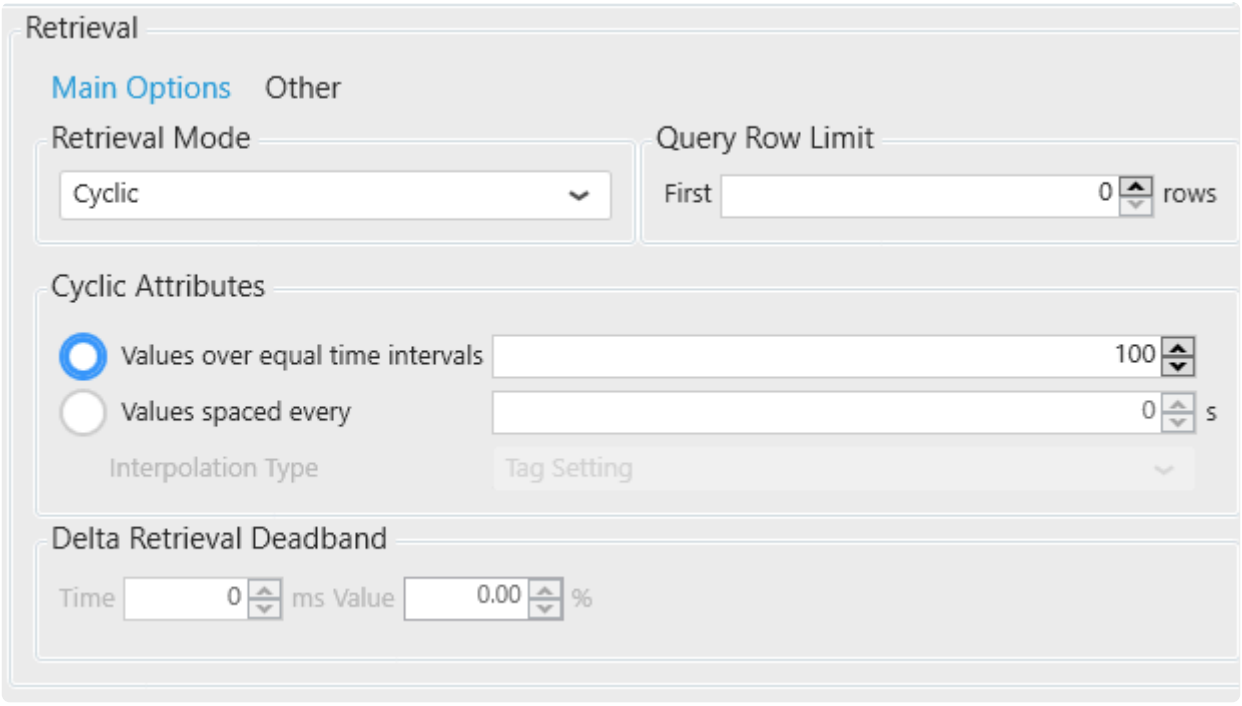
Last modified: Jun 18, 2019

2.4.2.2.4. History Values : query parameters

Introduction

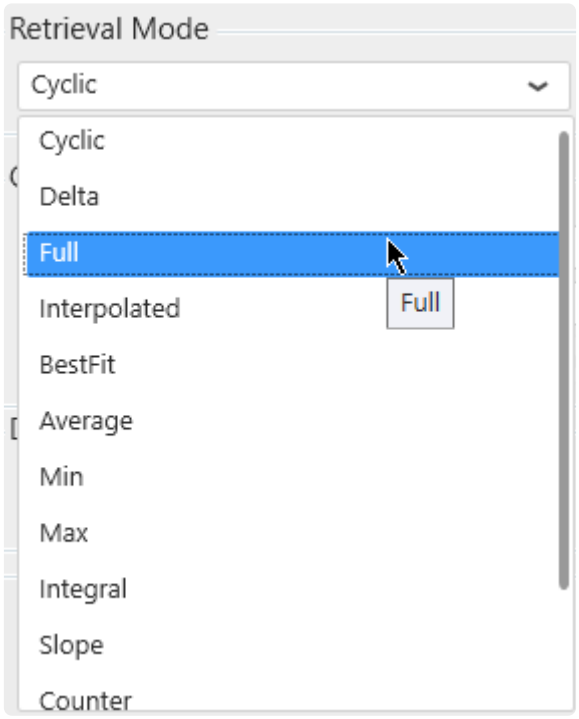
The History Values query has many parameters to configure.
Below is a brief description on how to configure them.
For more information, please see the corresponding help topics in the Wonderware Historian documentation.

In the *Retrieval* part of the window you can set all retrieval parameters :



Retrieval mode

You can select a *Retrieval Mode* (Cyclic is the default) :

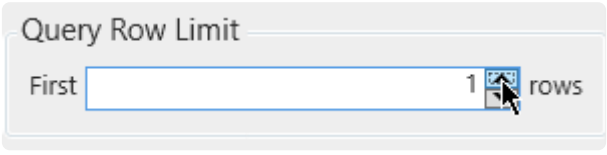


All Wonderware Historian retrieval modes are available :

- Cyclic
- Delta
- Full
- Interpolated
- BestFit
- Average
- Min
- Max
- Integral
- Slope
- Counter
- ValueState
- RoundTrip

Query row limit

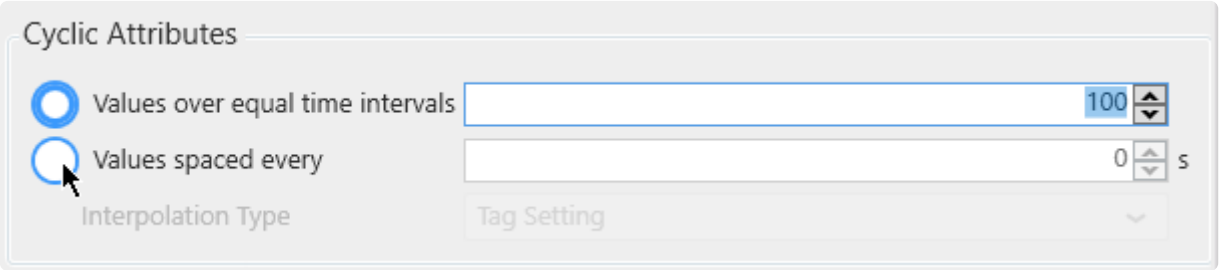
A *Query Row Limit* can be set :



This allows to limit the number of data rows returned.

Cyclic Attributes

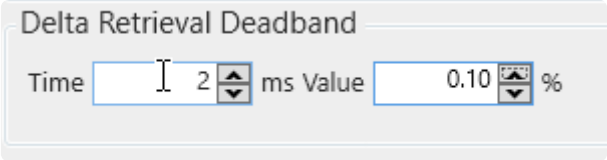
The *Cyclic Attributes* can be configured :



> **Note**
> The Cyclic panel control is accessible only when one of the following Retrieval Modes is selected: Cyclic, BestFit, Average, Min, Max, Integral, slope, Counter, ValueState, Roundtrip

Delta Attributes

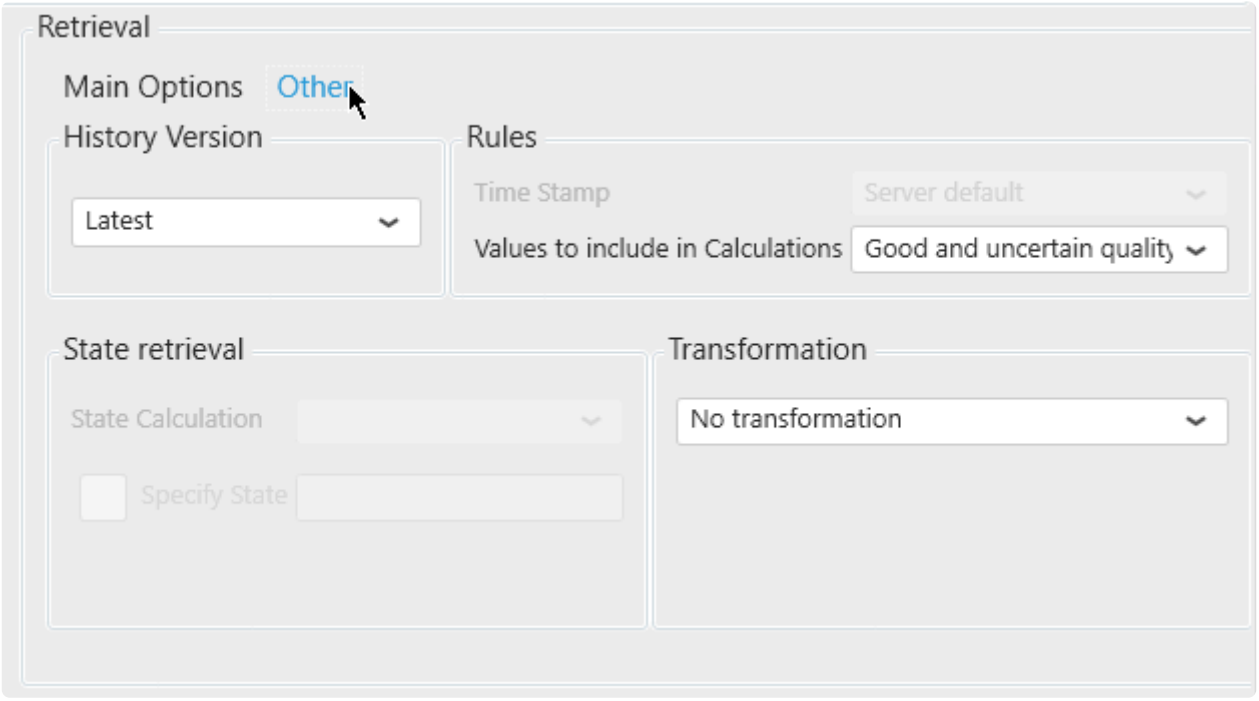
The *Delta Retrieval Deadband* can be configured :



> **Note**
> The Delta panel control is accessible only when *Delta* Retrieval Mode is selected.

Other Attributes

To display the other retrieval options click on the *Other* button :



Here, you can configure the History Version, the Rules Settings, the State Retrieval (available depending the retrieval mode chosen) and the Transformation options.

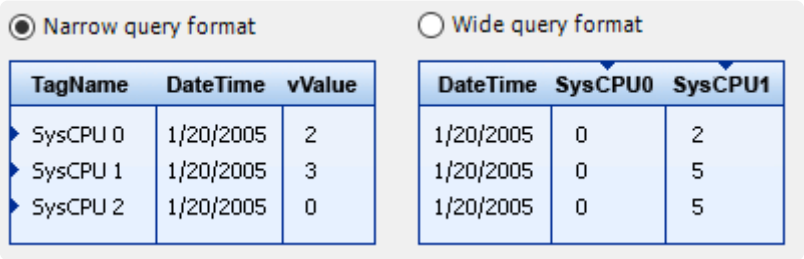
Limitation

> **Note**
> The Phantom Cycle : Do not include boundary values feature is currently not supported

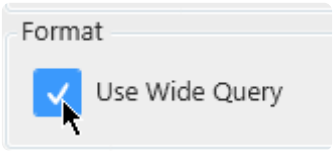
Last modified: May 15, 2019

2.4.2.2.5. History Values: query Format

As with the Wonderware Historian Query client, the Alpana query to Historian is by default in *Narrow* format.



Alternately, you can select the **Wide** Query Format by checking the checkbox under *Format* :



> Note : This is only available when using the *History Values* Query type.

Instead of the default narrow query, it will act as a pivot table to compare your selected tags into different columns over time.

> This is *very useful* to achieve some widget configurations.
> For example, it will allow you to display 2 different tags on 2 different Y Axis on the same Chart.

Last modified: May 15, 2019

2.4.2.2.6. State Statistics : query parameters

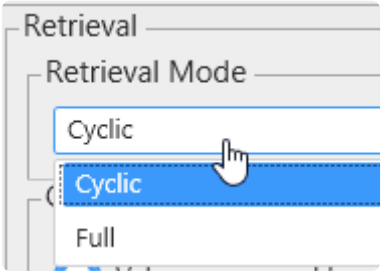
Parameters

Introduction

The State Statistics query has a few parameters to configure.
Below is a brief description on how to configure them.
For more information, please see the corresponding help topics in the Wonderware Historian documentation.

Retrieval Mode

The Retrieval Mode can be changed with the corresponding listbox :

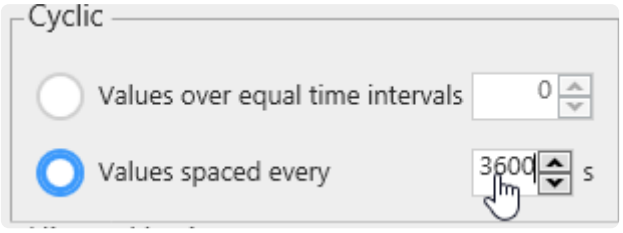


The following Retrieval Modes are available :

- Cyclic
- Full

Cyclic parameters

For Cyclic retrieval, the value spacing can be selected from fixed number or fixed duration :



Full parameters

For Full retrieval, a row limit can be set :

History Version

The History Version listbox allows to select “Latest” or “Original” :

Common usage example

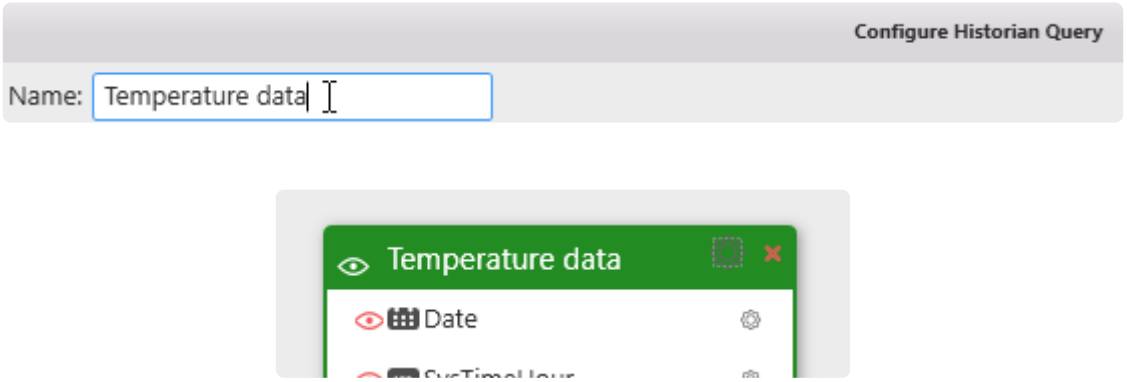
A common usage is to set Cyclic retrieval with values spaced over 1h or 1 day, and a start time at a full hour like midnight.

This will return state statistics with hourly (or daily) information of how much time (total, %, …) was spent in each state.

Last modified: May 15, 2019

2.4.2.3. Renaming the Query Table

The Query Table can be renamed so that it has a meaningful name inside the Data Source configuration :



Last modified: May 15, 2019

2.4.3. Wonderware Historian query result

After configuring the query, Alpana is able to send it to the Wonderware Historian server and get the result.
This result will be modeled as a Table inside the current Data Source.

For details about the schema and data of this query result, see the next chapters.

Last modified: May 15, 2019

2.4.3.1. Connection Schema

The Wonderware Historian Connection presents its Result in a table which can be renamed at the query level.

Data Fields

The fields of the Historian table depend on the query Type.
For a detailed description of these fields and their meaning, see the corresponding documentation from Wonderware.

History Values

For a History Values query in *narrrow* Format (the default), the following fields are returned :

- TagName
- Date
- Value
- vValue
- OPCQuality
- QualityDetail
- QualityDescription

For a History Values query in *Wide* Format, the following fields are returned :

- Date
- (first tag)
- (second tag)
- (…)
- wwResolution

State Statistics

For a State Statistics query, the following fields are returned :

- TagName
- StartDateTime
- EndDateTime
- Value
- vValue
- OPCQuality
- StateCount
- StateTimeMin
- StateTimeMax

- StateTimeAvg
- StateTimeTotal
- StateTimePercent

Last modified: May 15, 2019

2.4.3.2. Important : data limits

> **Important**
> In order to limit performance issues for huge Historian queries, some limitations have been set.

The Wonderware Historian Connector has the following intentional limitations :

- 10000 max data rows per query
- 30s timeout

When a Historian query returns more than 10000 results, it returns the first 10000.

Last modified: May 15, 2019

2.5. Aveva Insight

(new in Alpana 2020 R1)

The Alpana connector for Insight allows to get data from the “time series” part of Aveva Cloud Operations (historically Historian Online).
Internally, it uses the Historian Data REST API.

Getting data is done in two steps :

1. creating the *Connection* : logging-in to Aveva Cloud Operations
2. using the Connection to create *Tables* : create queries to time series inside Aveva Cloud Operations and fetch data into tables

Last modified: Apr 24, 2020

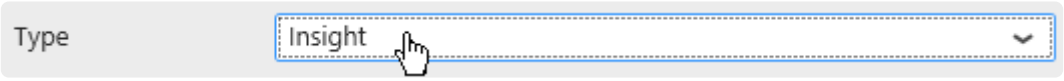
2.5.1. Connecting to Insight

As with other *Connection* types, connecting to Insight means defining how Alpana will find the data : what is the server address and what is the login.
See the following chapters for detailed information.

Last modified: Apr 07, 2020

2.5.1.1. Selecting Connection Type

In the “Create Connection” dialog change the Connection Type by selecting “*Insight*” :



Last modified: Apr 07, 2020

2.5.1.2. Login to Aveva Cloud Operations

Connection Parameters

To login to Aveva Cloud Operations, there are two main modes :

- *Basic Auth*, or “Basic Authentication” : using login credentials to access Historian Data REST API.
- *Bearer Token* or “API Key Authentication” : using authentication keys to access Historian Data REST API

To connect, fill-in the corresponding fields :

Solution Name and *Domain*

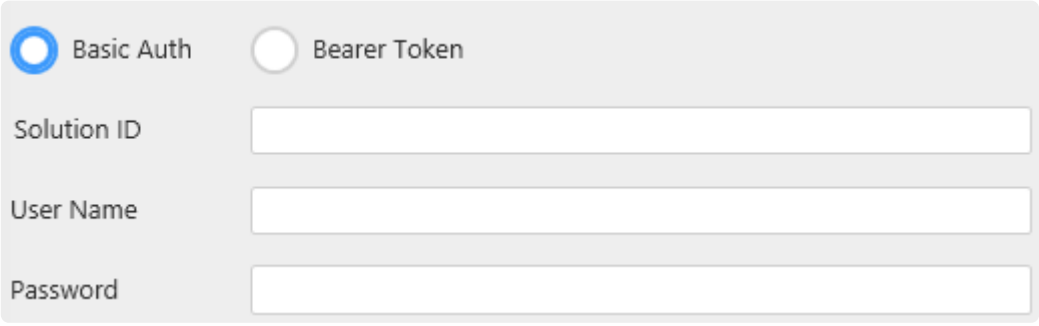
The *Solution Name* is informational only, and doesn’t participate in the actual connection.
It will be displayed at the root of the tag selection.
It is preferable to set your actual solution name, so that the tag tree resembles what you’re used to.

The *Domain* allows to select the server connection :

- `com` are the U.S. servers, with a URL like `https://online.wonderware.com/`
- `eu` are the Europe servers, with a URL like `https://online.wonderware.eu/`

Basic Auth : Basic Authentication

Select this if you want to authenticate using login credentials (username and password) to connect to Insight.
Fill-in the corresponding fields :



Solution ID

This is the identifier of your solution found in the API URL's.
To find the value, login to Insight from your web browser, and navigate to *Administration* > *Integration Settings* > *REST API*, and open the "Basic Authentication" parameters.
There you will find an *Endpoint* similar to `https://online.wonderware.com/s/a9jers/apis/Historian/v2/` (the actual endpoint will be different in your case).

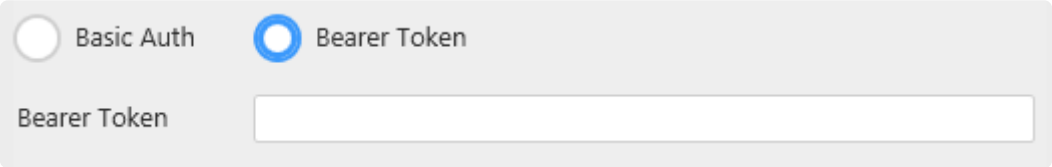
Your solution ID is that `a9jers` part for example (the actual value is specific to your Solution).
Paste it into the Alpana setting.

User Name* and *Password

This is the User Name (Email) and Password you use for connecting to Aveva Cloud Operations.

Bearer Token : API Key Authentication

Select this if you want to authenticate using authentication keys (API Key) to connect to Insight.
Fill-in the corresponding field :



Bearer Token is the API Key (OData bearer token) used to authenticate.
To find the value, login to Insight from your web browser, and navigate to *Administration* > *Integration Settings* > *REST API*, and open the "API Key Authentication" parameters.
This is where you generate an new API Key, or copy the value of an existing key.

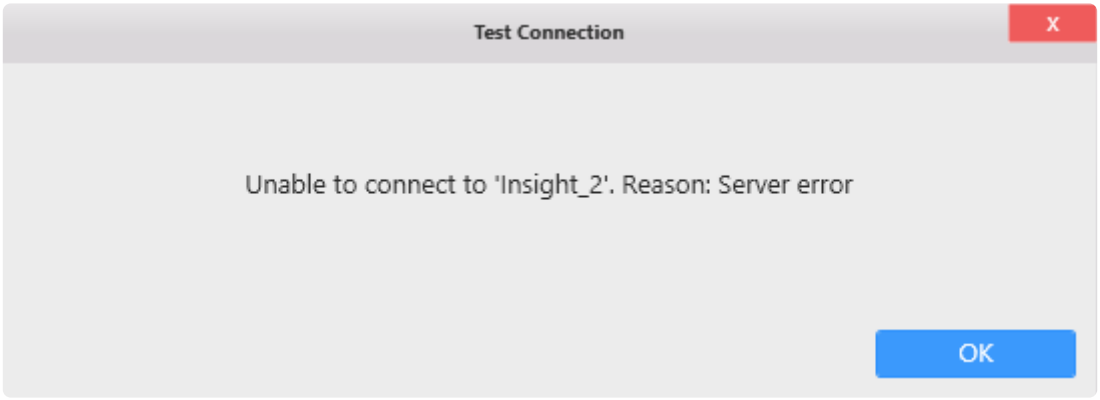
The Bearer Token is this `Bearer ejzuNiz_sl63aOeyI...` value. Paste it into the Alpana setting, including the "@Bearer @" part.

Connecting

When all fields are filled, click OK to create the Connection.

Connection errors

If a connection error happens, a pop-up window appears :



Check the connection settings and try again.

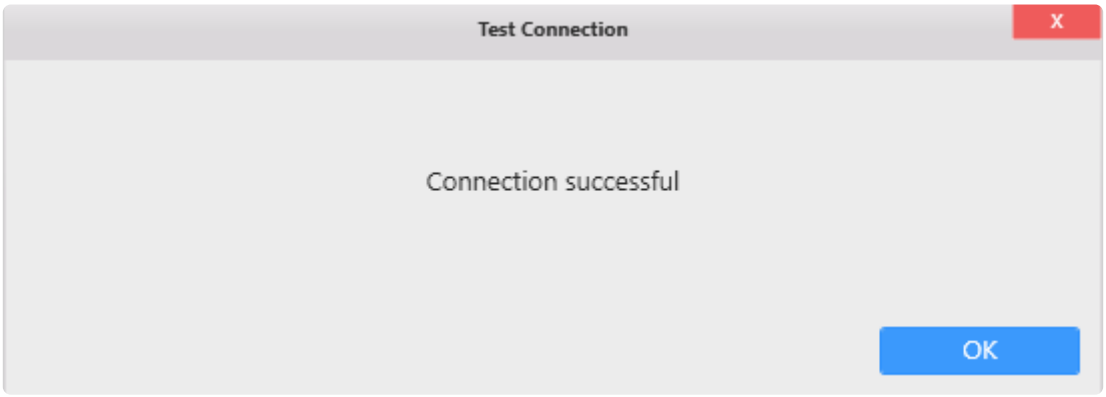
Editing Connection

As with other Connection types, the Insight Connection can be modified later, see the corresponding chapter.

Last modified: Apr 16, 2020

2.5.1.3. Test connection

To test the connection, click *Test Connection* button. The following message will confirm its validity :



If the connection is invalid, an appropriate error message will be displayed instead.

Then you will need to fix the Connection parameters (credentials, …) or check network connectivity (are the Insight servers accessible from here ?)

Last modified: Apr 07, 2020

2.5.1.4. Data refresh and Buffer

The Insight Connector is bufferized.

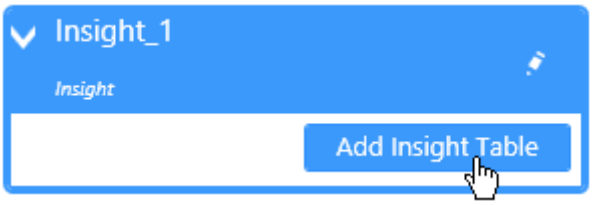
If you need to get new data, see the corresponding chapter.

Last modified: Apr 07, 2020

2.5.2. Querying Insight

Once the Connection to Insight is created, you can re-use it to make several different queries.

Select the desired Insight Connection, and click on *Add Insight Table* :



This will pop-up the *Create Insight Query* dialog.
When the query configuration is complete, a query Table is added to the current Data Source.

Last modified: Apr 07, 2020

2.5.2.1. Picking Tags

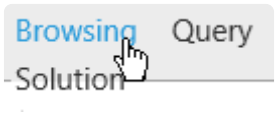
The first part of the configuration is selecting Insight tags.
This is documented in the next chapters.

Last modified: Apr 07, 2020

2.5.2.1.1. Browsing for Tags

Start browsing for Tags

Click on the *Browsing* tab (open by default) to start browsing for tags to pick :

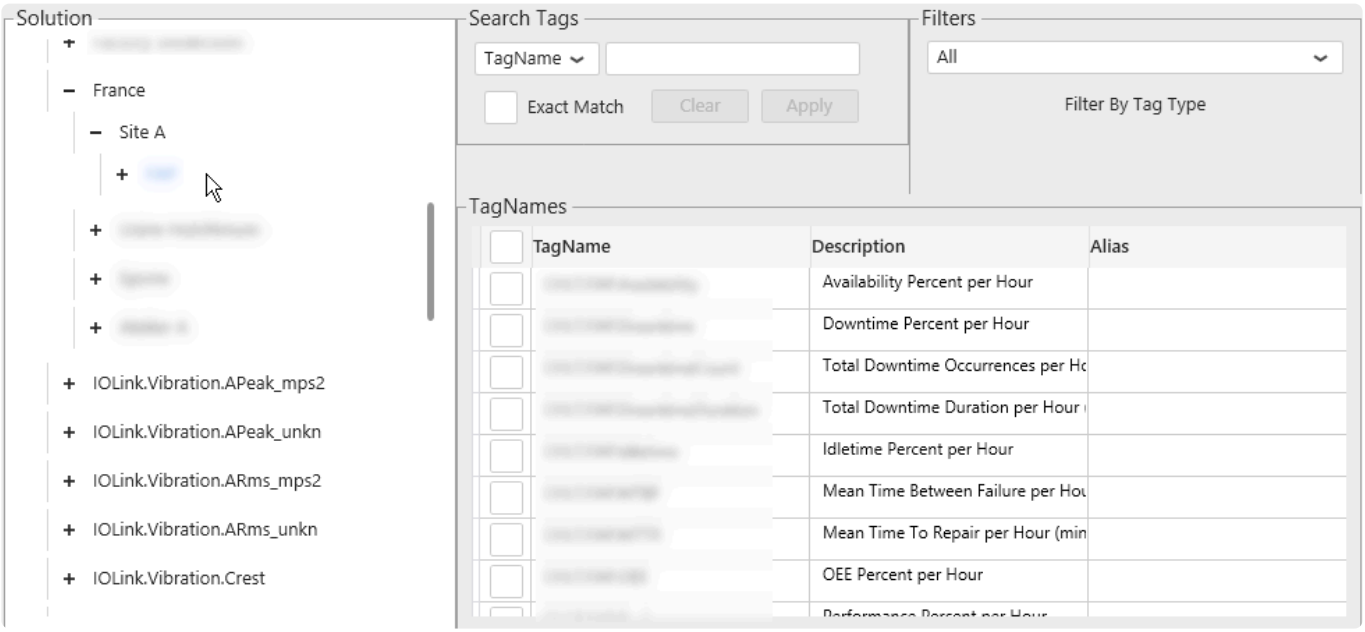


Navigating the Tag hierarchy

The Insight Tag hierarchy is presented in a tree on the left, organized by the *Location* property of the Tag.

Click on a +/- in the tree to unfold/fold a Tag group (grouped by location).

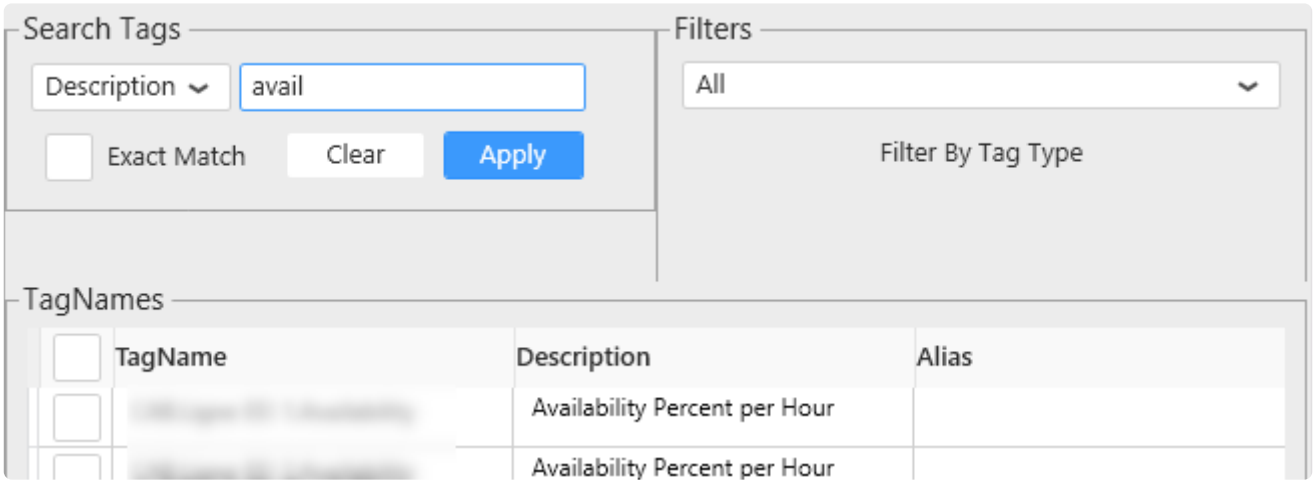
Click on group name to view the list of Tags under that group.



The list of Tags in the currently selected group is presented in a list on the right.

Filtering the Tag list

> Search and Filter only apply to the currently selected Tag group. If no Tag group is selected, then the result is always empty.



Filter by Tag name or property

In “Search tags”, select a property to search :

- FQN : the fully qualified name for the tag
- TagName : the name of the Tag
- Source : the data source
- Description : the Tag description
- Alias : the Tag alias
- TagDataType : the data type

Then enter the desired text to search and click “Apply” to filter Tags whose property contain that text.

You can also check for an “Exact Match” only.

Filter by data type

In “Filters”, you can filter the tag list by data type :

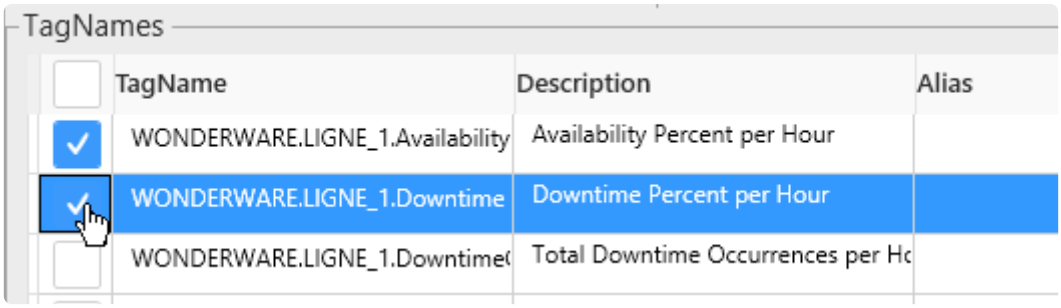
Last modified: Apr 07, 2020

2.5.2.1.2. Selecting Tags

> Important
> To make a correct Insight query you have to select one tag at least

Select Tags

To select a Tag, you can click on its checkbox to select it :



Unselect Tags

Once a tag is selected, it appears in the *Selected Tags* panel.
You can unselect a Tag by clicking again on its Selected check box or you can click on the cross button in the Selected Tags panel like below:

Note : you can hover the tag in the *Selected Tags* panel to see its full name in a tooltip.

Last modified: Apr 07, 2020

2.5.2.2. Configuring the query

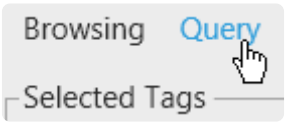
Insight allows many ways to query the data.
To make users feel at home, Alpana replicates these configuration options in the query configuration.

See the next chapters for details.

Last modified: Apr 07, 2020

2.5.2.2.1. Going to Query Configuration

To set all the main Insight query parameters, click on the *Query* Tab shown below



Note that the list of currently selected tags appear at the top of the panel.

Last modified: Apr 07, 2020

2.5.2.2.2. Query Type

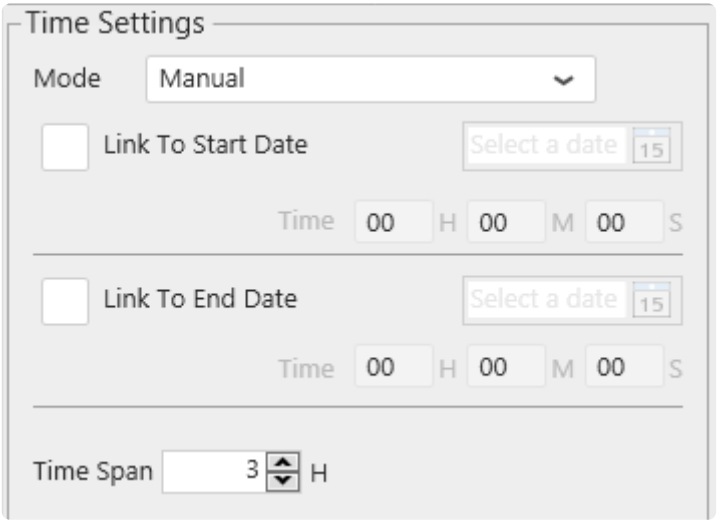
Currently in Alpana, only one Query Types is possible for Insight : *ProcessValues*

> Retrieves a set of process value records (where each record includes value + time + quality, or VTQ) for the specified tags.

Last modified: Apr 07, 2020

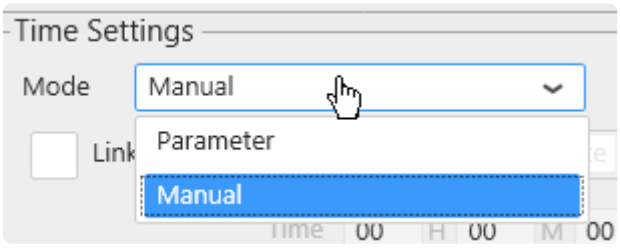
2.5.2.2.3. Time Settings

In the panel on the left, you can configure the Time settings for the query



The time interval of the query can be :

- *Manual* : User interaction at runtime cannot change the time interval
  - a frozen time range : the same time interval is always retrieved
  - a time span : data is retrieved in a a moving time window always ending **now**
- *Parameter* based : User interaction at runtime can dynamically change the time interval of the query



Last modified: Apr 07, 2020

2.5.2.2.3.1. Manual Time Settings

What are Manual Time settings

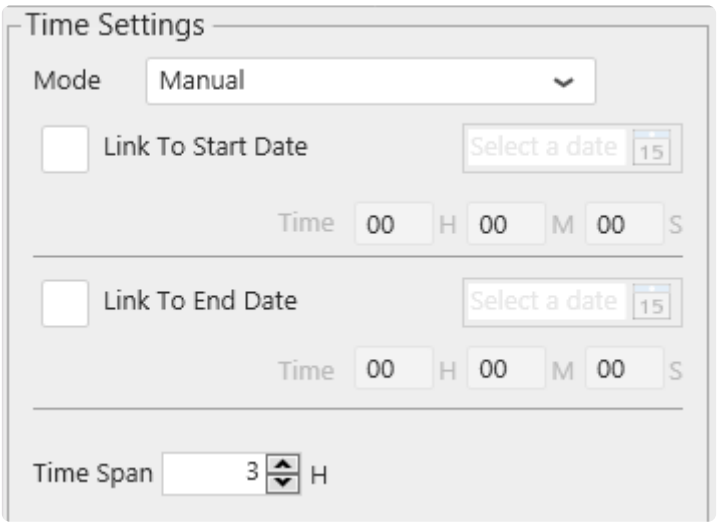
When using *Manual Time Settings*, the start date and end date of the Insight query cannot be changed by user interaction at runtime.
Note that you can still create a moving time frame, see below.

This allows to make a pre-defined query to show users precisely the data you want to show.
You can still let users filter this data by time, but they cannot explore outside the pre-defined query.

> If you want to let users explore any time frame at runtime, use Parameter-Based Time Settings instead.

How to configure

Use the panel on the left to configure the Time settings for the query :

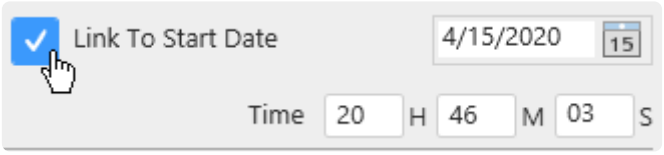


Moving time window

By default, the time range is a moving time window that ends **now** and starts some time ago depending on the *Time Span*.
You can set the *Time Span* by using the corresponding control.

Frozen time range

Instead, you can choose to freeze a time range : the data returned will always be between a pre-defined *start date* and *end date*.
You have the possibility to freeze the *start date* or the *end date* here by checking it like below the desired checkbox :



Then you can configure the Datetime from the text box or by clicking on the calendar

> Note :
> If you select both, the interval option will be disabled

Last modified: Apr 07, 2020

2.5.2.2.3.2. Parameter Based Settings

What are Parameter-Based Time Settings

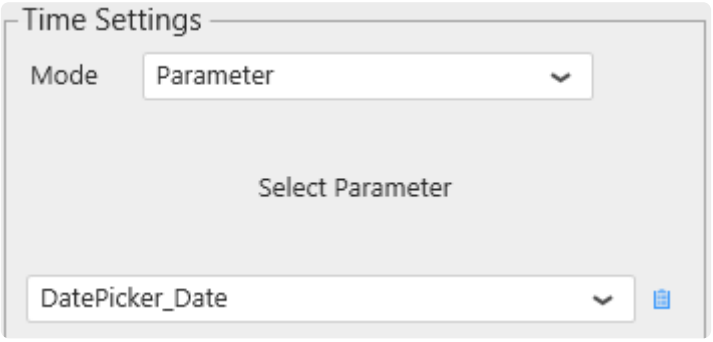
With *Parameter-Based Time Settings*, the start date and end date of the Insight query are contained in a Alpana Parameter.
This means that the values can be changed at runtime using User interaction (filters, URL parameters).

This is useful for letting users explore any time range in the Insight data.

How to configure

To configure a parameter-based Insight query, the Alpana Parameter containing the values must already exist and have multiple values (exactly 2 values) : the start date and end date.

Use the panel on the left to select an existing Parameter from the dropdown list, or create a new Parameter using the shortcut button :



Hint

A recommended way to proceed is to :

1. have a separate DataSource with field containing a list of possible dates/times for the user to select from (this is easy to do in Excel, CSV, SQL). Note this is optional with some widgets like Date Picker which support creating a list of dates without a Data Source.
2. create a Master Widget for the user to select the date/time, and configure it with the above DataSource
3. make a default selection in the Master Widget, possibly using relative time default selection (if the Widget type allows it), so that the Parameter always has a value
4. configure the Insight query with Parameter-Based Time Settings, using the Parameter that was automatically generated by the Master Widget

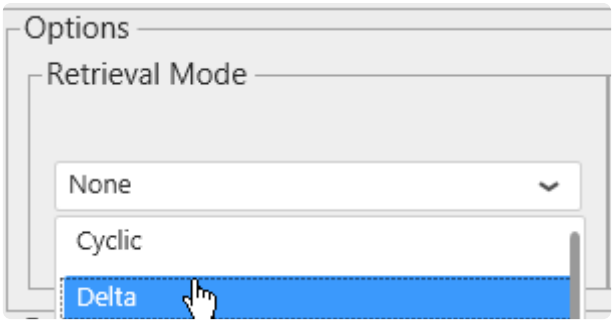
Last modified: Apr 07, 2020

2.5.2.2.4. Process Values : query parameters

Introduction

The Process Values query exposes some configuration options.
Below is a brief description on how to configure them.
For more information, please see the corresponding help topics in the Insight documentation.

In the *Retrieval* part of the window you can select the Retrieval Mode :



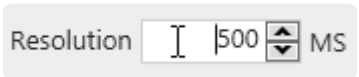
RetrievalMode and *Resolution*

You can use the *RetrievalMode* and *Resolution* parameters in retrieval queries to better control the returned data.

RetrievalMode specifies how the resulting data is calculated for Raw and ProcessValues entities. Valid values are:

- *Cyclic*
- *Delta*
- *Full*
- *Interpolated*
- *BestFit*
- *Average*
- *Minimum*
- *Maximum*
- *Integral*
- *Slope*
- *Counter*

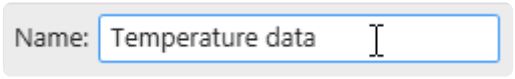
Resolution specifies the granularity of data returned :

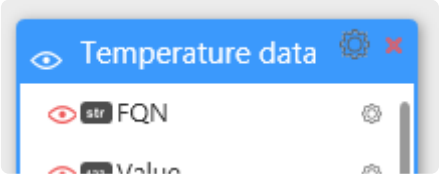


Last modified: Apr 07, 2020

2.5.2.3. Renaming the Query Table

The Query Table can be renamed so that it has a meaningful name inside the Data Source configuration :





Last modified: Apr 07, 2020

2.5.2.4. Configuring the query result

The Insight Connection presents its Result in a table.
Some default fields are always displayed, and some additional fields can be added, to help add more context to the data.

Data Fields

The following fields are returned by default :

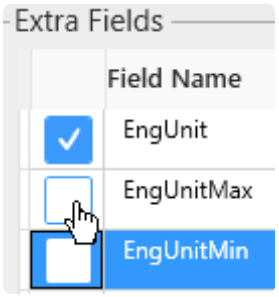
- *FQN* : the fully qualified name for the tag
- *Value* : the process value as a number
- *DateTime* : the process measurement time
- *Text* : the process value as text
- *OPCQuality* : the measurement quality

Additionally, you can select the following fields to add metadata to the result :

- *EngUnit* : the measurement unit
- *EngUnitMax* : the measurement high limit
- *EngUnitMin* : the measurement low limit
- *InterpolationType* : the type of interpolation used when returning data
- *MessageOff* : the text to display when the boolean is false
- *MessageOn* : the text to display when the boolean is true
- *SourceTag* : the TagName of the source Tag
- *Namespace* : the namespace
- *Raw@odata.navigationLink* : a link to more JSON data about the tag, which requires authentication
- *Minutely@odata.navigationLink* : a link to more JSON data about the tag, which requires authentication
- *Hourly@odata.navigationLink* : a link to more JSON data about the tag, which requires authentication
- *Daily@odata.navigationLink* : a link to more JSON data about the tag, which requires authentication
- *Description* : the Tag description
- *TagName* : the name of the Tag
- *TagType* : the type of tag (Analog, etc)
- *Source* : the data source
- *Alias* : the Tag alias
- *TagDataType* : the data type (double, etc)
- *Location* : the *Location* property of the tag

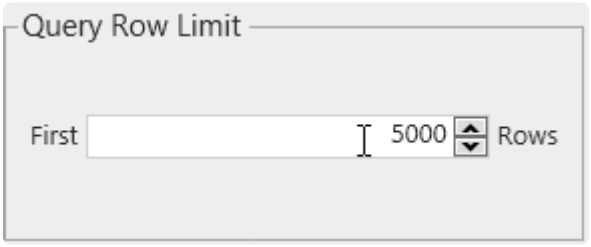
How to select additional fields

The *Extra Fields* presents a list of fields to select. Just check/uncheck the boxes corresponding to the desired fields :



Query limits

You can apply a limit to the number of records retrieved, by using the *Query Row Limit* field :



When a Insight query returns more results than this limit, it returns the first X results only.

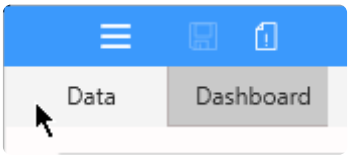
Last modified: Apr 07, 2020

2.6. Transforming Data

This Chapter is a reference on how to create and configure a Data Source to prepare data for your Widgets and Dashboard.

To understand the Alpana Data Model, see the corresponding chapter in this documentation.

Since Alpana Designer v3.0, all data preparation is made inside the *Data* tab.



Last modified: Jun 17, 2019

2.6.1. Introduction

Data Sources in the Dashboard design workflow

You don’t need to be able to create a Data Source in order to create a Dashboard : someone may do it for you !

The workflow to design Dashboards is built around the idea that Dashboards should be easy to create and edit. As such, the data preparation steps can be made by another person and shared to you.

Example :

- User1 needs to make a dashboard but doesn’t have the skills or credentials to prepare data
- User2 is skilled in IT/SQL/data/… and has the credentials to access the data, but has no process knowledge and won’t know how to make the Dashboard. She creates and shares a DataSource.syds file to User1

In this Course, you will learn all about Data Sources and how to share them with your “User1” if needed.

Last modified: Apr 16, 2019

2.6.1.1. Learning Pre-Requisites

To create Data Sources, you will need to connect to your data first. Please see the corresponding chapter : Connecting to Data.

More generally, it is also important to understand the Alpana data model.

Last modified: Apr 16, 2019

2.6.1.2. What are Data Sources

In Alpana, data preparation happens in **Data Sources**.

Each Data Source prepares a query to the original data to be presented inside Widgets.

Data is fetched through the Alpana Connections, see Course Connecting to Data.

Each Widget displays data from a single Data Source (but a Data Source can be re-used in any number of Widgets).

Alpana Designer provides a graphical editor that allows to make this data preparation using simple mouse interaction.

Last modified: Apr 16, 2019

2.6.1.3. Content : Data Preparation Activities

The main data preparation activities in Alpana Designer include :

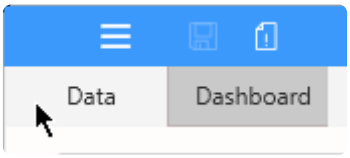
- Fetch : select an object (table, view, stored procedure, …) from which to get data
- Merge : join tables from one or more Connections
- Filter : remove data based on multiple criteria
- Calculate : create Expressions to make calculations
- Format : format, rename and hide fields to make it easier for another user to re-use the Data Source.
- Share : export/import Data Sources

Last modified: Apr 16, 2019

2.6.2. Creating and Managing Data Sources

This chapter describes how to create and manage Data Sources.

Since Alpana Designer v3.0, all data preparation is made inside the *Data* activity tab.



Last modified: Jun 17, 2019

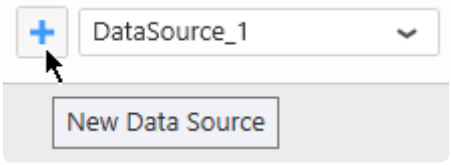
2.6.2.1. Create a Data Source

New Dashboard : first Data Source

When creating a new Dashboard, a new Data Source called “DataSource_1” is created and opened in Alpana Designer.

Creating a Data Source

After creating a Connection, you can create a Data Source by clicking the **+** button next to the Data Sources list in the top bar :



Then, the new Data Source is automatically open for edition.

Data Source storage

A Data Source contains only the information on how to prepare the data. It does not contain the data itself.

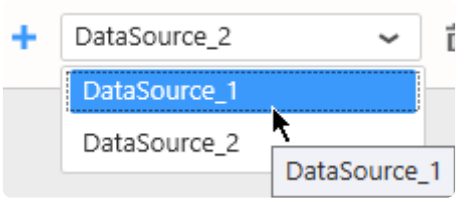
When a Dashboard is saved, all the Data Sources are saved inside.

You can create as many Data Sources as needed.

Last modified: Apr 17, 2019

2.6.2.2. Browsing Data Sources to Edit

To browse Data Sources, open the Data Sources list in the top bar, and select by name :



All the actions to actually edit the Data Source are explained in the following chapters of this Course.

Last modified: Apr 17, 2019

2.6.2.3. Rename a Data Source

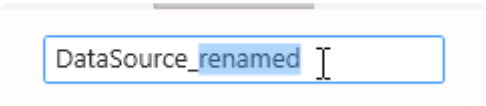
Why rename a Data Source?

The Data Source name will appear in widget data binding.

Giving good names can help the user who will make the Dashboard based on this Data Source.

How

Use the text box on the left of the top bar to rename the currently open Data Source :



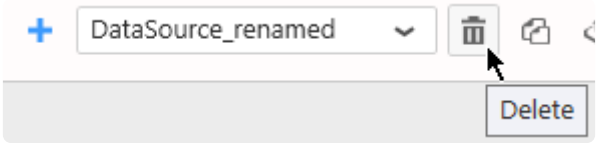
The new name is immediately validated as you type it.

Last modified: Apr 17, 2019

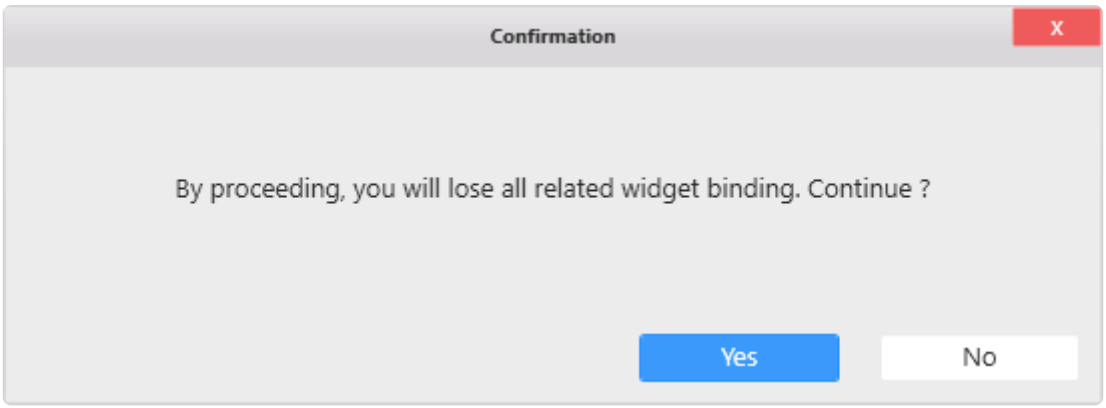
2.6.2.4. Delete a Data Source

How to Delete a Data Source

To Delete an existing Data Source, left click then click on the trash icon in the top bar :



If the Data Source is used by any Widget, the following message will appear :



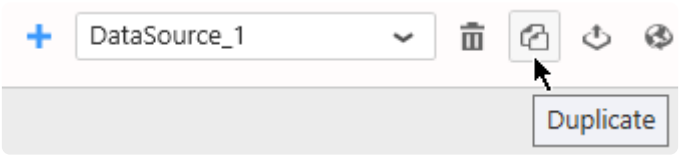
If you choose “Yes”, the Data Source is deleted and all dependent Widgets lose their data binding.

Last modified: Apr 17, 2019

2.6.2.5. Duplicate a Data Source

You can Duplicate an existing Data Source to start working on a copy.

In the top bar, click on the Duplicate_ icon :



A copy of your Data Source is created, and you can start working on it independently.

Last modified: Apr 17, 2019

2.6.3. Export / Import Data Sources

Sharing Data Sources allows to collaborate between users with different skill sets : an IT engineer creates a Data Source, exports it, then a process engineer only has to import it and start creating visualizations.

This chapter describes how to proceed.

Last modified: Apr 25, 2019

2.6.3.1. What is a Data Source Export

A Data Source can be exported to a file with format `*.alps` .

This DataSource.syds file contains the definition of the Data Source and all the dependent Connections, including their credentials.

This is useful for splitting roles in Dashboard creation :

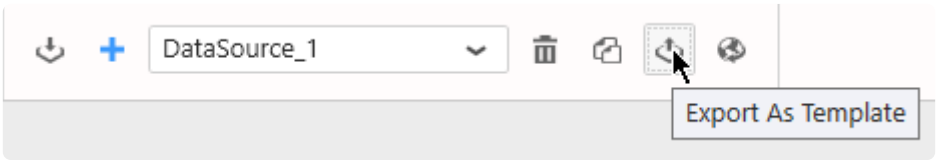
- User1 needs to make a dashboard but doesn’t have the skill or credentials to prepare data
- User2 is skilled in IT/SQL/data/… and has the credentials to access the data, but has no process knowledge and won’t know how to make the Dashboard. She creates and shares a DataSource.alps file to User1

Last modified: Mar 14, 2019

2.6.3.2. Export a Data Source to a file

How to Export to a file

In the top bar, click the *Export as Template* icon :



Select a file name and save locally.

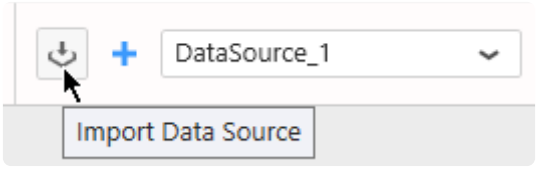
Last modified: Apr 17, 2019

2.6.3.3. Import a Data Source from a file

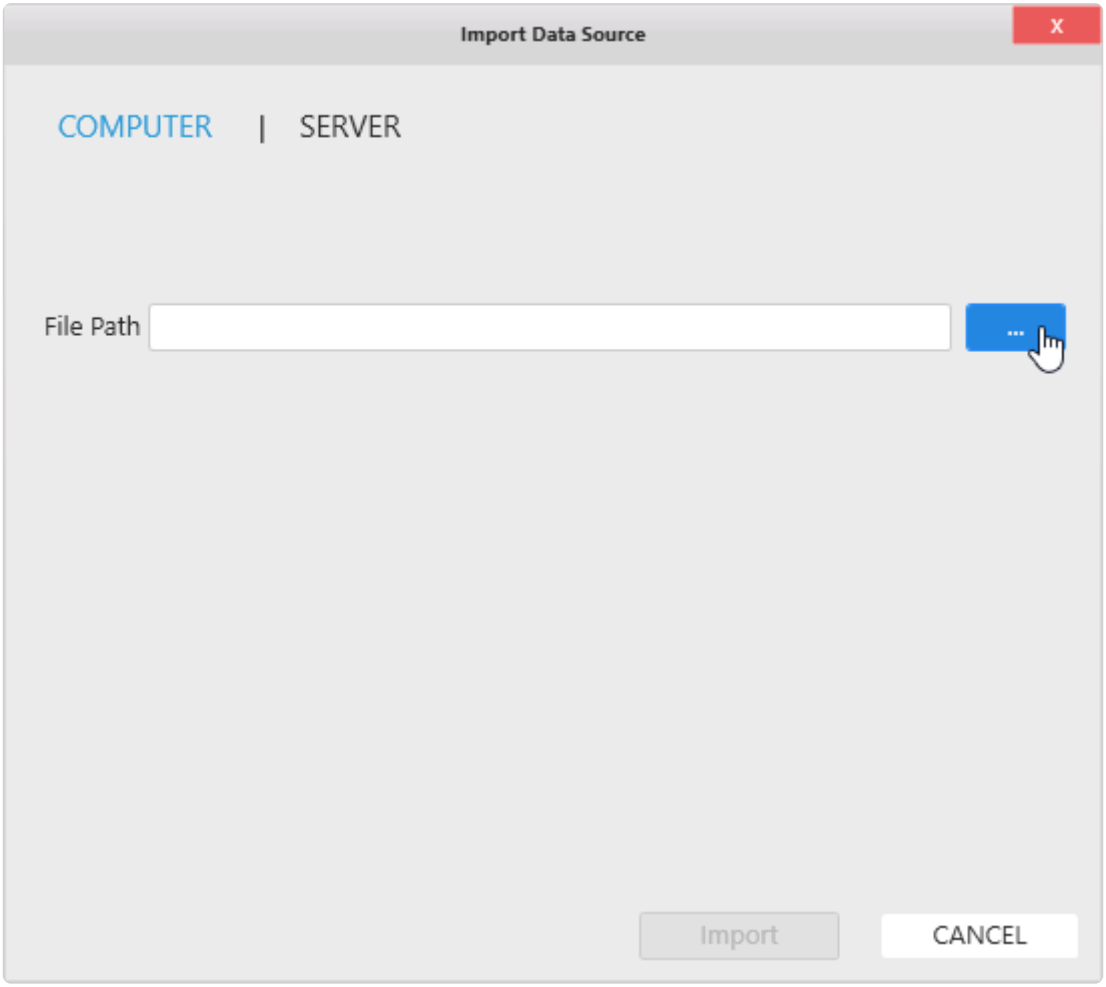
A Data Source can be imported from a file in a few ways :

Import in an existing Dashboard

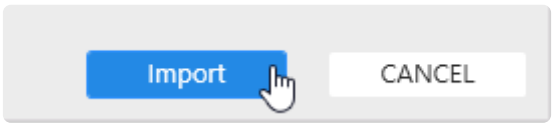
Click on the “Import” button in the Data Sources list :



In the “Import Data Source” dialog, under “Computer”, click “…” to browse for a DataSource.alps file :



Then click “Import” :



The Data Source is now imported in your current Dashboard.

Create a new Dashboard based on a DataSource file

In Windows Explorer, you can double click or open the DataSource.alps file with Alpana Designer.

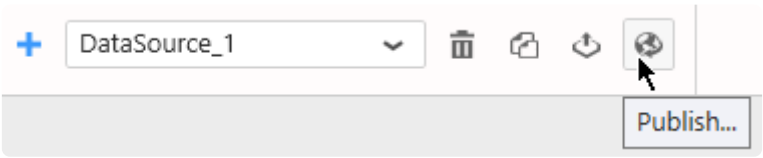
This creates a new Dashboard with the corresponding Data Source imported.

Last modified: Apr 17, 2019

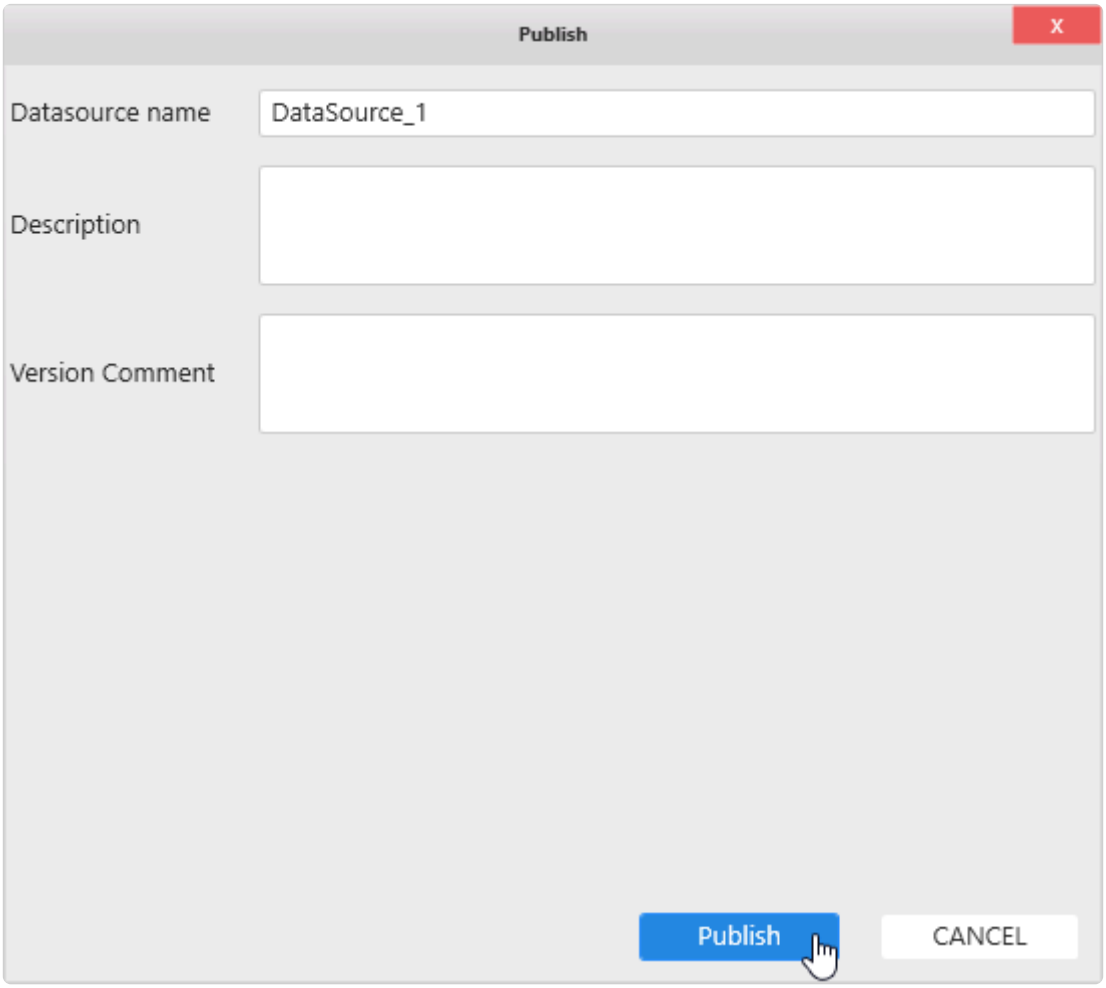
2.6.3.4. Publish a Data Source to Alpana Server

In order to Publish a Data Source to Alpana Server, you need to be **logged in** from Alpana Designer to that Alpana Server, and have the necessary **permissions** to create Data Sources.

In the top bar, click *Publish* :



In the “Publish” dialog, fill in details and click “Publish” :



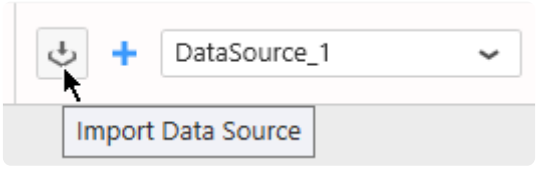
The Data Source is now published to Alpana Server and available to other users depending on permissions.

Last modified: Apr 26, 2019

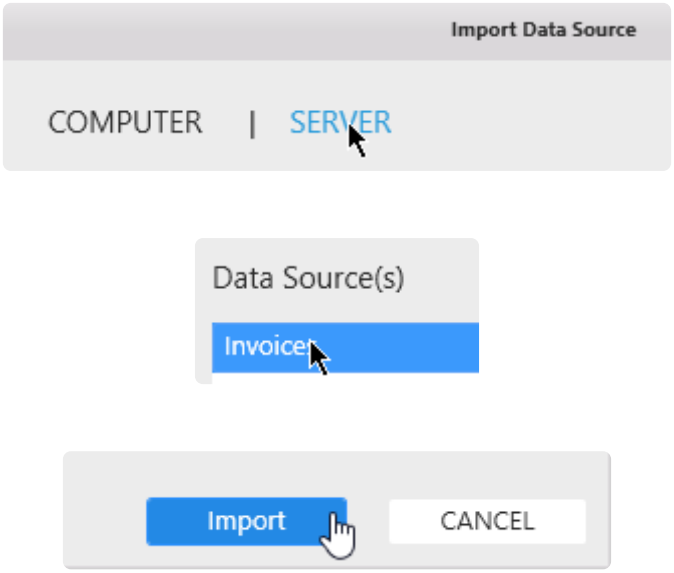
2.6.3.5. Import a Data Source from Alpana Server

In order to Import a published Data Source from Alpana Server, you need to be logged in from Alpana Designer to that Alpana Server, and have the necessary permissions to access the Data Source.

Click on the “Import” button in the Data Sources list :



In the “Import Data Source” dialog, click “Server”, select a published Data Source, and click Import :



The Data Source is now imported in your current Dashboard.

Last modified: Apr 25, 2019

2.6.4. Fetching Data

Introduction

A Data Source can fetch data from Connections by selecting the corresponding object in the Connection schema :

- a data Table
- a data View
- a Stored Procedure
- a Table-Valued Function
- an Alpana Custom Query Table (see next Chapter Custom Query Tables)

For details on how these elements exist or not based on the Connection type, please see the dedicated Course Connecting to Data.

Pre-requisite

In this Chapter, we will edit a Data Source.

To learn how to create or edit a Data Source, see the previous Chapter Creating and Managing Data Sources.

In this Chapter, it is assumed that you have created a Data Source and that it is open for edition.

Last modified: Apr 17, 2019

2.6.4.1. using the Data Preview

Previewing data

When the Data Source definition is updated, the bottom main area shows a data preview :

Data Preview | Interactions in this grid will only be reflected in the preview and will not affect the underlying data | Rows Count 100 | Load More

continent	Country	FactoryName	FactorySize	ProfitObjective	RatingO
Europe	Denmark	Viborg	8	6100000	2.4200
Europe	Denmark	Viborg	8	6100000	2.4200
Europe	Denmark	Viborg	8	6100000	2.4200
Europe	Denmark	Viborg	8	6100000	2.4200

Loading more data rows

In order to improve performance while editing the Data Source, only a few data rows are fetched.
By default, only the first 100 data rows are fetched.
If your data set contains more than 100 rows, you can click the “Load More” button on the top right to get up to 1000 rows.

Then if your data set exceeds 1000 rows, the Data Preview will still limit to 1000 rows :

Data Preview | Interactions in this grid will only be reflected in the preview and will not affect the underlying data | Rows Count 1000

FactoryName	FactorySize	ProfitObjective	RatingObjective	Enabled	Da
Beijing	50	17600000	8.19999980926514	True	1/1/20
Beijing	50	17600000	8.19999980926514	True	1/2/20
Beijing	50	17600000	8.19999980926514	True	1/3/20
Beijing	50	17600000	8.19999980926514	True	1/4/20

Refreshing the Data Preview

You can force to get new data by clicking the Refresh button on the top toolbar :

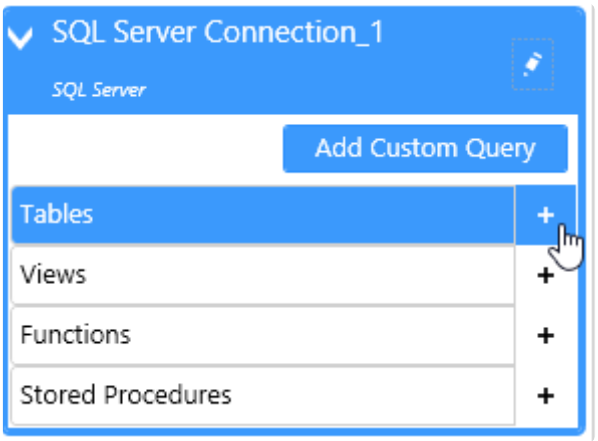
Last modified: Mar 13, 2019

2.6.4.2. getting and browsing the Connection schema

Browsing the Connection schema

In order to get data from an object in the Connection, you will need to browse the Connection schema.

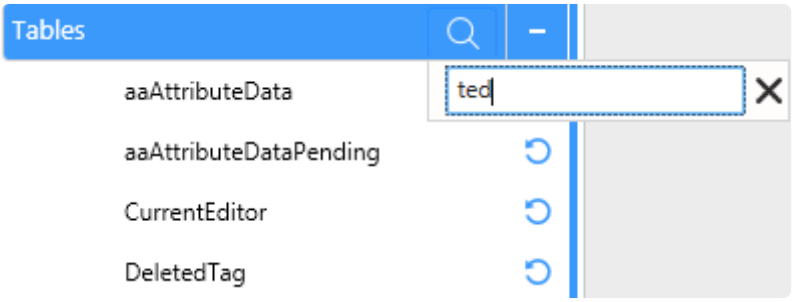
To do so, click on the Connection on the left, and use the **+** icons to navigate the schema tree :



Searching for an object

When the database contains many objects (tables, views, stored procedures, etc…), you can filter the list by clicking the magnifying glass icon :

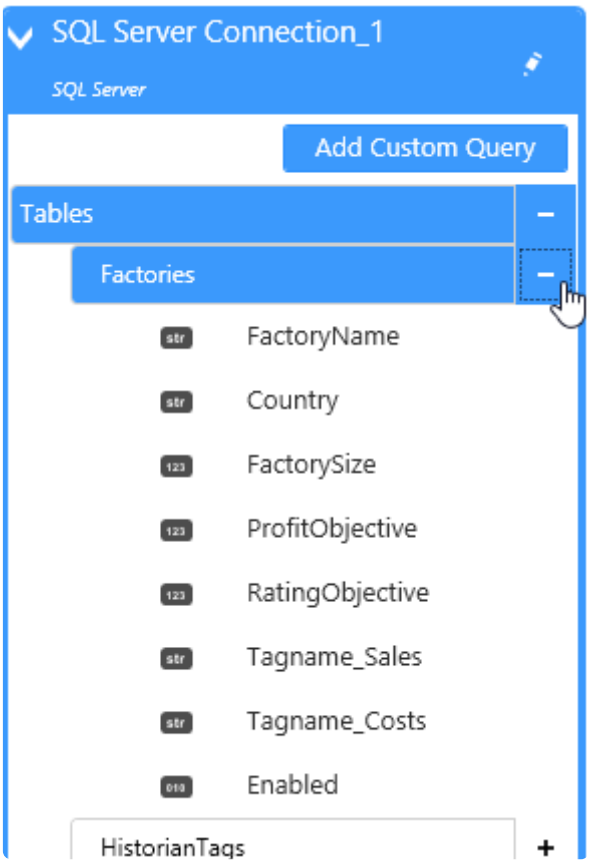
In the text box that pops up, write the text that is contained in the desired object names :



This will filter to display only the objects whose name contain the text (the search is case insensitive). Click the cross button to clear the filter.

Showing the fields of an object

Clicking one last time the + icon will show the list of fields in that object :



Field Type icon

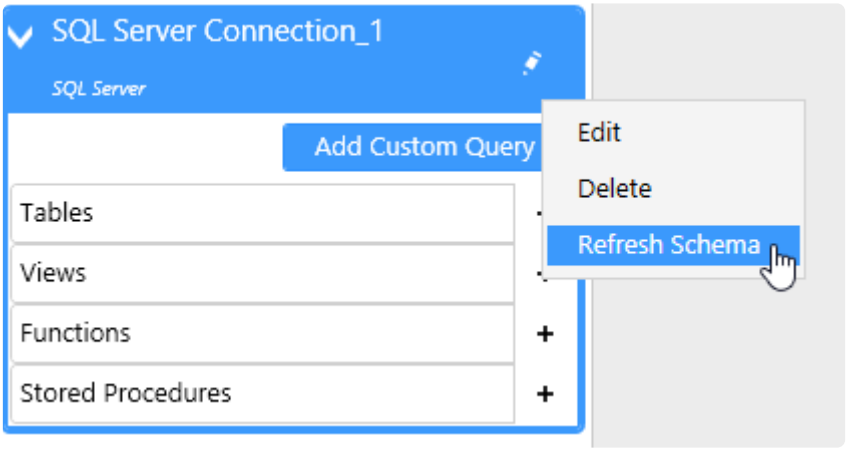
Next to each field is an icon showing the data type of that field :

- str : string
- 123 : number
- : date
- 010 : boolean

Refreshing the Connection schema

If the Connection schema has changed since you started browsing (maybe a table was added or removed, etc…), you will need to refresh it manually.

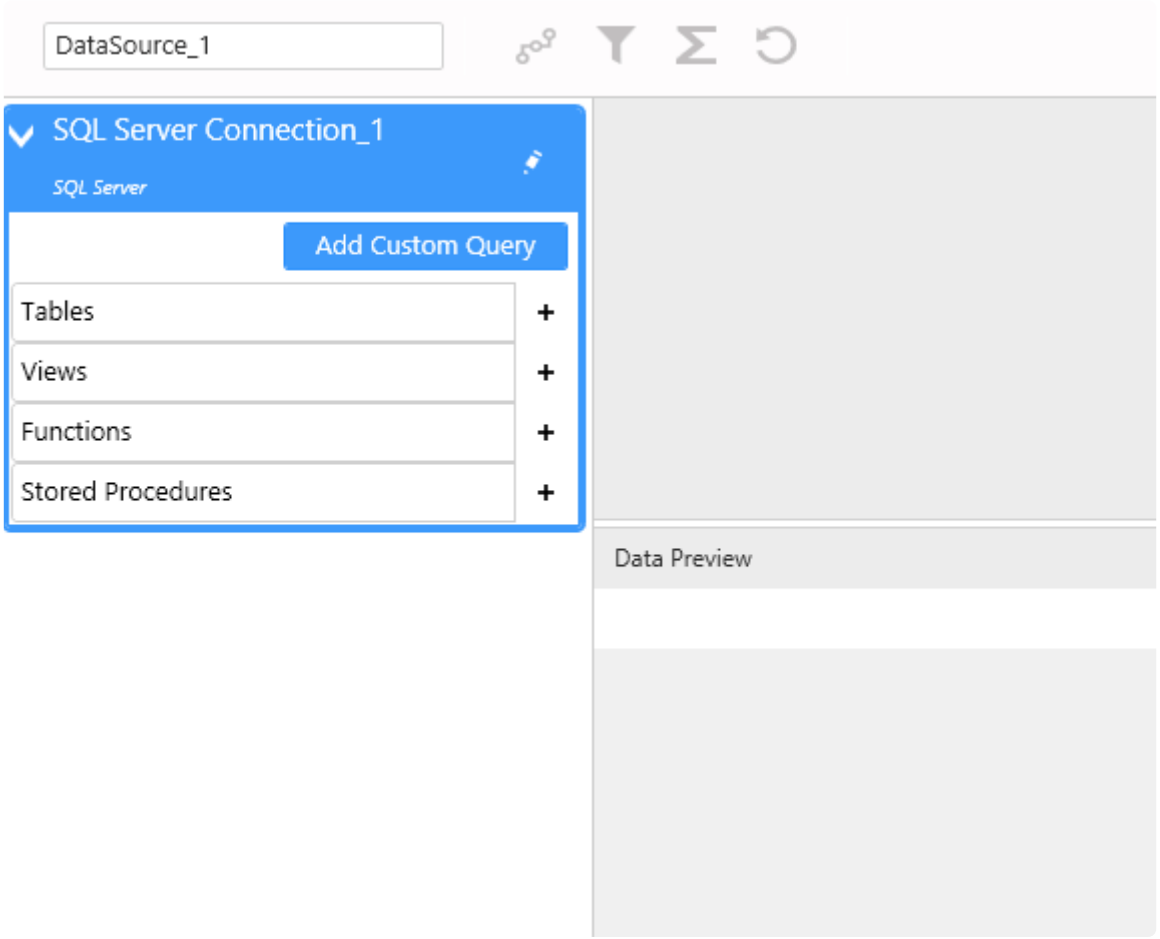
To do so, click on the pen icon on the Connection on the left, and select “Refresh Schema” :



Last modified: Dec 24, 2019

2.6.4.3. fetching data from a data Table

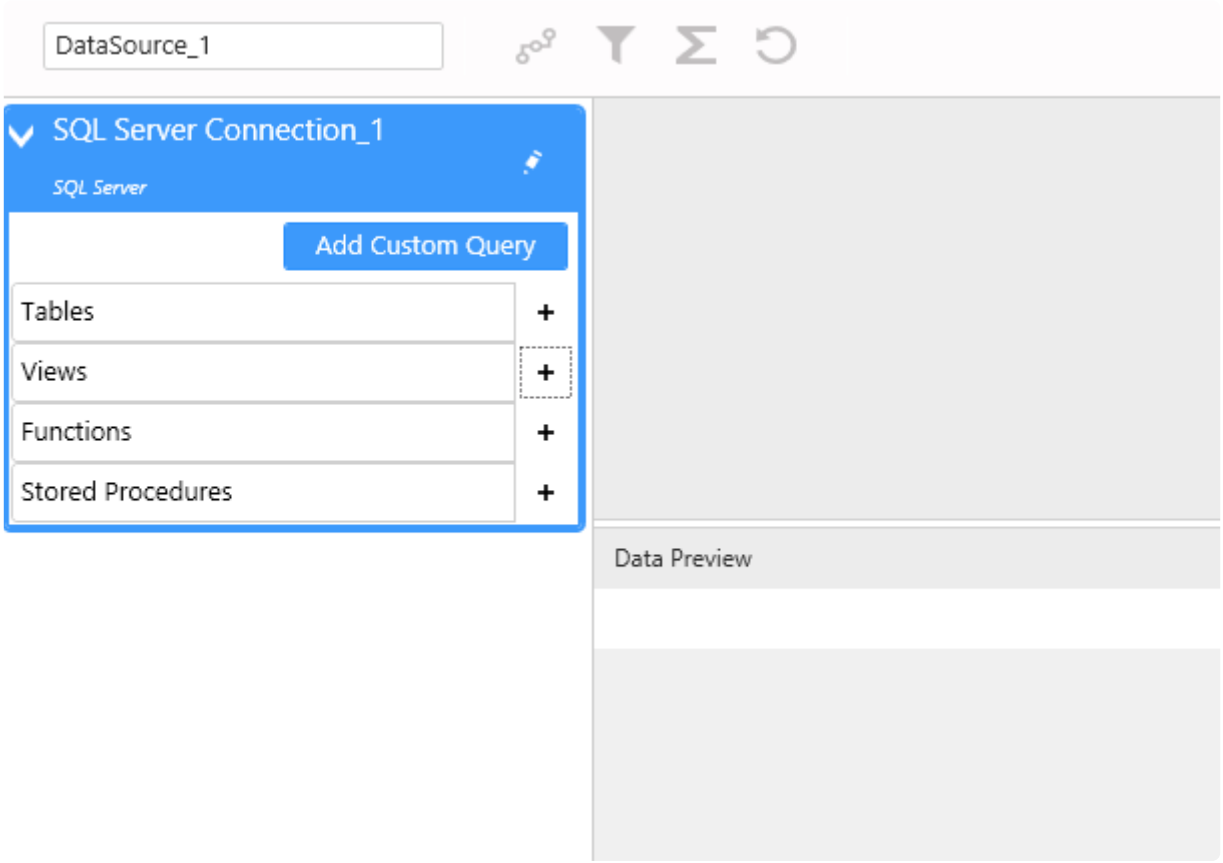
To get data from a data Table, browse to the desired Table in the schema and drag it to the top main area :



Last modified: Mar 13, 2019

2.6.4.4. fetching data from a data View

To get data from a data View, browse to the desired View in the schema and drag it to the top main area :



Last modified: Mar 13, 2019

2.6.4.5. fetching data from a Stored Procedure

Important notice

Stored Procedure can perform destructive actions when called : deleting tables, inserting data, etc…

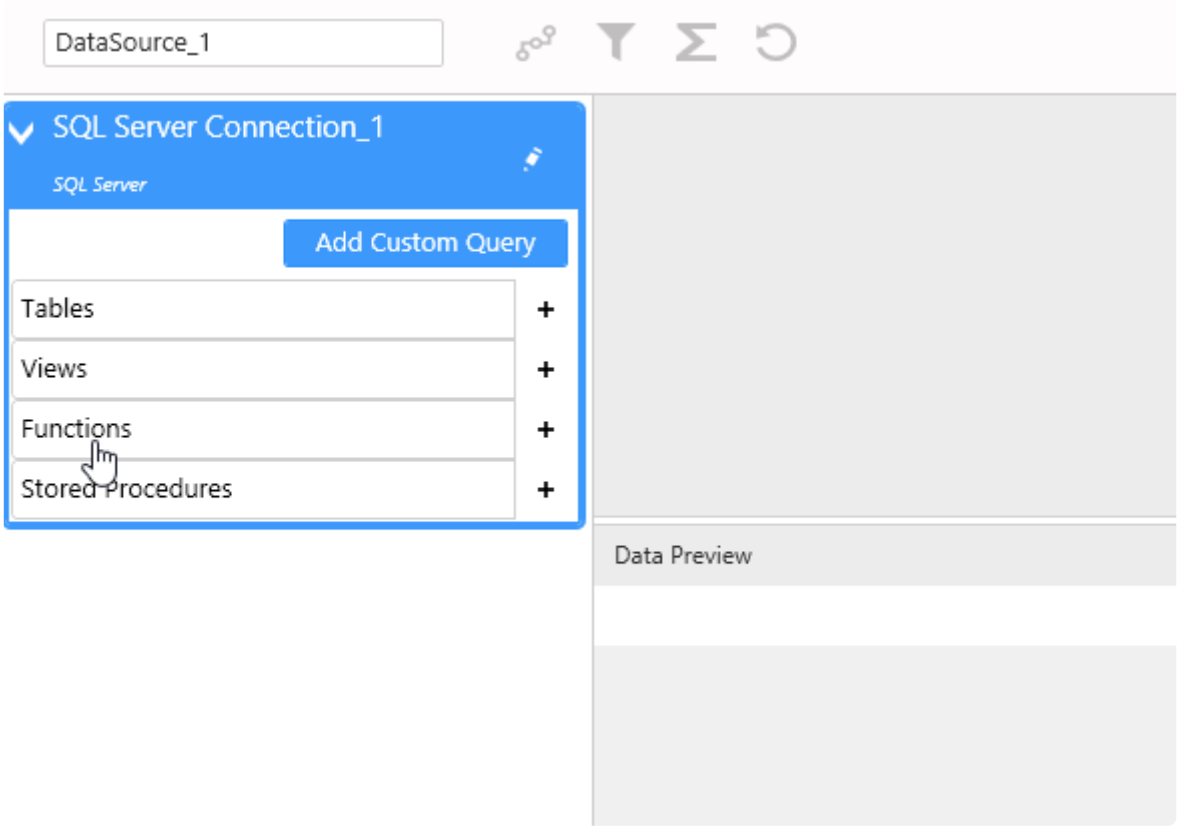
In Alpana, the Stored Procedure will be called every time data is fetched : every time anyone loads a dashboard or refreshes a filter pointing to such a Data Source, the Stored Procedure will be executed.

> ! Before executing a Stored Procedure, please make sure that it doesn't cause unwanted side effects.
> If necessary, instead of calling existing Stored Procedures, you can :

- create Table-Valued Functions or Stored Procedures that only SELECT data and don't make changes to the data.
- create dedicated Stored Procedures for your dashboarding and reporting tools, or Alpana specifically, and make sure that they are safe
  Ask your database administrator for advice.

Stored Procedure without parameters

To get data from a Stored Procedure, browse to the desired Stored Procedure in the schema and drag it to the top main area :

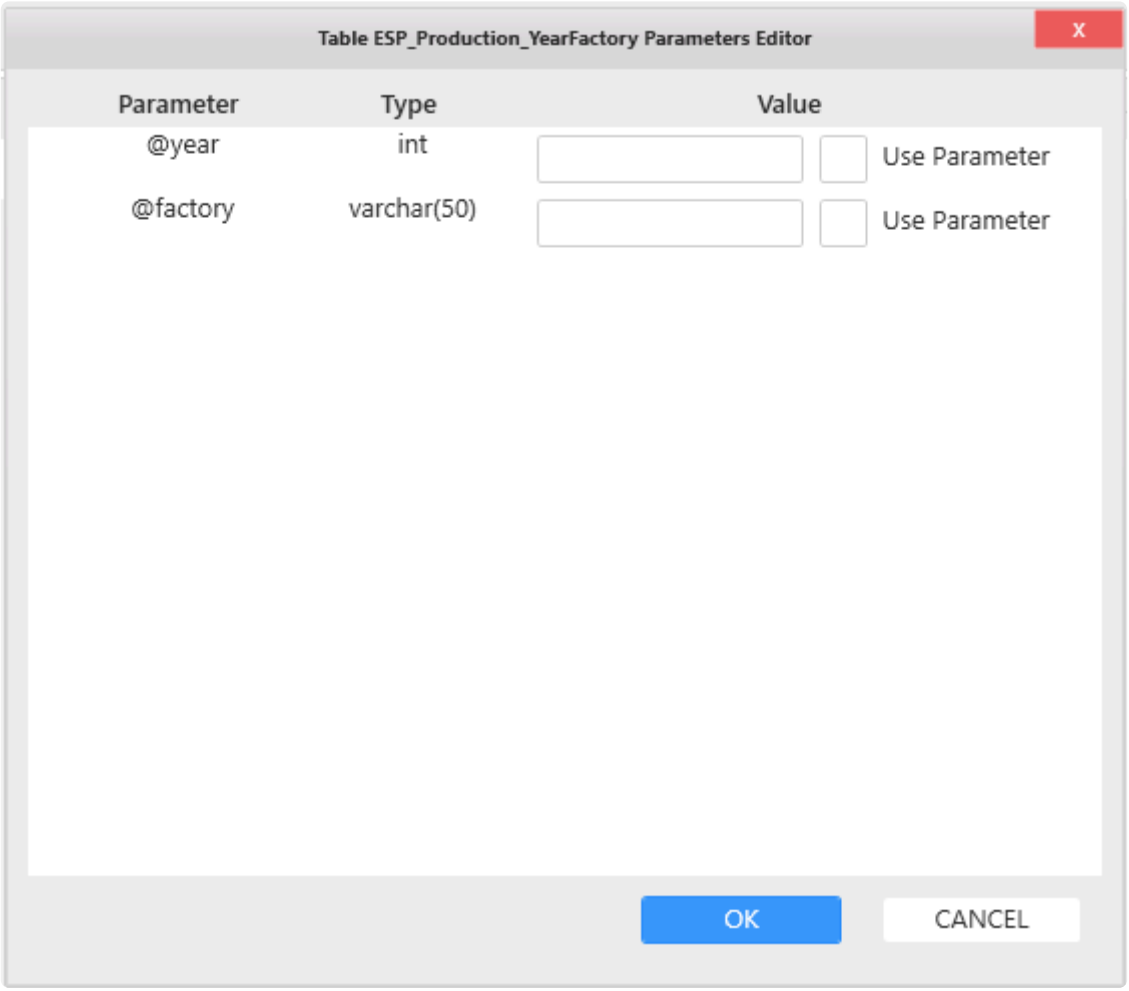


If the Stored procedure takes no input parameters, then that's all.

If the Stored procedure does take input parameters, then please proceed below :

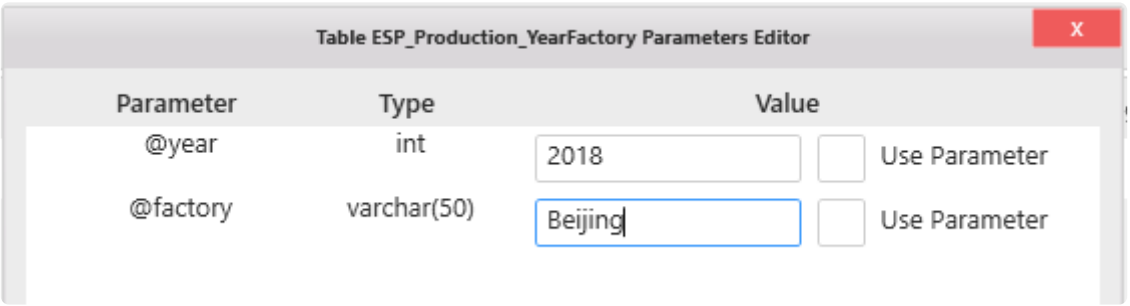
Stored Procedure with parameters

If the Stored procedure takes input parameters, a dialog appears to let you set the parameters. It lists the input parameters and their data type :



Stored Procedure with static parameters

To fill in static values for the Stored Procedure parameters, simply fill the text boxes and click “OK” :



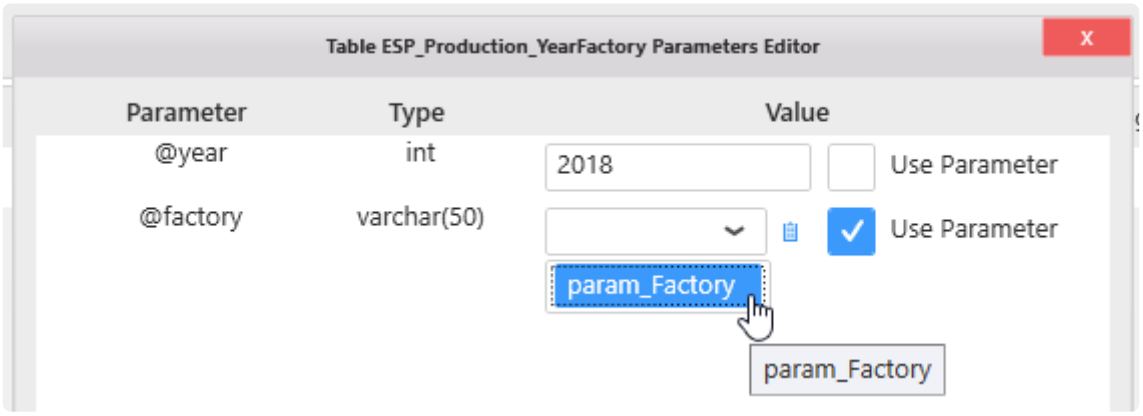
With a static value, the Stored Procedure will always be called with the value you entered.

Stored Procedure with dynamic parameters

Instead, you can bind a Stored Procedure parameter to an Alpana Dashboard Parameter (see the dedicated Course Chapter on Parameters).

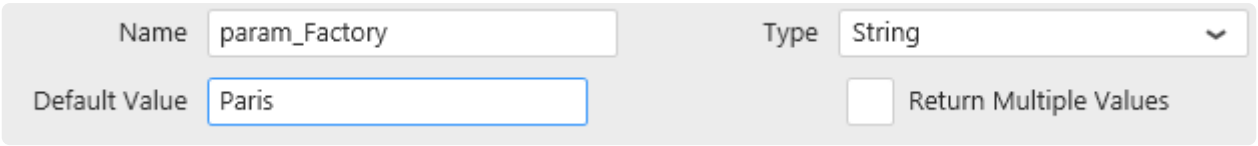
This will allow you for example to change the Stored Procedure call depending on Master Widget clicks on the dashboard.

To do so, in the “Stored Procedure Parameters Editor” dialog, check “Use Parameter” and select an existing Alpana Dashboard Parameter :



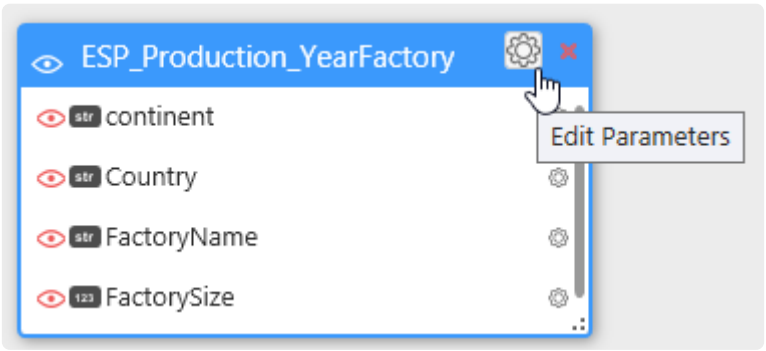
You can also create a Parameter directly form this dialog by clicking the parameter list button :

Note :
Be careful that some Stored Procedures require non empty parameters. In order to complete this step, you may have to set a default value to the Alpana Parameter :



Editing existing Stored Procedure parameter binding

Once the Stored Procedure parameters are bound, you can change them at any time by clicking the “Edit Parameters” cog icon on the corresponding table :



Stored procedure limitations

SQL Stored Procedures can do a lot of things, and some of them don’t even return data.

Alpana can only deal with certain types of stored procedures :

Single data set

Alpana supports stored procedures that return a single data set.

Stored procedures that return several data sets are not supported. For example :

```
SELECT 123;  -- first data set
SELECT 465;  -- second data set
```

… is not supported.

No output parameters

Stored procedures that use output parameters are not supported. For example :

```
CREATE PROCEDURE output_param
@outparam INT OUTPUT
AS
BEGIN
SELECT @outparam = 123;
END
```

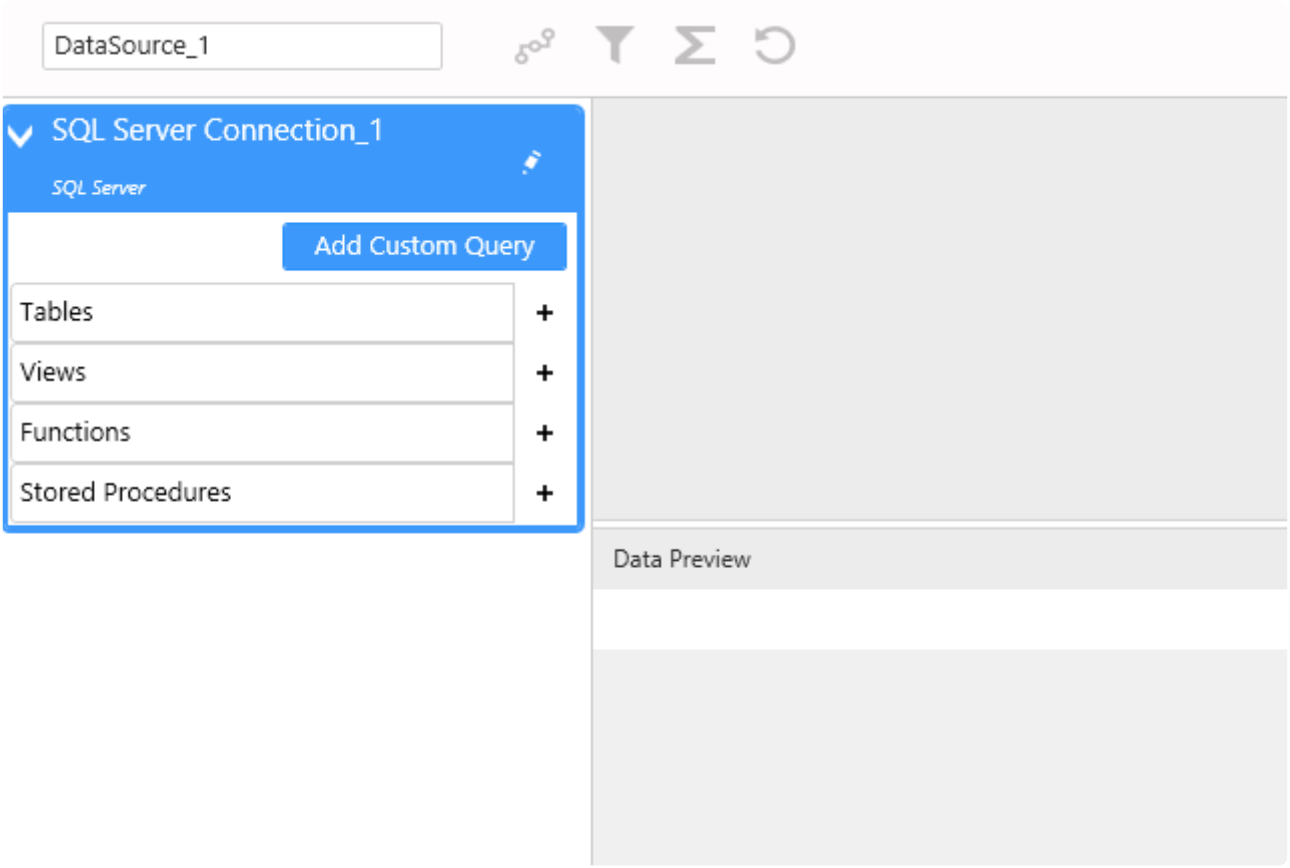
… is not supported.

Last modified: Mar 13, 2019

2.6.4.6. fetching data from a Table-Valued Function

Table-Valued Functions without parameters

To get data from a Table-Valued Function, browse to the desired Table-Valued Function in the schema and drag it to the top main area :



If the Table-Valued Function takes no input parameters, then that's all.

If the Table-Valued Function does take input parameters, then please proceed below :

Table-Valued Function with parameters

If the Table-Valued Function takes input parameters, a dialog appears to let you set the parameters. It lists the input parameters and their data type :

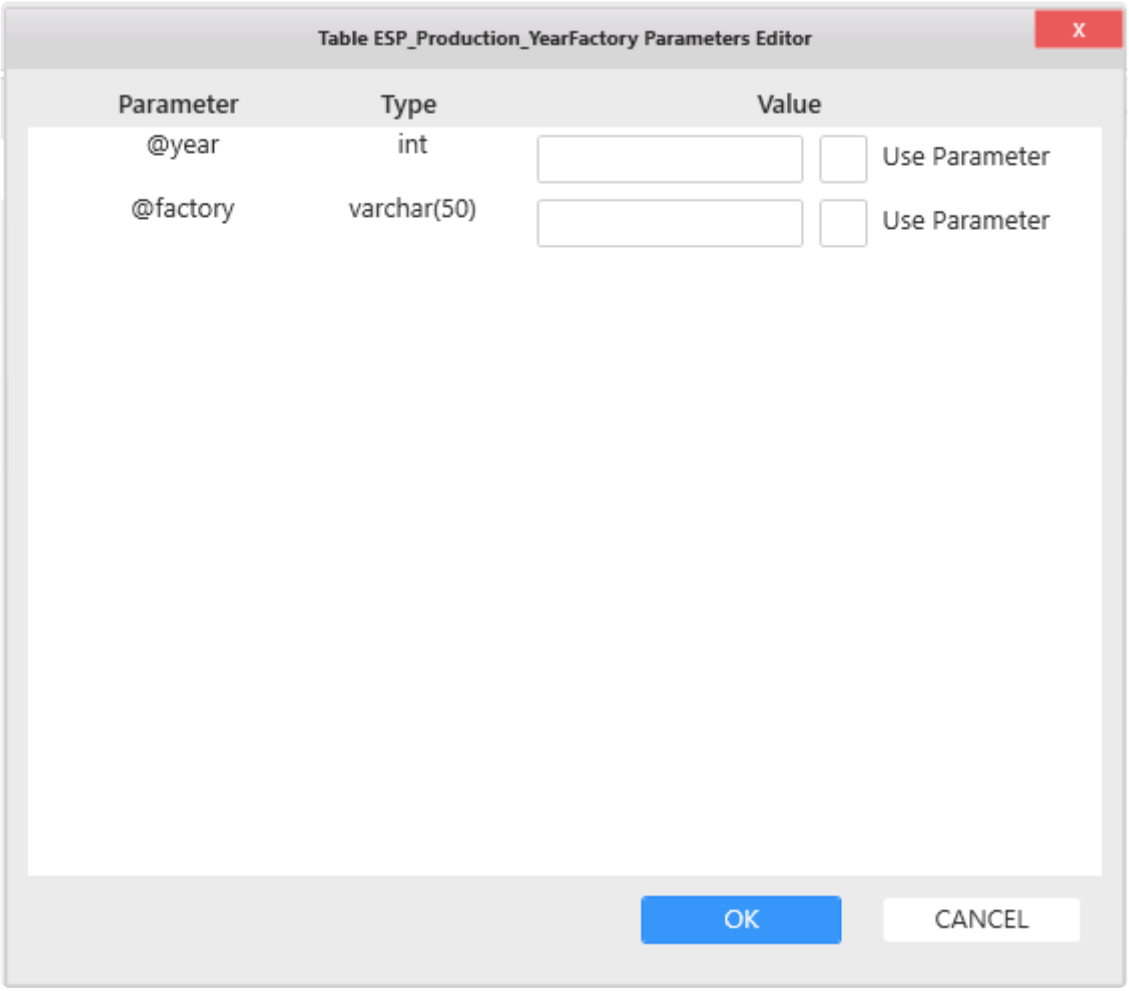
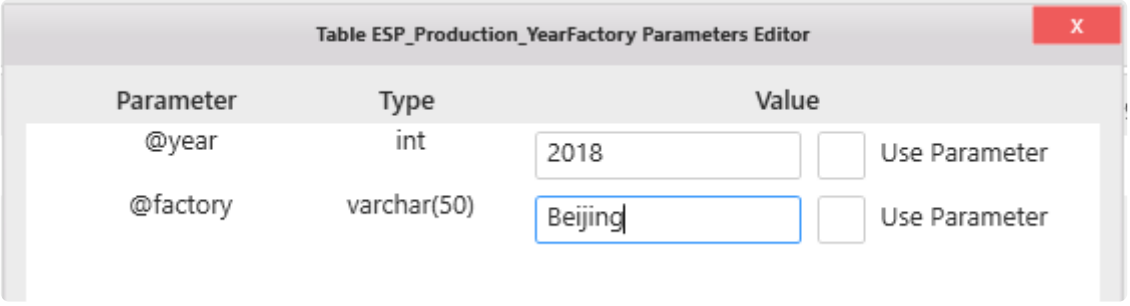


Table-Valued Function with static parameters

To fill in static values for the Table-Valued Function parameters, simply fill the text boxes and click “OK” :



With a static value, the Table-Valued Function will always be called with the value you entered.

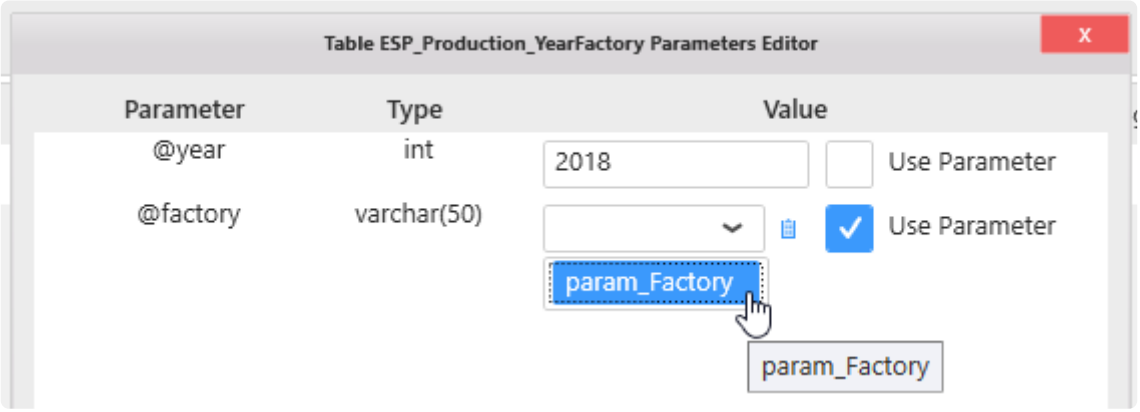
Table-Valued Function with dynamic parameters

Instead, you can bind a Table-Valued Function parameter to an Alpana Dashboard Parameter (see the dedicated Course Chapter on Parameters).

This will allow you for example to change the Table-Valued Function call depending on Master Widget

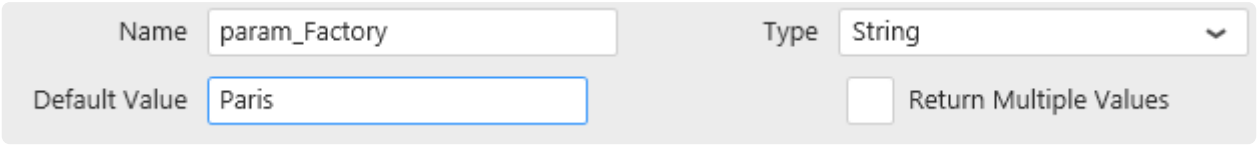
clicks on the dashboard.

To do so, in the “Table-Valued Function Parameters Editor” dialog, check “Use Parameter” and select an existing Alpana Dashboard Parameter :



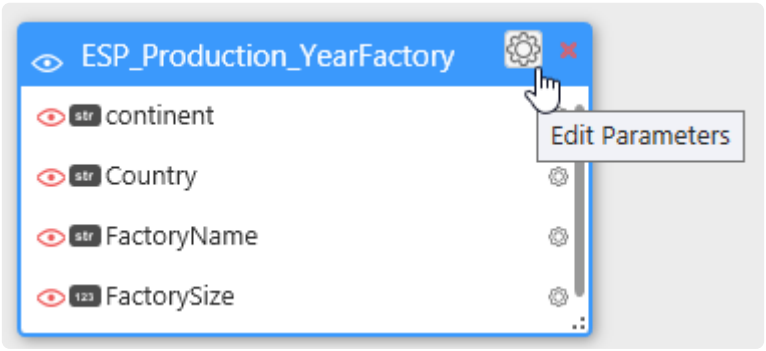
You can also create a Parameter directly form this dialog by clicking the parameter list button :

Note :
Be careful that some Table-Valued Function require non empty parameters. In order to complete this step, you may have to set a default value to the Alpana Parameter :



Editing existing Table-Valued Function parameter binding

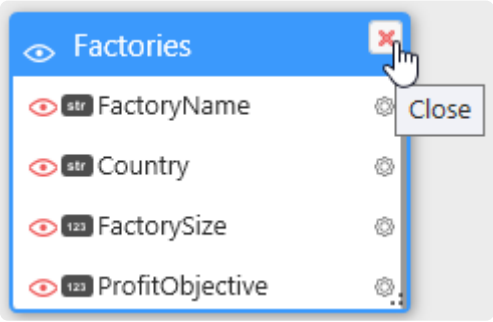
Once the Table-Valued Function parameters are bound, you can change them at any time by clicking the “Edit Parameters” cog icon on the corresponding table :



Last modified: Mar 13, 2019

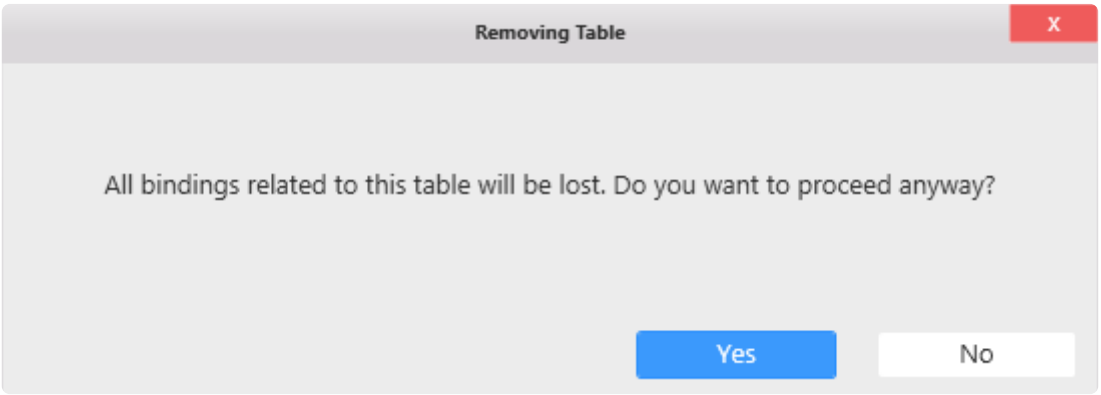
2.6.4.7. Removing a table from the Data Source

You can remove an object (Table, View, etc, …) from the Data Source by clicking the cross on the top right of the table :



If the table is in a relationship with other tables and removing it breaks Join conditions, a confirmation dialog appears :

All bindings related to this table will be lost. Do you want to proceed anyway ?



If you press OK, the Join Editor dialog appears and lets you fix the missing relationships (see below Chapter “Merging Data”).

Last modified: Mar 13, 2019

2.6.5. Custom Query Tables

What is a Custom Query Table ?

When composing a Data Source, you are not limited to using Tables or Views from your Connection.

You can also add a **Custom Query Table** : similarly to a SQL View, you write a SQL query and a Table will be created in the currently opened Data Source.

However, this can be more than a simple way to create a custom View, as we will see further in the Using Parameters chapter.

Pre-requisite

In this Chapter, we will edit a Data Source.

To learn how to create or edit a Data Source, see the previous Chapter “Creating and Managing Data Sources”.

In this Chapter, it is assumed that you have created a Data Source and that it is open for edition.

Last modified: Mar 07, 2019

2.6.5.1. Language

When connecting directly to a database, the language of the Custom Query must be compatible with the database.

When connecting to a bufferized Connection, the Custom Query must be written in the language used by the **Buffer** database.

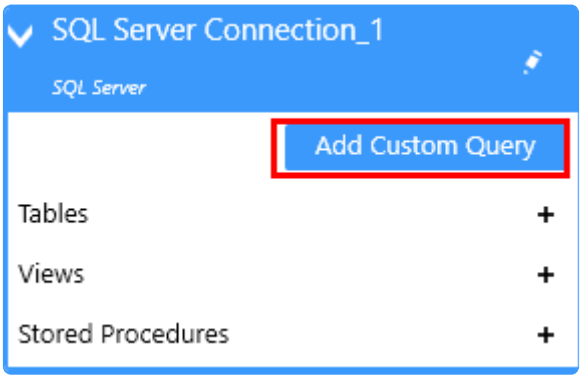
> ! Important
> When changing the underlying buffer database technology (for example when switching from SQL Server to Oracle), the Custom Query Table will probably need to be updated accordingly, since there are some differences in their SQL syntaxes.

Last modified: May 11, 2020

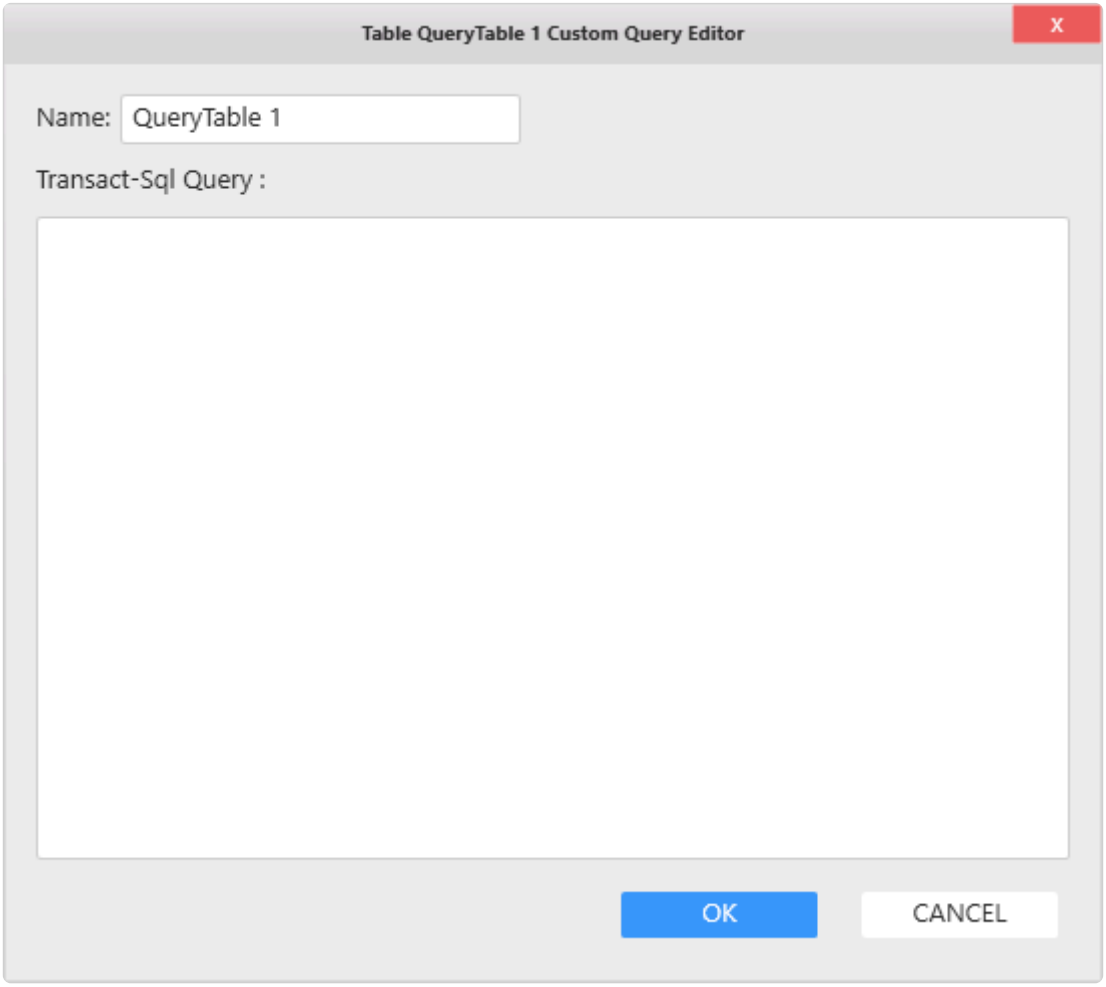
2.6.5.2. Creating a Custom Query Table

To do so, open the Data Source where you need to add a **Custom Query Table**.

From the Connection schema below, click on Add Custom Query.

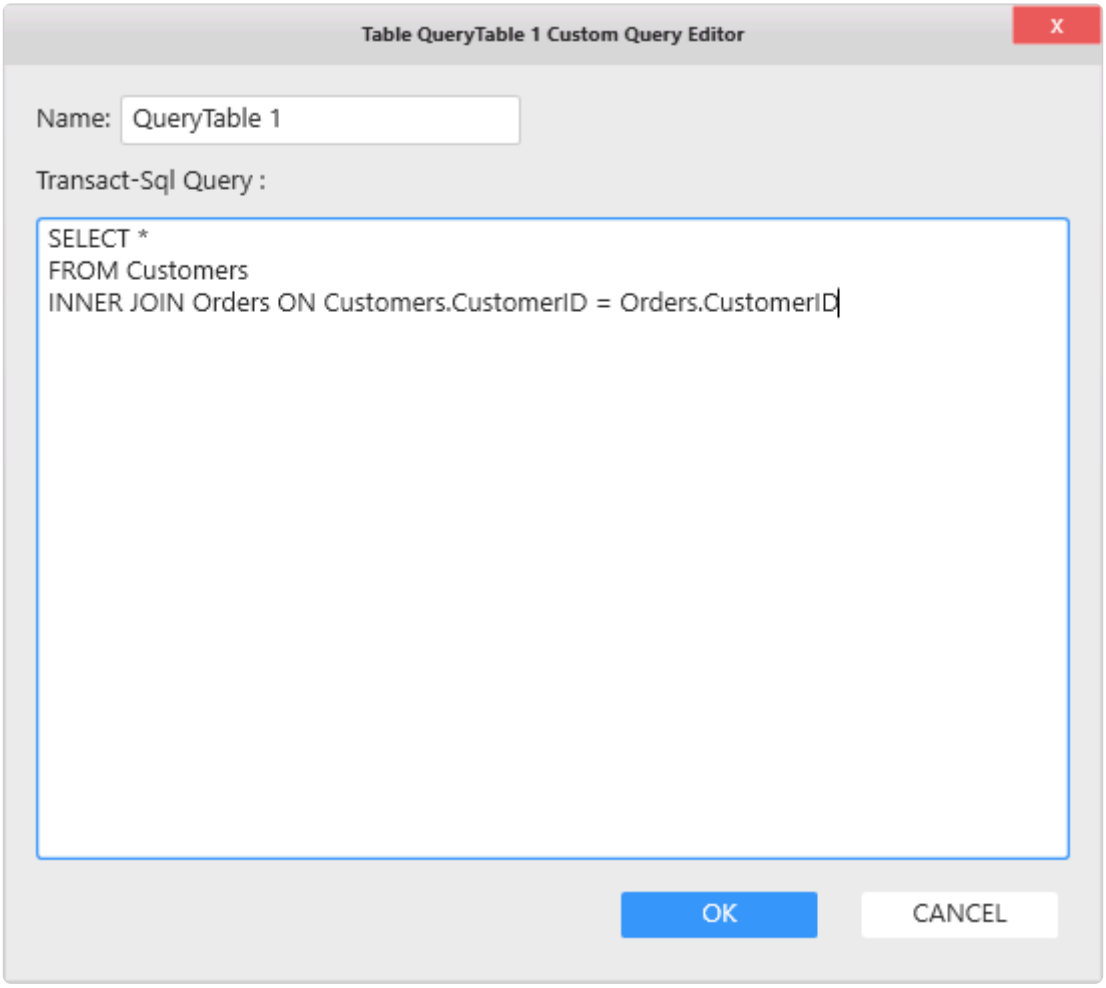


This will bring up the Custom Query Editor frame.



Write the query in the correct SQL language (see the previous chapter “Language”).

For bufferized Connections, you can use the table names as listed in the schema.



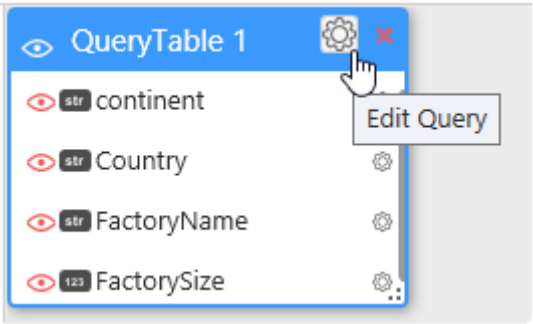
Click OK to create the Custom Query Table. Fix any SQL error if necessary.

Now the Custom Query Table is created in the Data Source currently opened. It can be used like a normal table from the same Connection and, for example, can be used in a Join.

Last modified: Mar 13, 2019

2.6.5.3. Updating a Custom Query Table

To do so, click on the cog icon to view and update the Custom Query.



> **Important**
> Be careful that any changes in the schema (column names and types) could impact the items that use this Custom Query Table, such as Widgets or Expressions. Changing the schema of the query could require to re-do some or all of the related data binding.

Click OK to update the Custom Query Table.

Fix any SQL error if necessary.

Last modified: Mar 13, 2019

2.6.5.4. Using Parameters

Parameters can be used to parametrize a Custom Query : the Custom Query changes dynamically.

Syntax

`${parametername}` will be replaced in-place by the values of "*parametername*".

Example :

```
SELECT *
FROM MyTable
WHERE MyColumn = ${parametername}
```

will get replaced at runtime by :

```
SELECT *
FROM MyTable
WHERE MyColumn = parametervalue
```

Multiple values

Parameters that return multiple values will return them separated with commas.

So that you can write :

```
SELECT *
FROM MyTable
WHERE MyColumn IN( ${multiparameter} )
```

and it will get replaced at runtime by :

```
SELECT *
FROM MyTable
WHERE MyColumn IN( Value1,Value2,Value3 )
```

NOTE
In Alpana Designer, the Custom Query will be executed with the default parameter values. This may cause unwanted syntax errors if the default value is empty for example. Setting a default value to the parameter fixes that issue.

Last modified: Mar 07, 2019

2.6.6. Merging Data

Last modified: Feb 21, 2019

2.6.6.1. Creating Joins

Why Join ?

When an object (a Table, a View, etc…) is already present in the Data Source and you want to add another object, you will have to specify how the two objects will be merged together to create a single data query.

In order to do this, a database JOIN operation is used.

> What is a join ?
> In simple terms, a join between a data Table1 and a Table2 expresses how an entry in Table1 describes the same information as another entry in Table2.
> “Joining” comes from SQL terminology. If you know SQL, this editor should expose familiar concepts.

This operation can be very useful for adding context to your data :

- your Table1 contains physical measurements
- your Table2 contains context information like where was data measured, what product you were producing at that time, etc…
  ⇒ This is all multiple aspects of the same information. By joining, you will be able to present all this data consistently in your visualizations.

What can I join ?

Joins can be created across all Connections, all Connection types (SQL, Excel, Historian, …), all object types (Tables, Views, Stored Procedures, …).
Any Number of join conditions can be created.

> Note : Multi-Connection
> When a Data Source joins tables from at least 2 different Alpana Connections, the Connections become bufferized.

How to Join tables

When an object (a Table, a View, etc…) is already present in the Data Source and you add another object, the Join Editor dialog automatically opens to let you configure the join relationship.

If you cancel this dialog, the new object is not added.

Last modified: Apr 17, 2019

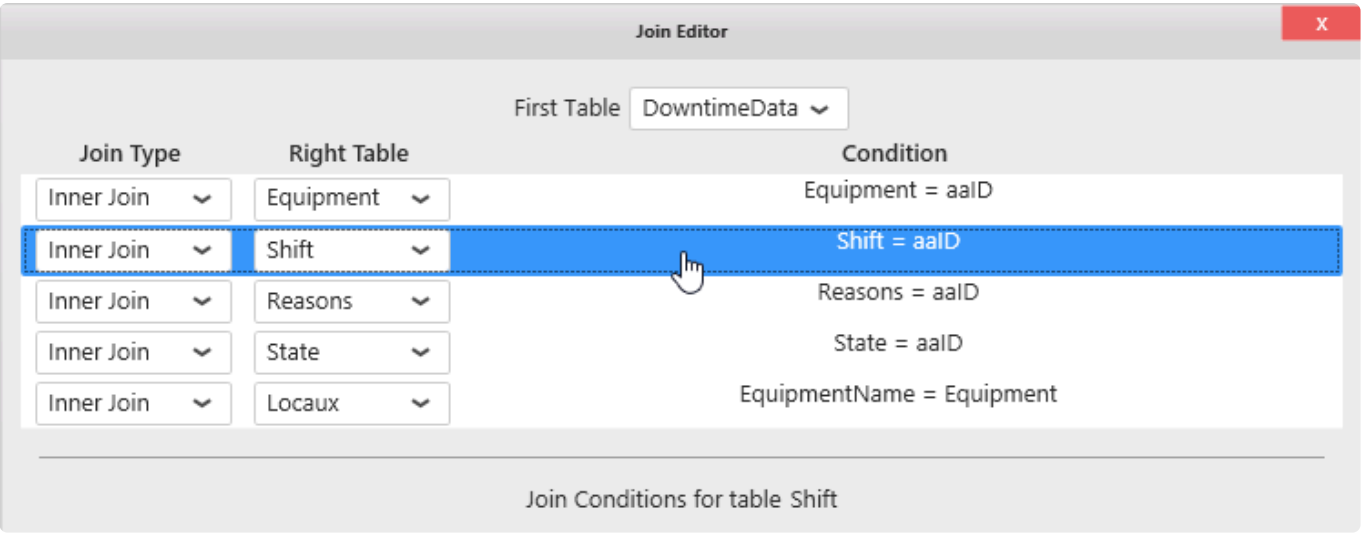
2.6.6.2. Configuring Joins

Configuring a relationship

The Join Editor presents a list of relationships similar to a SQL join :
Data is retrieved from a first table joined with X other tables.

This relationship expresses how an entry in Table1 will be on the same row as another entry in Table2.

Each relationship can be selected by clicking in the list :

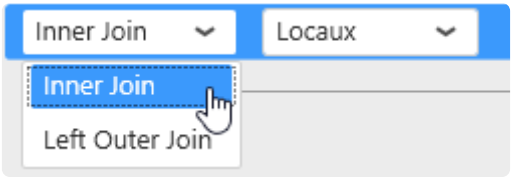


> There is 1 relationship for each table added beyond the first. If this is your first join, there is only one relationship.

Selecting the Join Type

A list allows to select the Join Type :

- *Inner Join* : only retrieve data where the join condition is true
- *Left Outer Join* : retrieve all data from the first table (“left table”) even if the join condition is not true. If the join condition is false and no data exists in the second table (“right table”), then *NULL* data is retrieved for the second table.



Join Conditions

In order to match data between tables, a condition must be expressed.
For example, if Table1 contains Customer orders and Table2 contains Persons profiles, you may want to join on “CustomerName = PersonsName”.
This means that the resulting dataset will show Persons information on the same row as Customer order information when the Name is the same.

By default a join condition is proposed in the editor.

Configuring a Join Condition

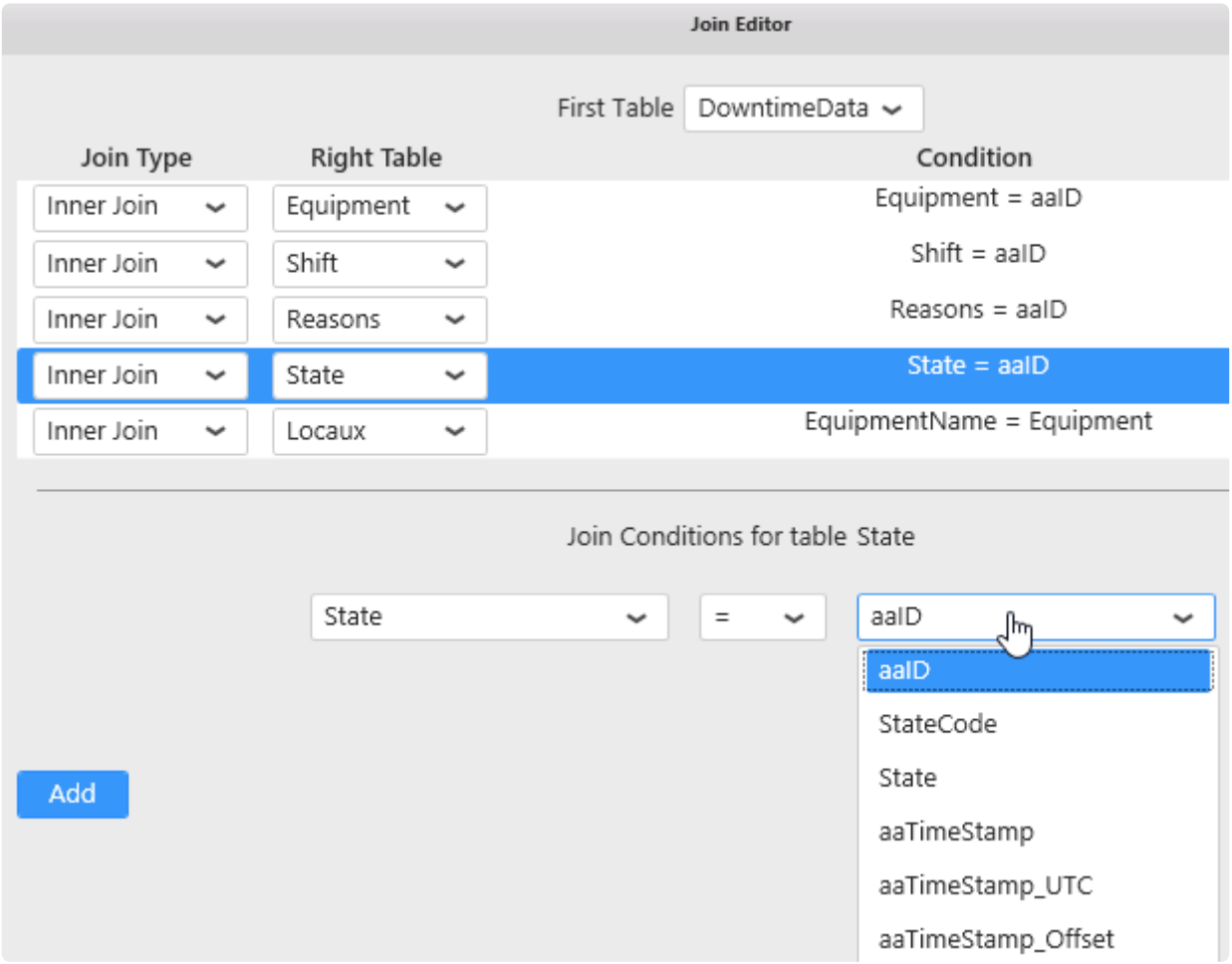
A condition is composed of 3 elements from right to left :

- a right member : a field from the currently selected Right Table. For your first join, this is the table you just added to the Data Source
- an operator : will express the relationship
- a left member : a field from any other table

To configure the Condition :

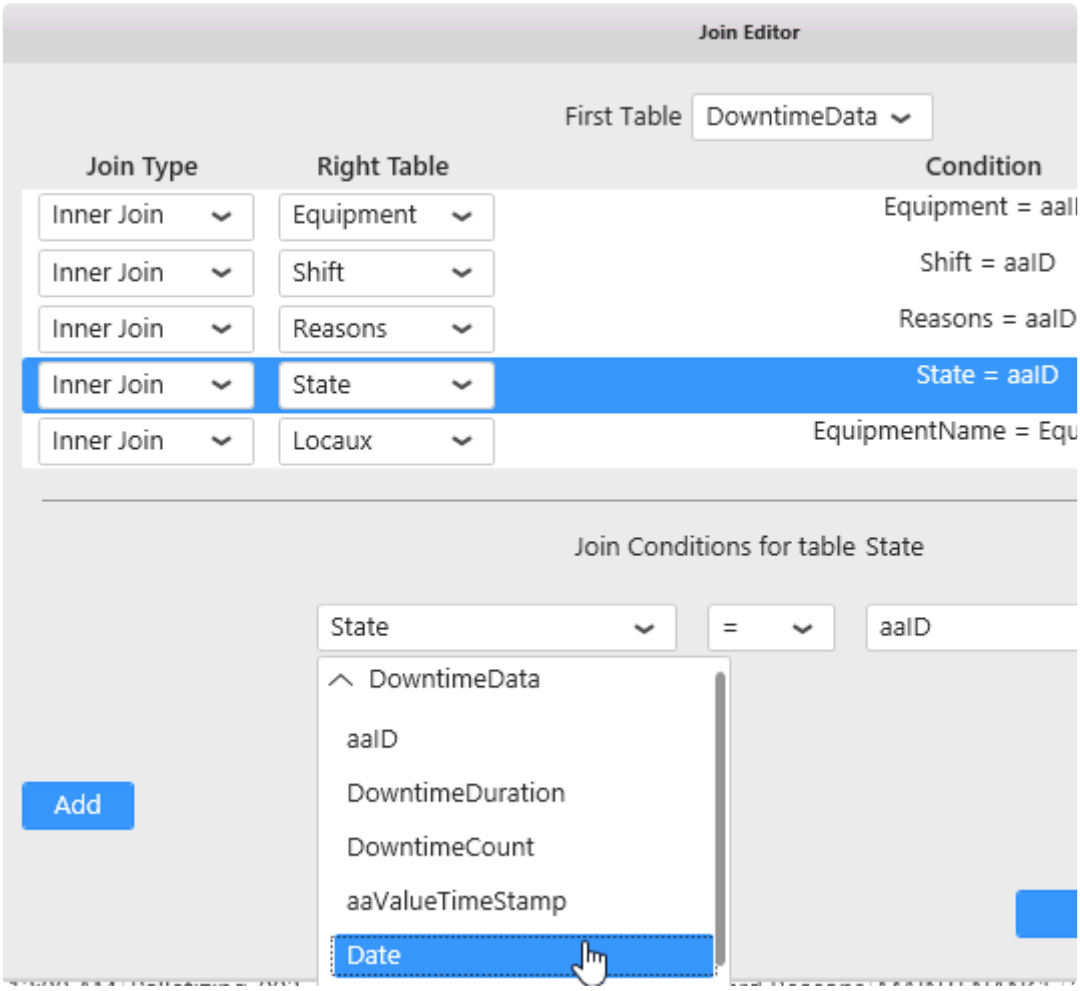
Select a right member

Select a field from the current table which will have to match the left field :



Select a left member

Select a field from any other table which will have to match the selected right field.
In the list, the fields are grouped by table names :



Select an operator

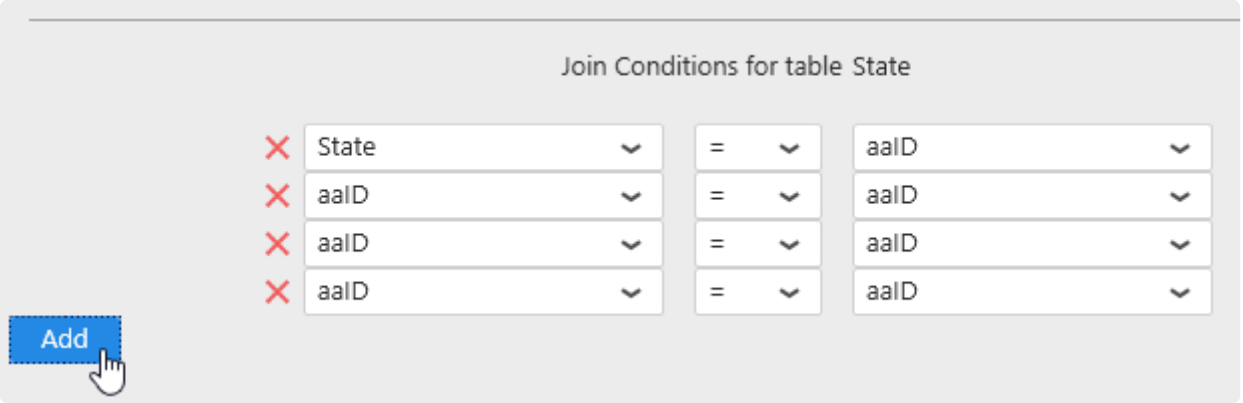
Select an operator from the list to express the desired relationship between the two selected fields :

Multiple join conditions

Sometimes a single condition is insufficient to express the desired relationship.

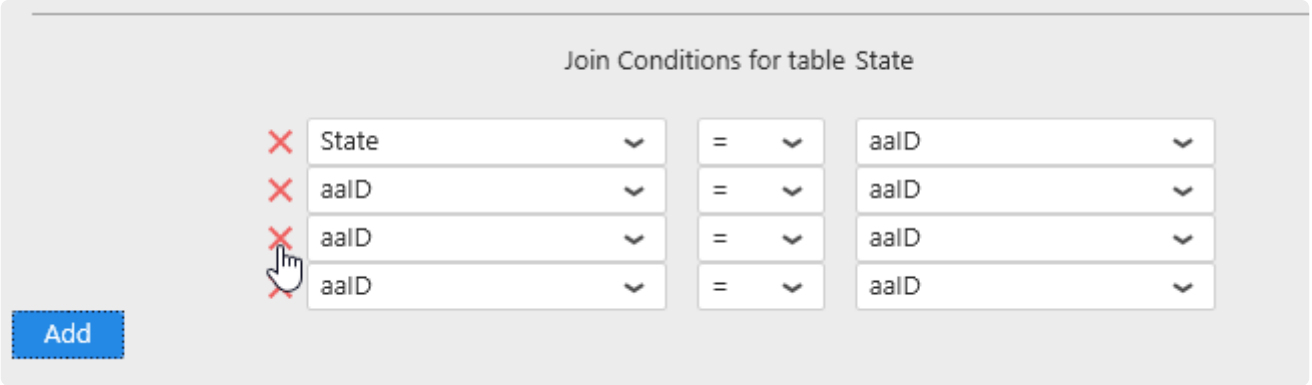
Adding a condition

It’s possible to add as many conditions as desired.
Conditions can be added by clicking the “Add” button :



Removing a condition

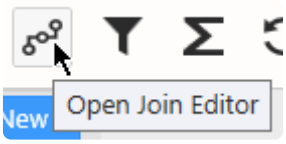
A minimum of 1 condition is required.
Each condition can be removed from the relationship by clicking on the corresponding red cross :



Last modified: Mar 14, 2019

2.6.6.3. Updating Joins

You can edit existing Join relationships by clicking on the “Open Join Editor” button in the top toolbar :



In the Join Editor, select a relationship, change it, and press OK.

Last modified: May 28, 2019

2.6.7. Filtering Data

Data Source Filters

You can apply filters at the Data Source level.
All Widgets that will use this Data Source will only see the data after the filter is applied.

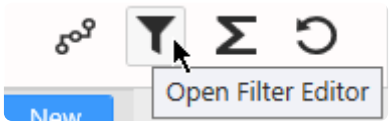
Note : If you wish to apply different filters to each Widget separately, see the corresponding chapter.

Last modified: Apr 30, 2019

2.6.7.1. Opening the Filter Editor

Add or Edit Filters

To add Filters to a Data Source or edit existing filters, press the “Open Filter Editor” funnel icon in the top toolbar :



See the presence of Filters

When a filter is applied on a DataSource, the Filter button is colored :

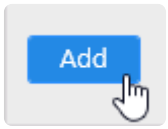
Each filtered table field appears with a yellow background and the word (Filtered) next to its name :

Last modified: Apr 30, 2019

2.6.7.2. Managing Filter Conditions

Adding a condition

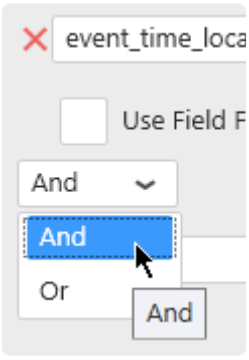
In the Filter Editor, click Add to add a filter condition :



Combining conditions

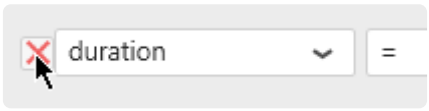
If filter conditions already exist, they will be chained using a logical operator.

By default, the operator is “And”, but it can be changed to “Or” :



Removing Conditions

Click the red cross on the left to remove the corresponding condition :



Last modified: Mar 15, 2019

2.6.7.3. Configuring Filter Conditions

Filter conditions are expressed with :

- a field name from the Data Source. This can also be an Expression.
- a boolean operator
- a value. This can also be a Parameter.

Depending on the Data Source field type, the settings are slightly different, please see for each type in the next pages.

Last modified: Mar 15, 2019

2.6.7.3.1. Filtering string fields

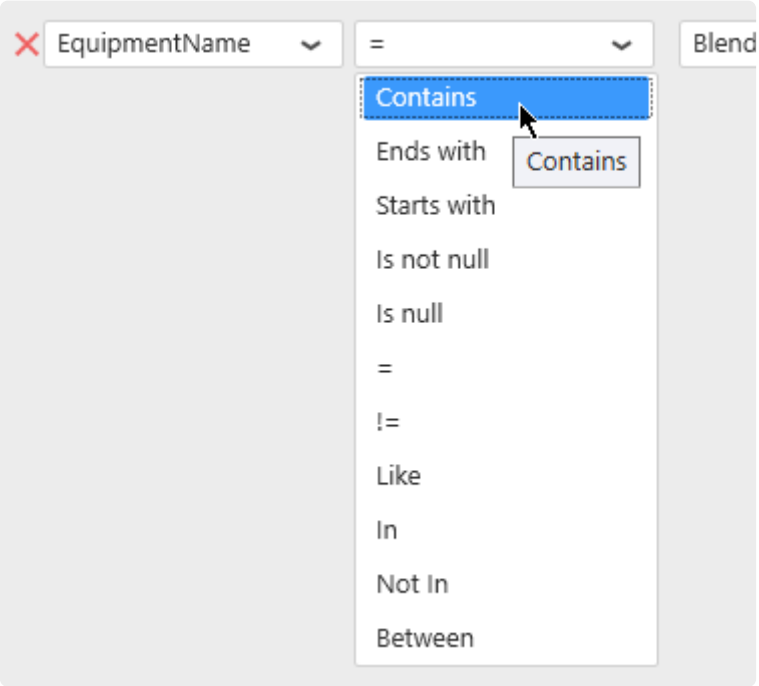
This is the most basic case.

The filter condition will be :
`<FieldName> <Operator> <Value>`

Example :
`EquipmentName StartsWith Blender`

Operators

Operators appropriate to Strings can be selected.

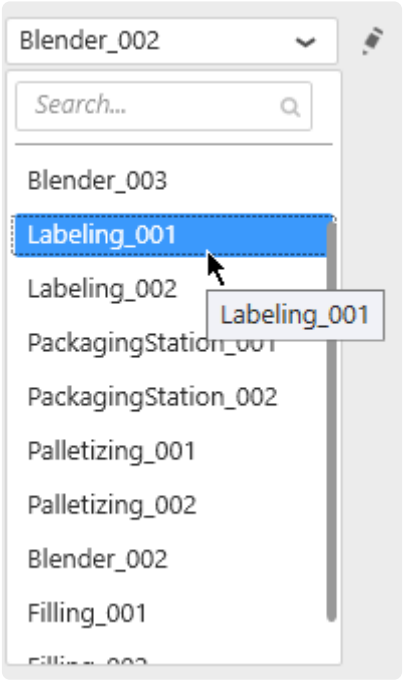


Values

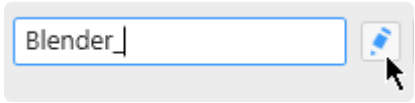
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text :



Depending on the operator selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend on actions inside the Dashboard.

For this, select “Use Parameter” and see the Chapter about Parameters.

Last modified: Mar 15, 2019

2.6.7.3.2. Filtering number fields

In a basic configuration, the condition is expressed as :

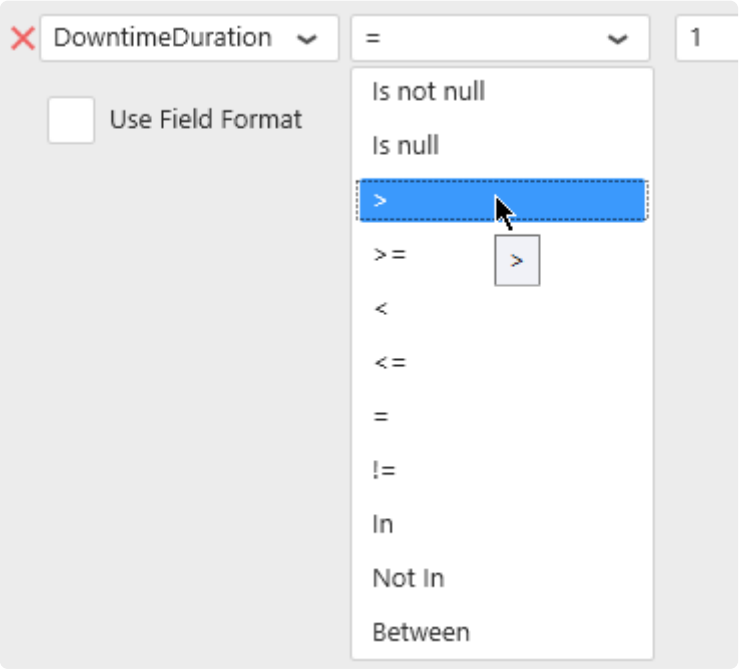
```
<FieldName> <Operator> <Value>
```

For example :

```
DownTimeDuration > 10
```

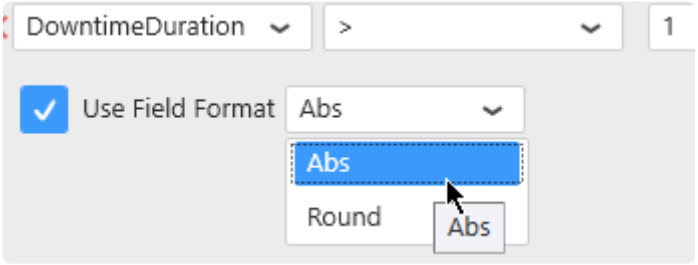
Operators

Operators appropriate to Numbers can be selected :



Field formatting

In addition to the basic configuration, a formatting function can be applied to the numerical field while the comparison is made :



This way, the filter condition will be :

```
<Function>(<FieldName>) <Operator> <Value>
```

For example :

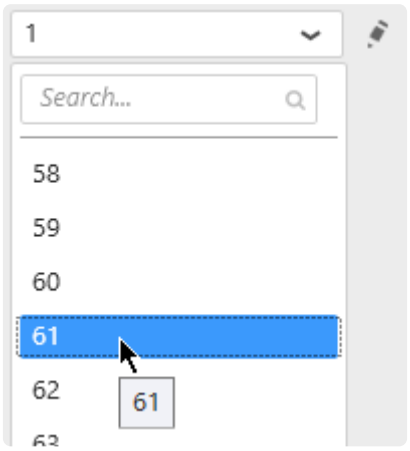
```
Abs(DownTimeDuration) > 10
```

Values

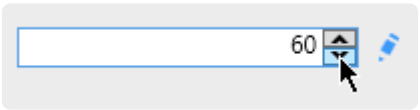
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text, or using the up/down arrows to increase/decerease :



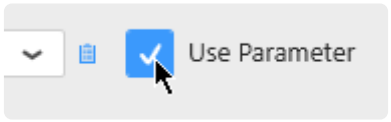
Depending on the operator selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend

on actions inside the Dashboard.

For this, select “Use Parameter” and see the Chapter about Parameters.



Last modified: Mar 15, 2019

2.6.7.3.3. Filtering date fields

In a basic configuration, the condition is expressed as :

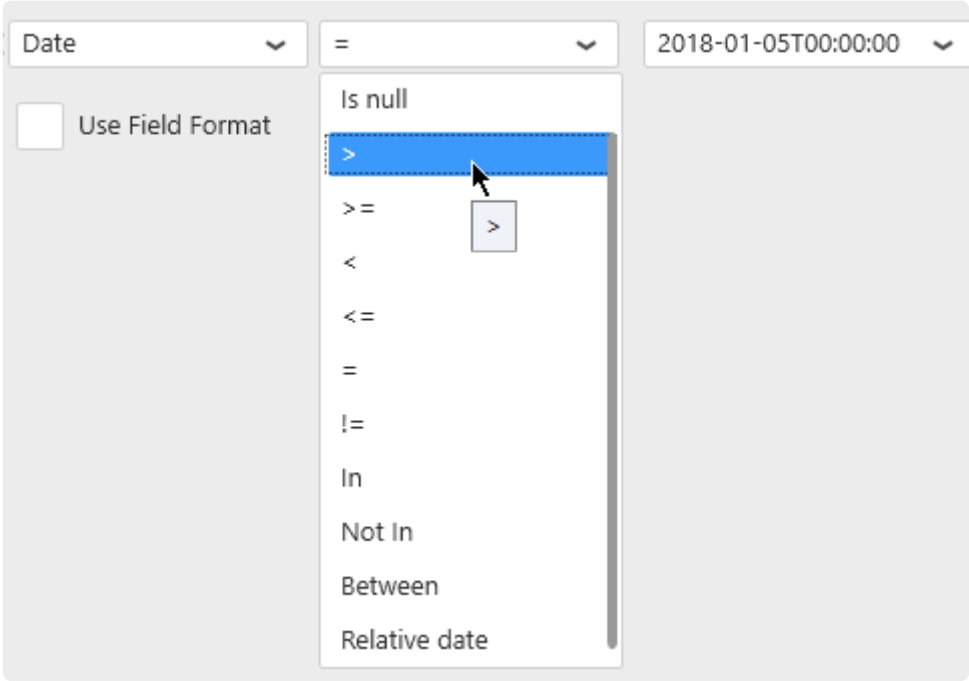
```
<FieldName> <Operator> <Value>
```

For example :

```
aaValueStartTime > 2018-01-05T00:00:00
```

Operators

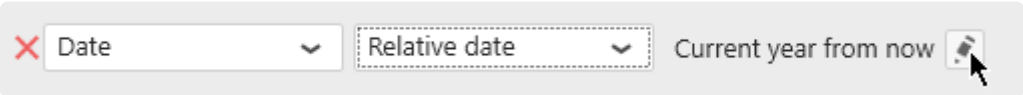
Operators appropriate to Dates and Times can be selected :



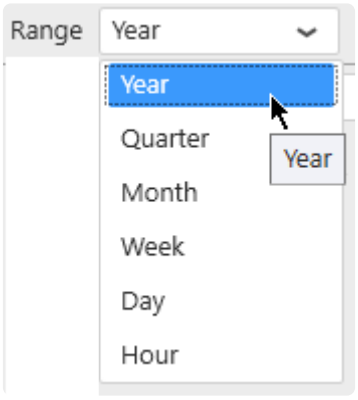
Relative date filtering

The special operator “Relative date” allows to select a “date window” / “time frame” that is moving over time relative to the current moment.

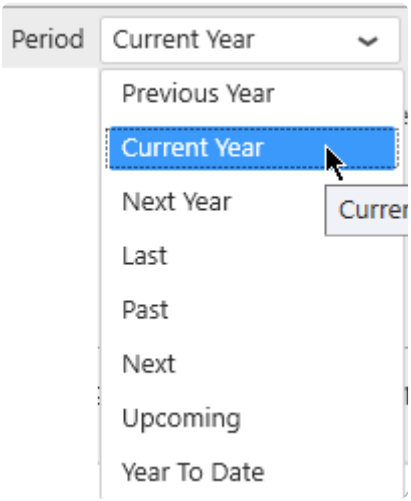
When selecting this operator, click the edit icon to define the relative date window :



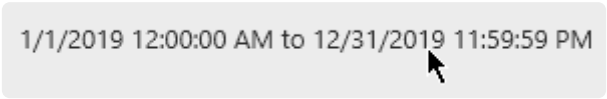
Then for each time range :
Year, Quarter, Month, Week, Day, Hour



… a Period can be selected :
Previous 1, Current, Next 1, Last N, Past N, Next N, Upcoming N, To Date

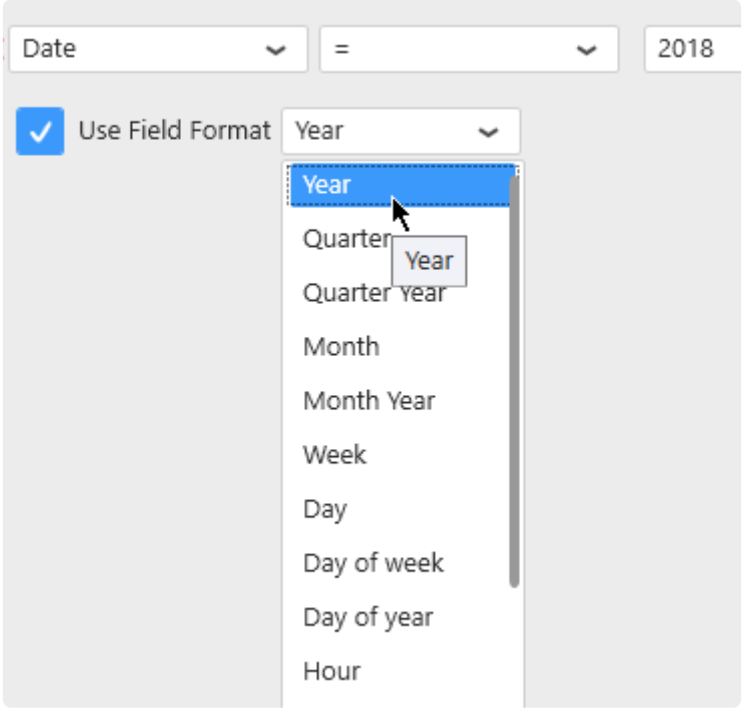


A preview of the selected date range for the current time appears at the bottom of the window to help you configure :



Field formatting

Similarly to the number fields, a formatting function can be applied to the date field :



This way, the filter condition will be :

```
<Function>(<FieldName>) <Operator> <Value>
```

For example :

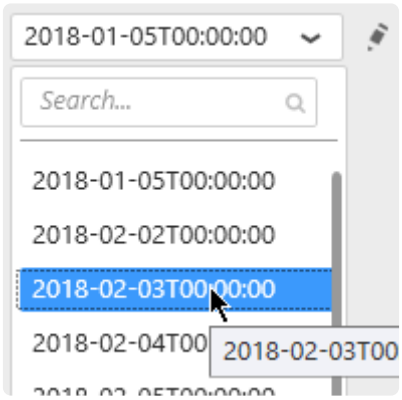
```
Year(Date) BETWEEN 2015 AND 2016
```

Values

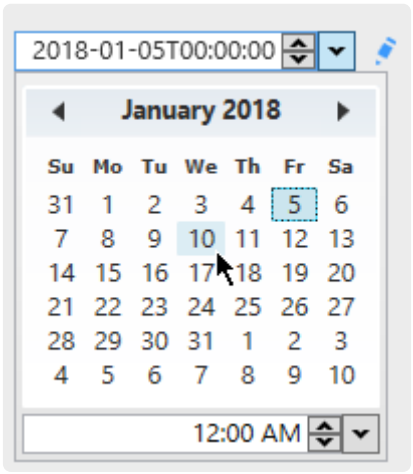
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text :



Depending on the operator and formatting function selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend on actions inside the Dashboard.

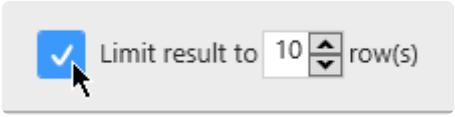
For this, select “Use Parameter” and see the Chapter about Parameters.

Last modified: May 03, 2019

2.6.7.4. Limiting dataset size

How to limit

Check “Limit result to” and select the number of rows you want to keep in the dataset :



Why limit ?

Checking this option with “Limit result to 10 rows” for example will apply a “TOP 10” to the query. When widgets are sorting the data, this can be a useful to display the 10 Items with the worst performance, or the 10 equipments with the highest consumption for example.

Last modified: Mar 15, 2019

2.6.8. Calculating Expressions

Alpana can add any number of calculated fields to a Data Source.
These are called *Expressions*.

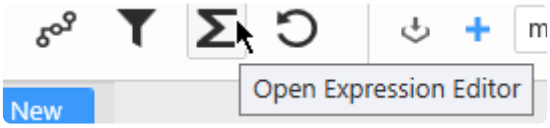
Expressions are stored inside the Data Source and exported along with it.

Expressions are evaluated when the Widget query is made, and will show accurate result depending on the Widget aggregation.
Because of this, using Expressions can be mandatory for certain types of calculations.

Last modified: Apr 17, 2019

2.6.8.1. Opening Expression Editor

Click on the Sigma Σ icon “Open Expression Editor” button :



The Expression Editor dialog opens to let you create and manage Expressions.

Last modified: Apr 17, 2019

2.6.8.2. Managing Expressions

Adding Expressions

Click the + button at the bottom left to add an Expression :

Deleting Expressions

Click the trash icon button at the bottom left to remove the selected Expression :

Hiding Expressions

Click on the eye icon next to an Expression name to hide it :

This can be useful for example when you create an intermediate Expression that will be re-used in other Expressions but not displayed to the end user.

Using Expressions

Expressions can be used in any other place where a data field can be found :

- Widget data binding
- Filter configuration
- …

In the Data Preview of the Data Source, you can also view a preview of the Expression as if it were another field in the data.

Please note that this is only a Preview. Since aggregation Expressions are calculated differently depending on aggregation, they will show the expected result in Widgets, which might be different.

Last modified: Apr 17, 2019

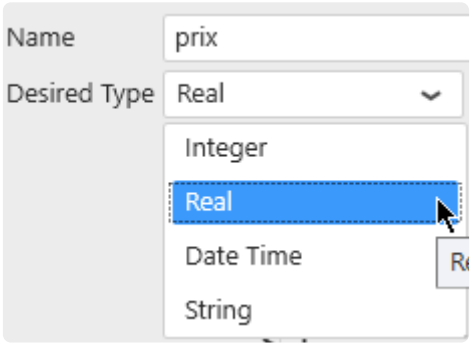
2.6.8.3. Expression properties

Expression Name

You can chose a name for the Expression.

Expression Type

Select a desired type for the Expression :



This will determine how the Expression can be used. For example, what kind of aggregation can be made when it is bound to a Widget.

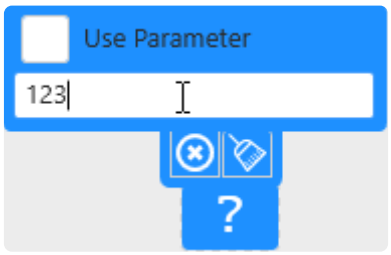
Last modified: Apr 17, 2019

2.6.8.4. Adding a constant

Constant value

By clicking on a **?** question mark, you can input free form text :

- number constant
- string constant



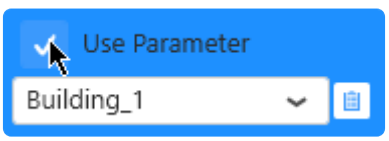
Note on dates

> To input a constant date, it is strongly advised to use the ISO date format :
> *YYYY-MM-DD* for dates, for example 2019-12-25
> *YYYY-MM-DDThh:mm:ss.sss* for datetimes, for example 2019-12-25T23:59:59.999

When inputting dates, you are actually writing a string that will be converted to date on the target database where the query is made.
Note : in case of a Bufferized connection, it is the Buffer database.
This means that your date string may be interpreted wrong depending on the language of the target database.
A date format that is not ambiguous is advised, for example ISO date format.

Dashboard Parameter

To use an Alpana Parameter instead of a constant value, check *Use Parameter* and select a Parameter (it is also possible to create a new Parameter from here by clicking the list icon) :



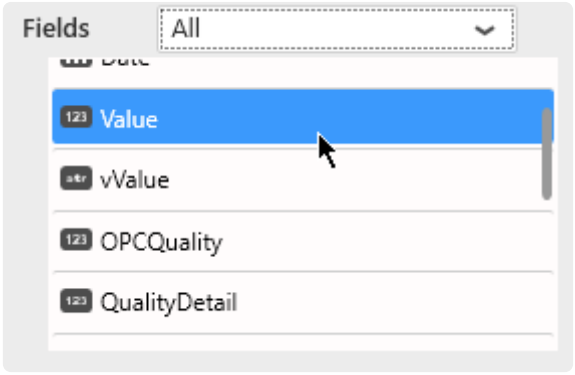
Last modified: Jul 02, 2019

2.6.8.5. Adding a field

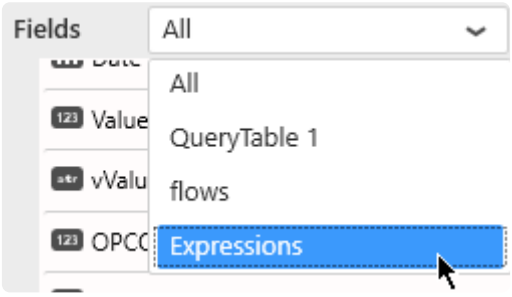
Any field from the Data Source can be used as function argument, including other Expressions

Browsing for a field

Fields can be browsed from the *Fields* container at the bottom :

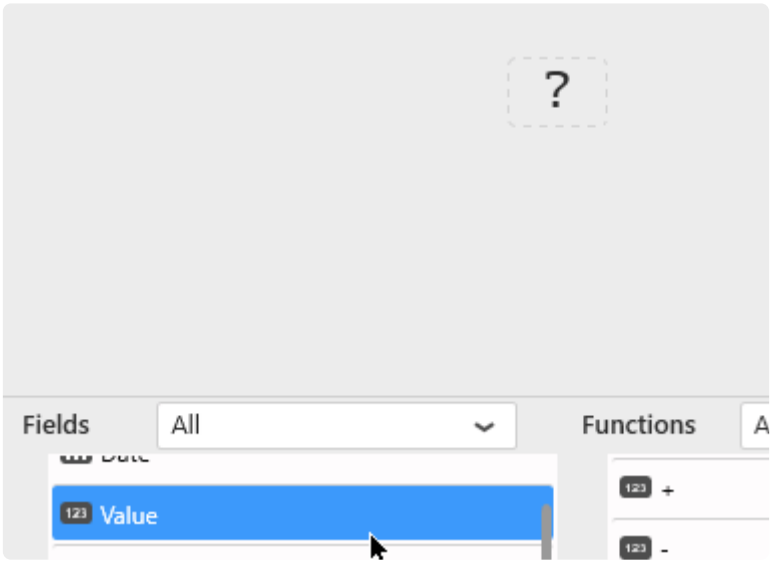


This list can be filtered by table name, or by selecting Expressions to list expressions only :



Adding a field to the Expression

To add a field to the Expression, simply drag it from the list into the Expression :

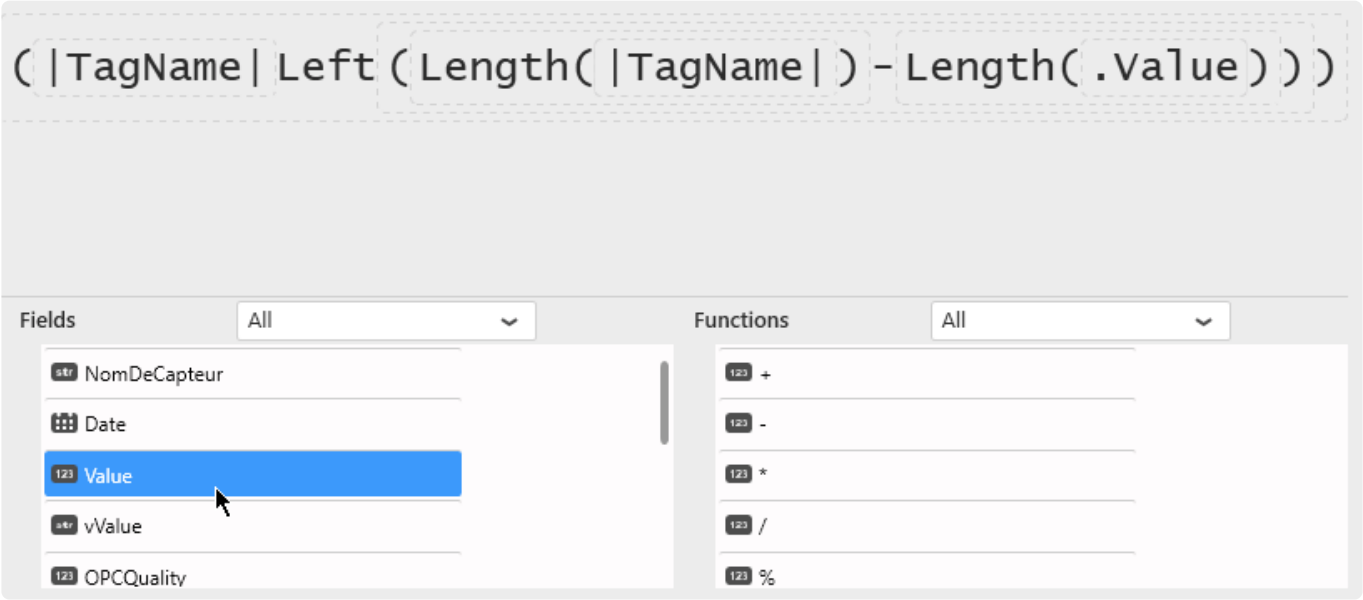


When adding a field on the **?** question mark, the field will take this place.

Replacing an existing field

Using drag and drop

To replace part of an Expression by a field, it's possible to drag the field from the list and drop it onto the expression part to be replaced :



Browsing on the field

To replace a field in an Expression, it's possible to click on the field to be replaced, and select a replacement field :

Last modified: Apr 17, 2019

2.6.8.6. Adding a function

Functions take part in the calculation and take other elements as argument.
There are two main types of functions :

- Calculation functions (like Arithmetic functions *, /, -, +, Abs, … or String functions Concat, Left, …)
- Aggregation functions (SUM, Count, AVG, …)

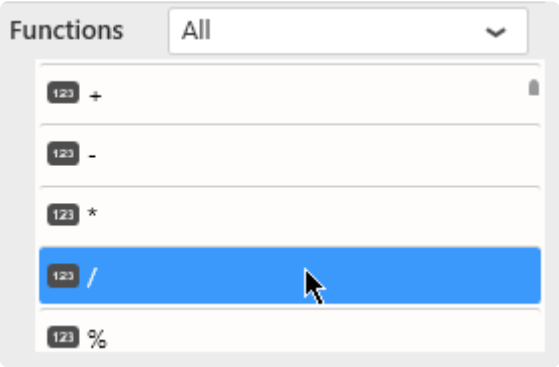
Most functions require arguments. A function argument itself can be anything :

- Function
- Field
- Value

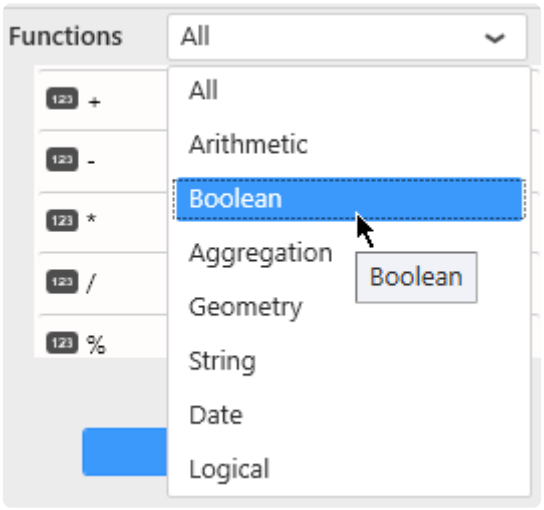
Any missing item will be displayed as a **?** question mark.

Browsing for a function

Functions can be browsed from the *Functions* container at the bottom :

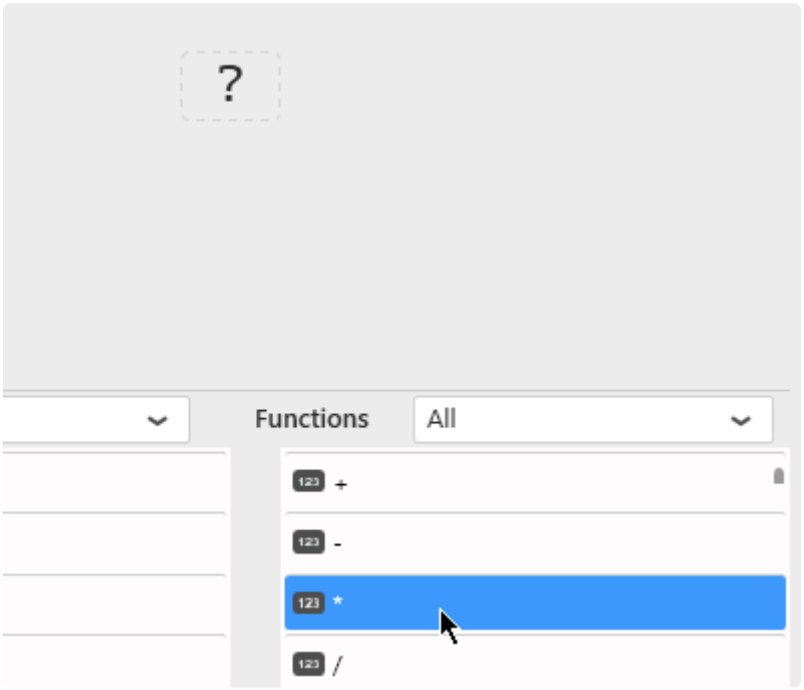


This list can be filtered by function type, or by selecting All functions :



Adding a function to the Expression

To add a function to the Expression, simply drag it from the list into the Expression :



When adding a function on the **?** question mark, the function will take this place.

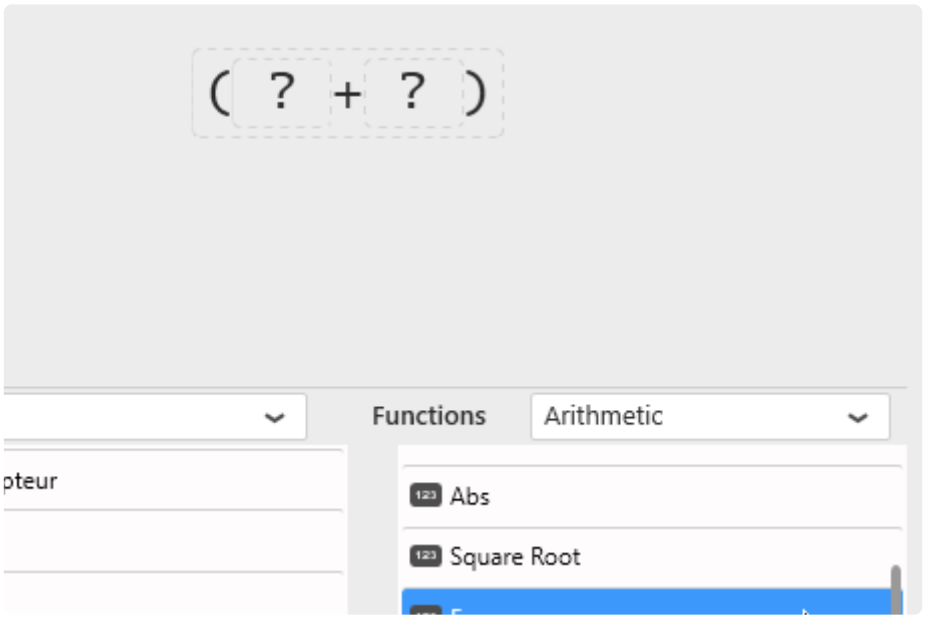
Replacing an existing function

Browsing on the function

To replace a function in an Expression, it's possible to click on the function to be replaced, and select a replacement function :

Wrapping an Expression into a function

Maybe you have started creating an Expression, but now you want to make it an argument to a new function.
To wrap part of an Expression into a function, it's possible to drag the function from the list and drop it onto the Expression part to be wrapped :



Last modified: Apr 17, 2019

2.6.8.7. Removing an existing item

To remove an item and all the dependent items, click the item and click on the “Reset” button :

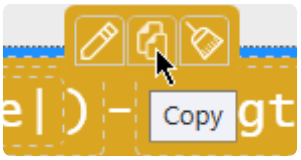
The item itself and all items underneath will be removed.

Last modified: Apr 17, 2019

2.6.8.8. Copying and Pasting an existing item

Copy

Click to select any item (function, field, value), then click on the *Copy* icon :



Paste

Click on the target item, then click on the *Paste* icon :

You can paste onto any item :

- to replace a **?** question mark as a function argument
- replace a single field
- replace a function and all its dependent items
- into a different Expression

Last modified: Apr 17, 2019

2.6.8.9. Aggregated Expressions

What are Aggregated Expressions

There is a special type of Expressions : *aggregated Expressions*.
An aggregated Expression is simply an Expression that uses a function from the "*Aggregation*" category of functions.

> ! These types of Expressions are **essential** to solve some common issues.
> For example, see Pitfall : Calculating ratios, averages, ….

Note : Aggregation Functions

Since Alpana v3.0, supported Aggregation Functions are :

- Avg
- Count
- Distinct Count
- Max
- Min
- Sum
- STDev
- STDevP
- Var
- VarP
  They are listed in the "*Aggregation*" category of functions in the Expression Editor.

Rules

Note : these rules come from the underlying SQL engine. You will find the same rules in SQL queries.

- Non-aggregated fields cannot be used in the same Expression as aggregated fields. See this chapter.
- Aggregation functions cannot wrap other aggregation functions. See this chapter.

In particular, when binding an aggregated Expression as Measure to a Widget, only the "aggregation" type "*Value of*" can be used.

Usage

Aggregated Expressions will always be calculated correctly, independently of how data is grouped inside the Widget.

Example

Imagine a simple dataset :

TagName	Building	Product	Price
FT101.Value	Building A	Product X	10
FT102.Value	Building A	Product Y	15
FT201.Value	Building B	Product X	20
FT202.Value	Building B	Product Y	25
FT301.Value	Building C	Product X	30
FT302.Value	Building C	Product Y	35

You create the following Expression :
`ExpressionTotal = SUM(Price)`

If you display this full dataset in Grid, you get :

TagName	Building	Product	Price	ExpressionTotal
FT101.Value	Building A	Product X	10	**10**
FT102.Value	Building A	Product Y	15	**15**
FT201.Value	Building B	Product X	20	**20**
FT202.Value	Building B	Product Y	25	**25**
FT301.Value	Building C	Product X	30	**30**
FT302.Value	Building C	Product Y	35	**35**

⇒ It seems like the `SUM()` part of the Expression didn't do anything.

Now if you display this same Expression in a Pie Chart grouped by "Product", you get :

Product	ExpressionTotal
Product X	**60**
Product Y	**75**

⇒ because **60** is the SUM 10 + 20 + 30 and **75** is the SUM 15 + 25 + 35

Now if you display this same Expression in a Card Widget, grouped across all the dataset, you get :

ExpressionTotal
135

⇒ because **135** is the total SUM 10+15+20+25+30+35

Last modified: Jun 17, 2019

2.6.9. Expressions Troubleshooting

Expressions are defined as objects in Alpana v2.0 and translated to SQL expressions when the widget queries are composed.

Some errors may happen from operations that are undefined in SQL, or when the calculation is not not well understood.

This Chapter will help you in these cases.

Last modified: Mar 08, 2019

2.6.9.1. Pitfall : Incorrect result in Expression with division

Your issue

In some cases, Expressions with a division will give an unexpected result : 0 everywhere, or imprecise value for example.

Example Expression :
`Expression1 = 100 * (SUM([isActive]) / COUNT([ID]))`
In this case I want the % of Active items.

[isActive] is 0 or 1 and ID is some ID for the items.

Why am I getting weird results for this Expression1 ?

(0 everywhere, or imprecise value for example)

Explanation

Expressions are executed as part of the final SQL query.

For buffer-based Connections (Excel, CSV, Historian, Web, …), the final queries are also **database queries**.

SQL division behaves differently if the two sides are Integer.

Test the following SQL query :

```
SELECT 123/456;
```

The result is 0

Why ?

We are performing an Integer Division (see https://en.wikipedia.org/wiki/Euclidean_division)

The quotient is 0 and the remainder is 123.

More info

To explore further, you can test the following SQL query (paste in SQL Management Studio for example) :

```
SELECT
        29/10 AS quotient,  -- 2
        29%10 AS remainder,  -- 9
        (29/10) * 10 + 29%10 AS result,  -- 29 = 2*10 + 9
        1.0 * (29/10) AS wrong,  -- 2.0
        (1.0 * 29)/10 AS real_division;  -- 2.9 because 1.0 * 29 = 29.0
```

In my example Expression1 above, SUM, and the result of COUNT is also an Integer.

So I am not going to get the % of Active items I expect

Solution

To fix it, I need to **force the conversion to a Real number** for at least one side of the division.

Example solution :

> Multiply one member by 1.0 to force the conversion to a Real number.

```
Expression2 = 100 * (SUM([isActive] * 1.0) /COUNT([ID]))
```

Last modified: Mar 08, 2019

2.6.9.2. Pitfall : Calculating ratios, averages, ...

Introduction

An Expression may seem to perform the correct calculations, but it is important to double-check results during the development process, and question the calculations themselves.

One crucial point to remember is that fields must be aggregated in order to be displayed inside a widget. This performs a second layer of calculation and needs to be thought out carefully.

The easy case : SUM

Sum is easy. Sum of sums gives the expected total.

If you configure an Expression like :

```
ExpressionSUM = [DurationLoading] + [DurationPacking]
```

You can easily use it in a widget with default aggregation SUM, and you should get the total durations as expected result.

The only things to look out for is :

- unit consistence (adding minutes with minutes)
- NULL handling (what happens when [DurationPacking] is NULL)

The general case

Consider the following dataset :

Day	GoodProduction	TotalProduction	ratio
1	8	80	0.1
2	2	2	1
3	3	15	0.2
4	3	3	1
(total)	**16**	**100**	**should be 16/100 = 0.16**

Assignment :
I need to make an Expression for the ratio GoodProduction / TotalProduction.

ratio : First Attempt (*wrong*)

I create :

```
Expression1 = [GoodProduction] / [TotalProduction]
```

It **seems** to give good results in the Data Preview grid in Alpana Designer. But **in the Data Preview, rows are not aggregated !**

When I add the expression to a widget, what aggregation do I choose ?

`SUM([Expression])` will give 2.3 which is **incorrect** ! The reason is that you calculate the sum of ratios (0.1 + 1 + 0.2 + 1) which has no meaning.

`AVG([Expression])` will give 0.575 which is **incorrect** too ! The reason is that you calculate the un-weighted average of ratios (0.1 + 1 + 0.2 + 1)/4 which has no meaning also.

ratio : Solution

The Expression must express what you really want to calculate, including aggregation :

```
Expression2 = SUM([GoodProduction]) / SUM([TotalProduction])
```

This will be used as a calculated column in all queries, including widgets and will always make the correct calculation.

For example, in the Total of the data above, you will correctly get : (8+2+3+3) / (80+2+15+3) = 0.16

There is only one fix needed :

Since we are making a ratio of integers, we should force to cast as float to make float division :

```
Expression3 = (SUM([GoodProduction]) * 1.0) / SUM([TotalProduction])
```

General case : Averages and other

In many cases, this phenomenon can be hidden.

If your database field is called [AverageDuration], it is probably wrong to make a SUM or AVG of it, for the same reasons as above.

In some cases, you will need to *expose to the dashboard Data Source the fields necessary to re-aggregate the calculation properly*.

Example on adapting your data

Your Data Source points to an SQL View with a calculated column “OEE”.

You made the calculation before reaching the dashboard, and you defined it as : `[OEE] = SUM([Uptime]) / (SUM([Uptime]) + SUM([Downtime]))`

Now in your Dashboard, how will you aggregate this OEE ? You will run into the issues described above as you will implicitly make wrong calculations.

> The solution here can be to expose the fields [Uptime] and [Downtime] inside your SQL View. Like that, in your Data Source, you will be able to recreate this calculation as an Expression, and get correct results in all cases.

see also

Aggregated Expressions

Last modified: Apr 29, 2019

2.6.9.3. Error when mixing aggregate fields and non-aggregate fields

The Expression engine (SQL) doesn't allow to mix aggregate fields with non-aggregate fields, as it results in undefined behaviour.

Example :

```
Expression = SUM(ValueField_A) + ValueField_B
```

ValueField_A is inside an aggregation expression (SUM) and ValueField_B is not.

This results in an error :

"*The following expression(s) contain(s) non aggregate field references and aggregate functions at the same time which is not allowed*"

To solve this, think about the calculation you really want to do, and what will happen when it will be included in a GROUP BY operation.

see also

Aggregated Expressions

Last modified: Apr 29, 2019

2.6.9.4. Error when wrapping aggregate fields inside aggregate functions

The Expression engine (SQL) doesn't allow to apply an aggregation function on top of another aggregation function.

Example :

```
Expression = AVG(SUM(ValueField_A))
```

This results in an error :
"*Cannot perform an aggregate function on an expression containing an aggregate or a subquery*."

To solve this, you should leave only 1 level of aggregation functions (think about what you really want to calculate and test on a small dataset).

see also

Aggregated Expressions

Last modified: Apr 29, 2019

2.6.10. Value-based Settings

In addition to data preparation itself, Data Sources also contain *information that is bound to the data*.

These “*value-based settings*” have a different value based on the value inside a Data Source field.

Since v2.0, the only value-based setting is Value-based Colors.

> Note :
> This is only available for text type fields.

Last modified: Jun 17, 2019

2.6.10.1. Value-based Colors

Goal

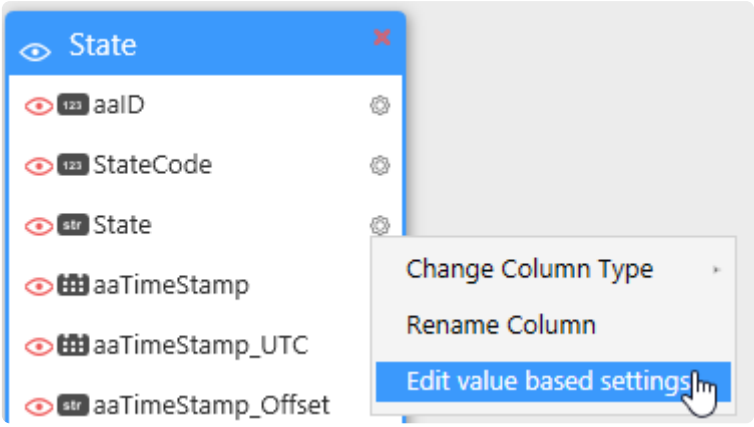
The “Value-based color” mechanism allows to bind a central unique value for a color based on a Data Source field value.

For example, you may want to associate colors to values in the “Machine State” field :

- value “Downtime” is always **red**
- value “Running” is always **green**

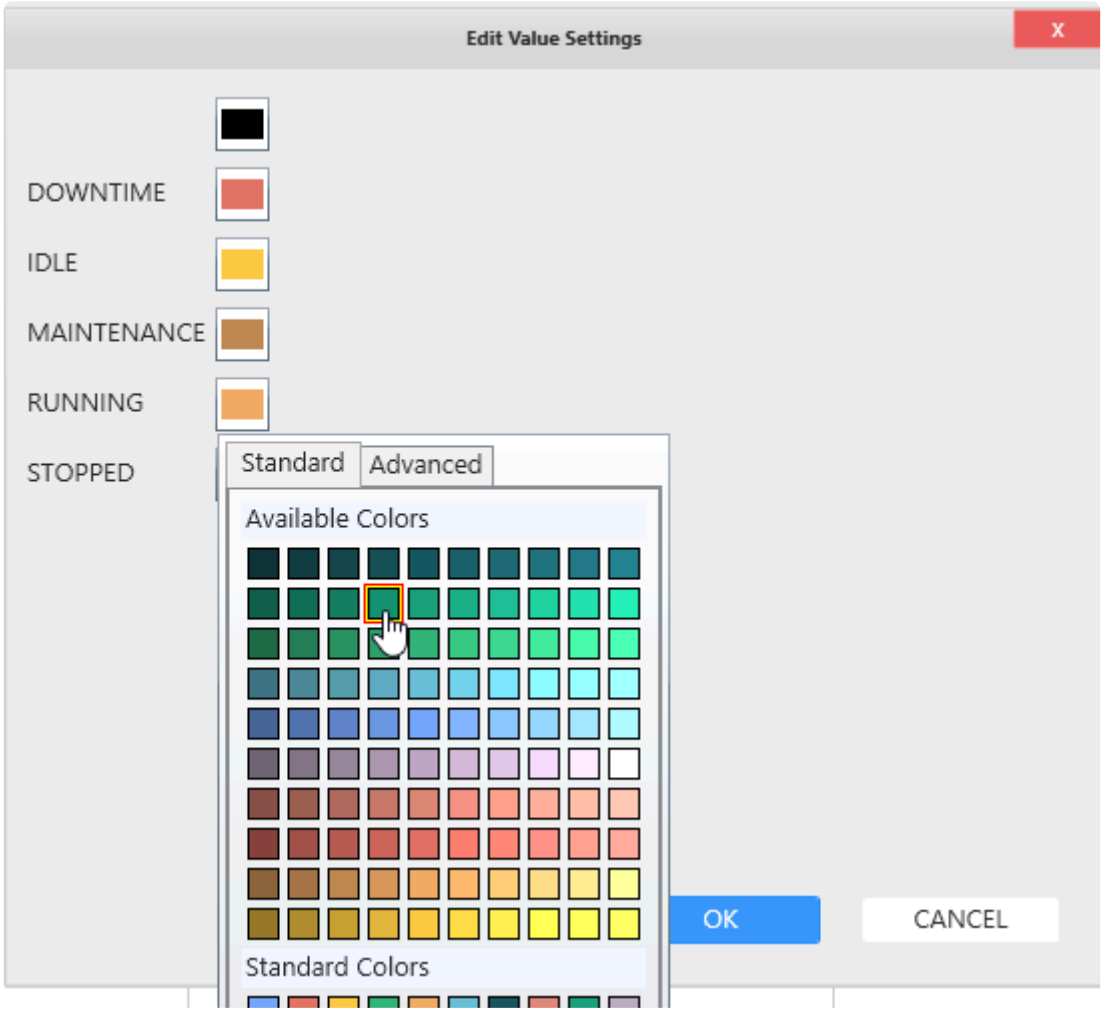
Edit value-based colors on a data field

To edit value-based colors on a data field, find the relevant field and click on the cog icon directly to its right, then select “Edit value based settings” :



Each value currently in the field is listed.

To assign a color, just click on the current color :



Change to your desired color and click OK to validate.

Edit value-based colors on an Expression

To edit value-based colors on an Expression, find the relevant field and click on the cog icon directly to its right, then select “Edit value based settings” :

Then proceed as above to assign colors.

Last modified: Mar 14, 2019

2.6.11. Formatting Columns

Last modified: Feb 21, 2019

2.6.11.1. Renaming Columns

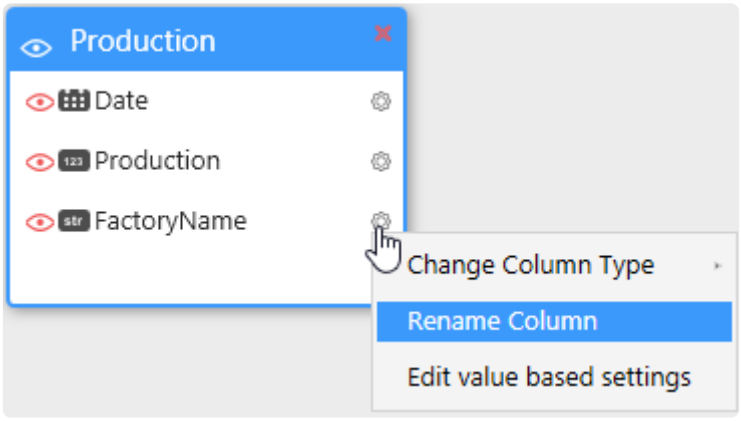
Why rename columns ?

The column names will appear in widget data binding.

Giving good names can help the user who will make the Dashboard based on this Data Source.

How

To rename a column, click on the cog icon and select “Rename Column” to enable the edit mode :

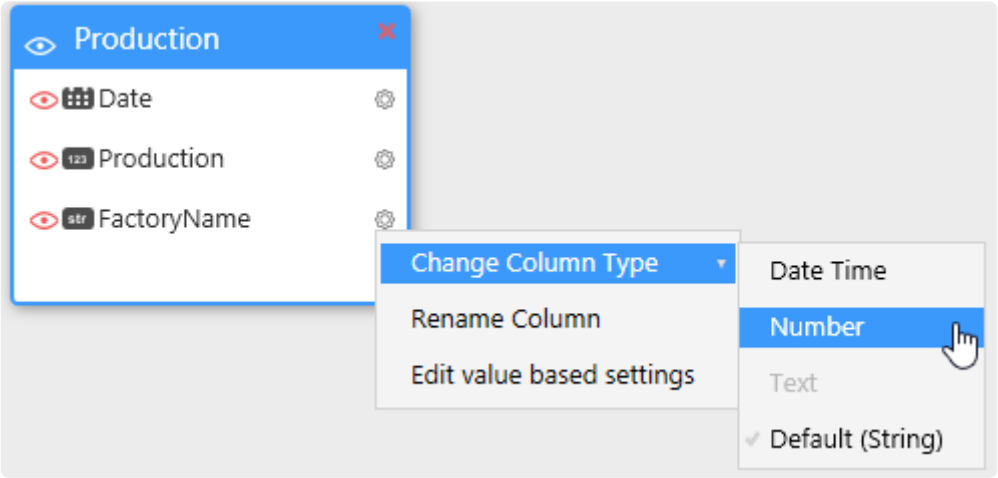


Press Enter to validate the new name.

Last modified: Mar 14, 2019

2.6.11.2. Changing Column Type

To change a column data type, click on the cog icon and select “Change Column Type”, then choose the new type of the column :



On Connections Types that don’t explicit types (like CSV), the default selected type may be incorrect. This allows you to select a better data type.

Last modified: Mar 21, 2019

2.6.11.3. Hiding Column

Why hide columns ?

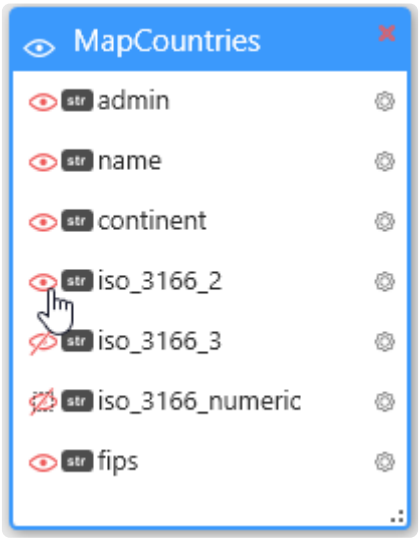
The visible columns will appear in widget data binding.

hiding un-necessary or confusing columns can help the user who will make the Dashboard based on this Data Source.

> NOTE
> If you hide a column, it won't be accessible for Widgets data binding.
> However, it will still be accessible for making Joins, Expressions and Data Source Filters.

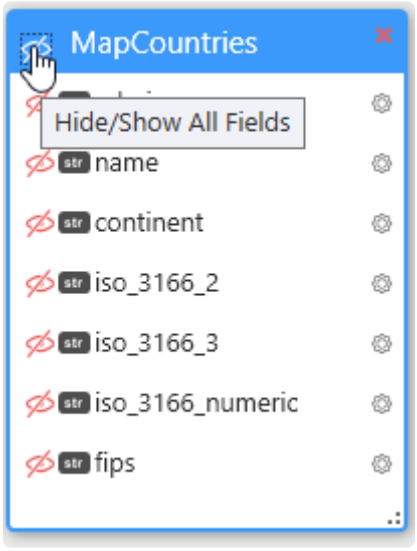
How

To Hide or Unhide a column, click on the red eye, just to the left of the desired column :



Hide all

To Hide or Show all Columns of a table, click on the eye icon just to the left of the table name :



Last modified: Mar 14, 2019

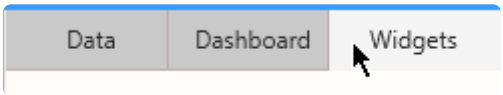
2.7. Configuring a Widget

The main Widget configuration activities in Alpana Designer include :

- binding data : what to display
- graphical properties : how to display it
- interactions : how visualizations interact with each other

> To create Widgets, you will need to connect and prepare your data first. Please see the relevant Courses : Connecting to Data and Transforming Data.

Since Alpana Designer v3.0, all Widget configuration activities are made inside the *Widgets* activity tab.

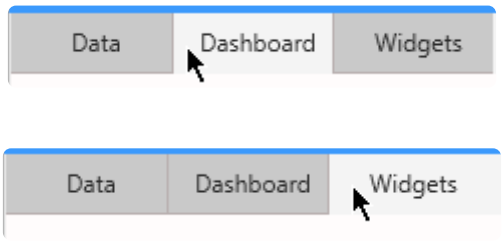


Last modified: Jun 17, 2019

2.7.1. Creating and Managing Widgets

This chapter describes how to create and manage Widgets.

Since Alpana Designer v3.0, Widgets are spawned in the *Dashboard* activity tab, and then configured inside the *Widgets* activity tab :

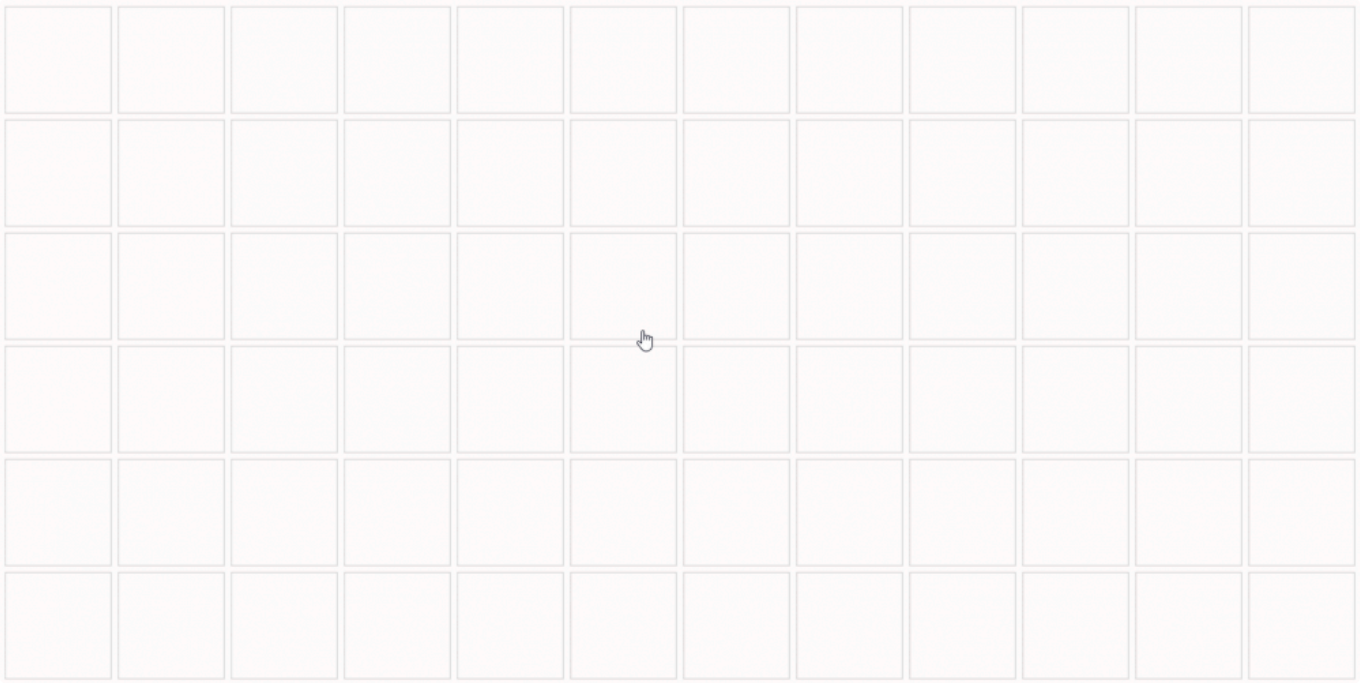


Last modified: Jun 17, 2019

2.7.1.1. Adding a Widget

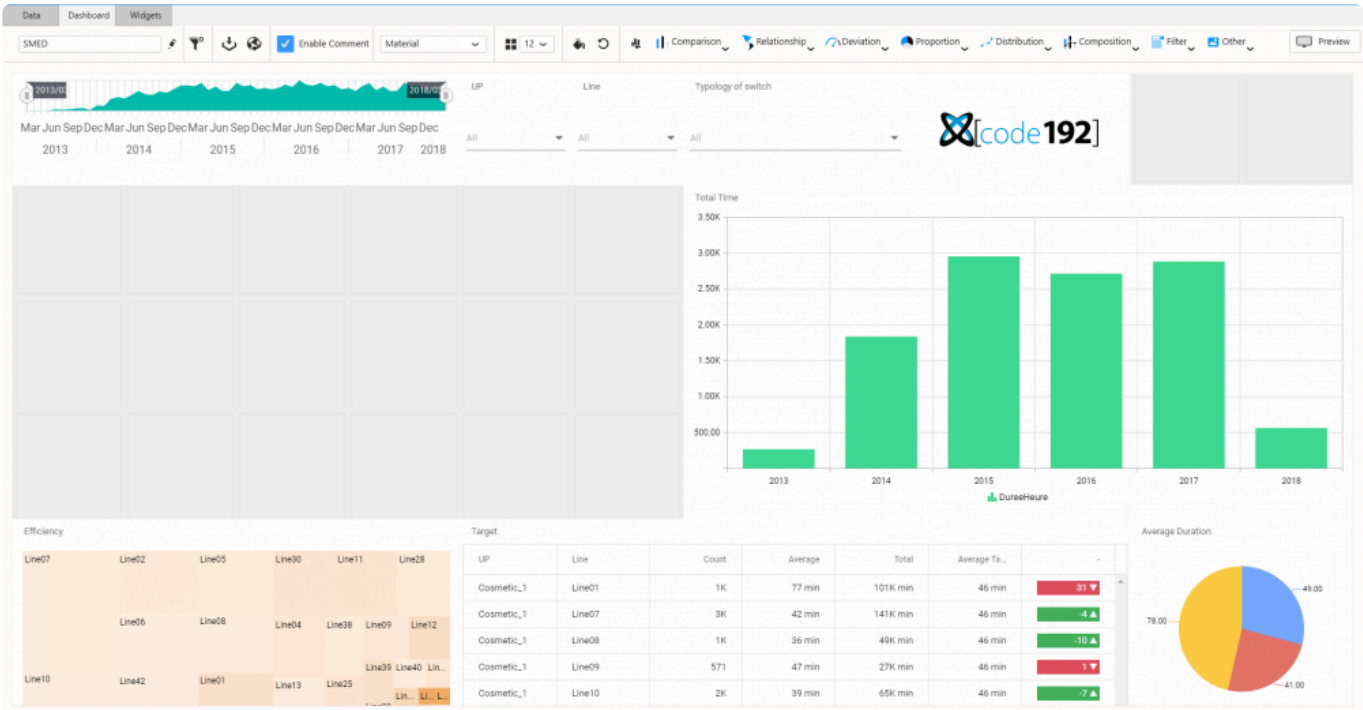
Dashboard Layout

In the *Dashboard* activity tab, the main area shows the Dashboard layout canvas where some square tiles are displayed to guide you.



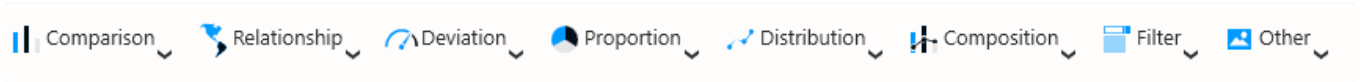
Created Widgets will appear on this layout canvas and can be re-arranged later.

For example, this is what a work in progress Dashboard will look like in this view :

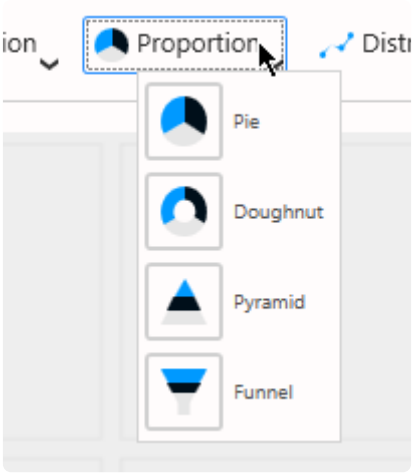


Browsing for available Widget types

In the *Dashboard* activity tab, the top bar displays drawers that contain the Widget library :

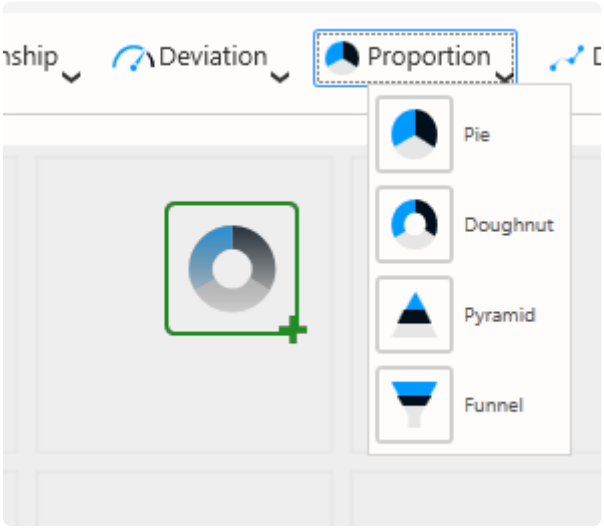


Click on a drawer to open it and list all widget types in this category :

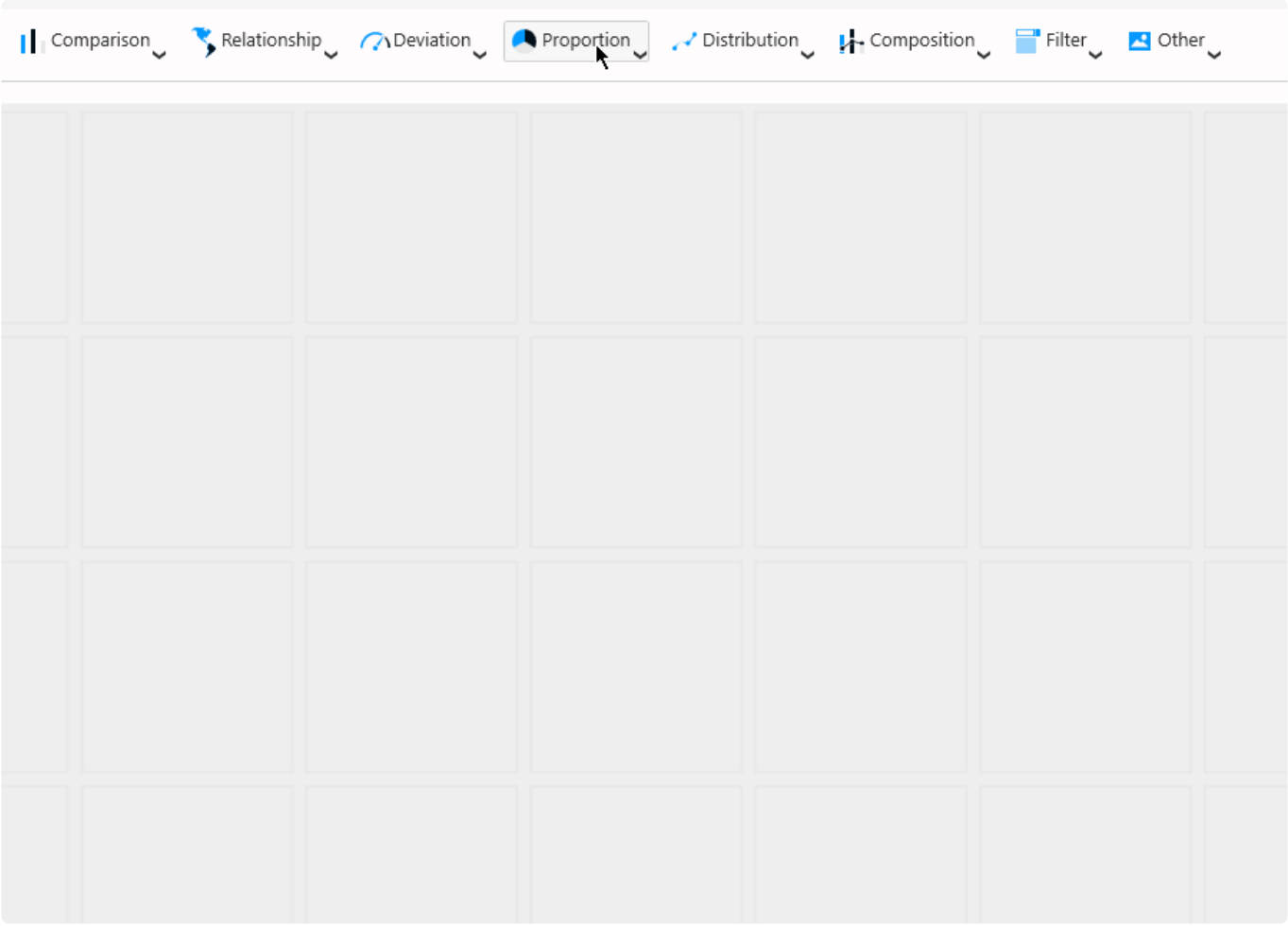


Adding a Widget

To add a Widget to the Dashboard, simply drag it from the drawer into any place in the layout canvas.



See the below animation for the full workflow :



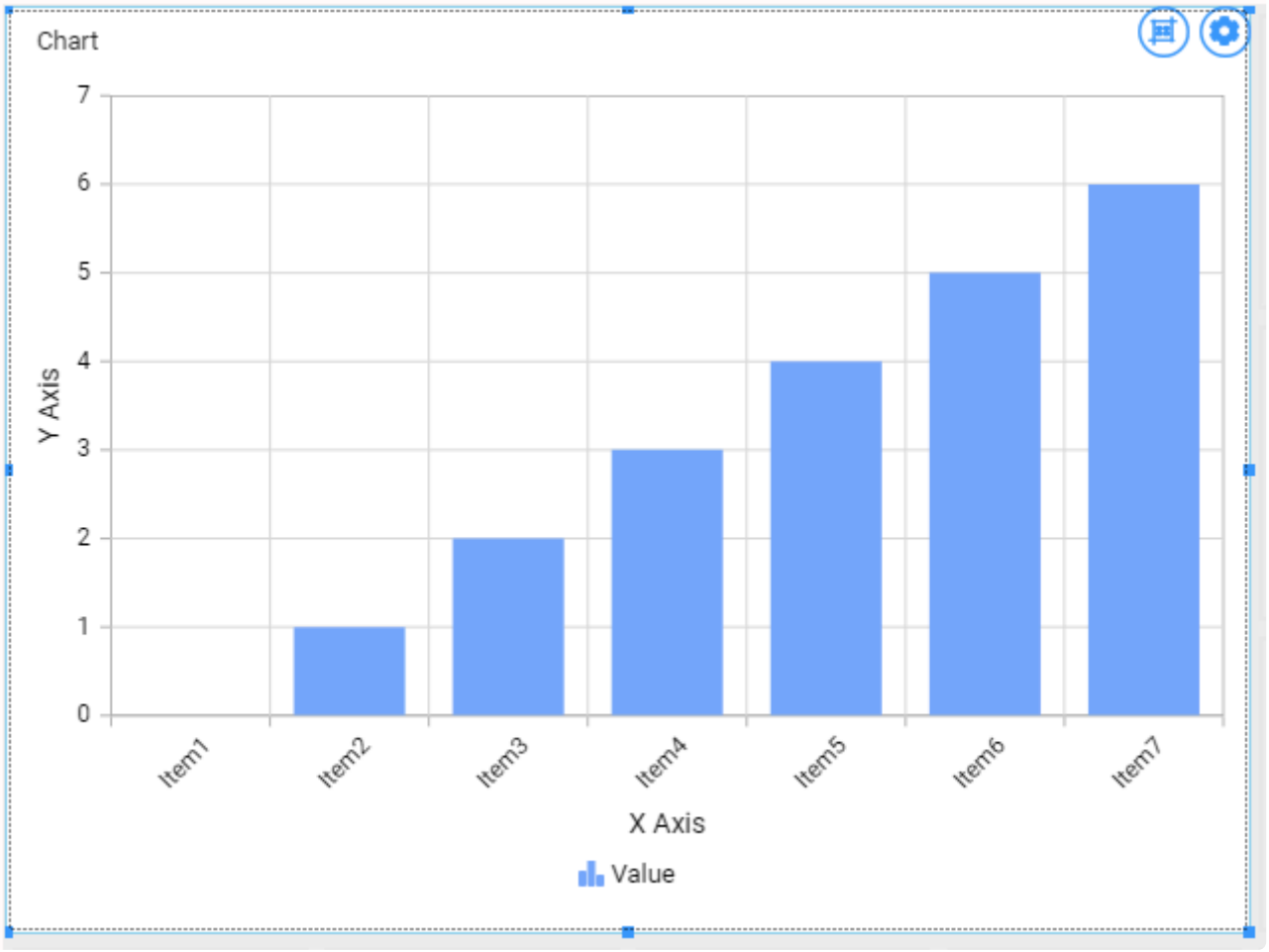
Don’t worry, you will be able to move, resize, delete, … this Widget at any time later.

Note : Sample values

When a Widget is created, it is not configured.

However, in order to help you choose a Widget Type and have an idea of what the Dashboard will look like, some sample data are displayed, like “Item1, Item2, …”.

This data is a placeholder only and is not related to any real data.



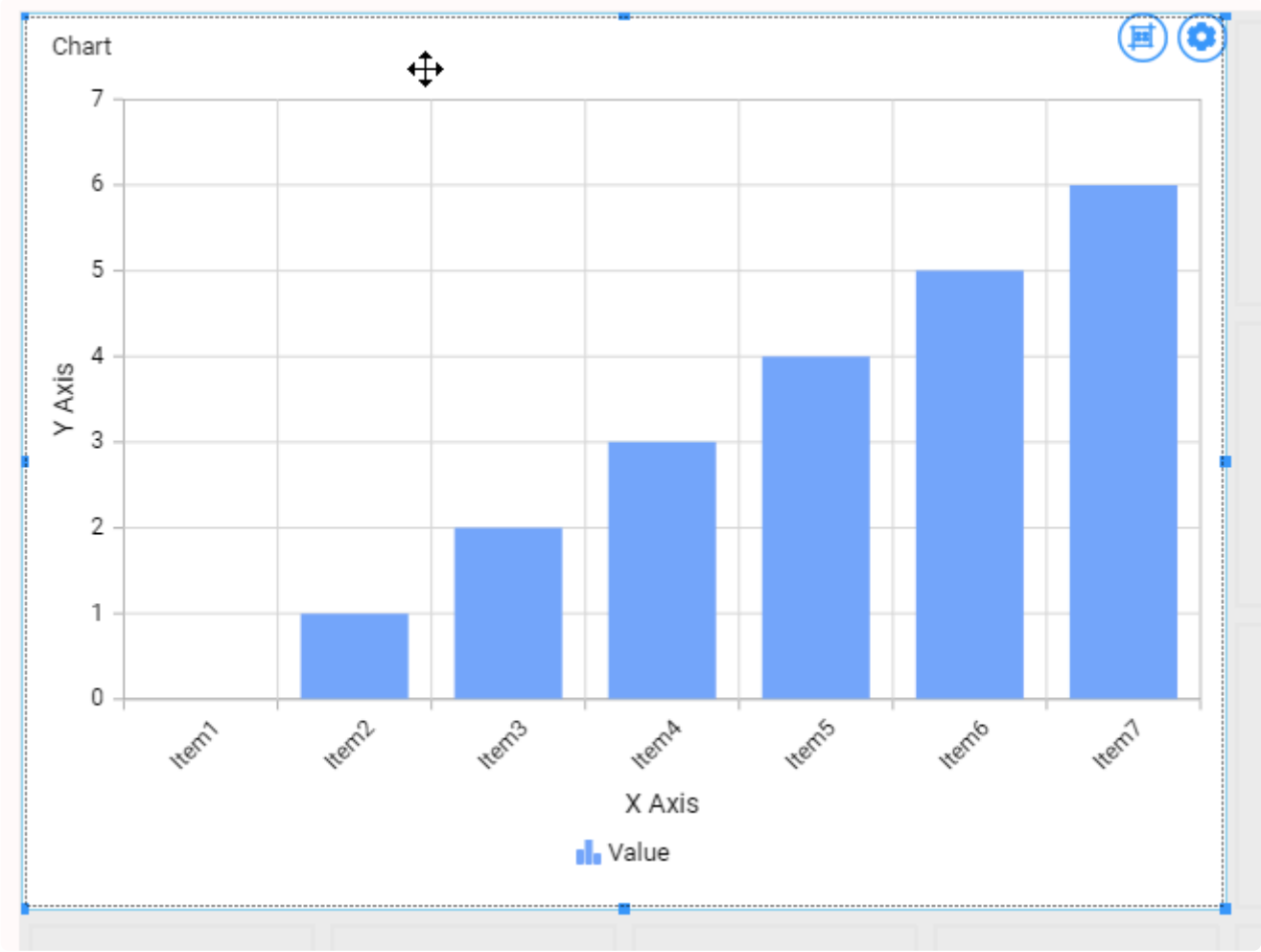
Last modified: Apr 19, 2019

2.7.1.2. Editting a Widget

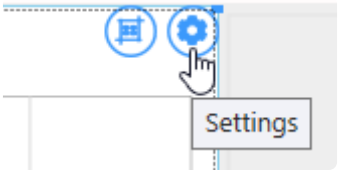
How to Edit a Widget

From the Dashboard activity tab

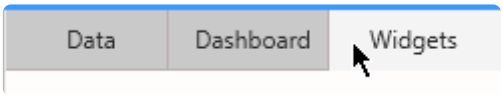
To open an existing Widget to start to Edit it, first left click on the Widget on *Dashboard* activity tab, so that it appears selected :



Then on the cog icon on the top right of the Widget itself :

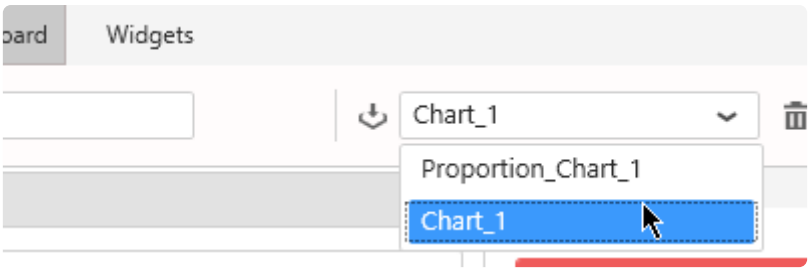


This will take you to the *Widgets* activity tab focused on this Widget



From the Widgets activity tab

When already inside the *Widgets* activity tab, you can select a Widget to edit from the list in the top bar :

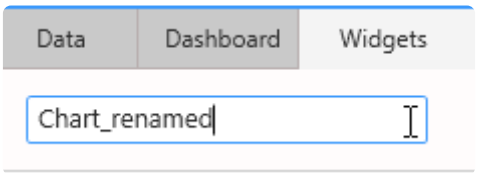


All the actions to actually edit the Widget are explained in the following chapters.

Last modified: Apr 26, 2019

2.7.1.3. Renaming a Widget

When the Widget is already open to Edit, you can use the top bar on the left to change its name :



No need to validate, the name is updated live.

> Note :
> This Widget name is used internally in the widgets list for example.
> However, this is not the Widget header displayed at runtime. To edit the header displayed at runtime, see the corresponding chapter in Header and Description.

Last modified: Jun 17, 2019

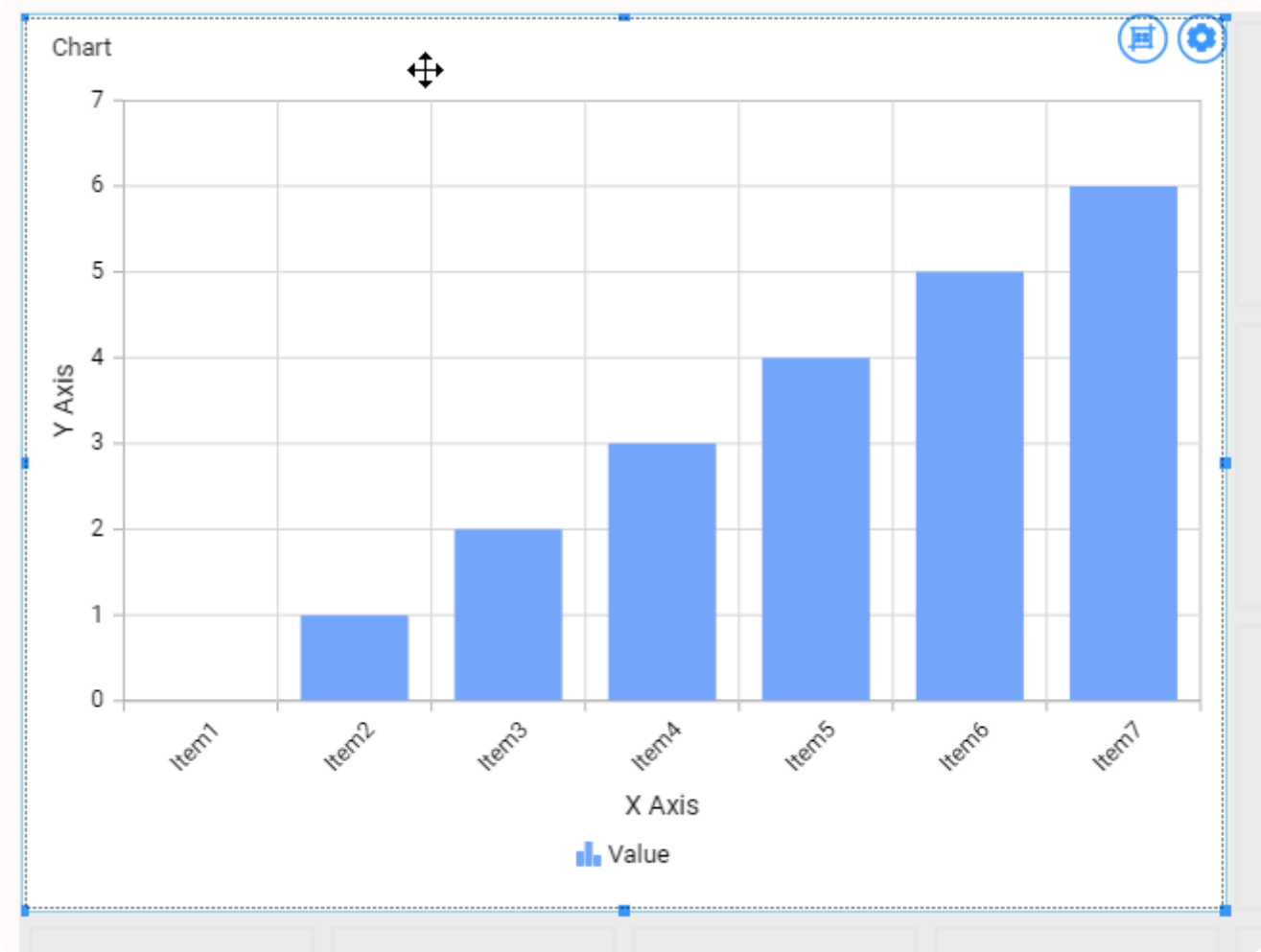
2.7.1.4. Deleting a Widget

From the Widgets activity tab

To delete a Widget, select it in the *Widgets* activity tab top bar, then click on button “Delete” :

From the Dashboard activity tab

To delete a Widget when you are in the main *Dashboard* activity tab, first left click on the Widget, so that it appears selected :



Then press the “Delete” key on your keyboard.

Last modified: Apr 19, 2019

2.7.1.5. Copying/Pasting a Widget

Instead of creating an empty Widget and configure it from scratch, you may duplicate an existing Widget and edit the copy.

Copy

To copy a Widget, select it on the layout canvas and use the `Ctrl+C` keyboard shortcut, or click the *Copy* button in the top window bar :

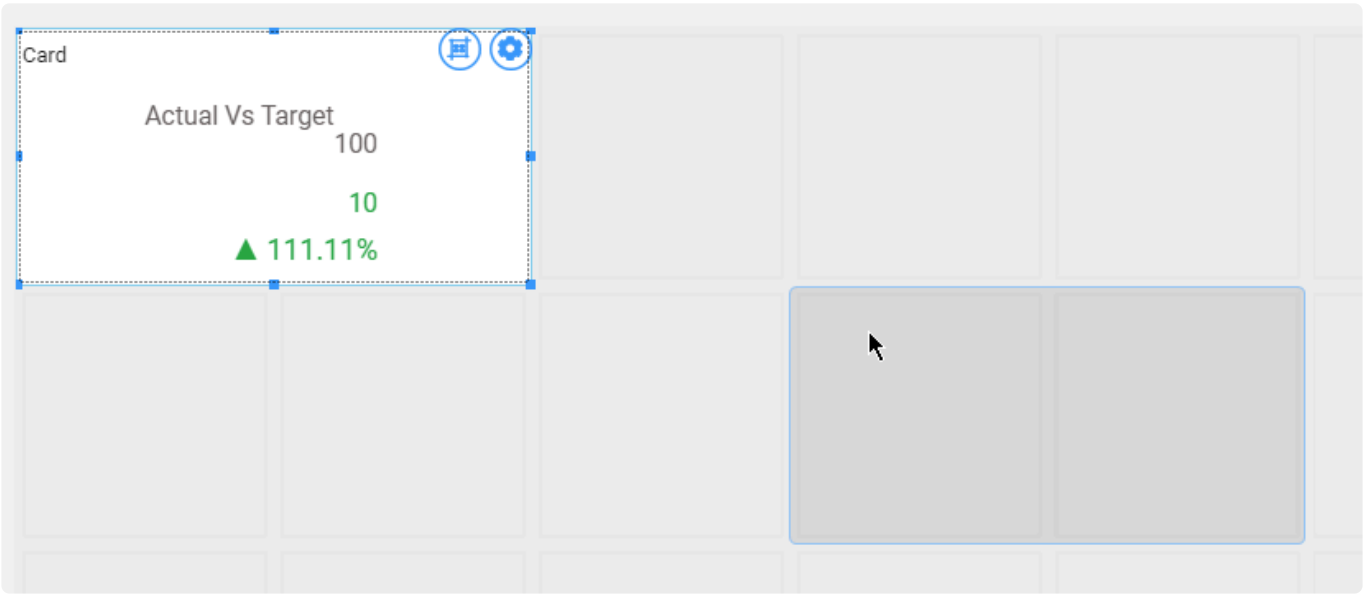
The Alpana Designer clipboard will now hold a copy of this widget, and you will now be able to paste it elsewhere on the layout.

Paste

After copying, you can paste the Widget copy somewhere else on the Dashboard layout.

To paste, use the `Ctrl+V` keyboard shortcut, or click the *Paste* button in the top window bar :

Now your mouse cursor will hold the Widget copy, ready to be placed on the layout.
As you move your mouse, you see the outline of the new Widget that will be created :



Click the desired location to paste the Widget.

Last modified: Nov 04, 2019

2.7.2. Export / Import Widgets

Sharing Widgets allows to re-use your work or collaborate between users.

This chapter describes how to proceed.

Last modified: Apr 26, 2019

2.7.2.1. What is a Widget Export

A Widget can be exported to a file with format `*.alpw`

This Widget.sydw file contains :

- the configuration of the Widget itself
- its position on the Dashboard canvas
- the dependent Data Source
- all the dependent Connections, including their credentials
- all related Parameter binding (for filters)

Data Sources and Connections use an internal unique identifier.

When importing several Widgets that were exported from the same Data Sources or Connections, the common elements are re-used instead of duplicated.

Last modified: Mar 14, 2019

2.7.2.2. Export a Widget to a file

How to Export to a file

In the top bar, click the *Export as Template* icon :

Select a file name and save locally.

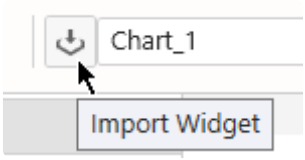
Last modified: Apr 19, 2019

2.7.2.3. Import a Widget from a file

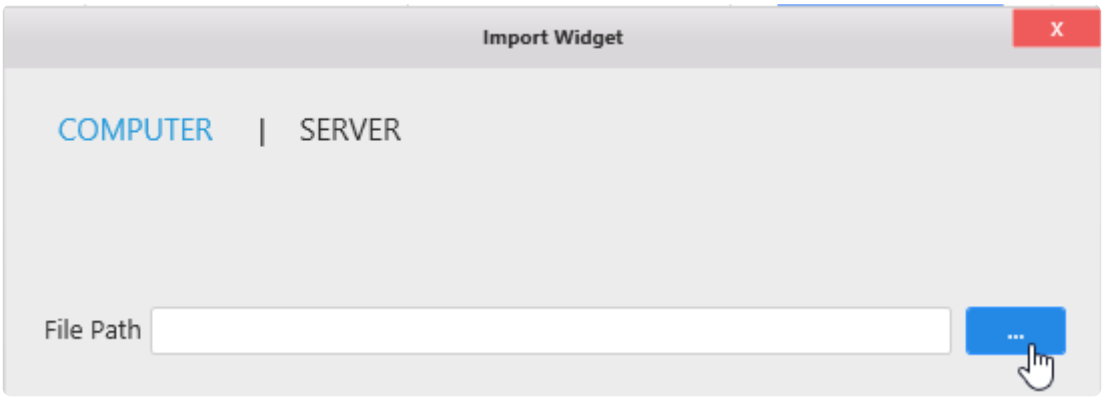
A Widget can be imported from a file in a few ways :

Import in an existing Dashboard

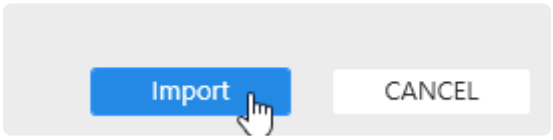
Click on the “Import” button in the Widgets list :



In the “Import Widget” dialog, under “Computer”, click “…” to browse for a Widget.alpw file :



Then click “Import” :



The Widget is now imported in your current Dashboard.

Create a new Dashboard based on a Widget file

In Windows Explorer, you can double click or open the Widget.alpw file with Alpana Designer.

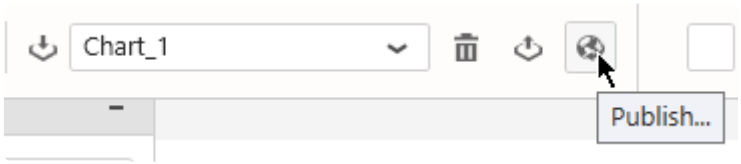
This creates a new Dashboard with the corresponding Widget imported.

Last modified: Apr 19, 2019

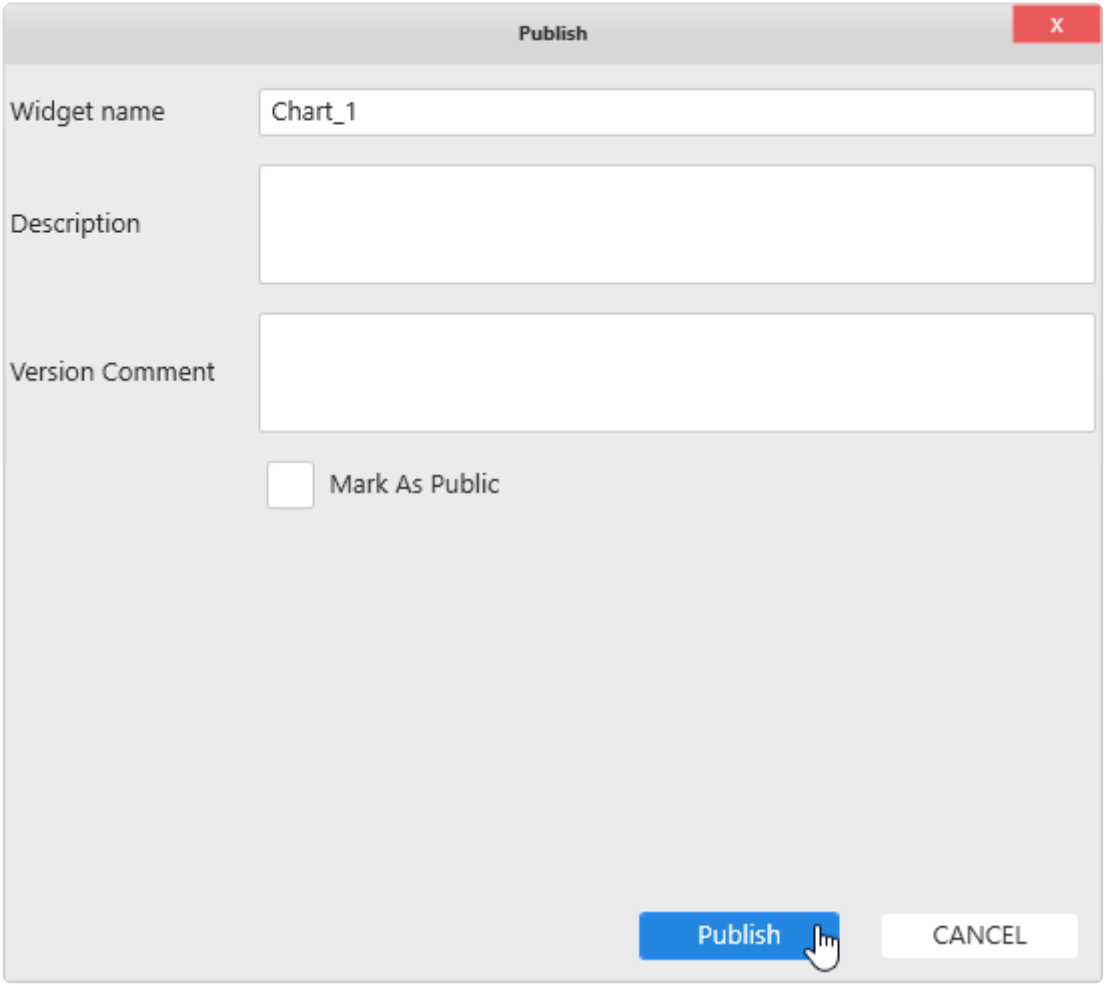
2.7.2.4. Publish a Widget to Alpana Server

In order to Publish a Widget to Alpana Server, you need to be **logged in** from Alpana Designer to that Alpana Server, and have the necessary **permissions** to create Widgets.

In the top bar, click *Publish* :



In the “Publish” dialog, fill in details and click “Publish” :



The Widget is now published to Alpana Server and available to other users depending on permissions.

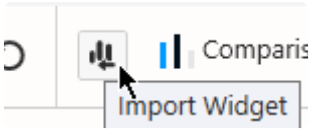
Last modified: Apr 26, 2019

2.7.2.5. Import a Widget from Alpana Server

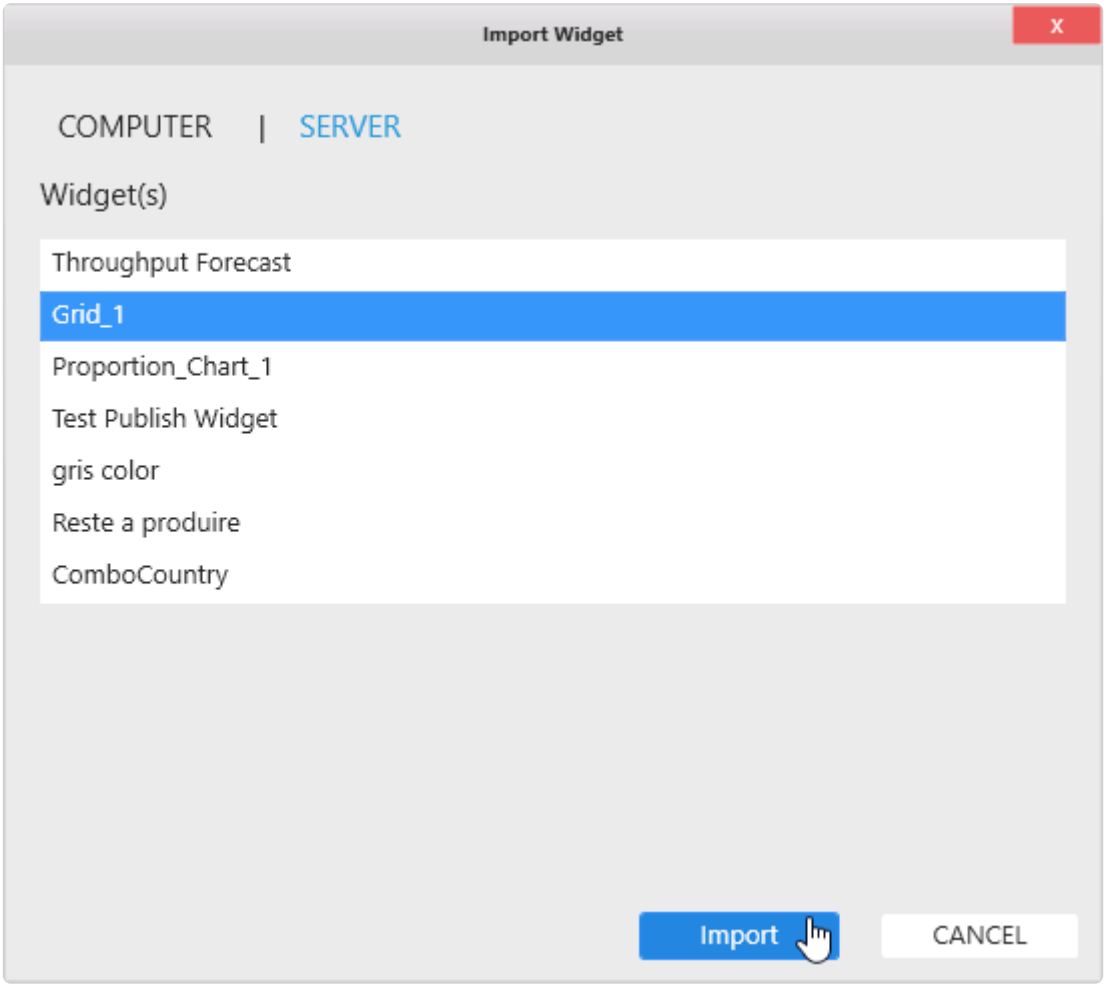
In order to Import a published Widget from Alpana Server, you need to be **logged in** from Alpana Designer to that Alpana Server, and have the necessary **permissions** to access the Widget.

From the Dashboard activity

Click on the “Import Widget” button in the top bar :



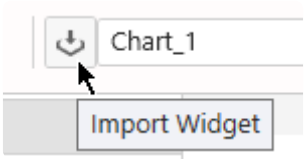
In the “Import Widget” dialog, click “Server”, select a published Widget, and click Import :



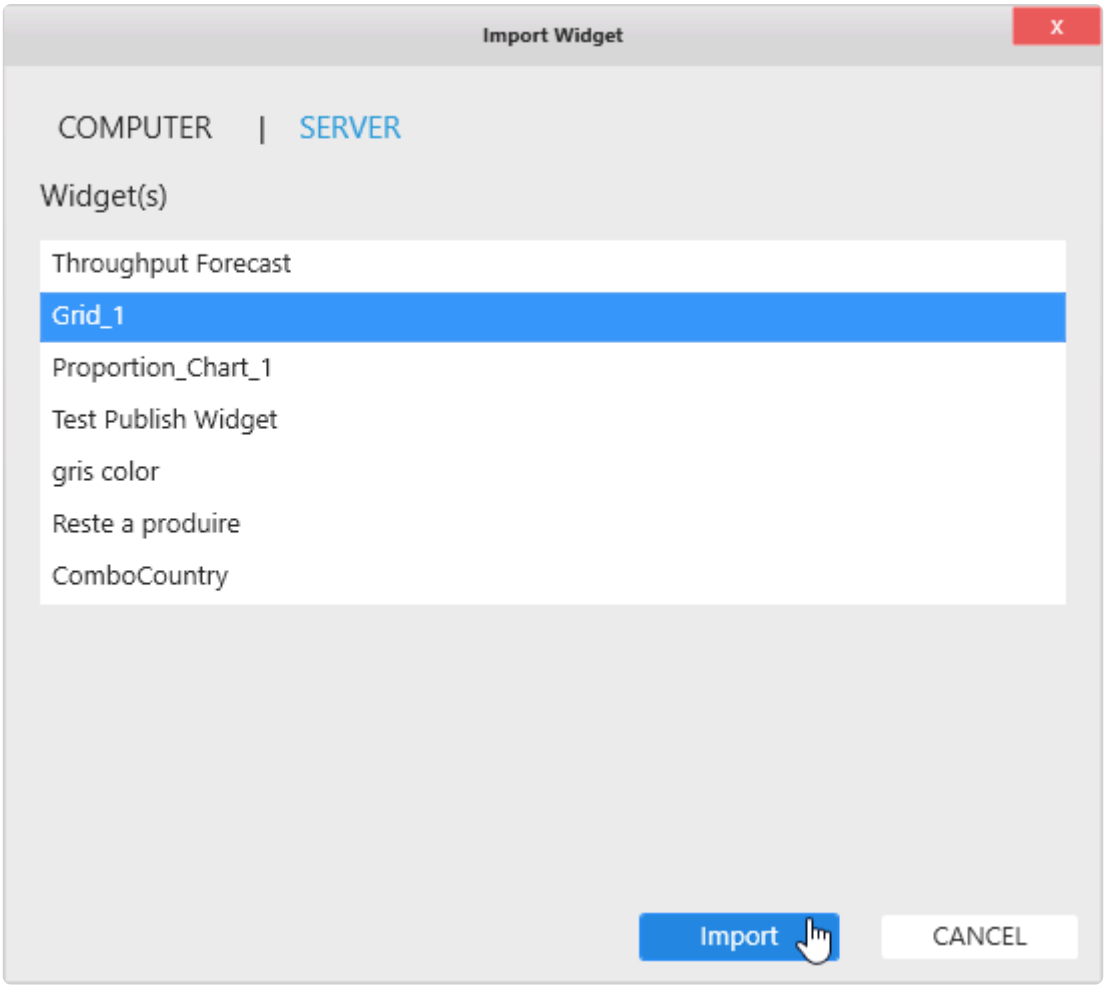
The Widget is now imported in your current Dashboard.

From the Widgets activity

Click on the “Import” button in the top bar :



In the “Import Widget” dialog, click “Server”, select a published Widget, and click Import :



The Widget is now imported in your current Dashboard.

Last modified: Apr 26, 2019

2.7.3. Widget Data Binding

Introduction

Each Widget displays data from a single Data Source.

Some Widget graphical properties are automatically generated from the values in the data.
As a result, it is preferable to start Widget configuration by binding data.

> In addition, data binding in Alpana Designer is very interactive and is a nice way to start exploring your data.

What is data binding ?

When binding data, you are in fact configuring a query to the Data Source.
Internally to Alpana, the result of this query is a table of values which will be used to dynamically change some graphical properties of the Widget.

For example, when you want to see the average temperature by Equipment on a Column Chart, the Widget will draw as many rectangles (columns) as there are Equipments, and the height of the rectangles (Y position) will be a number of pixels proportionate to the average temperature.

Why aggregate Quantities ?

In Dashboard visualization, we want to turn many data points into few simple indicators that allow to take a decision efficiently.
Several data rows will turn into a single shape (this is called “*grouping*”), and a single data value will be associated to this shape (this is called “*aggregating*”).

Even if you “know” that for each group your data contains only a single data row right now, your KPI definition needs to account for the fact that there could be many.

Example

You are trending the temperature from a sensor. Right now you know you only have 1 data point at each instant.
So why would you need to choose an aggregation for the Temperature ?

Well, tomorrow you will need to integrate the temperature measurements from 3 sensors.
Then, at each instant you will have 3 data points, but your chart is configured to show only 1 trend.

To solve this, you will need to create an aggregation expression.
Maybe a simple `AVG(Temperature)`, or maybe something more complex involving Alpana Expressions.

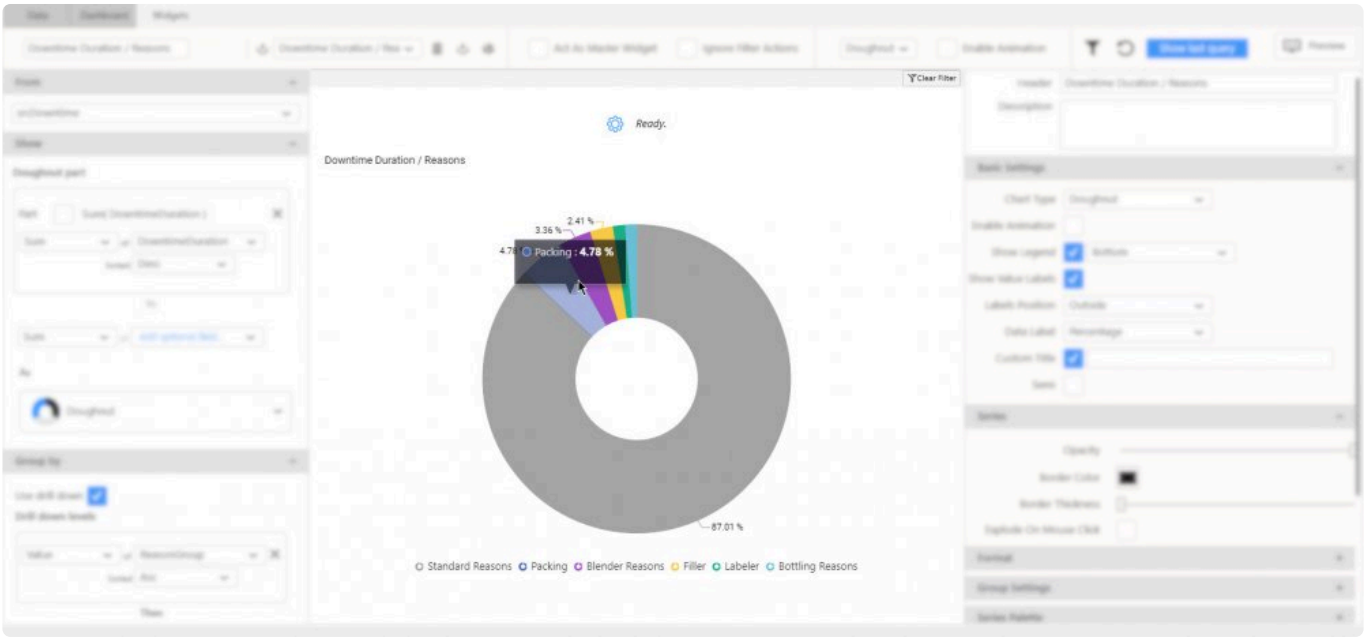
Last modified: Apr 30, 2019

2.7.3.1. using the Widget Preview

Widget Preview

When editing a Widget, the middle part of the window contains a live Widget preview.

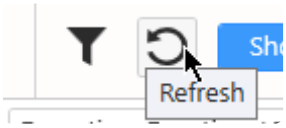
Every time you make a change to the Widget configuration, the Widget Preview is refreshed to reflect that change.



This preview is interactive and can be used to show tooltips, click to drill-down, or scroll to zoom on maps.

Refreshing the Widget Preview

This Widget Preview can be manually refreshed using the refresh button in the top bar :



This makes a new query to the data, and refreshes the preview.

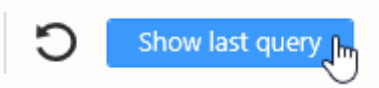
Last modified: Apr 29, 2019

2.7.3.2. Troubleshooting Widget data

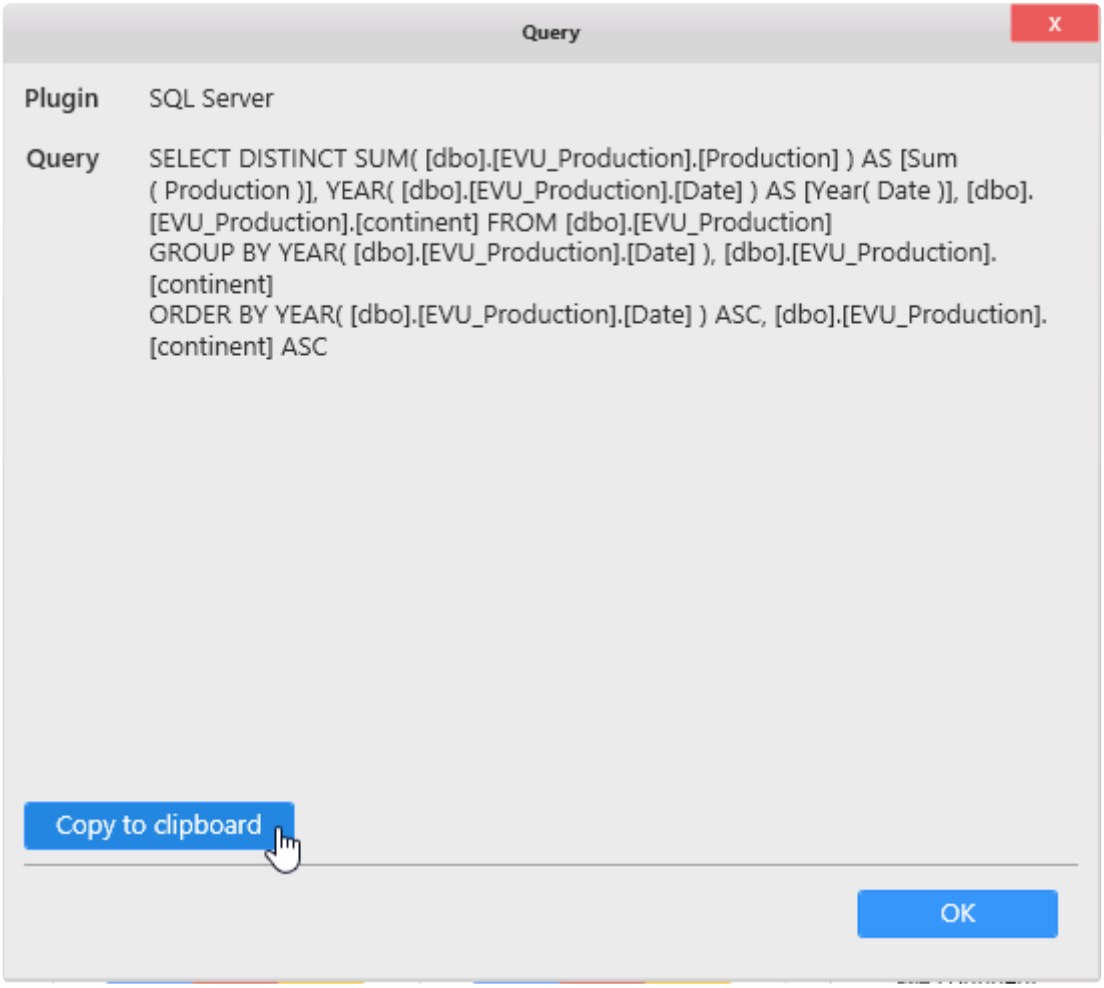
Show last query

In order to troubleshoot widget data, it is possible to get the last data query that was successfully composed for this Widget.

Select the desired Widget, and click the “Show last query” button on the top :



A dialog pops up and lets you see the last successful query for this Widget. You can also press “Copy to clipboard” in order to paste this query in your source data tool and investigate the query’s results manually :

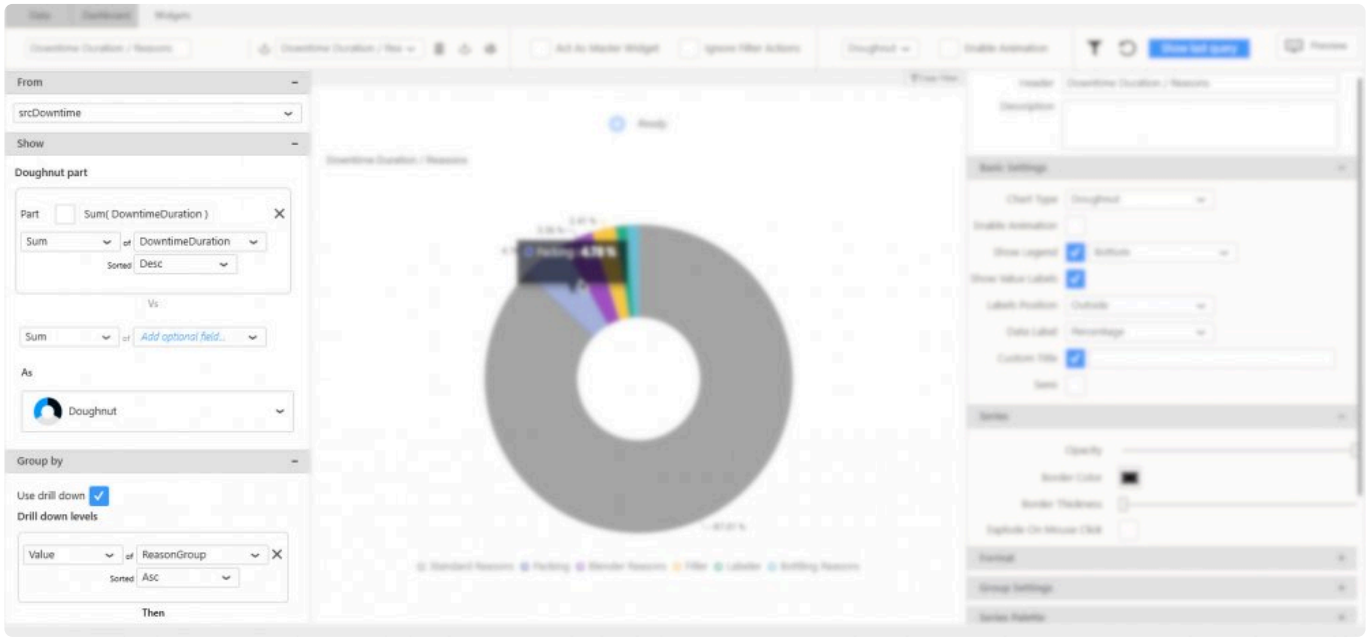


Last modified: Apr 29, 2019

2.7.3.3. Editing Widget data binding

Accessing the data binding

Starting with Alpana v3.0.0, all Widget data binding happens in the data panel on the left of the Widgets activity :



Folding / Unfolding sections

Different parts of the data binding interface are grouped into sections.
For better readability, each section can be :
… folded …

… and un-folded :

Data binding workflow

The data binding should read like a sentence :
> From *My Data Source*, Show *My Aggregated Measure Field* Group By *My Dimension Field*.

From : Data Source selector

This is where you will select the Data Source to bind to this Widget.

(see chapter Selecting a Data Source)

Show : Quantities

This is where you will configure the KPI : what **quantity** you want to visualize.
This will be an **aggregated *Measure***.

(see chapter Configuring the aggregation of number fields and chapter Binding a data field)

Group : Context

This is where you will define what **context** you will use to group data together.
This will be an un-aggregated **Dimension** with unique separate values.

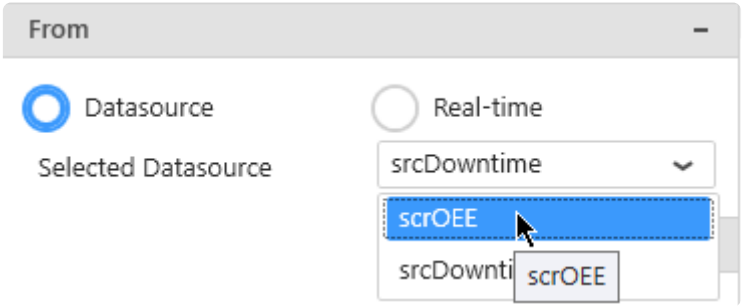
(see chapter Configure the grouping of category fields and chapter Binding a data field)

Last modified: Apr 30, 2019

2.7.3.4. Selecting a Data Source

Each Widget binds data from a single Data Source.

In order to select a Data Source for this Widget, open the Data Source listbox on the top of the Data panel, under the **From** title :



Warning
If the Widget had any data binding from another Data Source, it will be lost !

Last modified: Oct 15, 2019

2.7.3.5. Configuring the aggregation of number fields

Introduction

Most Widgets have configuration fields that require to generate a single number for each corresponding shape in the Widget.

Examples :

- Chart : the chart line Y position of each point
- TreeMap : the rectangle relative size for each rectangle
- …

Since data is grouped in a Widget query (“for each point”, “for each rectangle”, …), numerical fields must be aggregated in order to give a single number for each Widget “shape”.

This operation is done with an aggregation function.

> An *Aggregation Function* is a function that can take any number of values and turn them into a single number.

For Measure fields

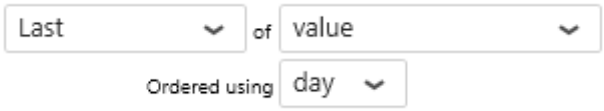
For Measure fields (numbers), the following aggregation functions are available : `Avg`, `Count`, `Distinct Count`, `Max`, `Min`, `Sum`, `STDev`, `STDevP`, `Var`, `VarP`, `Last`.

These functions absolutely require a number (except `Count` and `Distinct Count`, see below).
They translate to the corresponding functions in your source data engine (SQL, Oracle, … depending on your Data Source). Please refer to the corresponding documentation if you need to understand better these functions.

Using function `Last`

(new in Alpana 2020 R1)

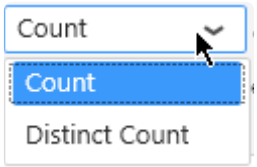
The `Last` aggregation function allows to select the last value of the Measure field, according to an ordering based on a secondary field.
Typically, if a value changes over time, "@Last@ *of* `value field` *ordered using* `date field`" will select the last value of `value field` when ordered according to the `date field`.
In some typical dashboards, this can be considered the "current value".



For Dimension fields

Dimension fields (strings, dates, booleans) can also be used to generate numbers for Widget queries.

The following aggregation functions are available : `Count`, `Distinct Count`.



These functions accept all data types.
They translate to the corresponding functions in your source data engine (SQL, Oracle, … depending on your Data Source). Please refer to the corresponding documentation if you need to understand better these functions.

For Aggregated Expressions

(see corresponding chapter)

Aggregated Expressions already guarantee they provide a single number, and no aggregation function can be used on top of them.

In order to use an Aggregated Expression as a number field, select `Value` instead of an aggregation function :

Last modified: May 11, 2020

2.7.3.6. Configure the grouping of category fields

Introduction

Most Widgets have configuration fields that are used to generate separate shapes. These fields are used to group numerical fields together and generate a single shape for each context combination.

Examples :

- Chart : the chart column category (x position) of each point ; the creation of each trend series
- TreeMap : the creation of each rectangle
- …

Grouping function

There is no particular grouping function.
Only "*Value* of" is available.

It creates a group for each distinct value of the selected field.

For string, number and boolean fields

There is no particular configuration for the following types of fields :

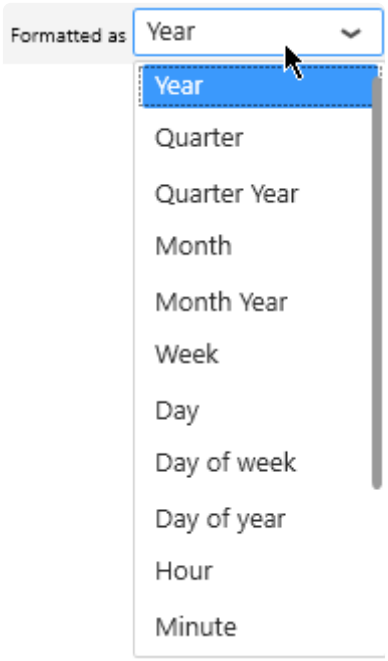
- string
- number
- boolean

When they are added to a category configuration, they group the query without any formatting : each separate value creates a separate group.

> Note : Aggregated Expressions cannot be used as category fields because they only generate a single value, which is not desirable.

For date fields

Date fields can be formatted to create different groups :



- Year (default when adding a field)
- Quarter
- Quarter Year
- Month
- Month Year
- Week
- Day Of Week
- Day Of Year
- Day
- Hours
- Minutes
- Seconds
- Date
- Date Time : the full datetime detail (no formatting happens, each different value is a different category)
- Time

Last modified: Apr 29, 2019

2.7.3.7. Binding a data field

Mandatory fields, optional fields

In order to complete the Widget configuration, some Widget properties **must** be configured : the Widget will not work without this field bound.
The corresponding field selector is marked with "*Select required field…*"

Some fields are optional : you may need to configure them in order to get what you want, but the Widget will still work without this field bound.
The corresponding field selector is marked with "*Select optional field…*"

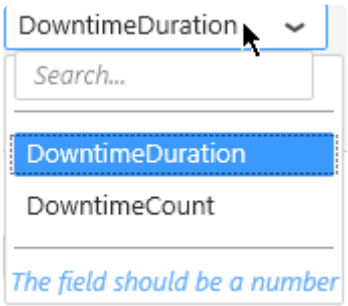
When you want to configure a field, just use those field selectors to select the desired field.

Selecting a field to bind

Depending on the aggregation function selected and the property you are trying to configure, not all field types may be available.
For example, if the aggregation function `Sum` is selected, then only the numerical fields and numerical Expressions are listed.

Number fields

If the configuration only accepts numbers, all number fields are listed in a single flat selector :

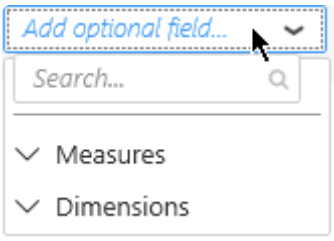


"Any" fields

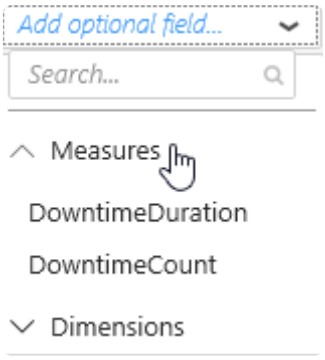
If the configuration accepts any type of fields, they are organized in 2 categories :

- *Measures* : the numerical fields from your data.

- *Dimensions* : the non-numerical fields from your data.

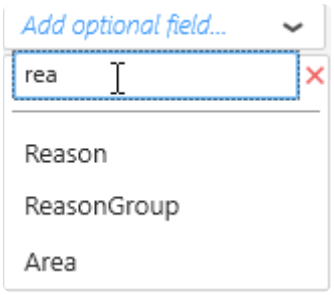


Just click a category to unfold it and list all fields of that type :



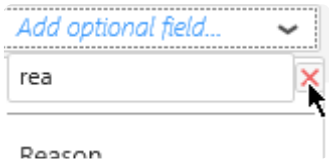
Searching for a field

Instead of browsing for a field, you can use the search box to start typing part of the field name :



This will incrementally list all fields partially matching the search pattern.

To clear the search, simply click the cross button next to the search box :



Last modified: Apr 30, 2019

2.7.3.8. Un-binding a data field

To remove a field binding from a Widget, click the cross inside the field container :

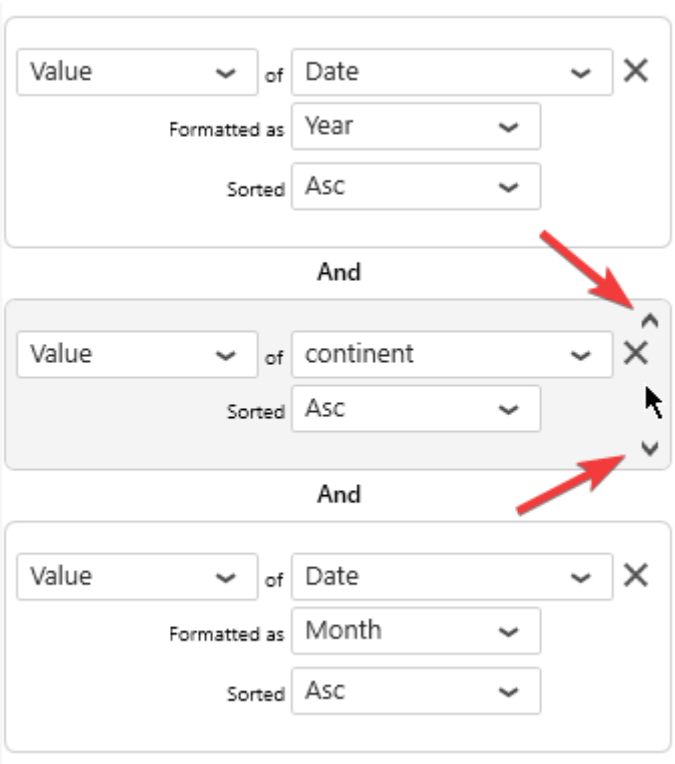
! All related configuration will be lost.

Last modified: Apr 30, 2019

2.7.3.9. Re-Ordering data fields

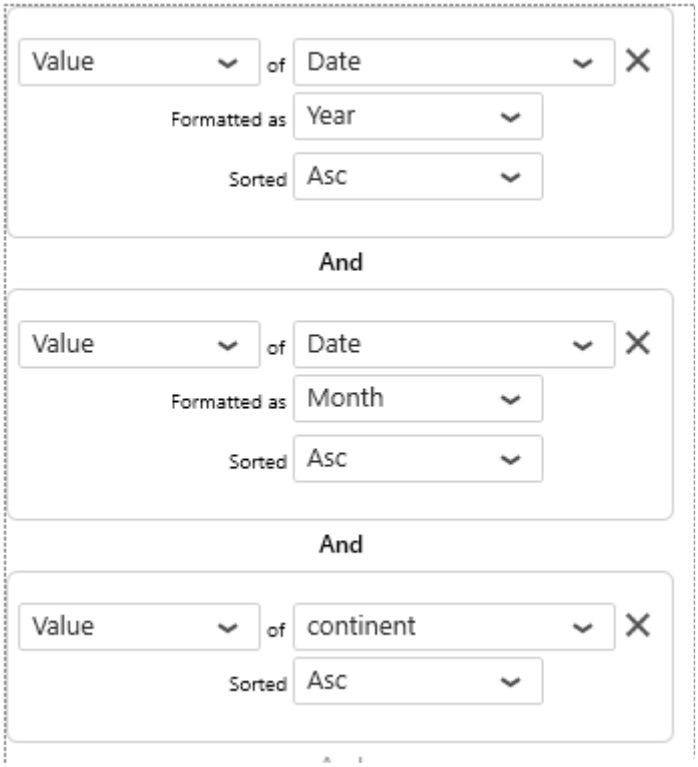
Some Widget configurations rely on the order of the list of bound fields.

In order to change the order of a field, hover it, and see small *up* and *down* arrows appear :



Click the *down* arrow to move the field down :

As a result, the field is swapped with the below field :



Click the *up* arrow to move the field up :

Last modified: May 07, 2019

2.7.3.10. Binding to Real-Time Data

Introduction

Instead of binding Widget data to a Alpana Data Source, you can configure it to accept Real-Time Data. Instead, it lets you create a Widget with no Data Source, and then you can push data directly into the Widget, using the Alpana web API.

The general process is :

1. From Alpana Designer, no need to create a Data Source for this Widget
2. Configure the Widget for Real-Time Data
3. Publish the Dashboard on Alpana Server
4. From an external tool (another web page, ArchestrA scripts, your own .NET application, etc), connect to the Alpana Server web API and push data into this Connector
5. ⇒ Your Widget displays the data live, instantaneously.

Configuring a Widget for Real-Time

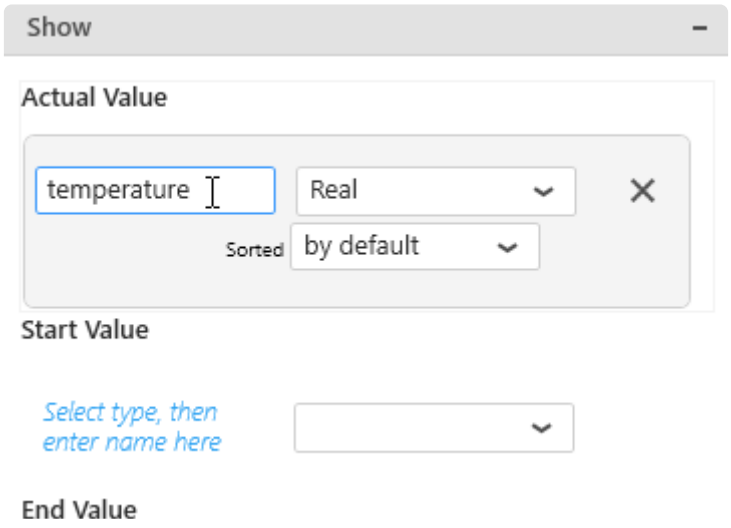
Select Real-Time

In order to this Widget to have Real-Time Data, select *Real-Time* on the top of the Data panel, under the **From** title :

Define fields

You need to define some fields for the widget to later accept data from the API.
The required fields depend on the widget type.

For each field, choose a type and a field name :



Last modified: Oct 15, 2019

2.7.4. Configure a sorting order

Introduction

Most Widgets will present data differently depending on how it is sorted in the query.

Example : a Chart is by default sorted by ascending X Axis values. It can instead be sorted by descending Y Axis values.

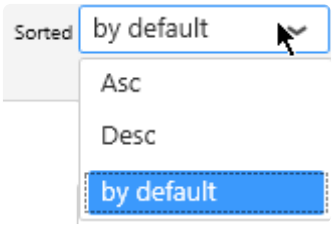
Sorting Priority

As a rule, the fields at the top of the Data binding panel will have sorting priority over the fields below.

Example : if you set a Chart Y Value field sorted descending, it will override the X Axis sorting because it is located above in the Data binding panel.

Configure sorting

On each bound data field, you can configure sorting :



- *by default* : do not take part in query sorting
- *Asc* : sorted in Ascending order
- *Desc* : sorted in Descending order

Last modified: Apr 30, 2019

2.7.5. Widget-level Filtering

Filtering data at the Widget level

You can apply filters at the Widget level.
The current Widget will only see the data after the filter is applied.

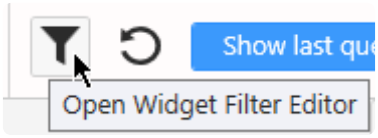
Those ***Widget-level*** **filters** are applied on top of any Data Source filters (if any) and apply only to the current Widget.

Last modified: Apr 30, 2019

2.7.5.1. Opening the Filter Editor

Add or Edit Filters

To add Filters to a Widget or edit existing filters, press the “Open Widget Filter Editor” funnel icon in the top toolbar :



See the presence of Filters

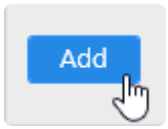
When a filter is applied on a Widget, the Filter button is colored :

Last modified: Apr 30, 2019

2.7.5.2. Managing Filter Conditions

Adding a condition

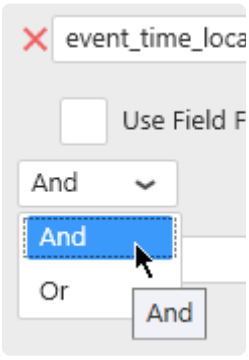
In the Filter Editor, click Add to add a filter condition :



Combining conditions

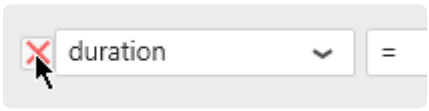
If filter conditions already exist, they will be chained using a logical operator.

By default, the operator is “And”, but it can be changed to “Or” :



Removing Conditions

Click the red cross on the left to remove the corresponding condition :



Last modified: Apr 30, 2019

2.7.5.3. Filtering aggregated values

Introduction

There are 2 ways to apply a widget filter :

- on the **raw** values : each row in the raw dataset is evaluated against the filter condition
- on the **aggregated** values : the raw dataset is first aggregated by the Widget, and only *then* the filter condition is applied

Example

The Data Source

Imagine a simple dataset :

TagName	Building	Product	Price
FT101.Value	Building A	Product X	10
FT102.Value	Building A	Product Y	15
FT201.Value	Building B	Product X	20
FT202.Value	Building B	Product Y	25
FT301.Value	Building C	Product X	30
FT302.Value	Building C	Product Y	35

The Widget

You are displaying this data in a Pie Chart with `Sum`(Price) grouped by “Product”, you get :

Product	Sum(Price)
Product X	**60**
Product Y	**75**

Filtering raw values

You apply a filter on the **raw** value to see only data where :
`Price > 20`

The raw dataset will be filtered like this :

TagName	Building	Product	Price
FT202.Value	Building B	Product Y	**25**
FT301.Value	Building C	Product X	**30**

FT302.Value	Building C	Product Y	**35**

And the Widget will display the following :

Product	Sum(Price)
Product X	**30**
Product Y	**60**

(because 60 = 25 + 35)

Filtering aggregated values

You apply a filter on the **aggregated** value to see only data where :
`Sum(Price) > 70`

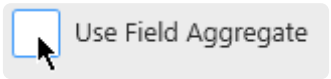
It is not the raw dataset that will be filtered.
The Widget will display the following :

Product	Sum(Price)
Product Y	**75**

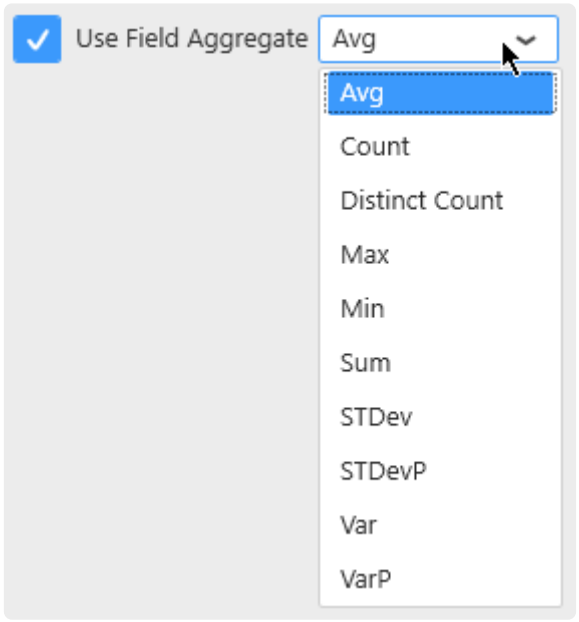
(Product X was filtered because its Sum(Price) was 60 <= 70)

How to filter by aggregated values

For each filter condition, you can check *Use Field Aggregate* :

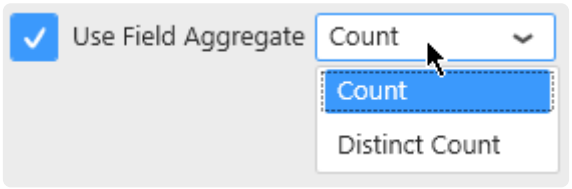


Then select the desired aggregation function :

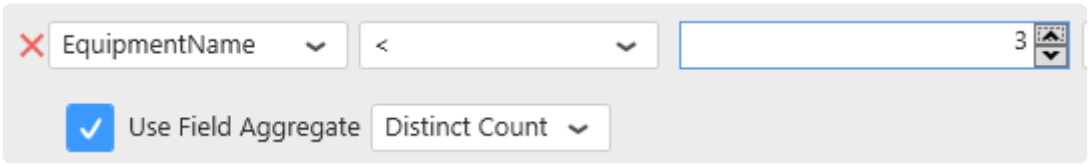


Note on non-numerical fields

On fields where the data type is not number (string, date, boolean), the aggregation functions will return a number, so the condition becomes a number condition.
For those fields, the list of available aggregation functions is also different :



For example, the resulting filter condition below reads as :
Distinct Count of EquipmentName < 3



Last modified: Apr 30, 2019

2.7.5.4. Configuring Filter Conditions

Filter conditions are expressed with :

- a field name from the Data Source. This can also be an Expression.
- a boolean operator
- a value. This can also be a Parameter.

Depending on the Data Source field type, the settings are slightly different, please see for each type in the next pages.

Last modified: Apr 30, 2019

2.7.5.4.1. Filtering string fields

This is the most basic case.

The filter condition will be :

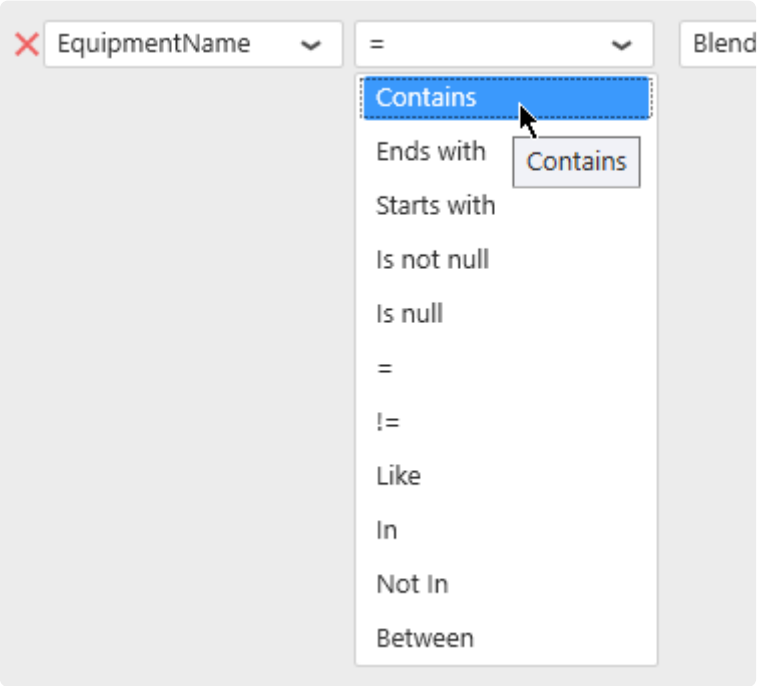
```
<FieldName> <Operator> <Value>
```

Example :

```
EquipmentName StartsWith Blender
```

Operators

Operators appropriate to Strings can be selected.

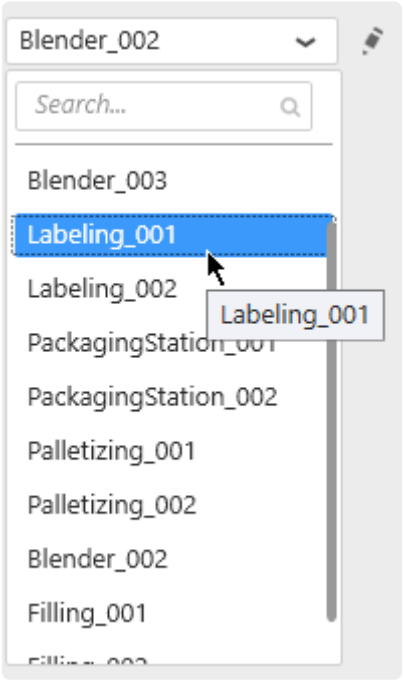


Values

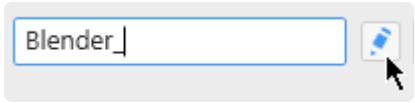
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text :



Depending on the operator selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend on actions inside the Dashboard.

For this, select “Use Parameter” and see the Chapter about Parameters.

Last modified: Apr 30, 2019

2.7.5.4.2. Filtering number fields

In a basic configuration, the condition is expressed as :

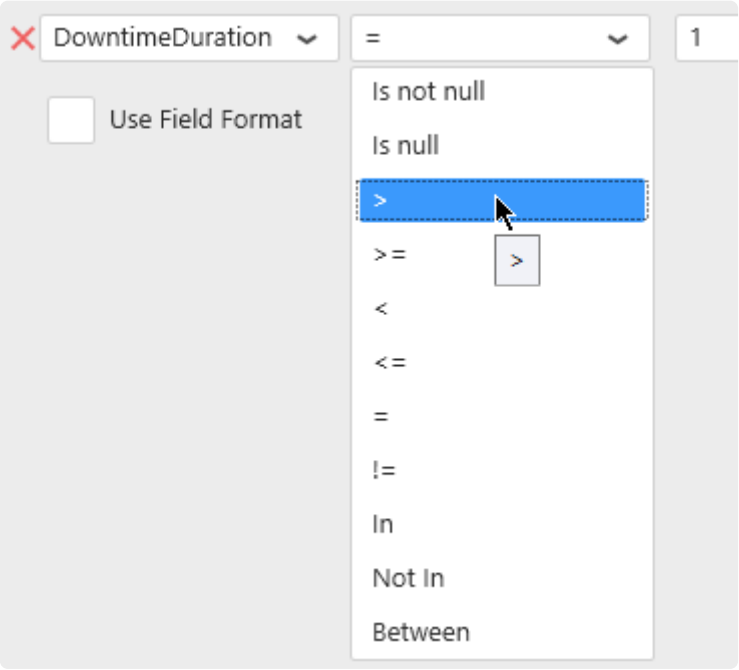
```
<FieldName> <Operator> <Value>
```

For example :

```
DownTimeDuration > 10
```

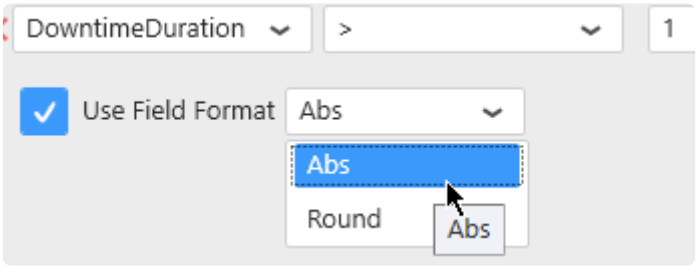
Operators

Operators appropriate to Numbers can be selected :



Field formatting

In addition to the basic configuration, a formatting function can be applied to the numerical field while the comparison is made :



This way, the filter condition will be :

```
<Function>(<FieldName>) <Operator> <Value>
```

For example :

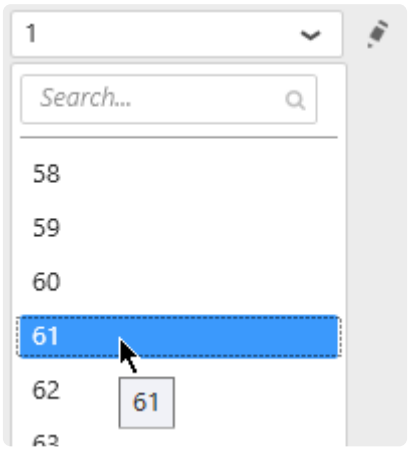
```
Abs(DownTimeDuration) > 10
```

Values

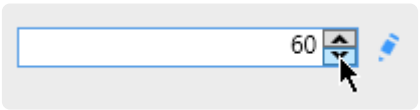
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text, or using the up/down arrows to increase/decerease :



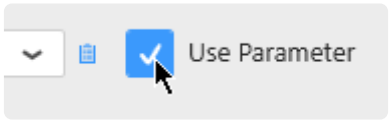
Depending on the operator selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend

on actions inside the Dashboard.

For this, select “Use Parameter” and see the Chapter about Parameters.



Last modified: Apr 30, 2019

2.7.5.4.3. Filtering date fields

In a basic configuration, the condition is expressed as :

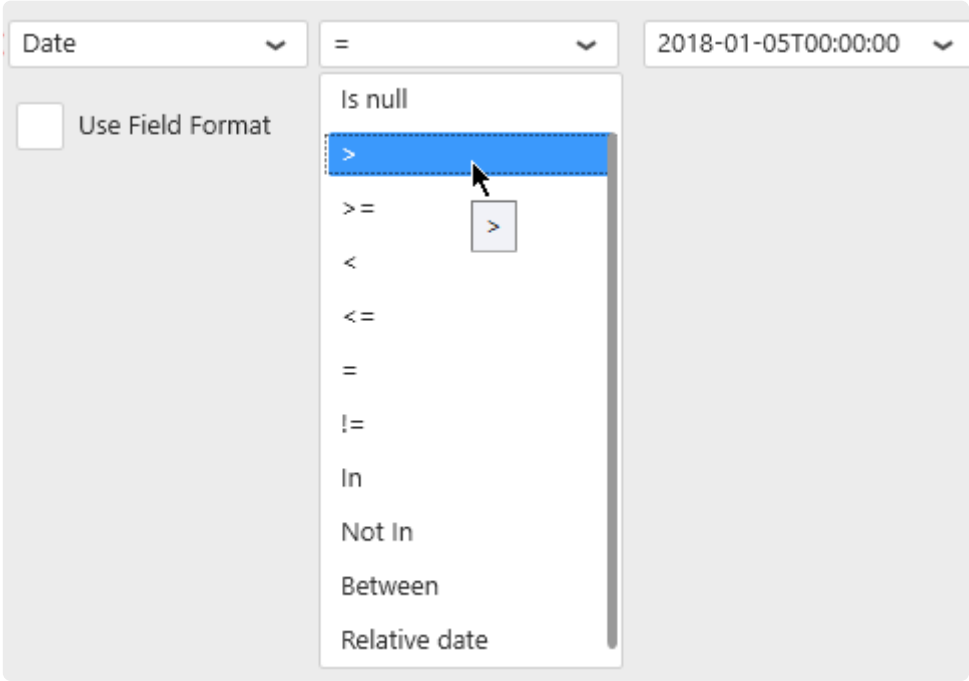
```
<FieldName> <Operator> <Value>
```

For example :

```
aaValueStartTime > 2018-01-05T00:00:00
```

Operators

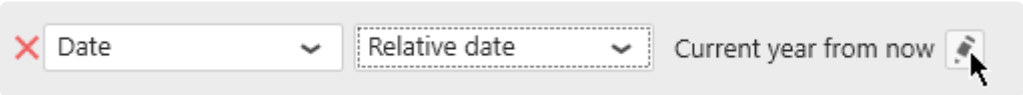
Operators appropriate to Dates and Times can be selected :



Relative date filtering

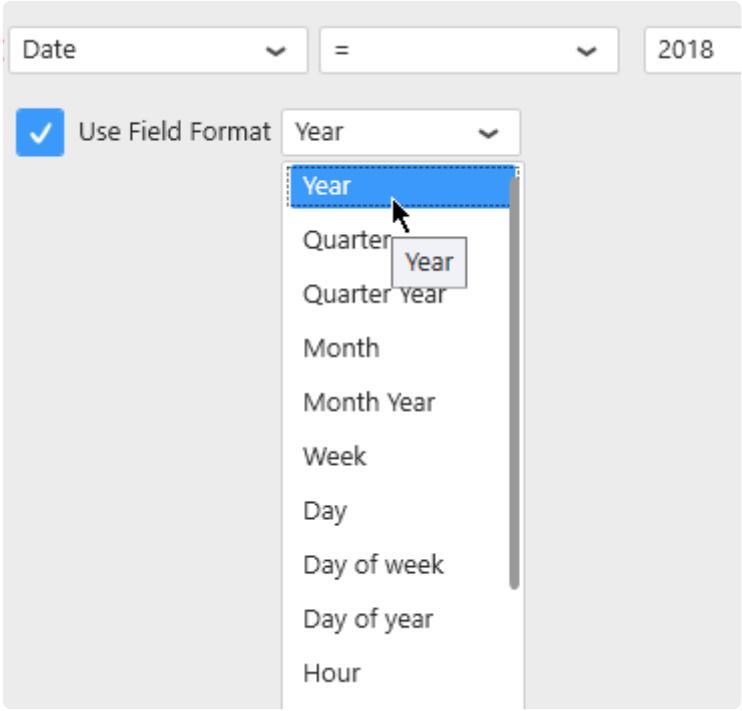
The special operator “Relative date” allows to select a “date window” / “time frame” that is moving over time relative to the current moment.

When selecting this operator, click the edit icon to define the relative date window :



Field formatting

Similarly to the number fields, a formatting function can be applied to the date field :



This way, the filter condition will be :

```
<Function>(<FieldName>) <Operator> <Value>
```

For example :

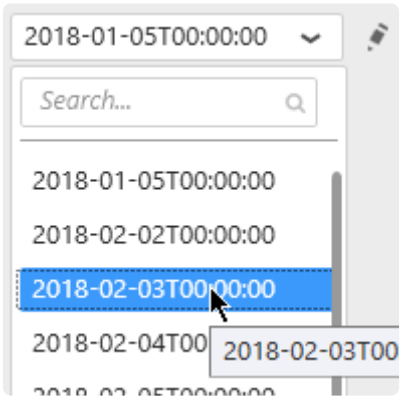
```
Year(Date) BETWEEN 2015 AND 2016
```

Values

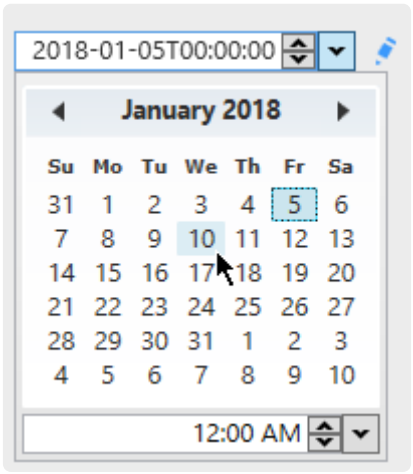
Static values

The “Edit values” button can be check/unchecked manually :

When the “Edit values” button is un-checked, a value can be selected from current values in the database :



When the “Edit values” button is checked, a value can also be entered as text :



Depending on the operator and formatting function selected, some of these options may not be available.

Dynamic filter values using Parameters

Instead of comparing to a static value, it is possible to compare to a dynamic Parameter that will depend on actions inside the Dashboard.

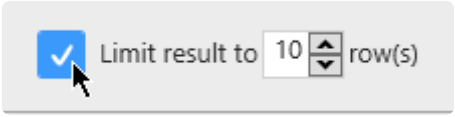
For this, select “Use Parameter” and see the Chapter about Parameters.

Last modified: Apr 30, 2019

2.7.5.5. Limiting dataset size

How to limit

Check “Limit result to” and select the number of rows you want to keep in the dataset :



Why limit ?

Checking this option with “Limit result to 10 rows” for example will apply a “TOP 10” to the query. When widgets are sorting the data, this can be a useful to display the 10 Items with the worst performance, or the 10 equipments with the highest consumption for example.

Last modified: Apr 30, 2019

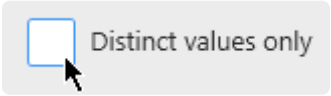
2.7.5.6. Getting Distinct Values

(new in Alpana 2020 R1)

Most Widgets show aggregated data, and so they necessarily display distinct values.
For example, in Charts, data queries are **grouped** in order to have a single Y value for each separate point on the X Axis.

However, some widgets allow to meaningfully display redundant values.
For example, in a Grid, you may want to list all values of “produced diameter”, displaying multiple rows even if there is the same value multiple times.

This is possible by un-checking *Distinct Values Only* :



Example with *Distinct Values Only* enabled (default) :

Produced Diameter
1.25
2.25
2.50

Only distinct values are displayed.

Example with *Distinct Values Only* disabled :

Produced Diameter
2.50
2.50
2.25
1.25
2.25
2.25
2.25

Now all values are displayed, including several times the same value (if the data contains such occurrences).

Last modified: Apr 24, 2020

2.7.6. General Widget Properties

All Widget types have specific Properties that are documented in the corresponding chapter.

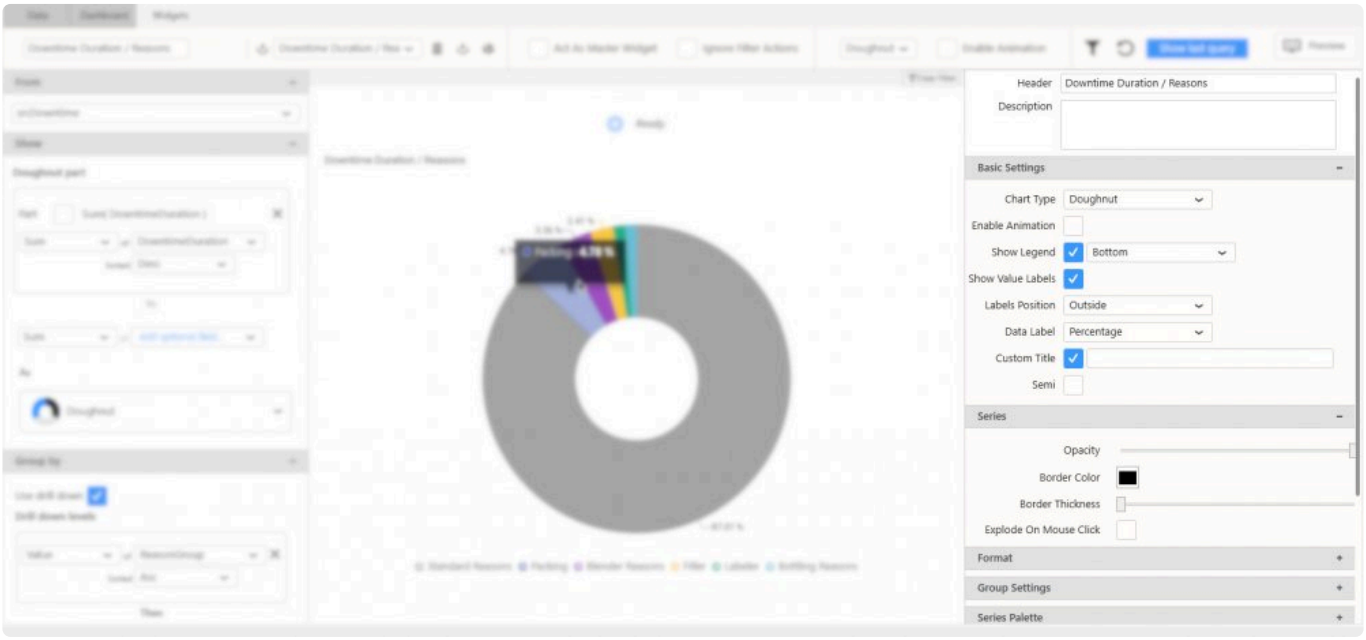
However, some Properties are common to several Widget types and listed in the current Chapter.

Last modified: Apr 30, 2019

2.7.6.1. Editing Widget Properties

Accessing the properties

Starting with Alpana v3.0.0, almost all Widget properties configuration happens in the properties panel on the right of the Widgets activity :



However, some properties are also located in the top bar for quick access.

Folding / Unfolding categories

Different parts of the properties configuration interface are grouped into categories.
For better readability, each category can be :
… folded …

… and un-folded :

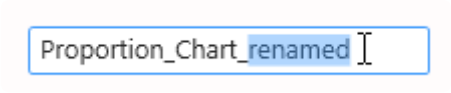
Last modified: Apr 30, 2019

2.7.6.2. Name, Header and Description

Widget Name

Each Widget has an internal Name used only at design-time, for example in the Widgets list.
This Name is not visible to the end user when the Widget is published.
It allows to have a clear useful name for the developer while the end user will want to see a different custom header, or even no header.

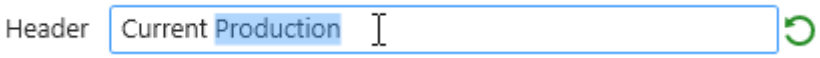
The Widget Name can be configured from the left of the top bar :



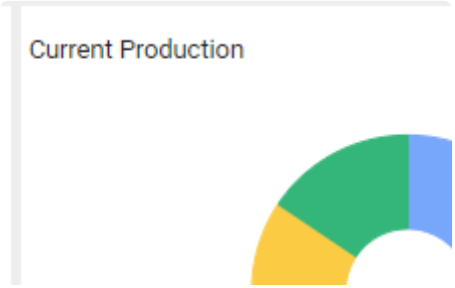
Header

Widgets can also have a Header visible to the end user.
This can be used to display a title that can be a short description of the Widget's purpose.

The Header can be configured from the top of the properties panel :

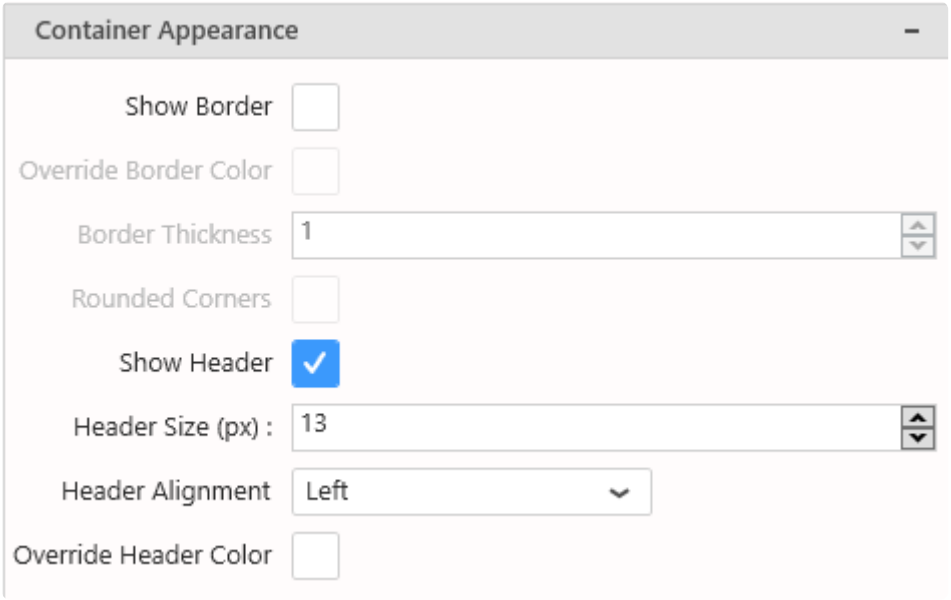


At runtime, this will be displayed at the top of the Widget container :



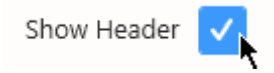
Header options

Some Header options are available in the *Container appearance* section :



Show / Hide Header

The *Show Header* option (check by default) allows to display the Header at runtime :



When the option is un-checked, the Header is invisible at runtime.

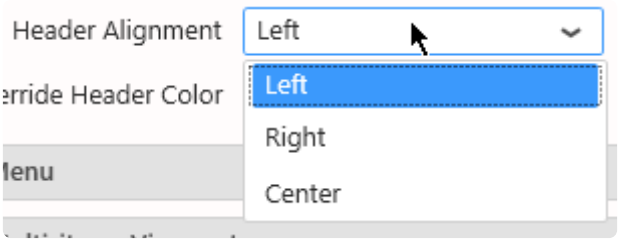
Header Size

The *Header Size* option allows to override for this Widget the Header Size set at the Dashboard level.

Header Alignment

Allows to align the Header :

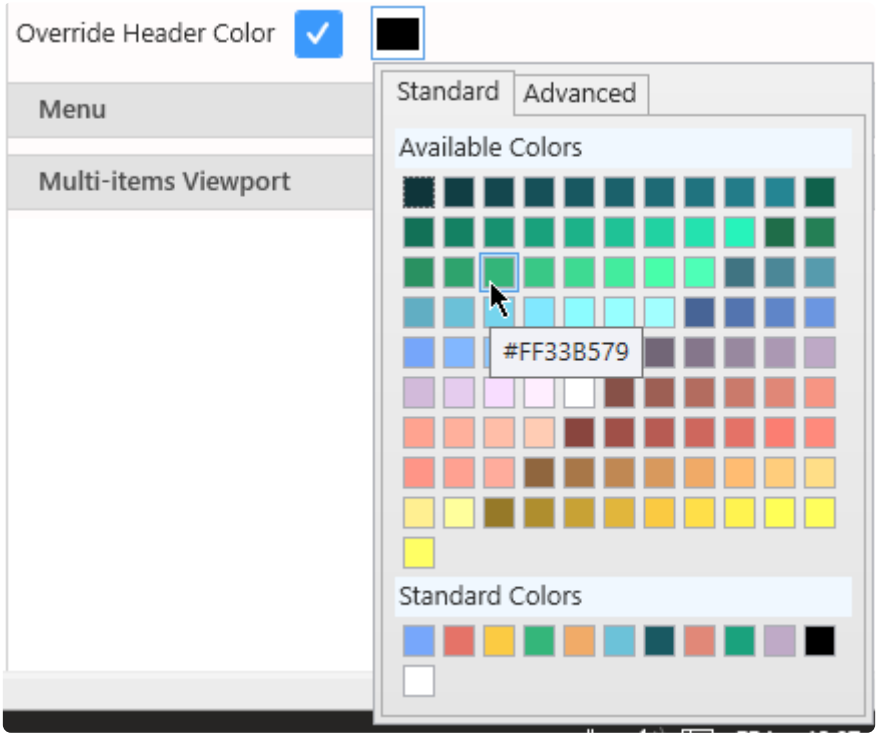
- Left
- Right
- Center



Header Color

By default, Header Color is managed by the currently applied Theme.
However, you can override the Theme color for the Header by checking *Override Header Color* and

selecting the desired color :



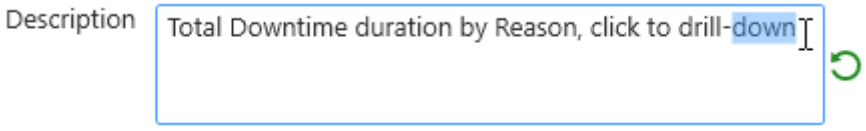
Label Parameters

The text content of the Header can include placeholders that will be replaced by Parameter values, see Label Parameters

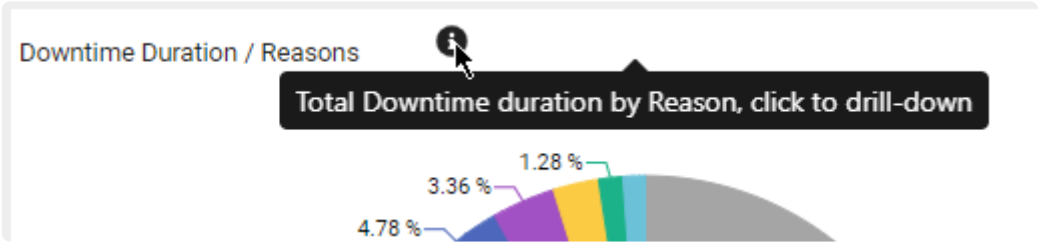
Description

The Description can be used to display a longer text describing the Widget's purpose, its data, how to use it, …

The Description can be configured from the top of the properties panel :



At runtime, the Description is hidden and a ⓘ icon appears near the Widget title to indicate the presence of a Description :
Hovering the ⓘ icon shows the description :



Last modified: Aug 26, 2020

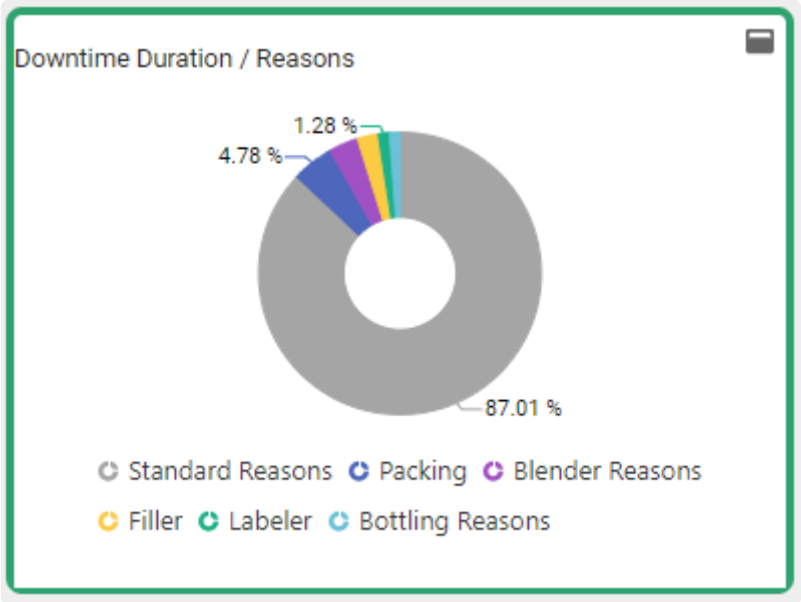
2.7.6.3. Container Appearance

Container Appearance

These properties rule the appearance of the *container* that contains the Widget within the Dashboard.

Example :

results in :



Border

Show border

Show a border line around the Widget.

Border Color

By default, Border Color is managed by the currently applied Theme.
However, you can override the Theme color for the Border Color by checking *Override Border Color* and selecting the desired color.

Border Thickness

Define the thickness of the line defining the border color, in pixels.

Rounded Corners

Round the corners of the Widget container, to turn it into a rounded rectangle.

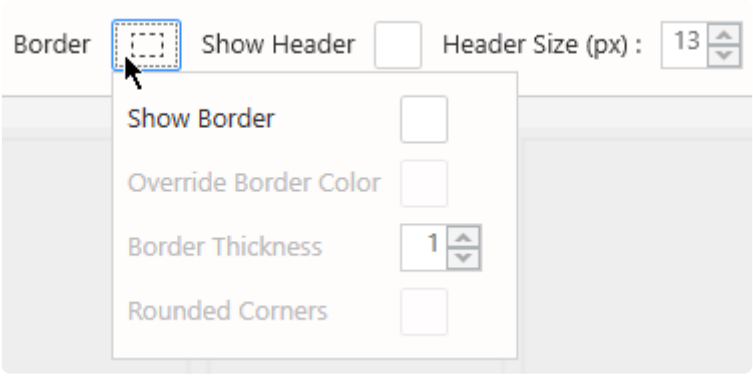
Header

Header settings : see previous chapter about Header and Description.

Note : accessing Border options on other Widgets

Some Widget types are not edited through the Widget activity tab, and don't have a "Container Appearance" section : Image, Label, Embed, …

For those Widgets, the Border options can be accessed in the top bar when the Widget is selected :



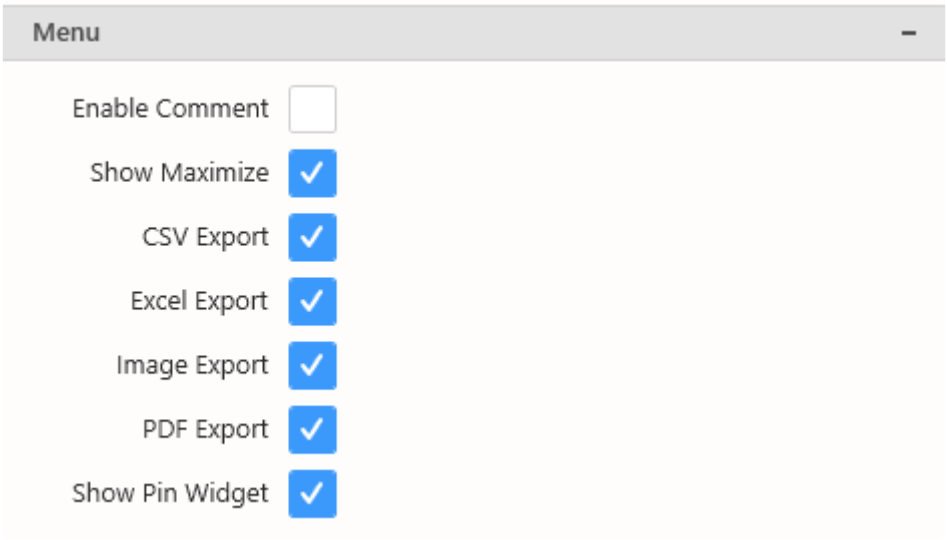
Last modified: Aug 26, 2020

2.7.6.4. Widget Menu items

Menu

At runtime, the top of the Widget Container will include buttons to run some actions on the Widget.

The *Menu* section allows to configure those buttons.



Note : Some of those properties are only visible when the Dashboard is published.

Enable Comment

Shows a control at runtime that allows to create comments at the Widget level.

Show Maximize

Shows a control at runtime that lets the user maximize the Widget full page.

Exports

Each options will allow at runtime the user to export the Widget in the selected format(s).

If none is selected, the *Export* button disappears at runtime.

- CSV
- Excel
- image

- PDF

Pin Widget

Shows a control at runtime that lets the user pin the Widget to a Homepage.

Last modified: Apr 30, 2019

2.7.6.5. Measure Formatting and Date Formatting

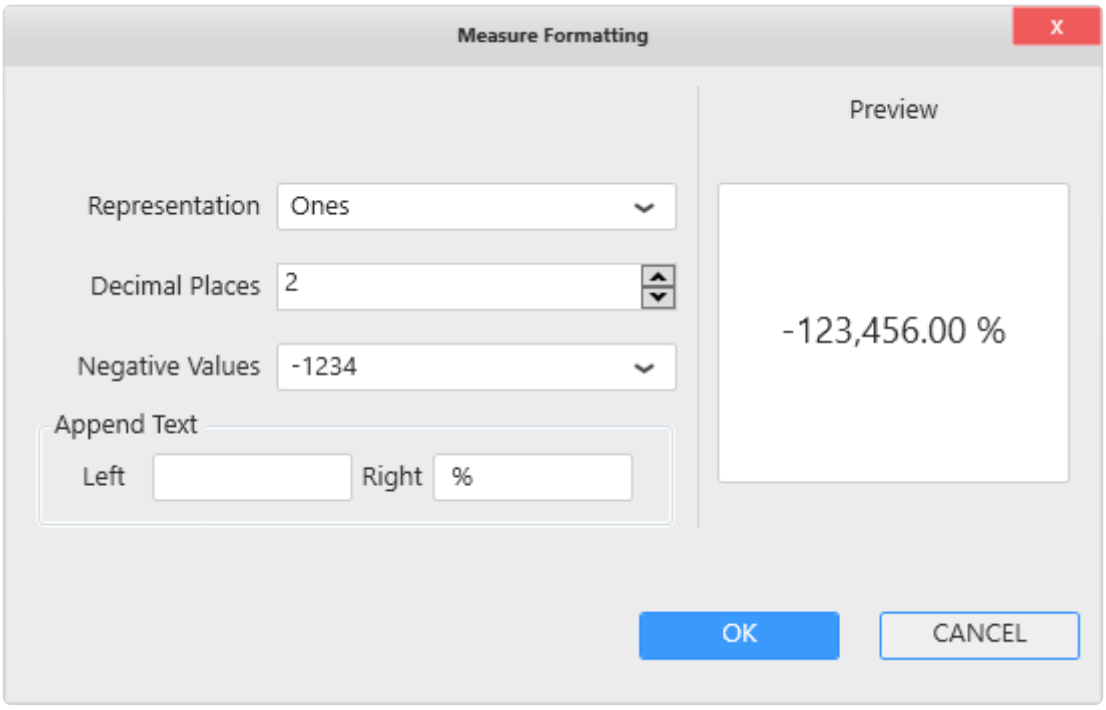
Introduction

Most Widgets show data labels for values and dates, which may depend on user culture and personal choices.

The properties panel has a "*Format*" section which allows to configure these labels. All number and date fields are listed there, and labels can be configured by clicking on the cog icon :

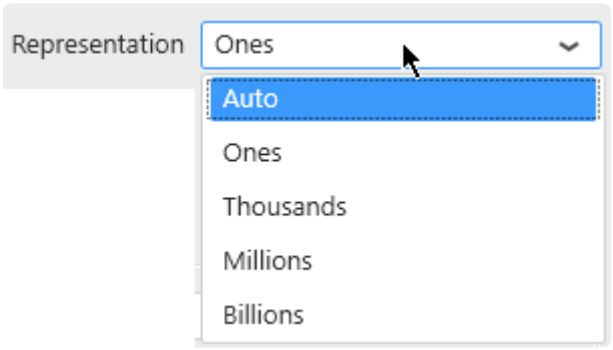
Measure Formatting

The Measure Formatting dialog allows to configure the label formatting for number fields :



Number Representation

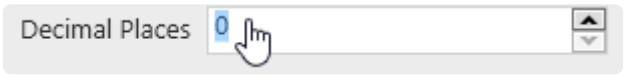
Numbers can be shortened in several ways :



- Auto (default) : depending on how large the number is, a multiplicator symbol can be added : “K” for thousand, “M” for million, “B” for billion
- Ones : the full number is displayed without multiplicator symbol
- Thousands
- Millions
- Billions

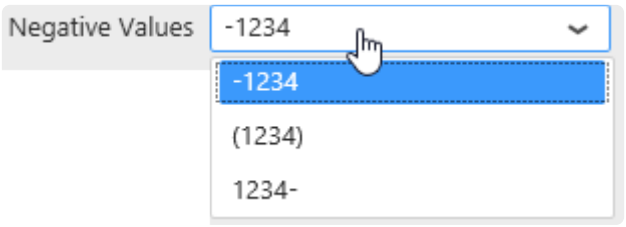
Decimal Places

The Decimal Places selector allows to force how many decimal places should be displayed for numbers :



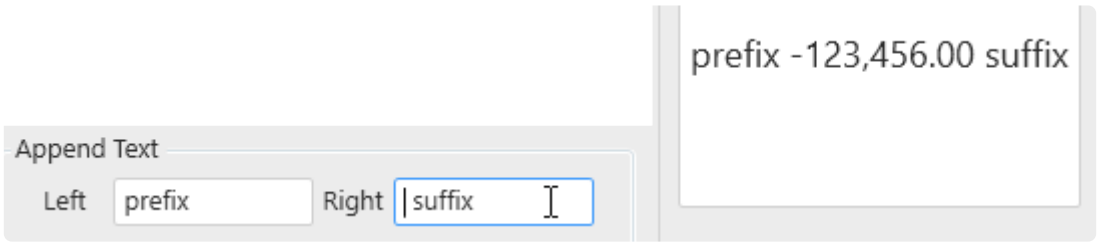
Negative Values

Alternate symbols can be used to represent negative values :



Prefix / Suffix

Prefix (before the number) and/or Suffix (after the number) text can be used :



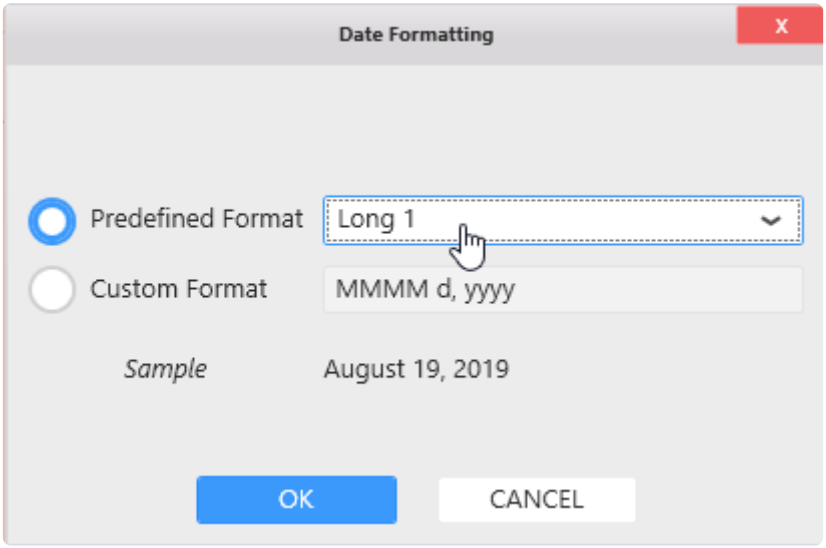
Hint : This is useful for example for showing units of measurement.

No space is added by default between the number and the prefix/suffix.

Date Formatting

The Date Formatting dialog allows to configure the label formatting for `Date` or `DateTime` fields :

Pre-defined formats



A selection of date format types is available, and a preview is shown at the bottom of the window as “Sample”.

Date format localization

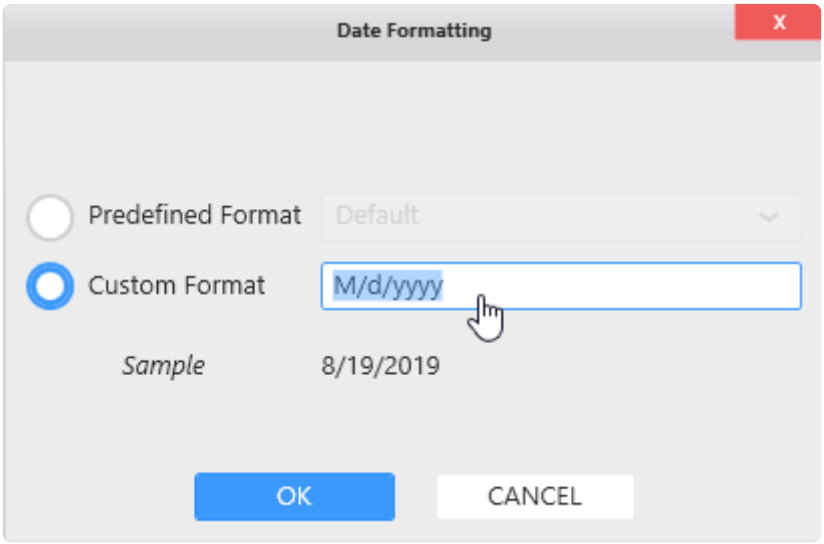
Only the *type* of date format can be configured (short, long, etc…).
At runtime, the date localization will depend on localization choices of the user.
In order of decreasing priority :

1. The Language selected by the user on **Alpana Server** contains culture information that will be used to localize the dates. Note : this is not available during Dashboard Preview from Alpana Designer.
2. The culture selected by the user as preferred language in the **web browser** will be used.
3. The culture of the display language of the **web browser** will be used.

In the Widget Preview, no locale is applied, and only the default EN-US English locale is applied.

Custom formats

Custom formats can also be configured, and a preview is shown at the bottom of the window as “Sample”. :



This allows to use text placeholders to define the custom date string.
There are 2 ways to configure a date format string :

- from a standard date format
- compose a custom string

Standard date formats

If the custom string is a single character, then you can use with one of the placeholders defined here as part of the .NET framework, and you will end up with one of the standard date formats that will adapt to your culture :

- `d` : Short date pattern
- `D` : Long date pattern
- `f` : Full date/time pattern (short time)
- `F` : Full date/time pattern (long time)
- `g` : General date/time pattern (short time)
- `G` : General date/time pattern (long time)
- `M`, `m` : Month/day pattern
- `O`, `o` : Round-trip date/time pattern
- `R`, `r` : RFC1123 pattern
- `s` : Sortable date/time pattern.
- `t` : Short time pattern
- `T` : Long time pattern
- `u` : Universal sortable date/time pattern
- `U` : Universal full date/time pattern

- `Y`, `y` : Year month pattern

Custom date strings

This allows to configure any date part in any order and use that as a label.
Use any text and include special placeholders to compose the date format.
The placeholders are defined here as part of the .NET framework, and a summary is below :

- `d`, `dd` : the day of the month
- `ddd`, `dddd` : the day of the week
- `f`, `ff`, `fff`, `ffff`, `fffff`, `ffffff`, `fffffff`, `ffffffff` : decimals of the second
- `h`, `hh` : the hour, using a 12-hour clock
- `H`, `HH` : the hour, using a 24-hour clock
- `m`, `mm` : the minute
- `M`, `MM`, `MMM`, `MMMM` : the month
- `s`, `ss` : the second
- `t`, `tt` : AM/PM
- `y`, `yy`, `yyyy` : the year
- `:` : the time separator
- `/` : the date separator
- `\` : the escape character allows to write a character without interpreting as a placeholder
- `"my text"`, `'my text'` : the escape quotes allow to write a text without interpreting as a placeholder

Last modified: Aug 30, 2019

2.7.6.6. Multi-items Viewport

Introduction

Some Widgets can be turned into a container for many Widgets.
These Widgets are displayed in a grid showing a certain number of Widgets horizontally (Columns) and vertically (Rows).

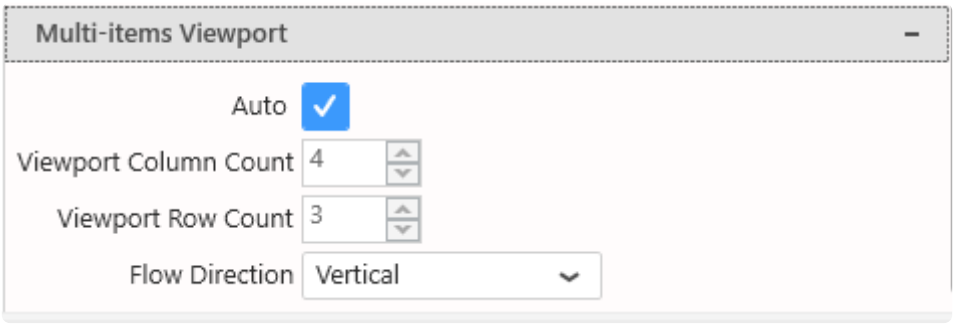
These many Widgets are generated automatically based on a data field, and their number may not be known in advance.
If there are too many Widgets contained, Alpana allows to view only some of them, and then use a scrollbar to see the others.

The properties sections "*Multi-items Viewport*" allows to configure how many Widgets can be displayed at the same time inside the Widget container.

Setting Multi-items Viewport

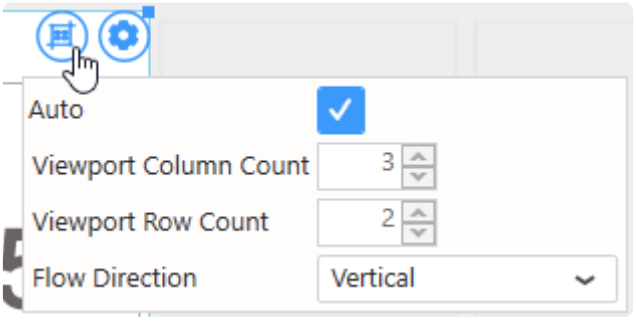
From the Widget properties

In the Widget properties, the section *Multi-items Viewport* allows to configure this setting :



From the Dashboard layout

In the *Dashboard* activity, each Widget has a shortcut to this setting :



Auto Mode

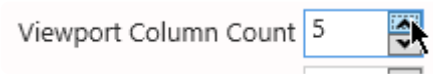
By default, *Auto* is checked and the Viewport is calculated automatically.

Manual Mode

By un-checking *Auto*, you can configure manually the Viewport.

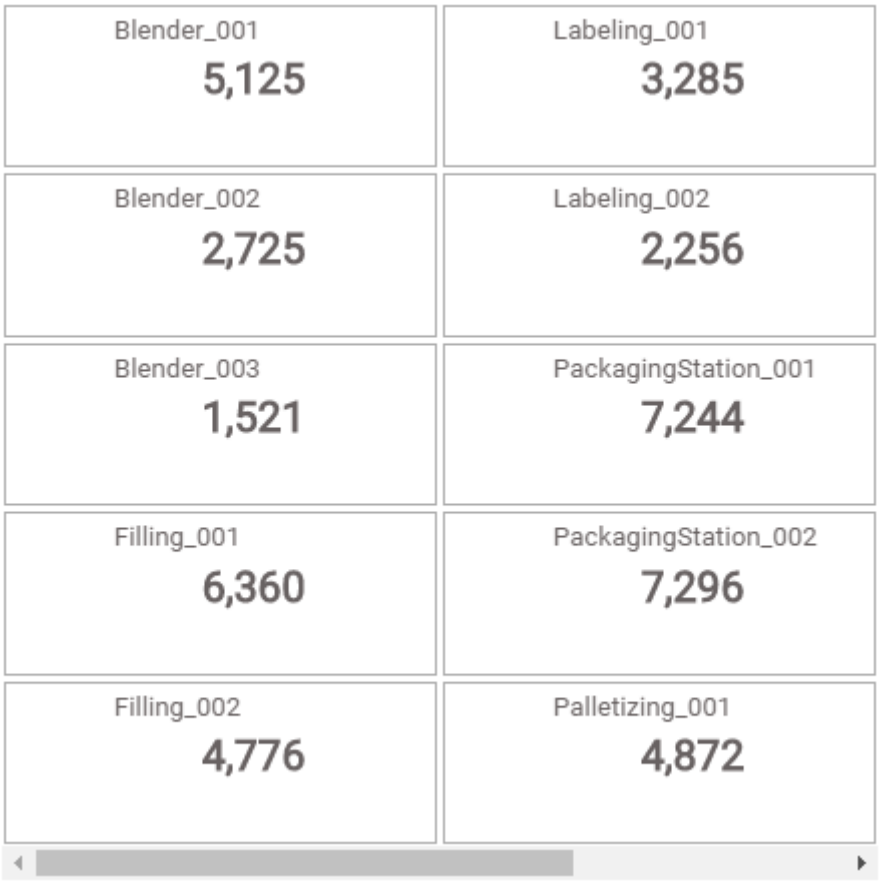
Columns

Viewport Column Count allows to configure how many Widgets are displayed horizontally (the number of columns in the grid) :



Rows

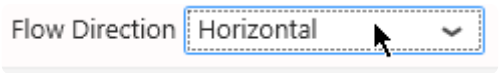
Viewport Row Count allows to configure how many Widgets are displayed vertically (the number of rows in the grid) :

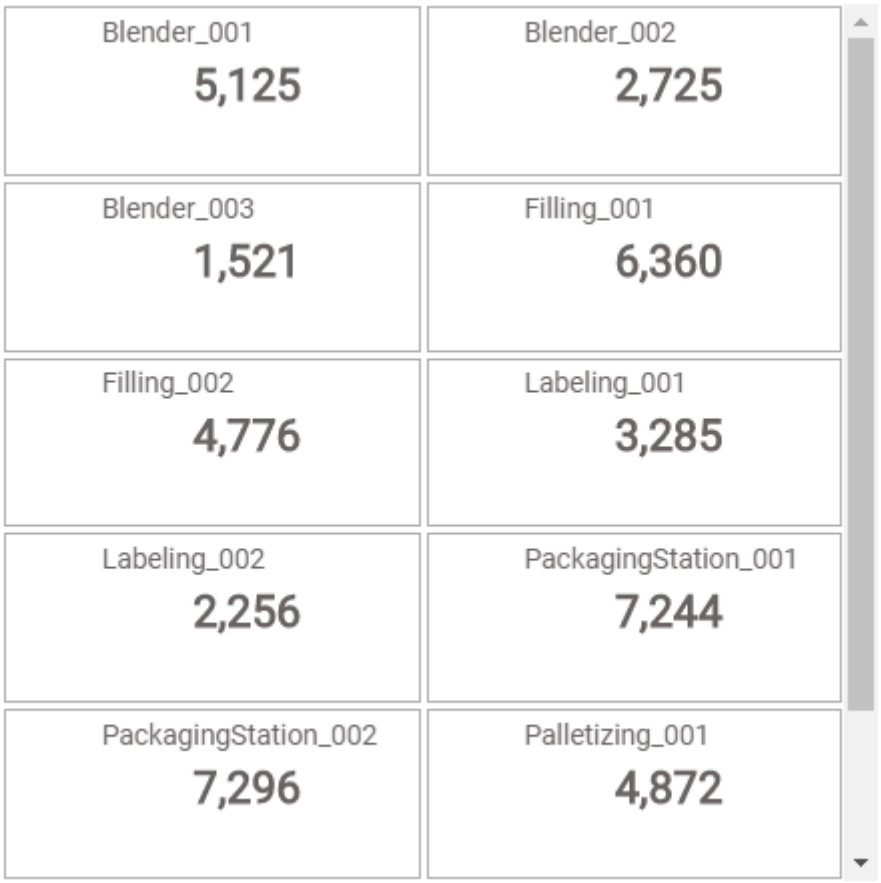


Flow direction

Flow direction allows to configure how Widgets are placed in the grid :

- *Vertical* : Widgets are placed from top to bottom, and then left to right. If there are too many Widgets to display, the next pages are on the right.
- *Horizontal* : Widgets are placed from left to right, then top to bottom (english reading direction). If there are too many Widgets to display, the next pages are at the bottom.



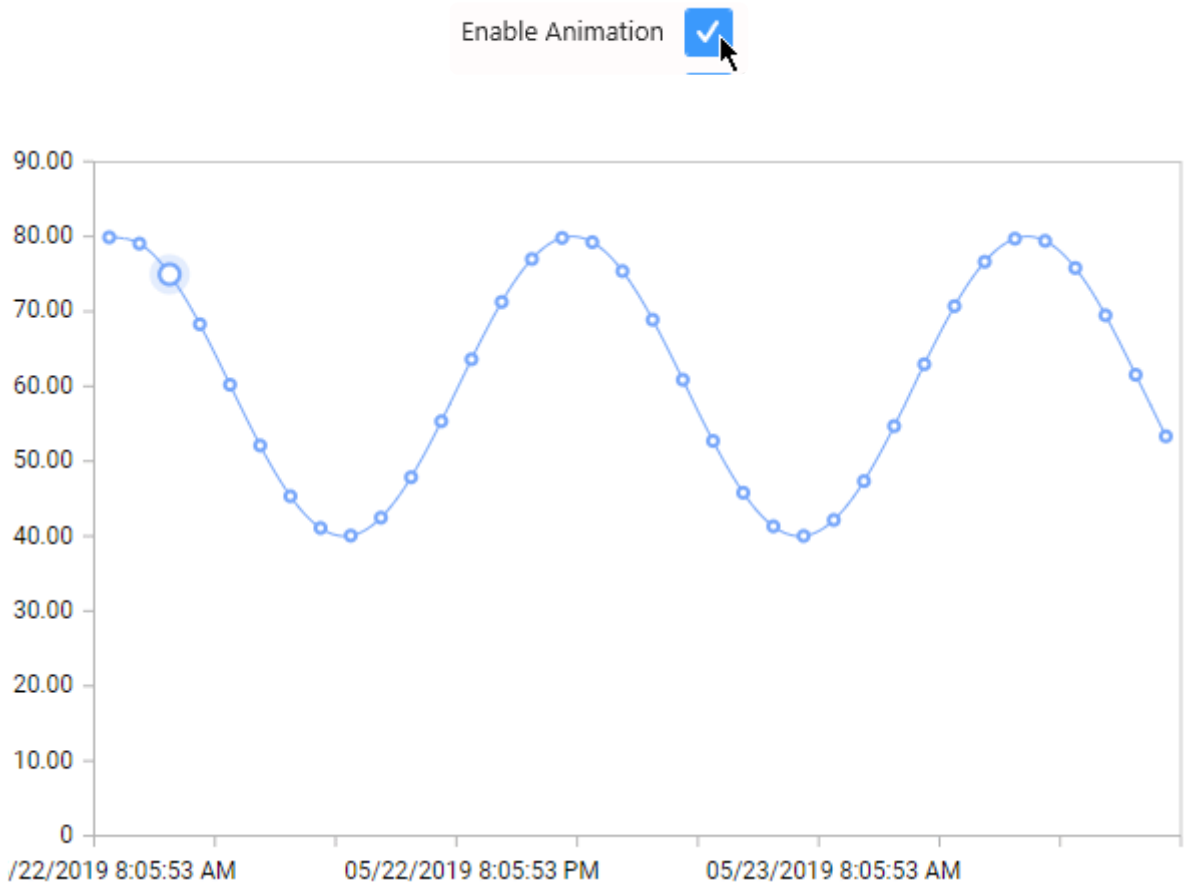


Last modified: Apr 30, 2019

2.7.6.7. Animation

Some Widget types can be configured to show an animation when displayed.

Under *Basic Settings*, check *Enable Animation* :



Last modified: May 24, 2019

2.7.7. Interactions

At runtime, users can click on widgets in order to perform an action depending on what data item is clicked.

In Alpana, three types of actions can be defined on widgets :

- Drill-Down (for the Widget types that support it)
- Link To URL
- Filter

When several of these actions are configured, only one is performed.

The order of precedence is the order in the list above.

Last modified: Apr 30, 2019

2.7.7.1. Drill-Down

Some Widgets define a Drill-Down action : when the user clicks on a data item, the Widget shows the underlying data as if the user were digging into it :

- all types of Cartesian Charts
- all types of Proportion Charts
- TreeMap

To learn to configure Drill-Down, see for each Widget in the Widget Reference.

Last modified: Apr 30, 2019

2.7.7.2. Link To URL

Introduction

Most Widgets can be configured as a link to a URL : when the user clicks on the Widget, the URL is opened.

It is possible to configure the URL to be generated dynamically : when the user clicks on different parts of the Widget, different URLs are open.

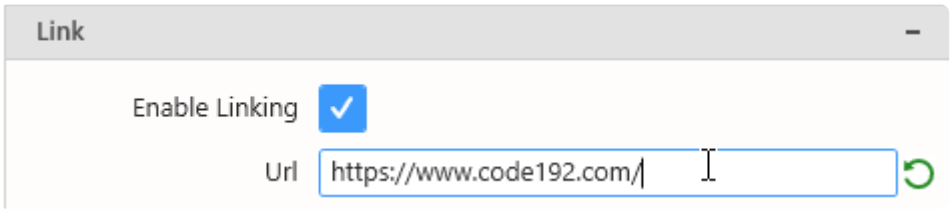
Use cases

This can be used to :

- open another Dashboard with different context (different URL Parameters values) depending on where the user clicked.
- open a different Dashboard depending on where the user clicked
- open a different website depending on where the user clicked
- open another website (for example a reporting solution) with different URL Parameters depending on where the user clicked

Configuring a Link To URL

The Properties panel has a “Link” section. Check the “Enable Linking” checkbox and write the desired destination URL :



With this configuration, when clicking the Widget, the same URL will always be opened.

Syntax for dynamically generated URLs

In the URL string, the special syntaxes below will be replaced by a value dynamically depending on the context on the widget at runtime.

Un-formatted field value

```
{{!MyFieldName}}
```

will be replaced by the corresponding field value depending on where the user clicked on the widget at runtime.

If the field is formatted (for example, a *MyDate* field formatted as *Month Year(MyDate)*), then the replaced value will be the raw value (un-formatted).
For example : `201702`

> This is the value that can be used as input to a URL Parameter.

Example

If the Widget category field is `Month Year( MyDate )`, then you can send its value to another Dashboard with :

```
https://aveva.alpana.io/viewer/dashboard/MyDashboard?MyTargetParameter={{!MyDate}}
```

At runtime, when the user clicks on the part of the Widget where the value corresponds to *February 2017*, then the link opened will be :

```
https://aveva.alpana.io/viewer/dashboard/MyDashboard?MyTargetParameter=201702
```

Formatted field value

```
{{:MyFieldName}}
```

will be replaced by the corresponding field value depending on where the user clicked on the widget at runtime.

If the field is formatted (for example, a *MyDate* field formatted as *Month Year(MyDate)*), then the replaced value will contain the format.
For example : `February 2017`
This is a value that can be used for display.

Example

If the Widget category field is `Month ( MyDate )`, then you can send its value to Wikipedia with :

```
https://en.wikipedia.org/wiki/{{:MyDate}}
```

At runtime, when the user clicks on the part of the Widget where the value corresponds to *February*, then the link opened will be :

```
https://en.wikipedia.org/wiki/February
```

Parameter value

```
${MyParameterName}
```

will be replaced by the current value of the Parameter.
If the Parameter contains multiple values, they are separated by commas ,

Example

If your dashboard contains a Parameter called `MySourceParameter`, then you can send its value to another Dashboard with :

```
https://aveva.alpana.io/viewer/dashboard/MyDashboard?MyTargetParameter=${MySou
rceParameter}
```

At runtime, when the user clicks the Widget, if the value of `MySourceParameter` is currently `Buildin g_A`, then the link opened will be :

```
https://aveva.alpana.io/viewer/dashboard/MyDashboard?MyTargetParameter=Buildin
g_A
```

Examples and tips

Dynamic e-mail creation

In this example, you have a Grid displaying current maintenance issues.
You want to let users click an item in the Grid to open an e-mail to support.

The solution is to use `mailto:` links, with the following syntax :

```
mailto:someone@yoursite.com?cc=someoneelse@theirsite.com, another@thatsite.co
m, me@mysite.com&bcc=lastperson@theirsite.com&subject=Need%20help%20with%2
0...&body=Hello...
```

Except the recipient, all parts are optional.
Of course, with Alpana you can now generate a link like this dynamically from fields in your data !
In the Grid Widget, those fields can even be hidden.

Example :

```
mailto:{{:support_email}}?subject=Issue%20with%20{{:equipment}}&body={{:Downti
me_cause}}%20is%20happening
```

Opening PDF at the desired page

In this example, you have a Grid displaying current maintenance issues.
You want to let users click an item in the Grid to open the correct maintenance document opened at the correct page.

On hosted PDF documents, the syntax is :

```
http://your-site.com/your-document-location.pdf#page=123
```

Of course, with alpana you can now generate a link like this dynamically from fields in your data !
In the Grid Widget, those fields can even be hidden.

Example :

```
http://your-site.com/{{:manual_name}}.pdf#page={{:manual_page}}
```

Last modified: May 29, 2019

2.7.7.3. Master Filter

Widgets can be configured as Filter : when a data item in the widget is clicked, other Widgets only show the corresponding data.

This behaviour is closely related to the notion of Dashboard Parameters and is documented in the corresponding Chapter.

Last modified: Apr 30, 2019

2.8. Widget Properties Reference

Each Widget type has specific properties and configuration options.
This chapter is a reference of all Widget properties.

A Widget is a visualization entity that uses data from a single Data Source and displays it in an interactive way.

Widgets are assembled to create a Dashboard.

> To learn about Widgets and their generic settings, see the corresponding generic chapter : Configuring a Widget.

Last modified: May 24, 2019

2.8.1. Cartesian Chart

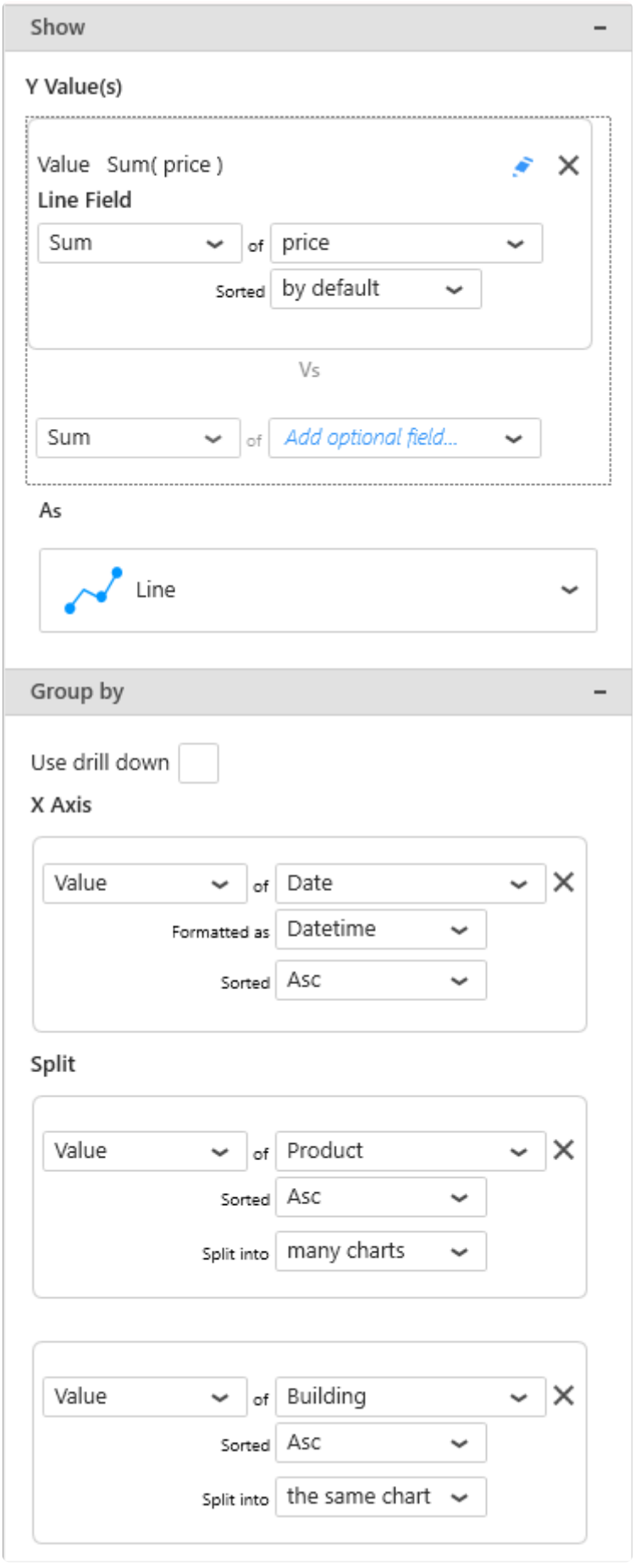
Cartesian Charts allow to view data on a 2D plane using perpendicular X-Y axes.
They are split into several types, but any type of chart can be changed to any other type of chart, and almost all types can be combined together.

Last modified: May 24, 2019

2.8.1.1. Binding Data

Introduction

The following fields are available for configuring data binding on a Cartesian Chart :



- *Y Value(s)* : the height of the chart line along the Y Axis
- *X Axis* : how we group values together on the X Axis to create points on the line

- *Split into the same chart* : how we split each series into different lines on the same chart
- *Split into many charts* : how we split series of different charts in the same container

Show : *Y Value(s)*

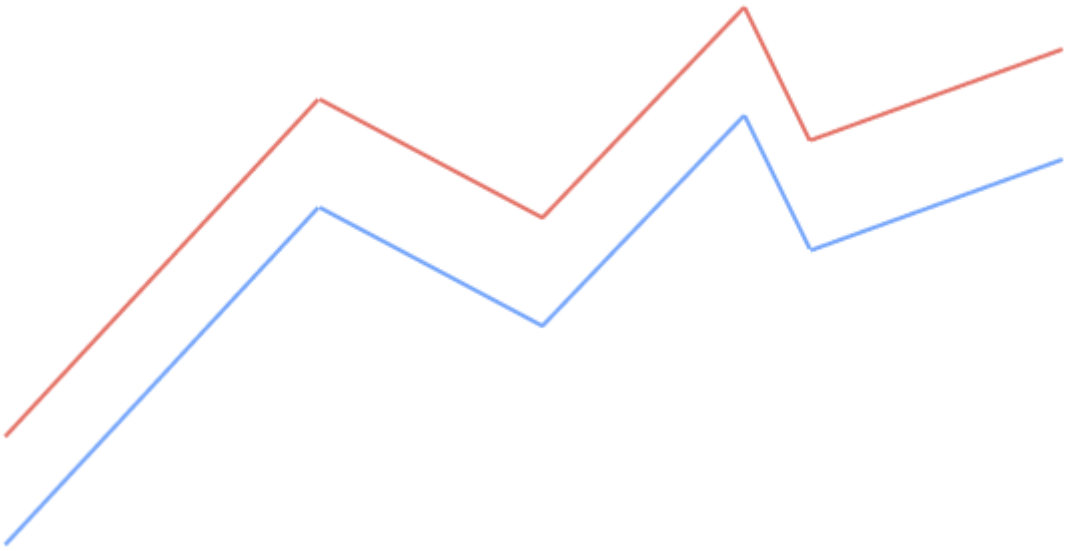
This represents the *quantity* we want to display on the Chart, and will bind for example to the height of the line (Line chart) or the height of the column (Column chart).

These fields **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)
At least one field must be bound.

> Some Chart Types require more than one field to define a point.
> For example the Range Area requires to define one field for each side of the area : Low and High.

Multiple fields : multiple Series

When binding multiple fields at the same time, each field will generate a separate Series.

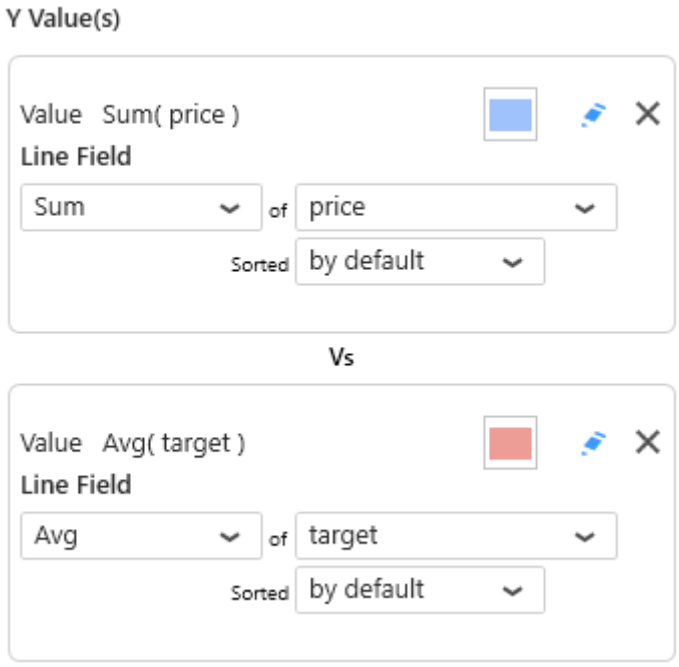


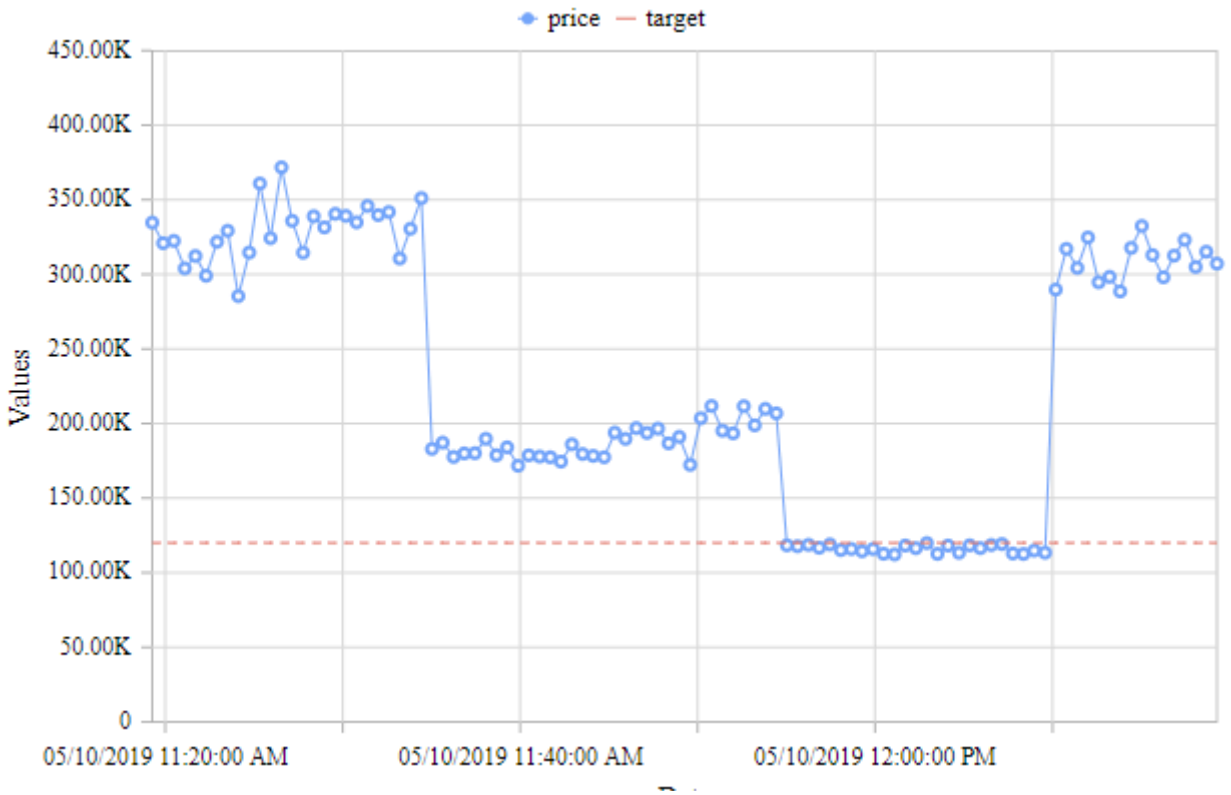
This way, each Series can have its own graphical properties, its own Y axis, etc… and even different Chart Types.
To change the order of fields, see the corresponding chapter.

> Note :
> This only works if your series are already into separate data fields. For example, with a Historian Wide Query.
> If you want to create series based on different values/names from a single data field, see below the "*Split into the same chart*" configuration.

Example

With a field called "Price" and a field called "Target", you will chart 2 separate series, one for each field.





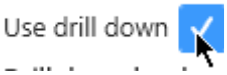
Group by : *X Axis*

This represents the *category* we want to use to separate data points.

These fields **cannot be aggregated** : they must have several separate values in order to group the “Values” aggregation.
At least one field must be bound.

Multiple *X Axis* fields : Drill-Down

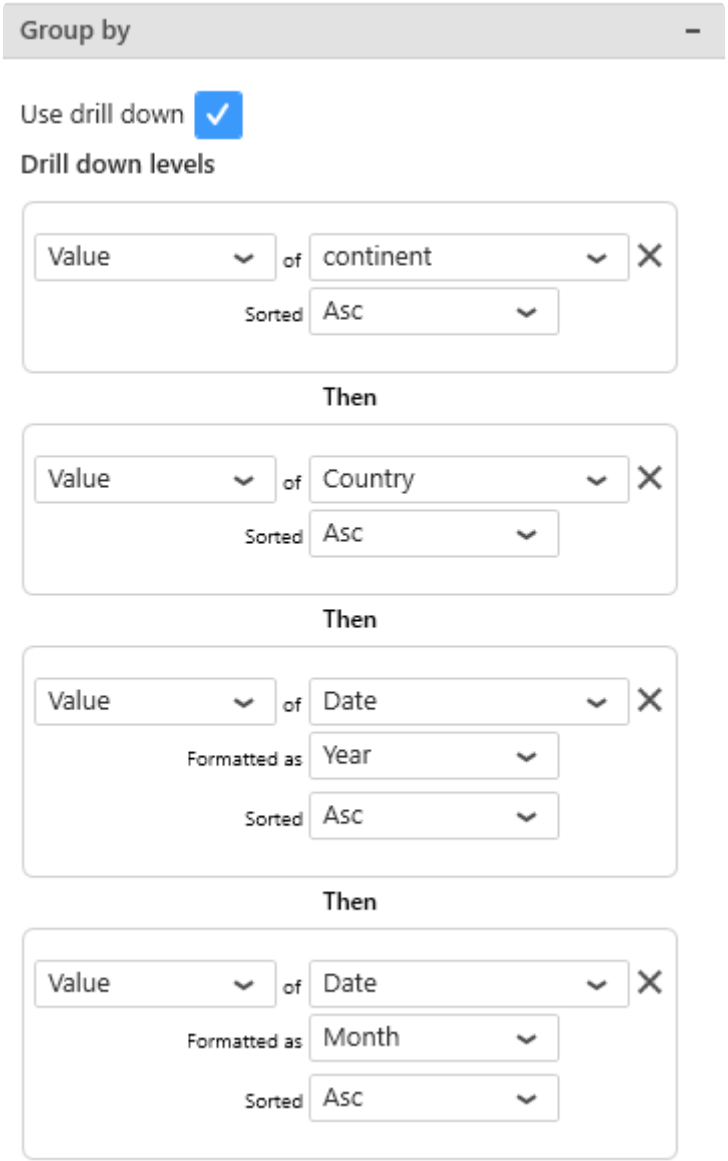
To enable Drill-Down, you can check the box *Use drill down* :



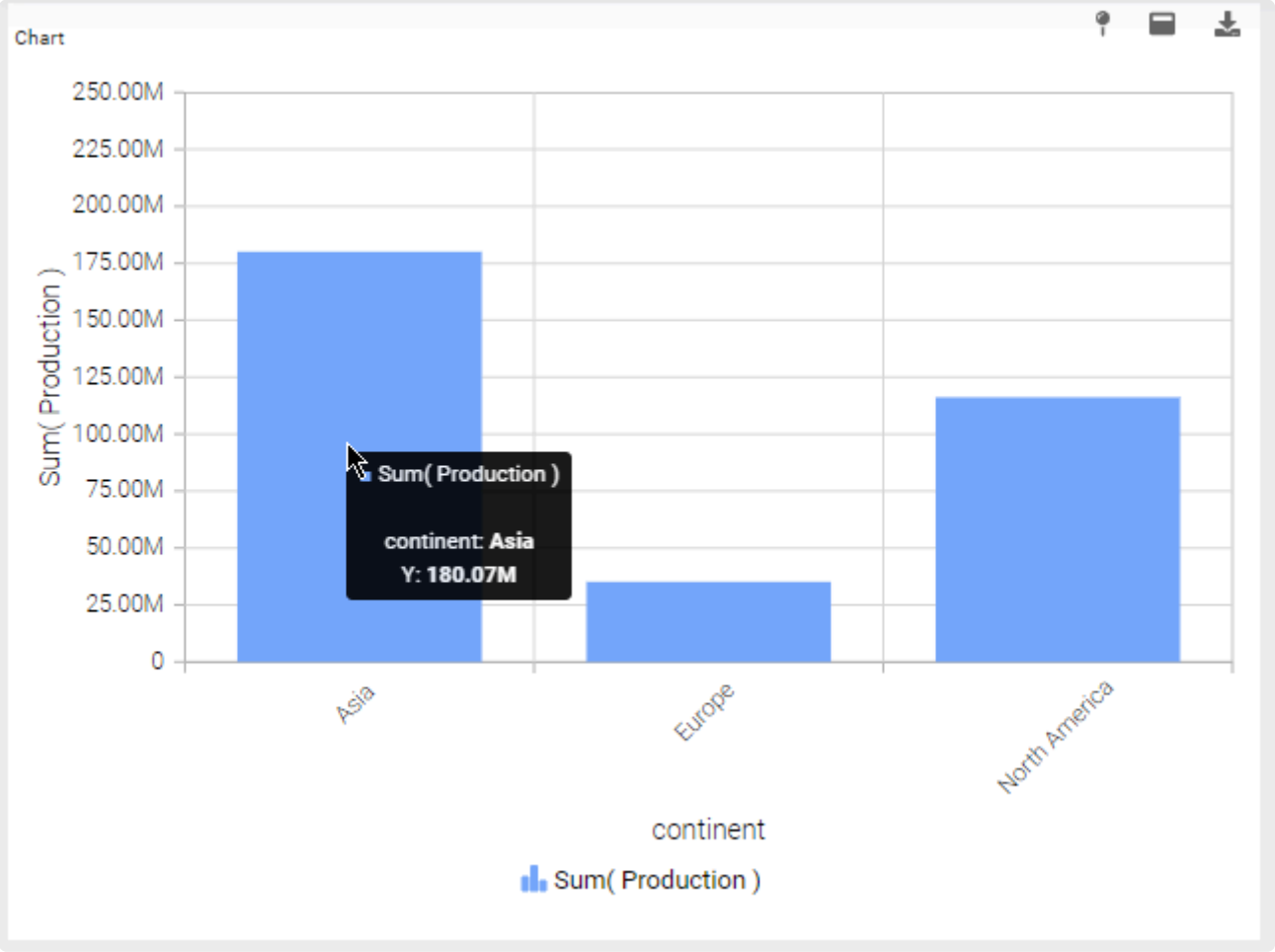
This allows to bind any number of fields to drill-down levels, in the same order as they are in the list. To change the order of fields, see the corresponding chapter.

Example

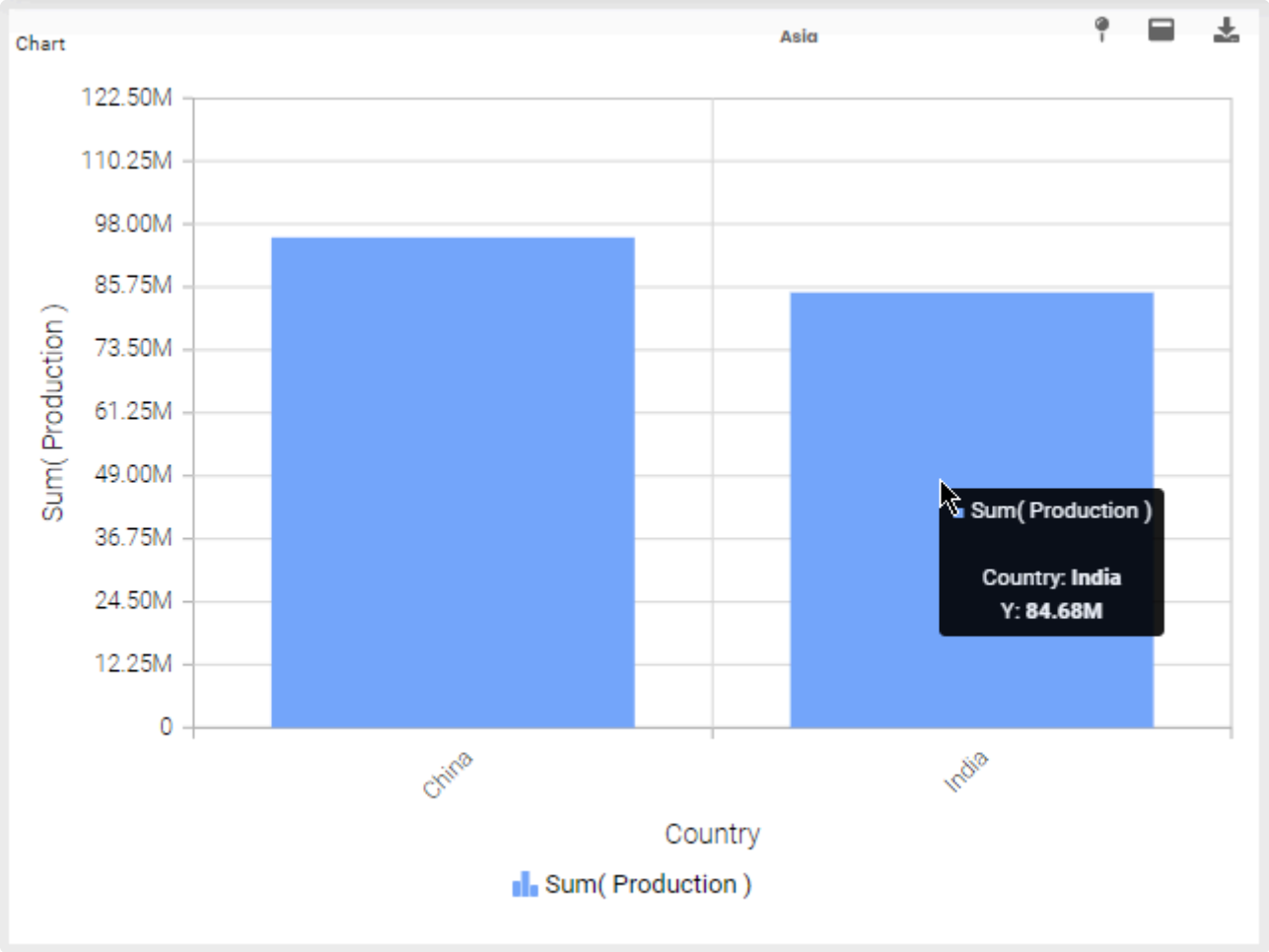
With fields called “continent”, “country”, and “Date”, the following configuration will configure the corresponding drill-down :



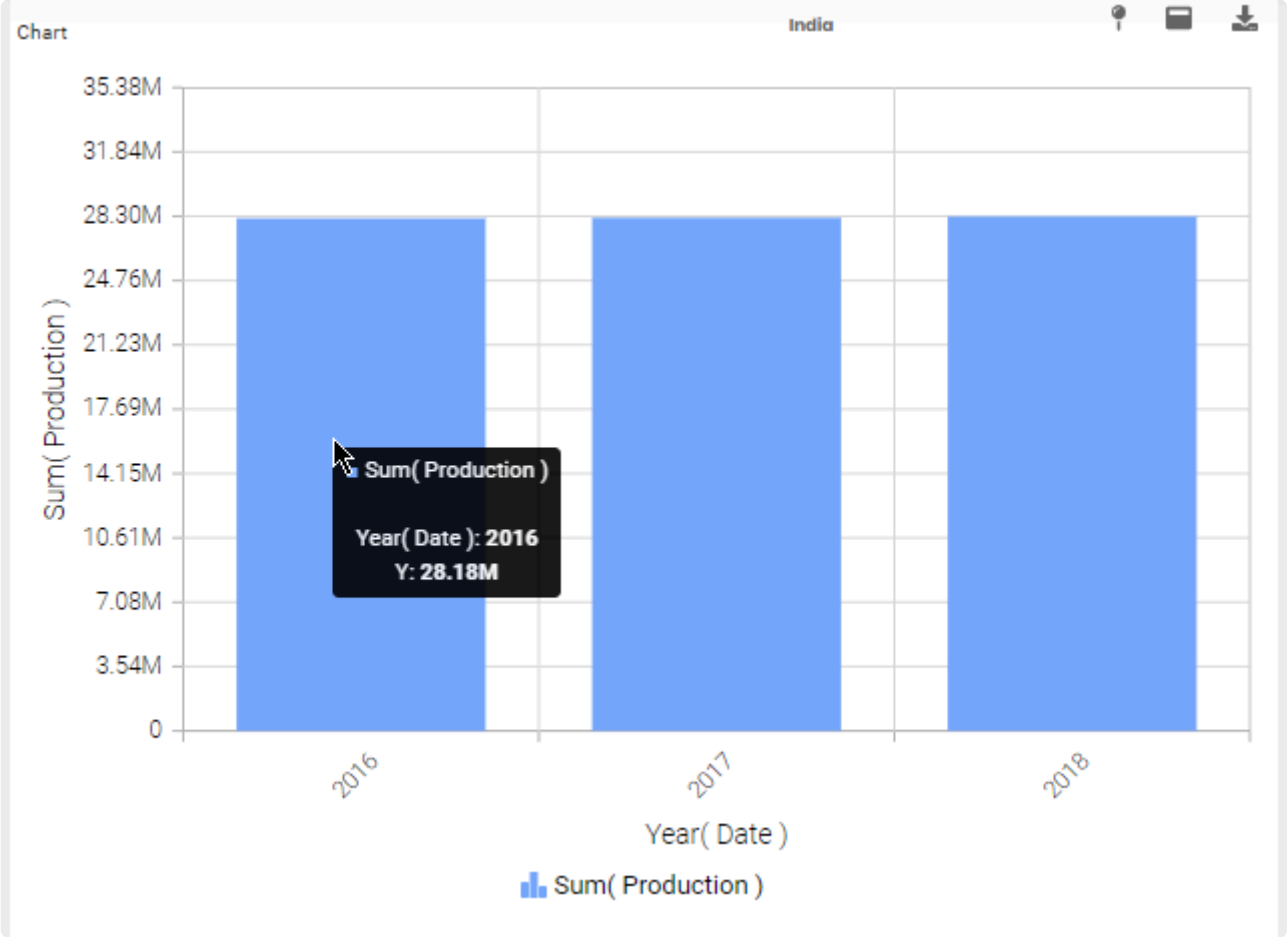
First, the data will be grouped by “continent” :



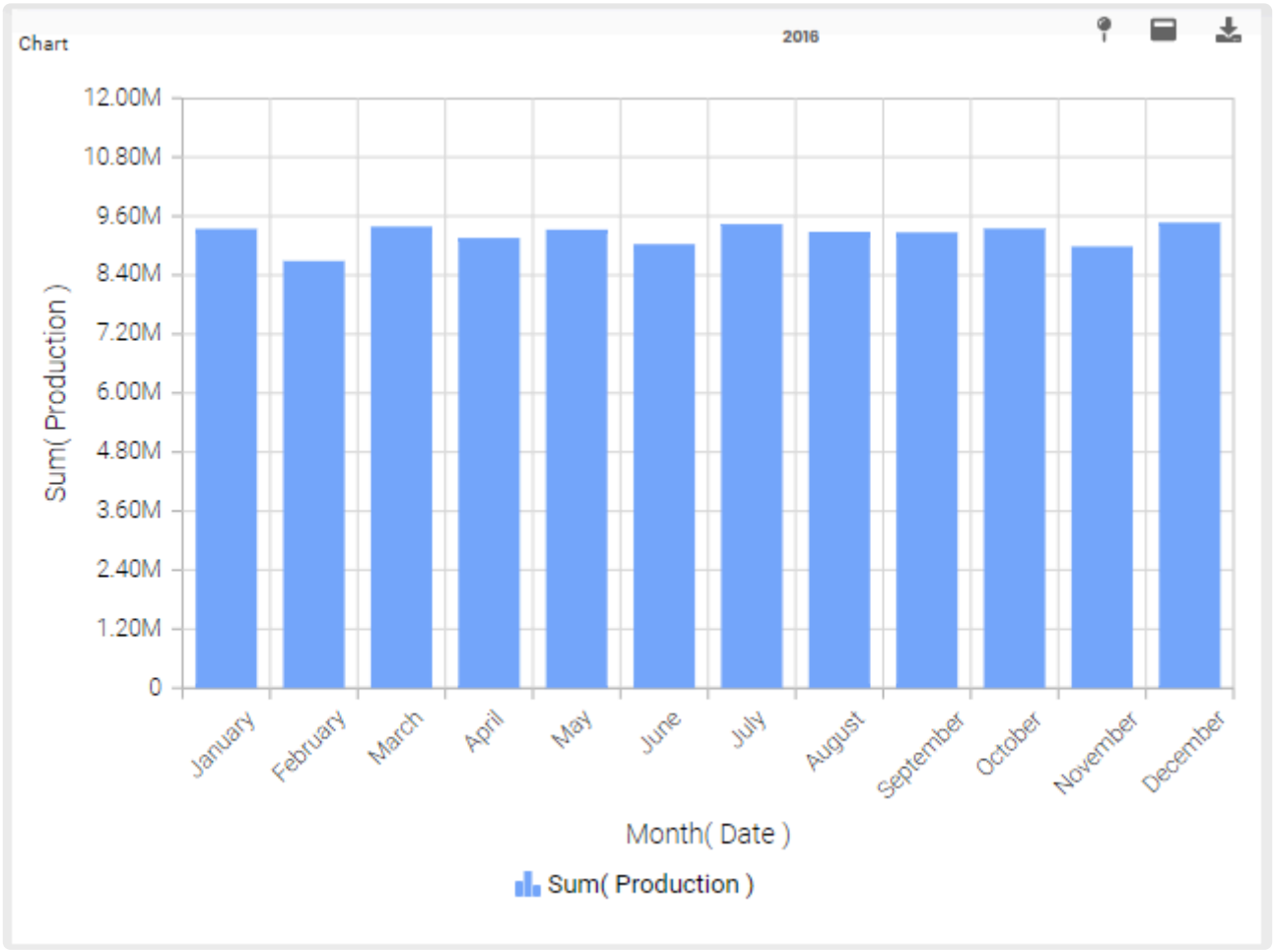
When the user clicks a continent (example : Asia), the Chart is drilled-down to the next level : the Countries of Asia :



Next, when the user clicks a Country (example : India), the Chart is drilled-down to the next level : the Years of India :



Next, when the user clicks a year (example : 2016), the Chart is drilled-down to the last level : the Months of 2016 of India :



Meanwhile, the Drill-Down navigation at the top shows the value of the current level.
When clicking it, you can drill back up to any level, or directly to the top by selecting *Close* :

Group by : *Split into the same chart*

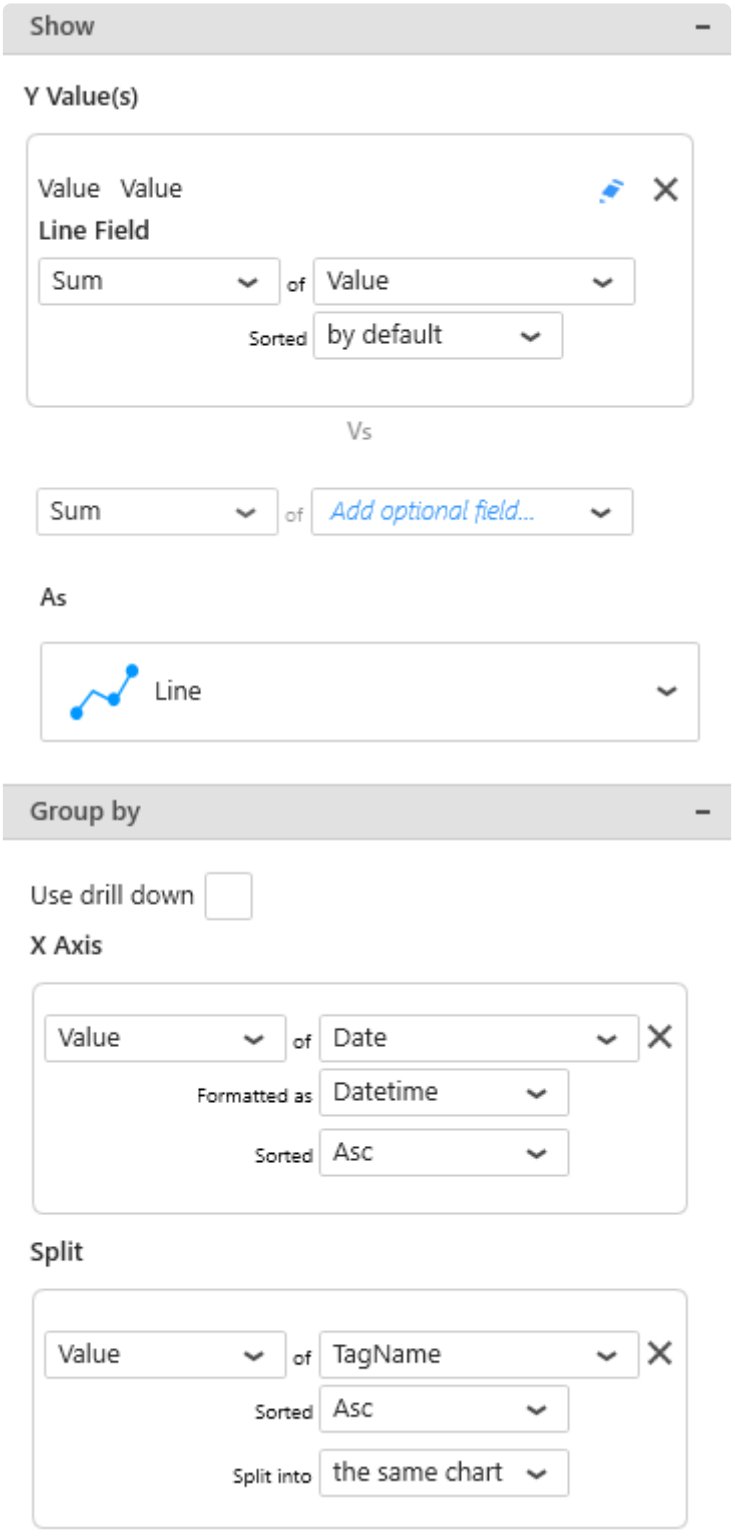
This represents the series field to generate a different chart series for each different value in this *Split into the same chart* field.

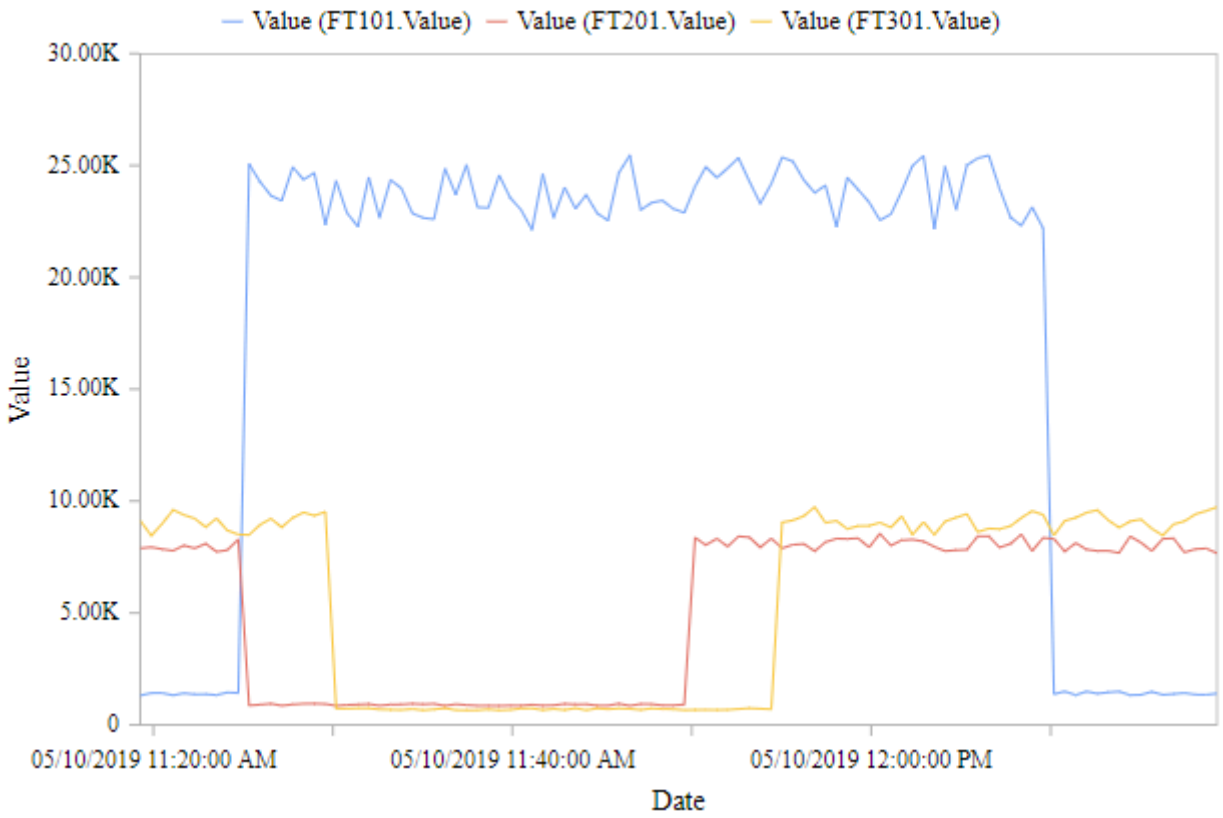
This field is optional.
It can be combined with *Split into many charts*.

> Note :
> This only works if your series are values into a single data field. For example, with a Historian Narrow Query.
> If you want to create series based on separate data fields, see above the “multiple Series” configuration.

Example

If you have Historian Narrow Query with a field called “TagName”, you may want to plot a separate Line chart for each value of TagName :


Show
Y Value(s)
Value Value
Line Field
Sum of Value
Sorted by default
Vs
Sum of Add optional field...
As
Line
Group by
Use drill down
X Axis
Value of Date
Formatted as Datetime
Sorted Asc
Split
Value of TagName
Sorted Asc
Split into the same chart



Group by : *Split into many charts*

This represents the series field to generate a different contained Chart for each different value in this *Split into many charts* field.

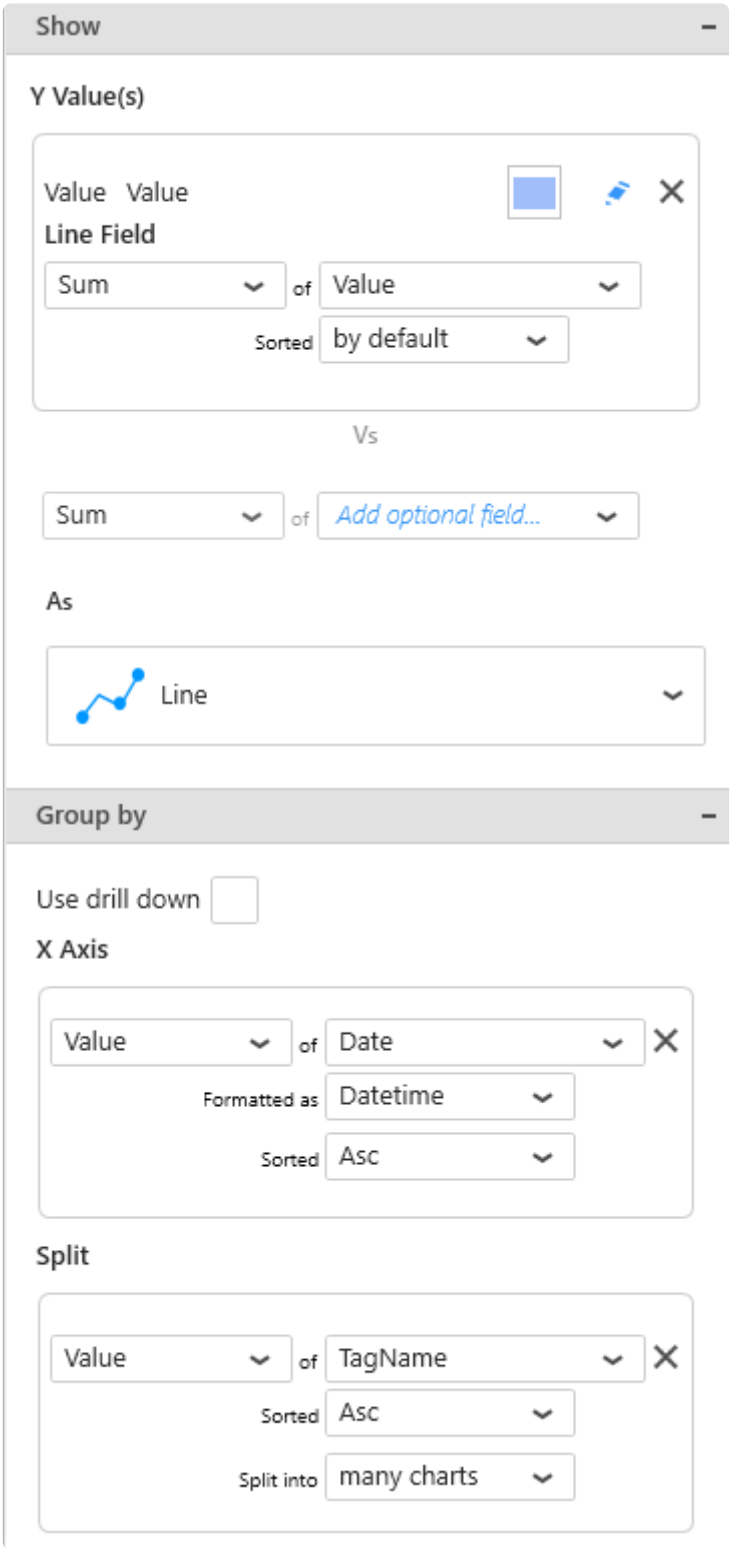
This will turn the widget into a container, and generate a complete different chart for each value in the series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

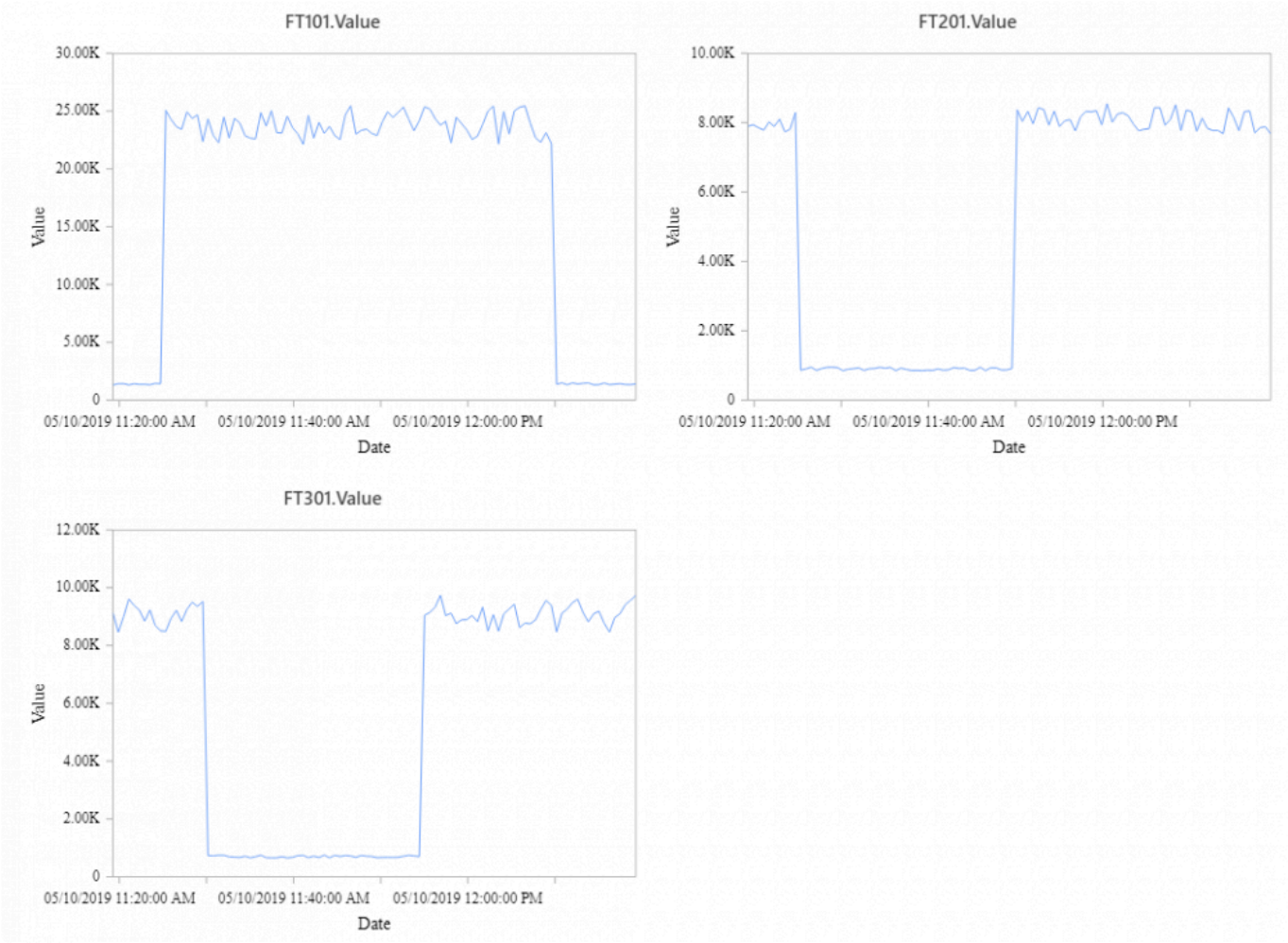
This field is optional.
It can be combined with *Split into the same chart*.

> Note : This only works if your series are values into a single data field.

Example

If you have a field called “continent”, you may want to plot a completely separate chart for each value of continent :


Show
Y Value(s)
Value Value
Line Field
Sum of Value
Sorted by default
Vs
Sum of Add optional field...
As
Line
Group by
Use drill down
X Axis
Value of Date
Formatted as Datetime
Sorted Asc
Split
Value of TagName
Sorted Asc
Split into many charts



Last modified: May 16, 2019

2.8.1.2. Chart Types

Categories

There are two main categories of Chart types :

- Horizontal X Axis : the grouping axis (X) is displayed from left to right
- Vertical X Axis : the grouping axis (Y) is displayed from bottom to top

Types

Cartesian Charts are split into several Widget types, listed below.

In addition, a special shortcut exists for combining different chart types :

Combo

Horizontal X Axis

1 Value Field

- Column
- Stacked Column
- 100% Stacked Column
- Area
- Stacked Area
- 100% Stacked Area
- Line
- Spline
- Spline Area
- Step Line
- Step Area

- Scatter
- Waterfall

2 Value Fields

- Bubble
- Range Area
- Range Column
- Hi/Lo

4 Value Fields

- Hi/Lo Open/Close
- Candle

Vertical X Axis

1 Value Field

- Bar
- Stacked Bar
- 100% Stacked Bar

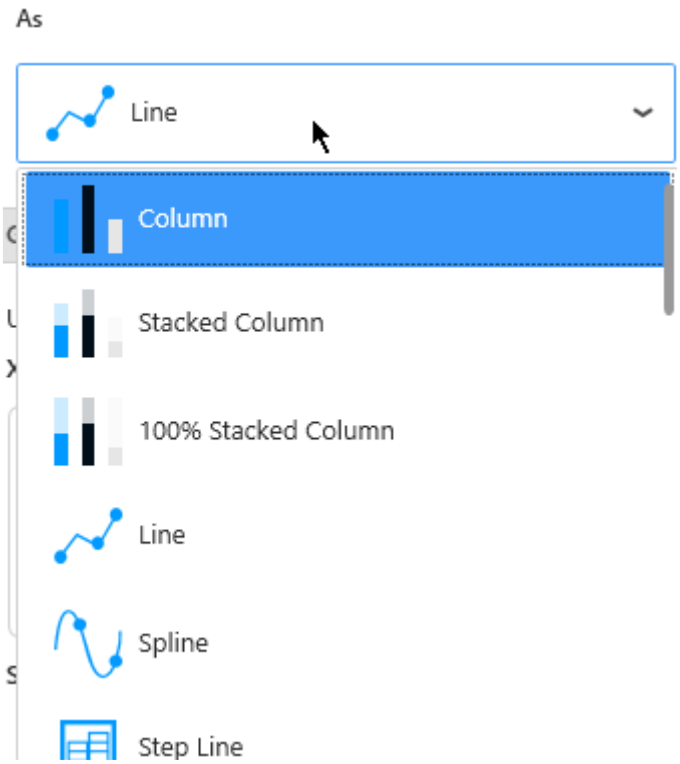
Last modified: May 16, 2019

2.8.1.3. Switching Chart Types

When adding a Widget, the Chart will have a single type globally : all series will have the same type.

Chart type can be changed on the fly while keeping the widget configuration.

To change the Chart type, use the selector under *As* in the data configuration :



Last modified: May 16, 2019

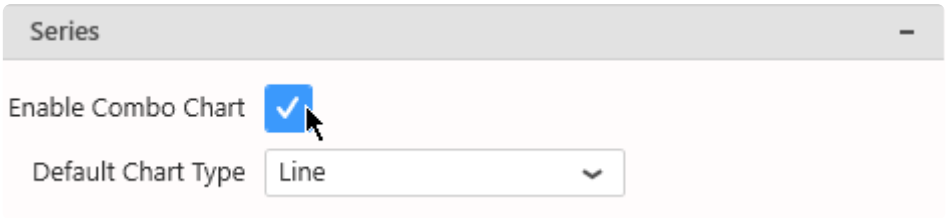
2.8.1.4. Combining Chart Types

Enabling Combo Chart

Any Chart can be turned into a Combo Chart.
This allows to combine multiple Chart Types into the same widget.

> ! *Horizontal X Axis* types of Charts cannot be combined with *Vertical X Axis* types.
> Example : a Bar and a Column cannot be combined together since they don't have the same orientation of the X Axis.

Click *Enable Combo Chart* under *Series* :

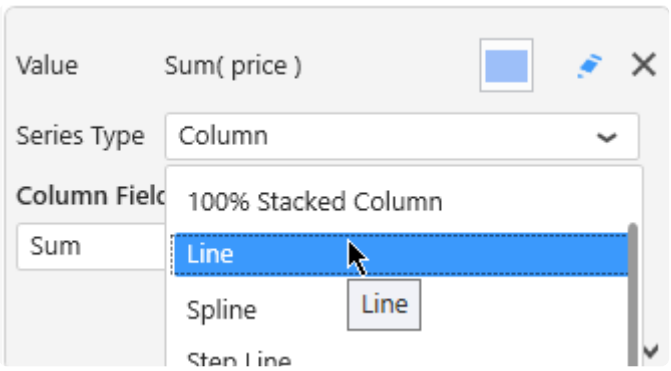


The *Default Chart Type* can be selected under *Series* also.
This will be the Chart type used by default when adding new *Y Value* Series to the data configuration.

In addition, a shortcut exists to create a Combo Chart immediately : the widget type Combo.

Selecting Types

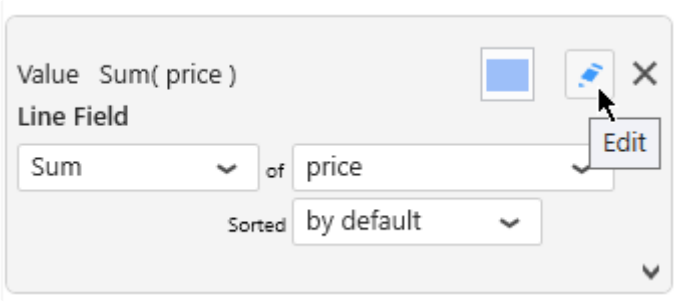
When Combo Chart is enabled, the Chart Type is no longer a global setting to the widget.
Instead, each *Y Value* Series has a new setting called *Series Type* :



Last modified: May 16, 2019

2.8.1.5. Series Settings

Each *Y Value* Series can be configured individually by clicking the pen icon next to its name :



This allows to fine-tune your chart by configuring exactly what you want on each series inside the widget.

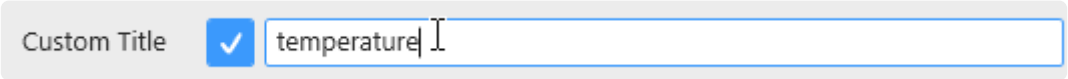
Keep in mind that some configuration options are not available for all chart types.

Last modified: May 24, 2019

2.8.1.5.1. Series Name

Each Series has a name displayed in various places.
By default, this name is automatically generated from the data field bound to the Series.

To configure this name, edit the Series Settings, in tab *Main*, check *Custom title* and write the desired name :

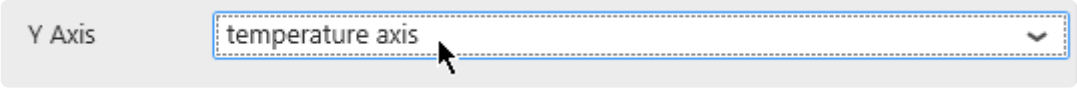


Last modified: May 24, 2019

2.8.1.5.2. Axis binding

Each Series can be bound to a different Y Axis.

Edit the Series Settings, in tab *Main*, select the desired *Y Axis*:



Listed are all the available Y axes.

Last modified: May 24, 2019

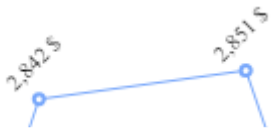
2.8.1.5.3. Data Point information

Value Labels

Text labels can be used to display the value next to each data point.

Edit the Series Settings, in tab *Advanced*, check *Value Labels* :

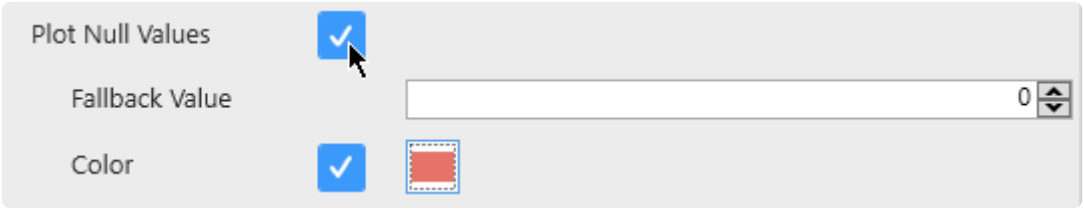
With the following result for example :



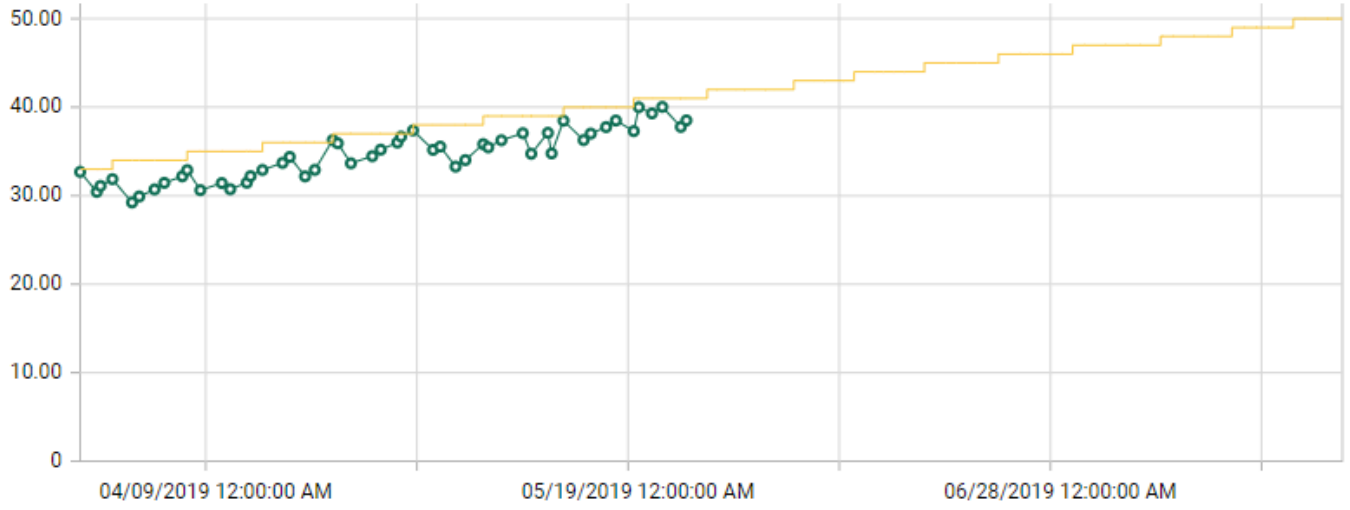
Null values

By default, when the *Y Value* is NULL, the chart doesn't display any shape.
Instead, you can force to display a point with a predefined value for NULL values.

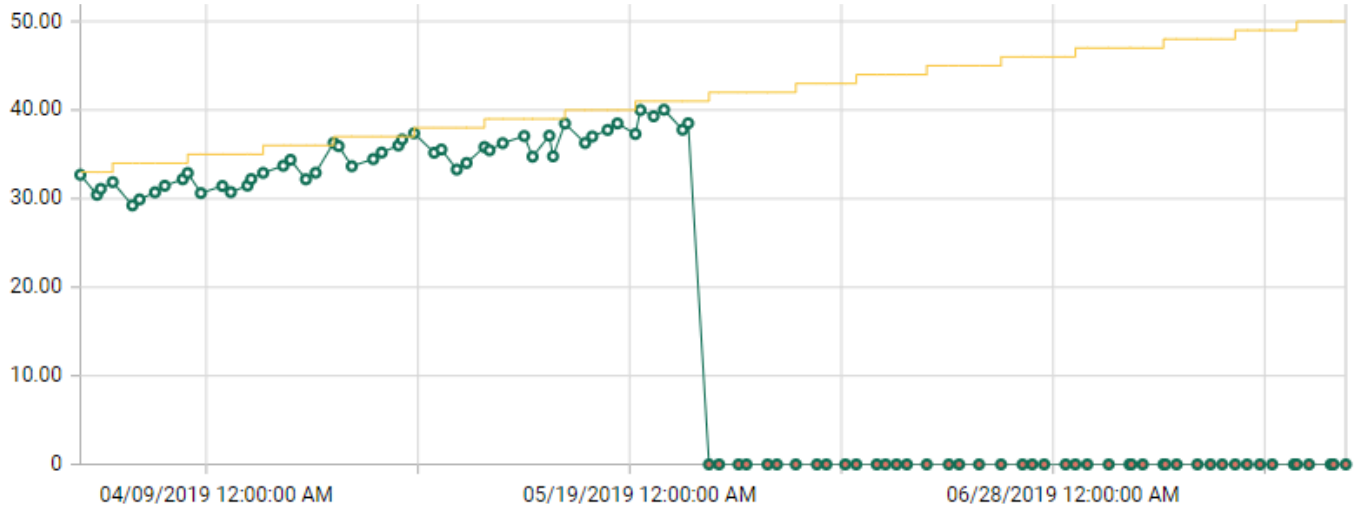
Edit the Series Settings, in tab *Advanced*, check *Plot Null Values* :



For example, in the following chart, there are NULL values after a certain date (data unknown) :



Instead of a broken line, you may choose to display those unknown values as 0 with a red point :

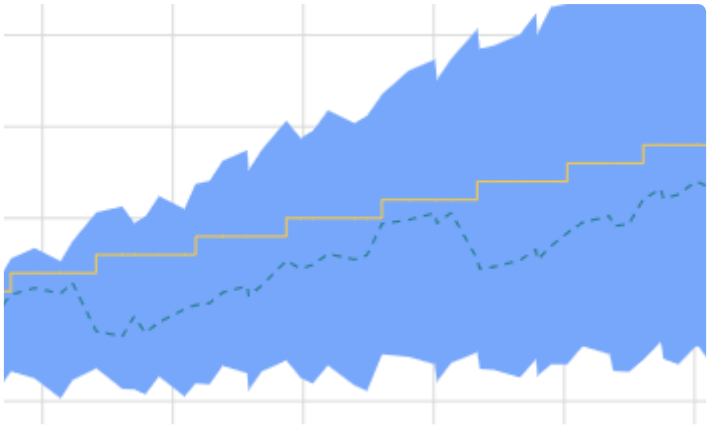


Last modified: May 24, 2019

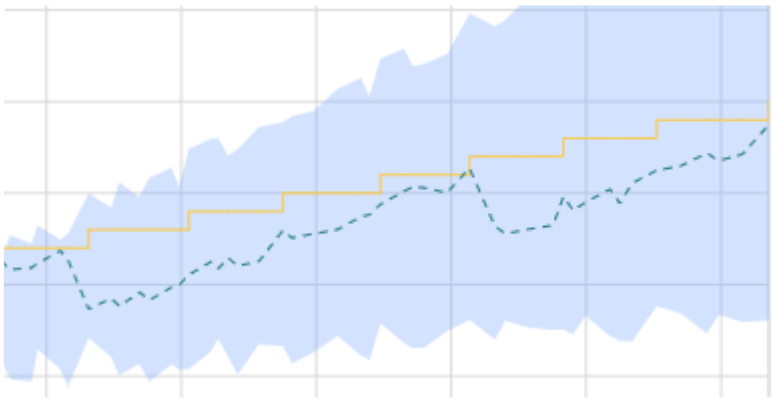
2.8.1.5.4. Style and Appearance

Opacity

Each chart series can be made semi-transparent to allow to see the other series behind it.
The *Opacity* of the Series can be configured in tab *Advanced* :



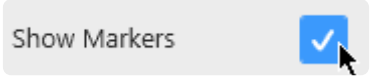
… will turn into :



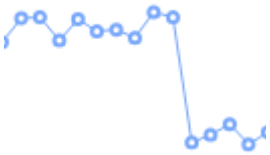
Markers

Some Chart types can display markers on the series.

Edit the Series Settings, in tab *Advanced*, check *Show Markers* :

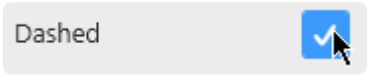


… will turn into :

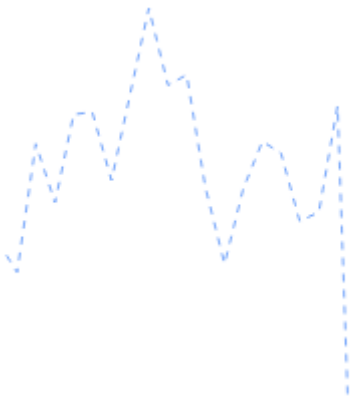


Dashed

Some Chart types can display a dashed line instead of a continuous line :



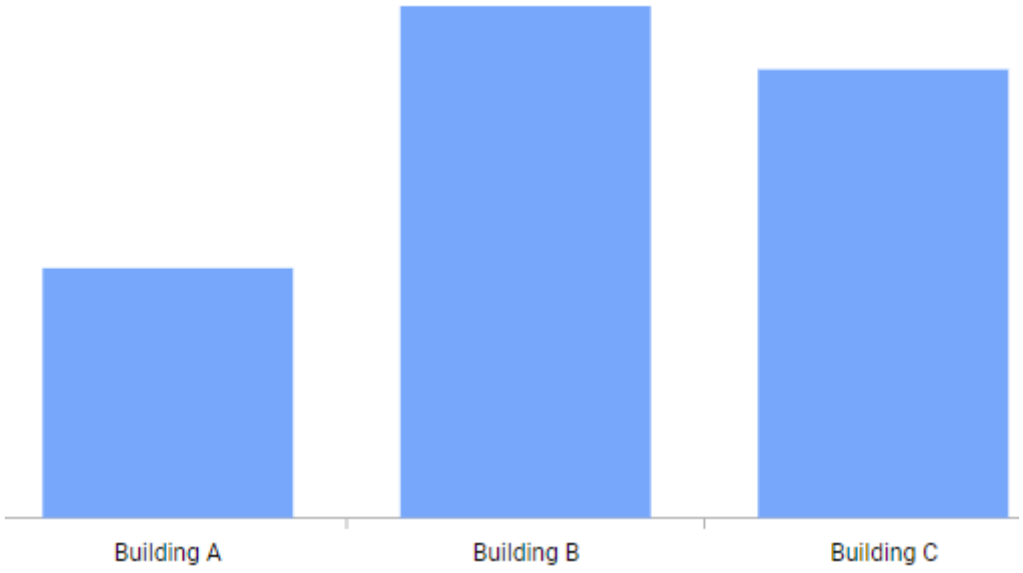
… will turn into :



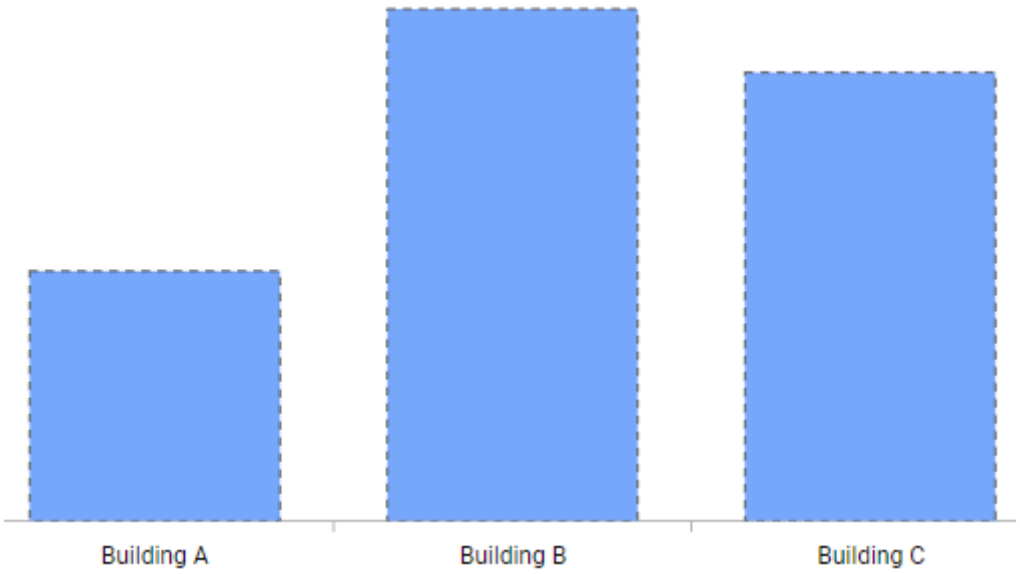
Border

Some Chart types can have a border around the shapes :

Border Color : select a color from the color picker
Border Thickness : give a value greater than zero to see the border, increase to have a thicker border
Border Dashed : if the border exists (Thickness > 0), you can display it as a dashed line instead of continuous



… will turn into :



Spline Type

Some Chart types are based on a Spline, which display an interpolated line.
The *Spline Type* controls the interpolation type depending on the calculation made :

- *Natural*
- *Monotonic* (Note : this style doesn't overshoot : it doesn't go over or under the actual values)
- *Cardinal*
- *Clamped*

Last modified: Jun 19, 2020

2.8.1.6. Axes

Since Alpana v3.0, Charts have 2 types of axes :

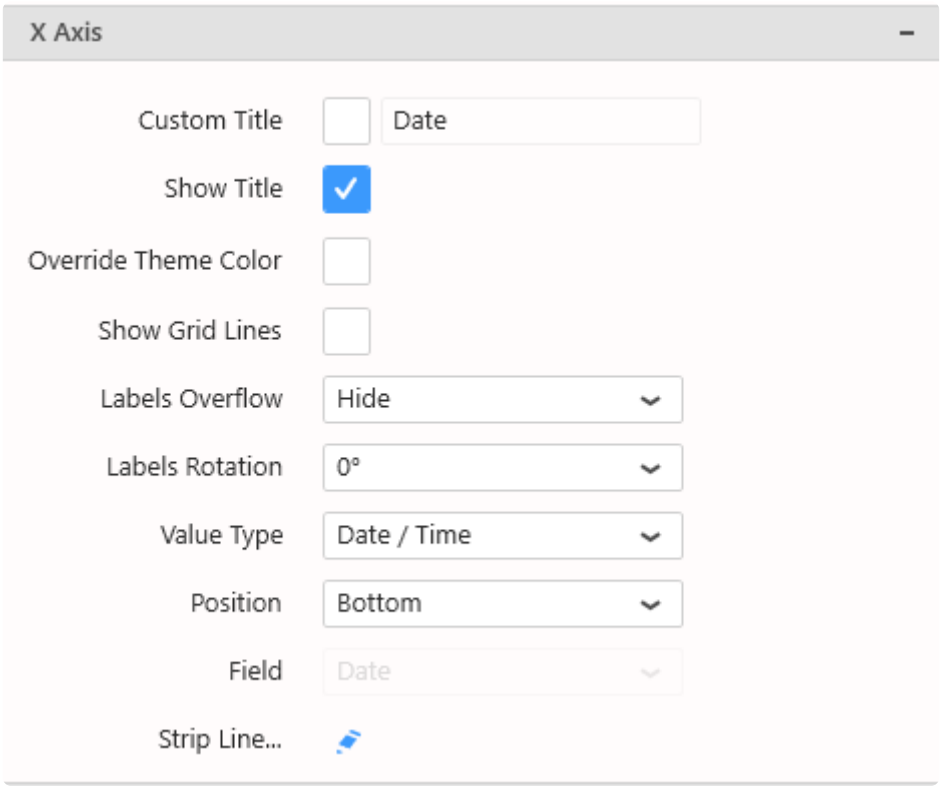
- one X Axis : for displaying values in the *X Axis* field. It's the grouping axis : grouping categories are displayed along this axis.
- one or more Y Axis : for displaying values in the *Y Values* Series fields. It's the value axis : aggregated values are displayed along this axis.

Each axis type has different configurations described next.

Last modified: Jun 17, 2019

2.8.1.6.1. X Axis

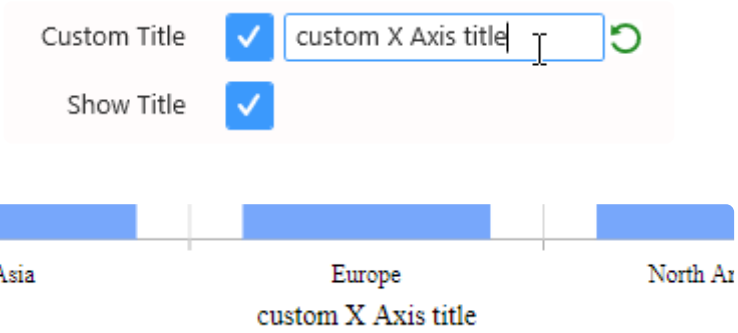
X Axis configuration is located in the right side in the properties, under section *X Axis* :



Title

X Axis has a default title generated from the name of the currently displayed data field.

If you wish to customize this title, check the box *Custom Title*, and write a new title :



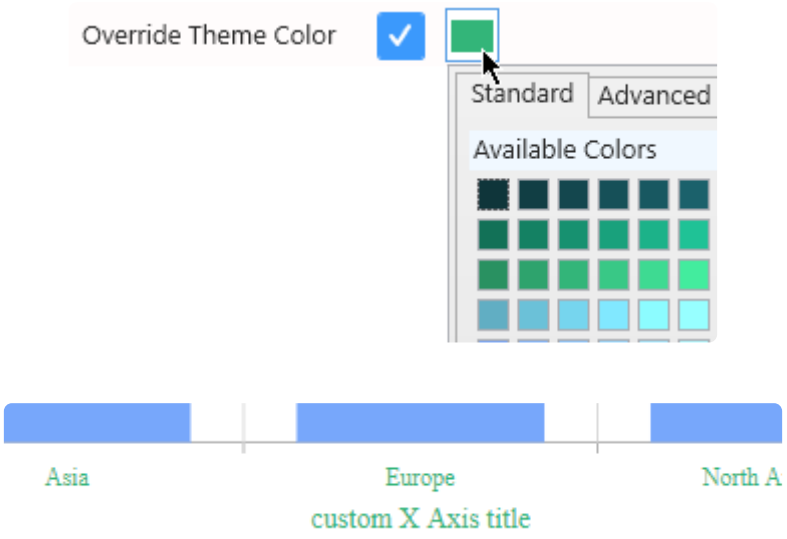
> ! When using Drill-down, the X Axis Title changes dynamically at each drill-down level to display the current X Axis field name.
> If you set a *Custom Title*, it will be the same at all Drill-down levels.

Hiding Axis Title

Use the *Show Title* check box to hide or show the title on the axis.

Color

By default, the Labels and Title colors are based on the Theme.
If you want to set a specific different color, check *Override Theme Color* and set the desired color :



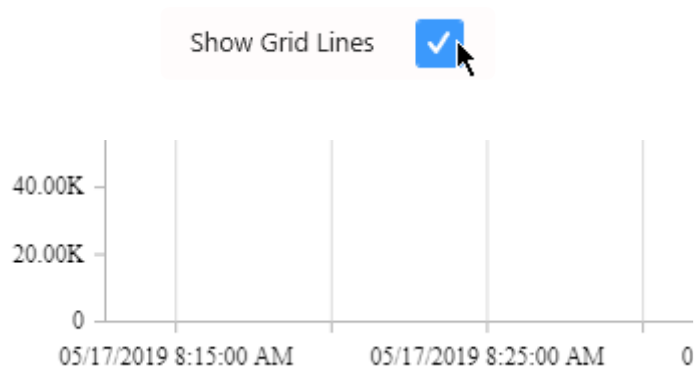
Axis Ticks and Labels

The Axis displays ticks at regular intervals with the X value of the corresponding position displayed as a text label.

Grid Lines

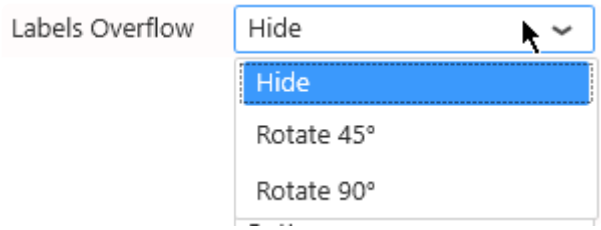
The Axis can display “Grid Lines” aligned with the Ticks, to let users have visual marks of what is the X value at this position.

Grid lines can be hidden or displayed with *Show Grid Lines* :



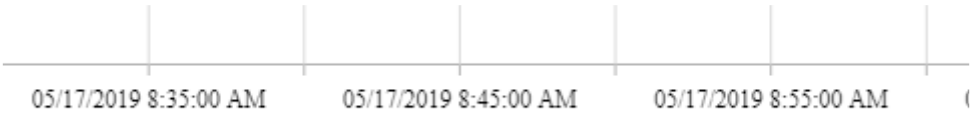
Labels Overflow

When Axis Labels are too long to be displayed, the following behaviors can be configured using *Labels Overflow* :



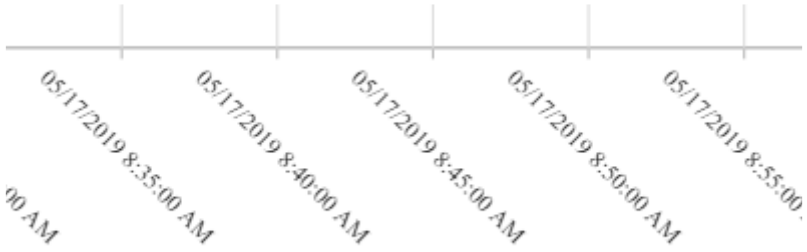
Hide

Labels are displayed entirely in their original orientation, but the labels that overlap are hidden :



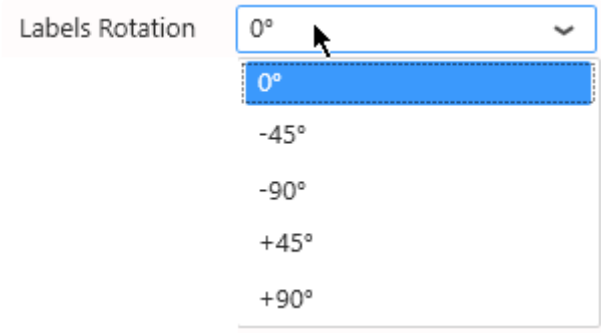
Rotate

Labels are displayed entirely, but if the labels overlap they are rotated to give more space :

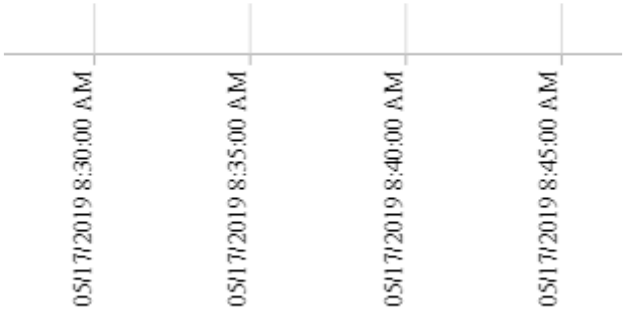


Labels Rotation

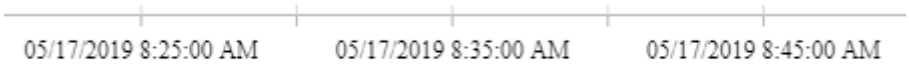
The Axis Labels orientation can also be rotated for better readability, by setting *Labels Rotation* :



It can allow for example to fit more labels when using the “Hide” Overflow Mode :



Or alternately to take less space vertically :



Ticks interval

On *Date / Time* type of Axis, by default, the interval between Ticks on the Axis is calculated automatically (depending on the widget size, value range, labels size, etc)
This allows to have a responsive behavior : the ticks will always fit nicely.

Instead of letting the interval be calculated automatically, you can input a *Custom Interval* :

This will force the the Tick interval to the value selected.

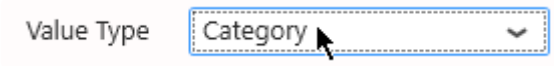
> Note : this is only possible on a *Date / Time* type of Axis (see below)

> Note : this means of course that the interval is no longer calculated automatically.
> As a consequence, on some screens the labels will overlap or otherwise be hard to read.

Value Type

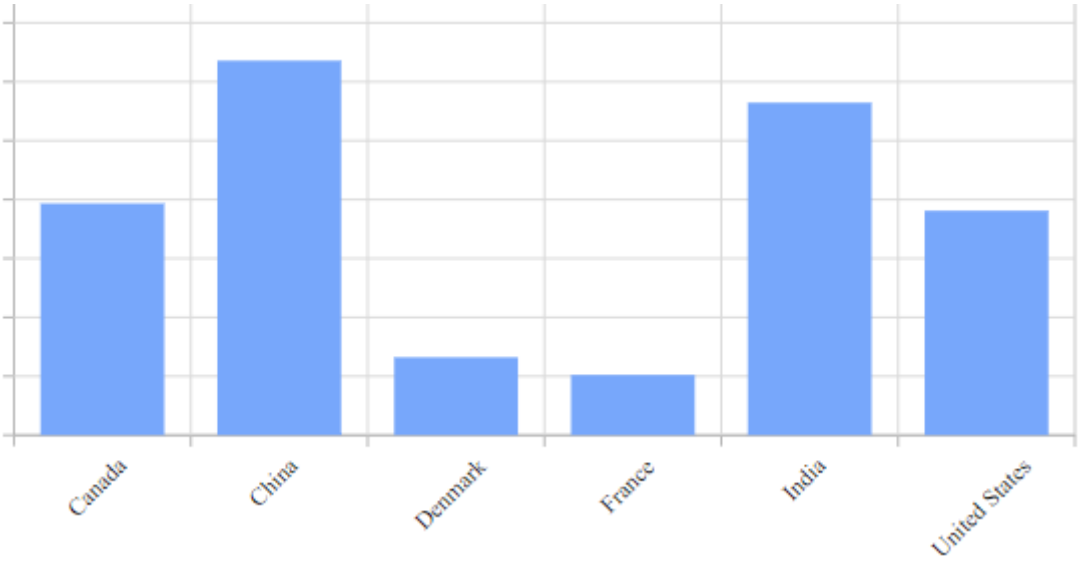
Sometimes you want to compare groups of values side by side, and the distance on the X Axis doesn't matter.
Other times, the distances on the X Axis are meaningful and you want to show how close or far the points are horizontally.

This is what the X Axis *Value Type* setting is for :



Category

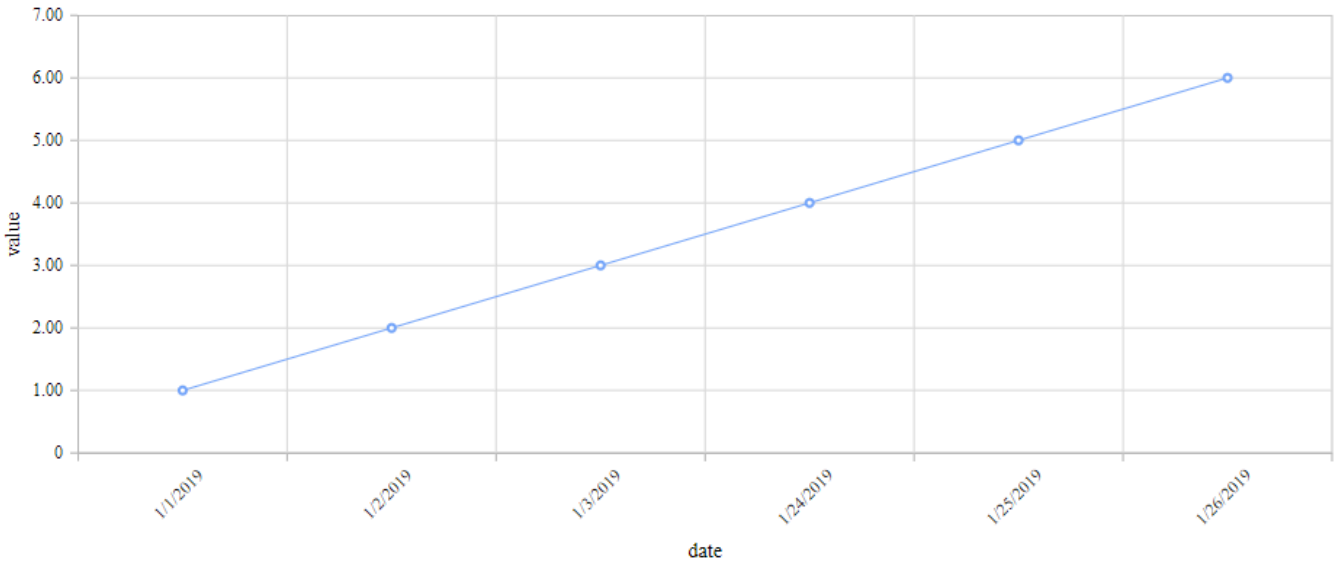
This is the default, and the only option available for text type of fields.
With *Category* X Axis type, all items on the X Axis are equally spaced.



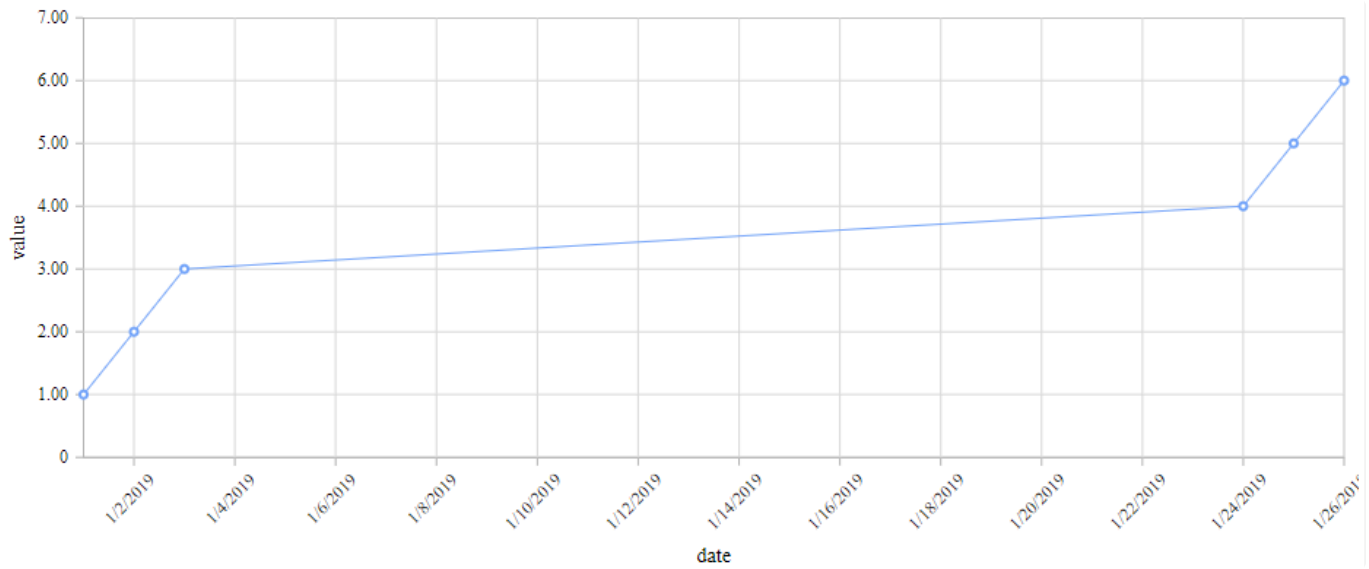
Date / Time

When the data field is of type Date, DateTime or Time, the *Date / Time* option is available.
The X position of points depends on the time difference in the data.

For example, the following data has missing points between day 3 and day 24 but this is not clearly visible and the progression looks linear :



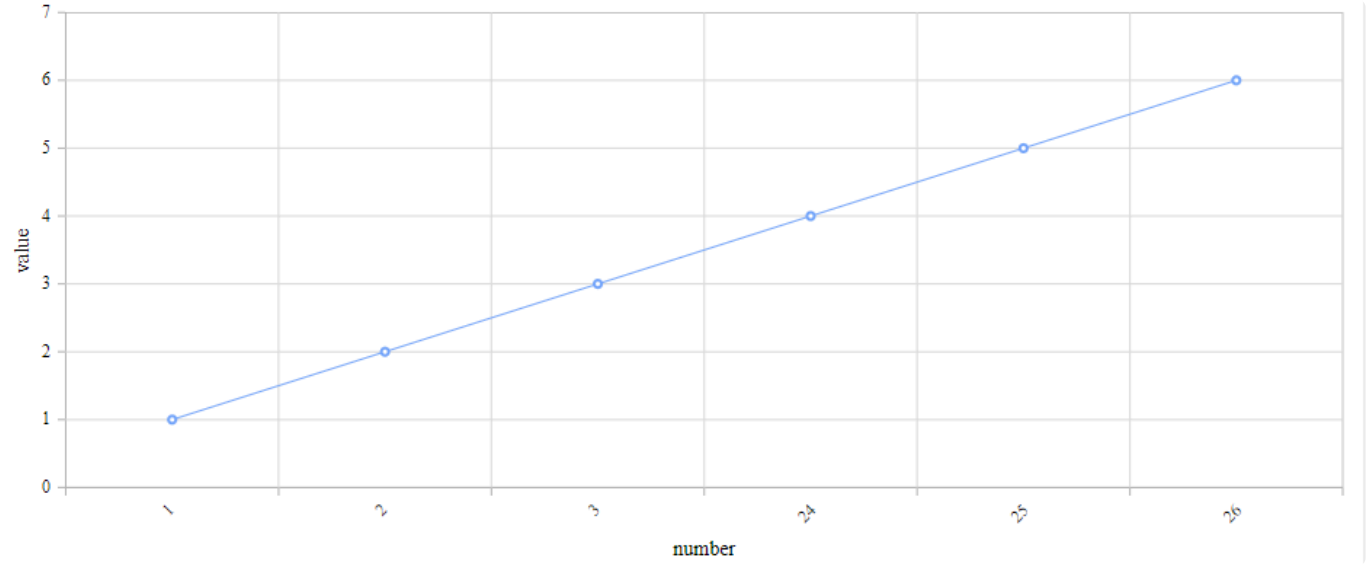
With *Date / Time* type, the real trend is visible :



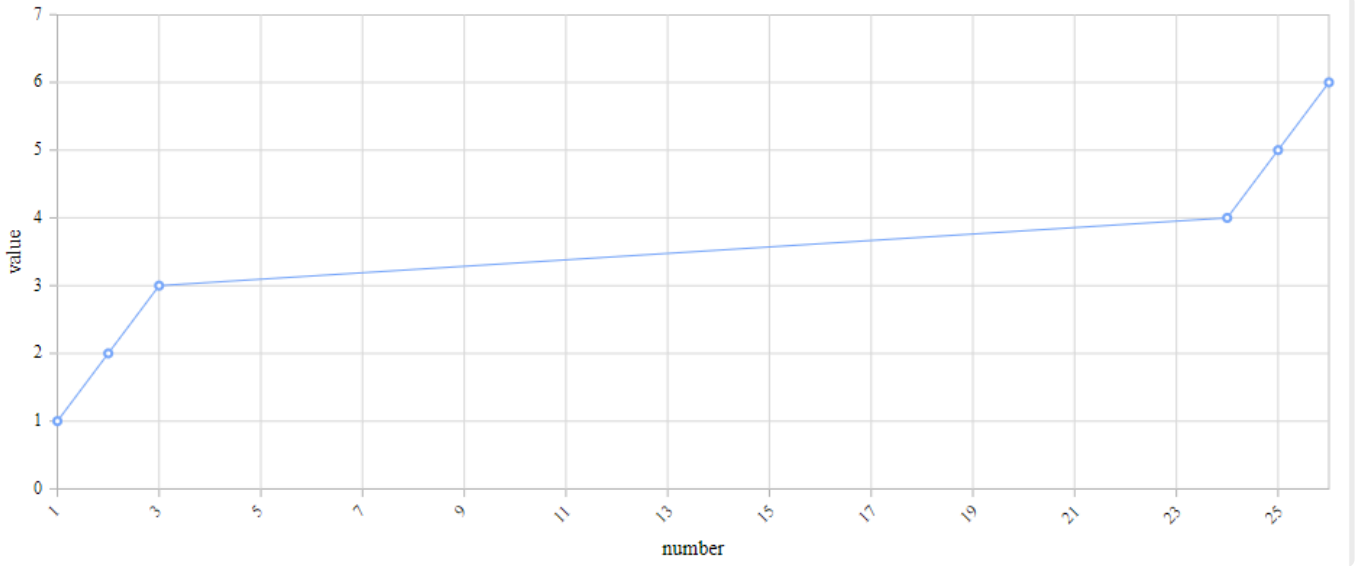
Numeric

When the data field is of type Number, the *Numeric* option is available.
The X position of points depends on the number difference in the data.

For example, the following data has missing points between X value 3 and X value 24 but this is not clearly visible and the progression looks linear :



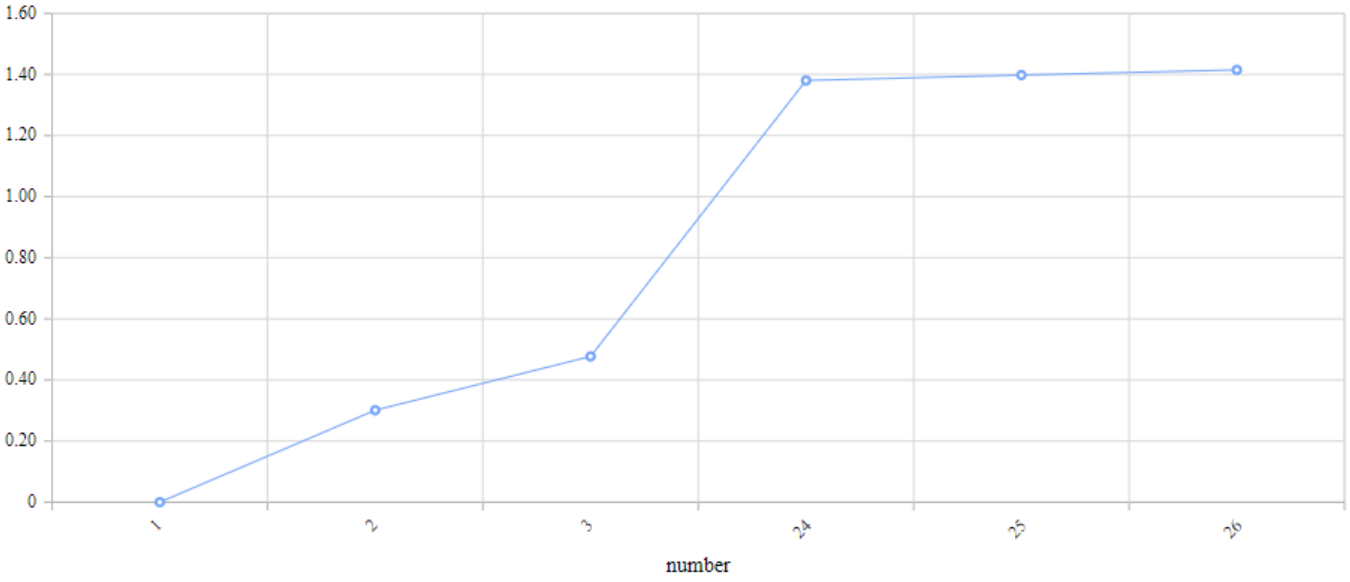
With *Numeric* type, the real trend is visible :



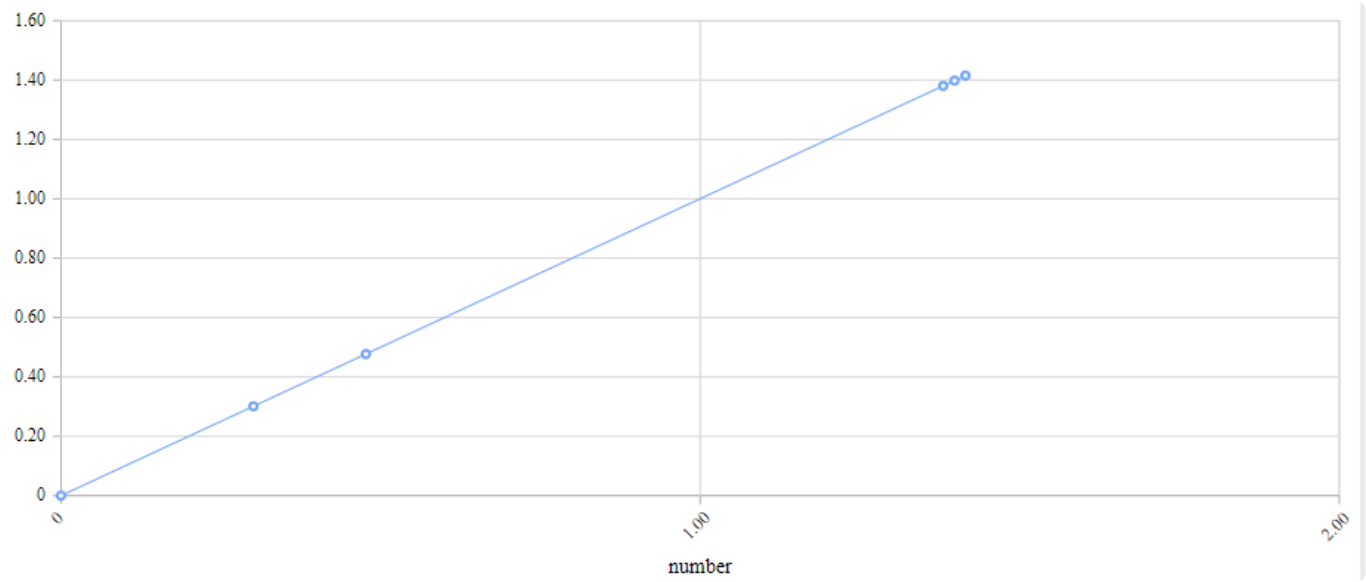
Logarithmic

When the data field is of type Number, the *Logarithmic* option is available.
The X position of points depends on the number difference in the data.

For example, the following data has missing points between X value 3 and X value 24 but this is not clearly visible and the progression looks random :

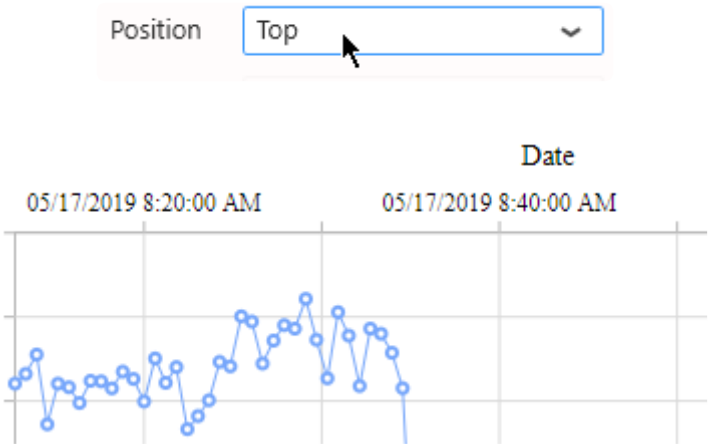


With *Logarithmic* type, the real trend is visible :



Position

The X Axis *Position* can be set to *Top* instead of the default *Bottom* :



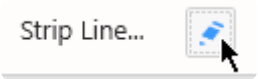
Last modified: Oct 30, 2019

2.8.1.6.1.1. Strip Line

Strip Lines allow to change the background color of the full chart between a minimum and maximum X value.

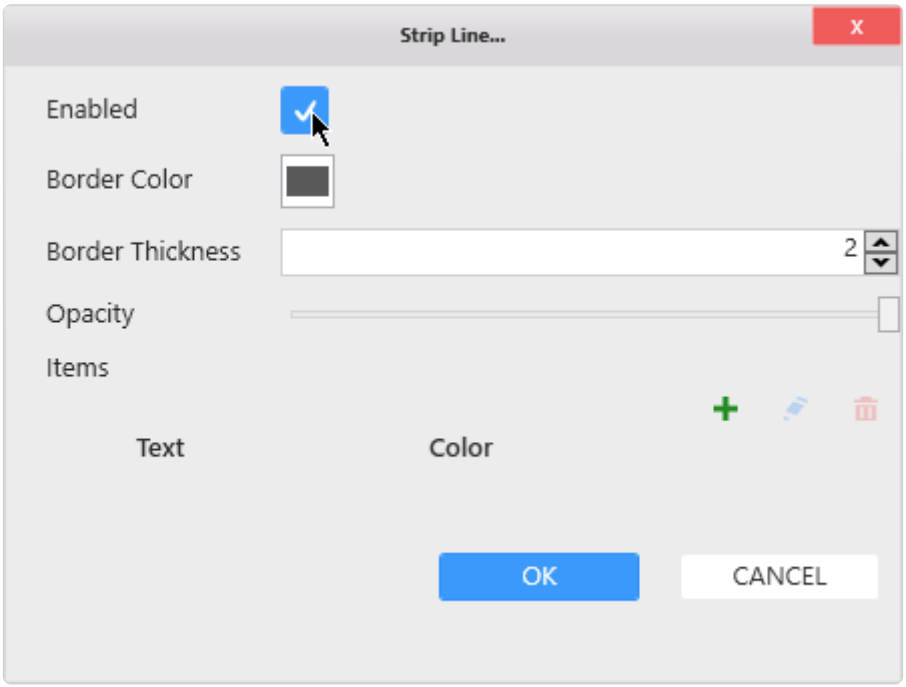
This can be useful for adding context information to a chart, for example the states of an equipment.

To open the Strip Line editor, click on *Strip Line…* :



Global Properties

In the *Strip Line…* pop up window, the top part lists some global properties of the Strip Line :



Enable

Check the *Enabled* box to enable to Strip Line and start configuring.

Border

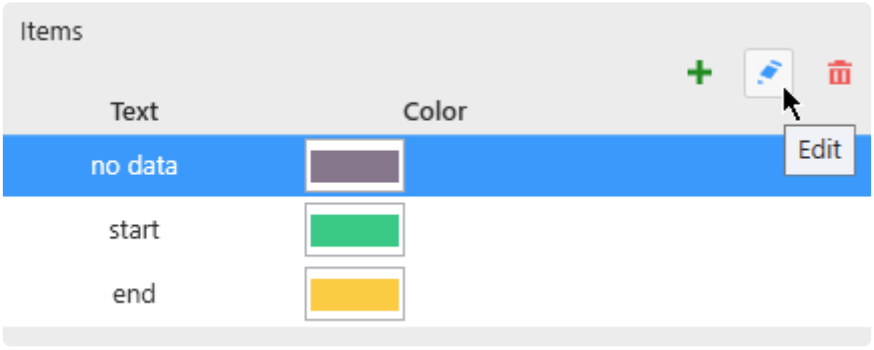
The Strip Line can have a border to better show where is starts and ends.
Use *Border Color* to set the color.
Use *Border Thickness* to set the thickness. A thickness of 0 means no border.

Opacity

The Strip Line can be made semi-transparent using the *Opacity* control.
This allows to integrate better with the background color of the theme and the rest of the chart.

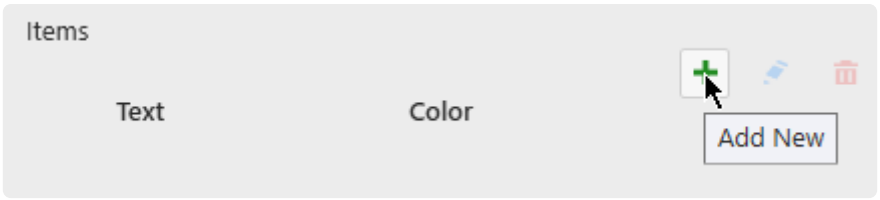
Items

There can be several X Axis Strip Lines, and they are all listed under *Items*.



Adding a Strip Line

To create a first Strip Line, click the + icon :



This opens the *Items* configuration pop-up window to let you configure this Strip Line.
See below for *Configuring*.

Later, to add more, use the + button again.

Editing a Strip Line

When Strip Lines exist, you can edit one by selecting it in the list and clicking the pen icon.

This opens the *Items* configuration pop-up window to let you configure this Strip Line.
See below for *Configuring*.

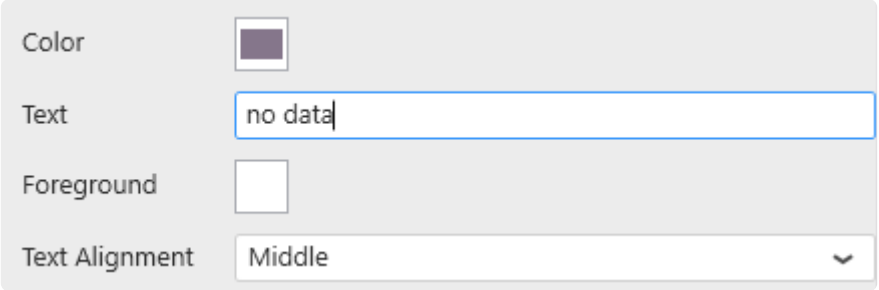
Removing a Strip Line

When Strip Lines exist, you can remove one by selecting it in the list and clicking the trash icon.

Configuring

Style and Appearance

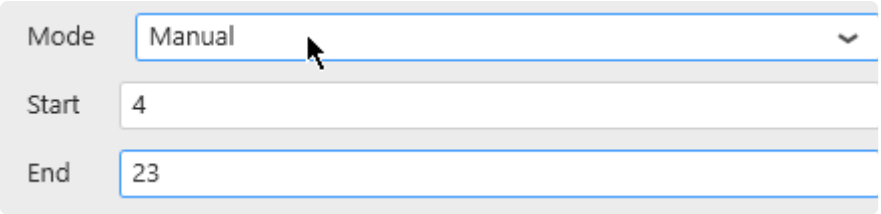
The Style following options are available for each Strip Line :



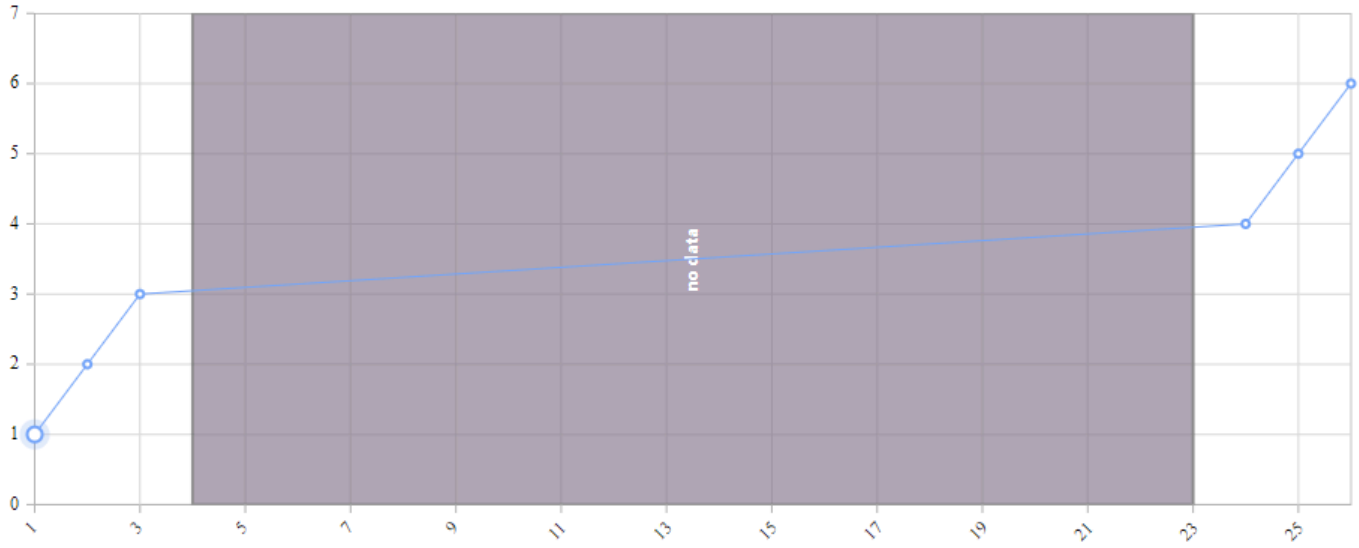
Color : Fill color of the Strip Line.
Text : Text label displayed on the Strip Line. Leave empty to hide text.
Foreground : Text color
Text Alignment : Text position : at the start, end or middle of the Strip Line.

Manual mode

The Start and End value can be configured statically by using “Manual” *Mode* :



In this example, it means the Strip Line will start on `X Value = 4` and end on `X Value = 23` :

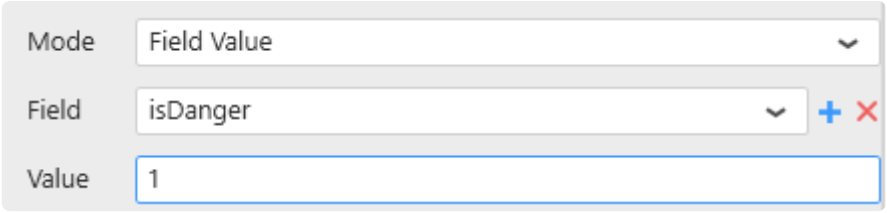


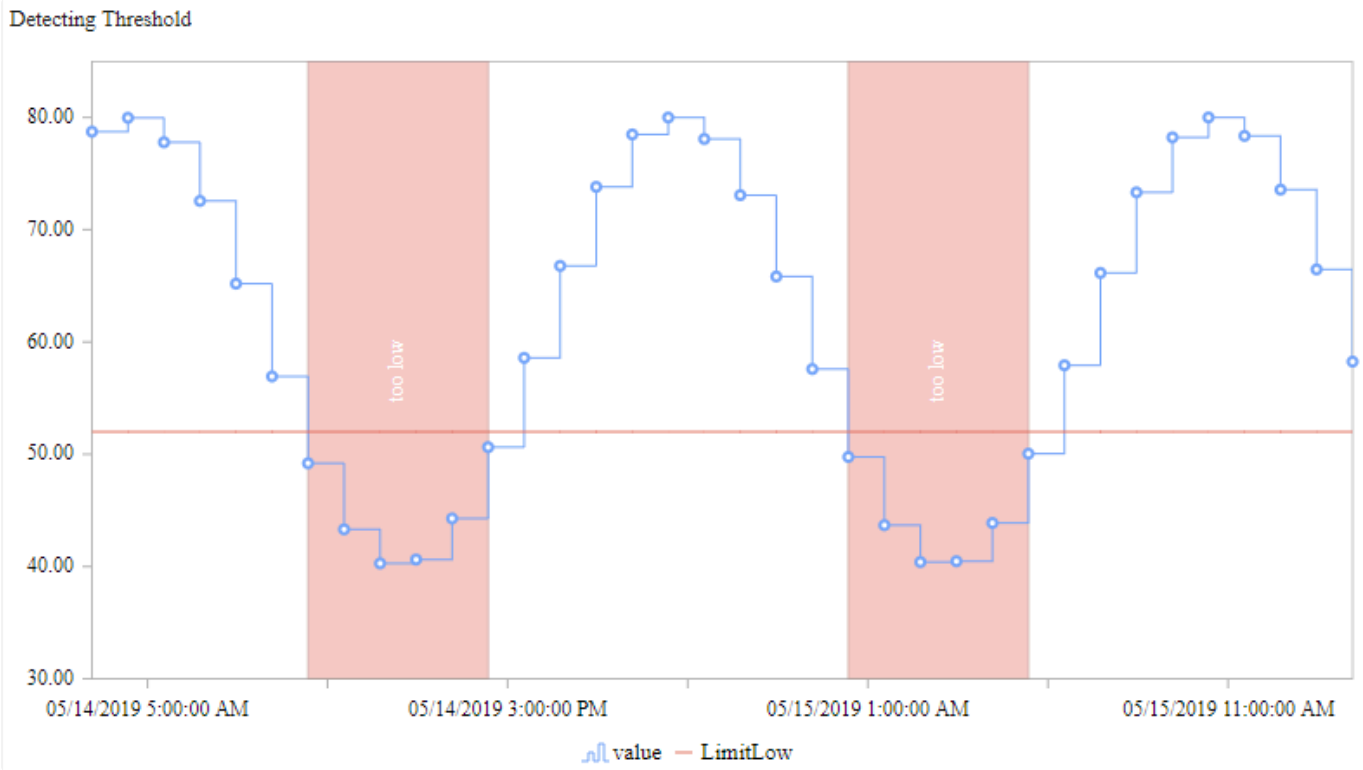
If the X Axis Value Type is “Category”, then the Strip Line *Start* and *End* values are the indexes of the points on the X Axis. The indexes start at 0.
Example : You have 6 X values : A, B, C, D, E, F in order
A Manual Strip Line starting at 1 and ending at 4 means that the first X value to start the Strip Line is B (zero-based index) and the End is at E.

Field Value Mode

The *Field Value* Mode is only valid if the X Axis Value Type is not “Category”.

Instead of static values, the Strip Line can be displayed depending on a condition based on the data. With *Field Value* Mode, you configure a condition, so that when `Field = Value` then the strip line is displayed.





Field

A list of fields is available to select from.

If your field is not available, click the + icon on the right to add a new field to the widget query :

Select the desired aggregation and field, and **click the check icon to validate** :

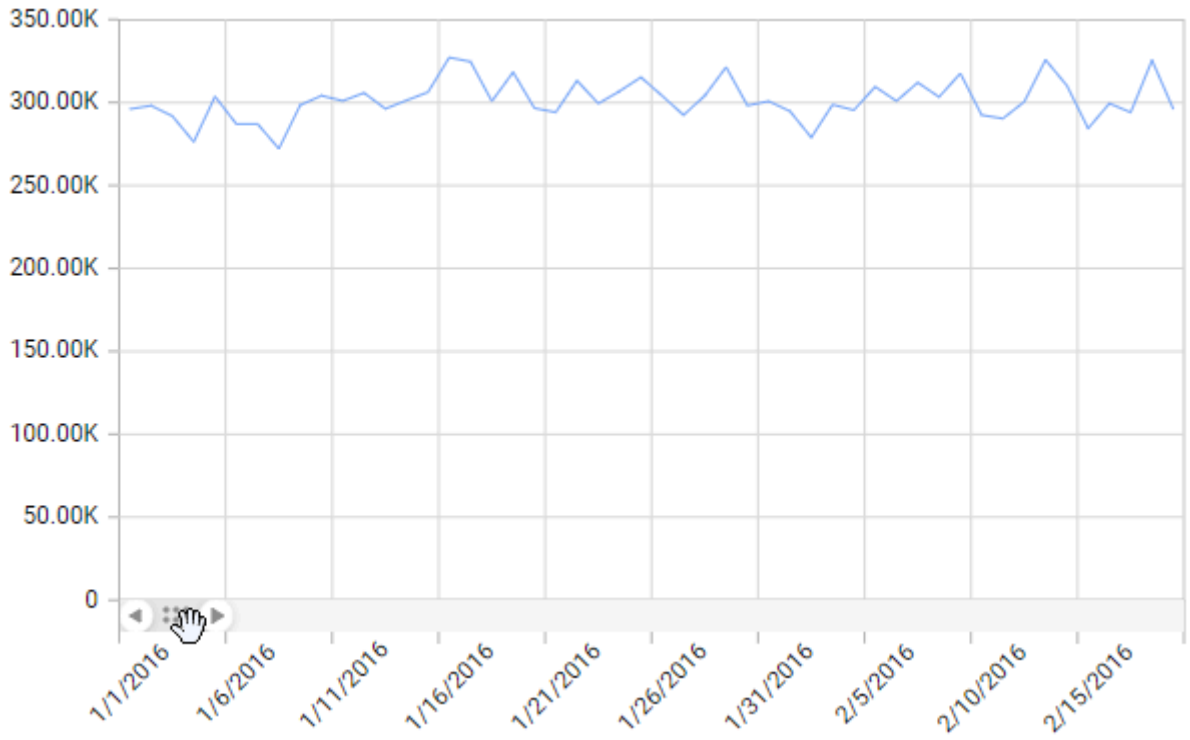
Value

As a Value you can enter any number.

Last modified: Jul 23, 2019

2.8.1.6.1.2. Paging / Scrolling

When too many data points are displayed in a chart, performance may be affected.
That's why a paging mechanism has been set up to limit how many data points are loaded at a time.
When this is active, a horizontal scrollbar is displayed on the X Axis to let you scroll to the rest of the data :



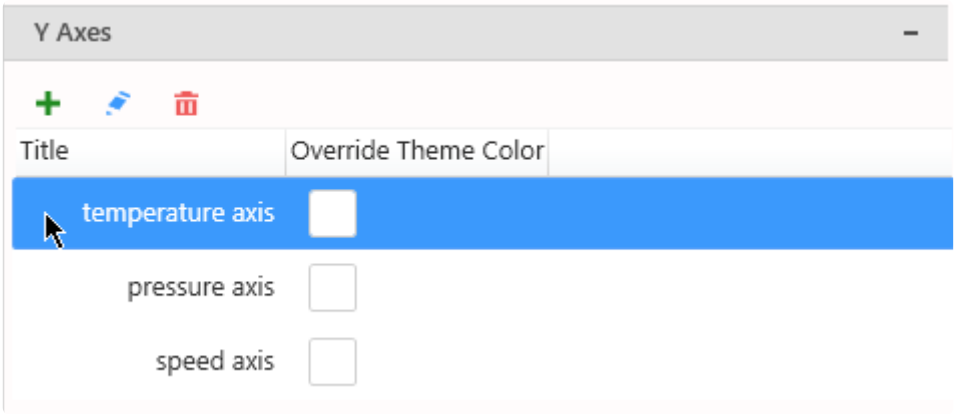
This default behavior can be configured under *Basic Settings* :

Enable Scrolling : enable / disable this behaviour. When disabled, all the data points are always loaded.
Page Size : the maximum number data points displayed per page

Last modified: May 24, 2019

2.8.1.6.2. Y Axes

All Y Axes are listed in the right side in the properties, under section *Y Axes* :



Managing the list of Axes

Adding

By default, only one Y Axis exists, named after the first *Y Value* Series field.

Click the + icon to add a new Y Axis.
This opens the *Edit Axis* configuration pop-up window to let you configure this new Y Axis.
See below for *Configuring*.

Editing

Click the pen icon to edit the currently selected Y Axis.
This opens the *Edit Axis* configuration pop-up window to let you configure this Y Axis.
See below for *Configuring*.

Removing

Click the trash icon to remove the currently selected Y Axis.

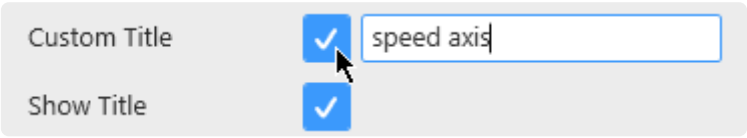
Binding a Series to a Y Axis

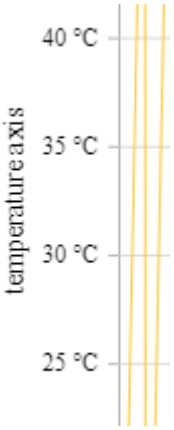
When a new Y Axis is created, no Series are bound to it.
To bind a *Y Value* Series to a specific Y Axis, see Series Settings.

Y Axes Properties

Title

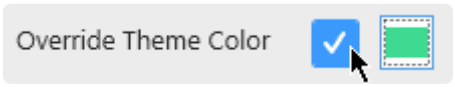
The Y Axis Title can be customized of hidden :

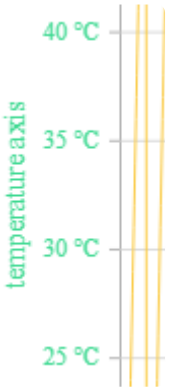




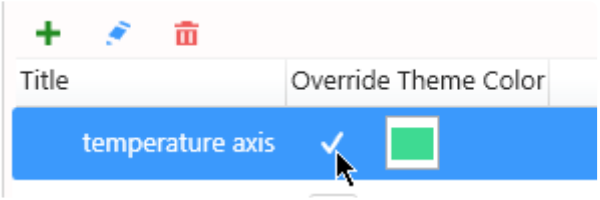
Color

By default, the Labels and Title colors are based on the Theme.
If you want to set a specific different color, check *Override Theme Color* and set the desired color :



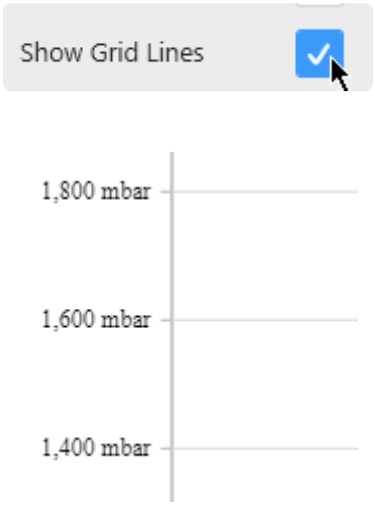


There is also a shortcut in the list of Y Axes :



Grid Lines

Grid lines can be hidden or displayed with *Show Grid Lines* :

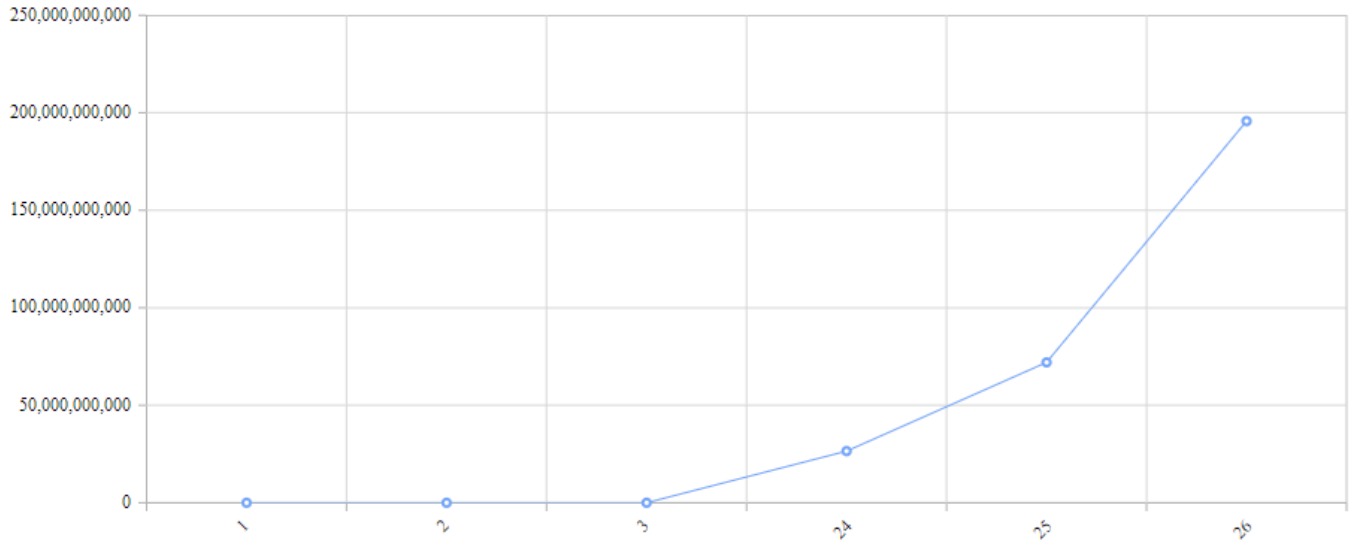


Value Type

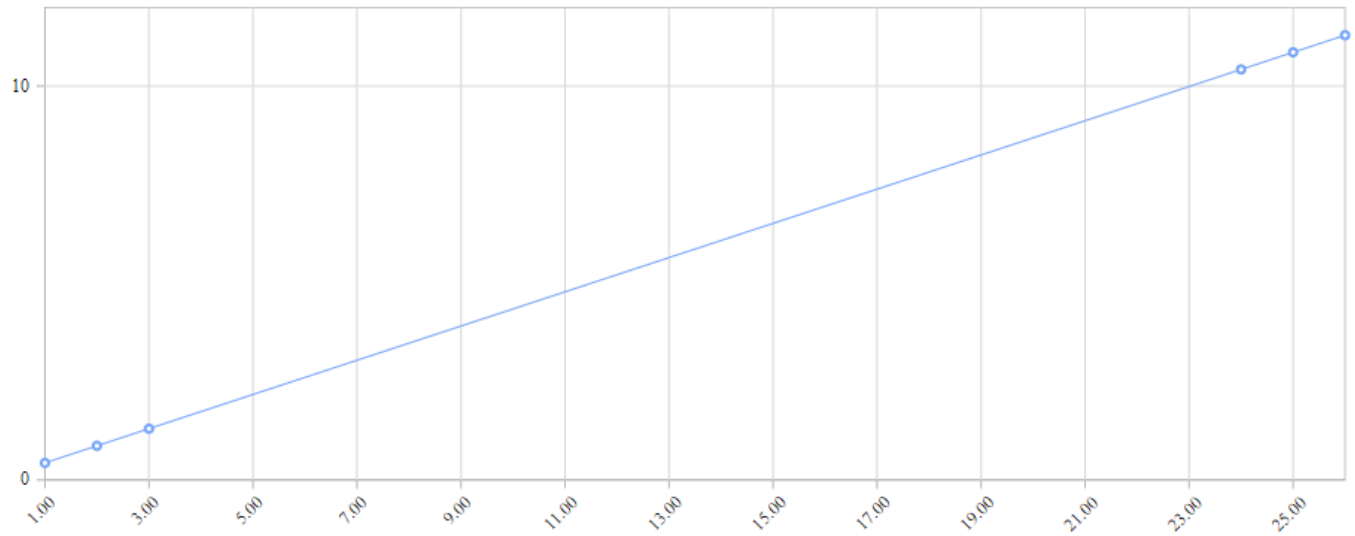
By default, the Y Axis *Type* is “Numeric” : this means that the Y position of points directly maps to the numeric value of the *Y Value* Series data field.

The *Type* “Logarithmic” can also be selected : this means that the Y position maps to the logarithm of the *Y Value* Series data field.

For example, here is a simple chart with Y Axis as “Numeric” (default) and X Axis as “Category” (default) :



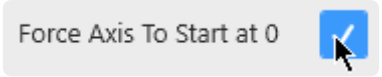
The variations happen on many orders of magnitude, and it's difficult to track what is the real trend. With a "Logarithmic" *Value Type* of the Y Axis, and Numeric for the X Axis, we see that the trend is exactly exponential :



Force Axis To Start at 0

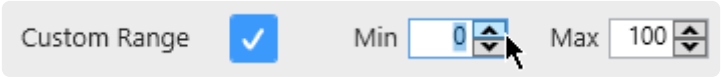
By default, the minimum and maximum value of the Y Axis is automatically calculated to scale to the data displayed.
Instead, you can force the zero to always be included on the Y Axis.
This is useful for example if you want to display absolute variations compared to the total value.

To enable this behavior, check *Force Axis To Start at 0* :



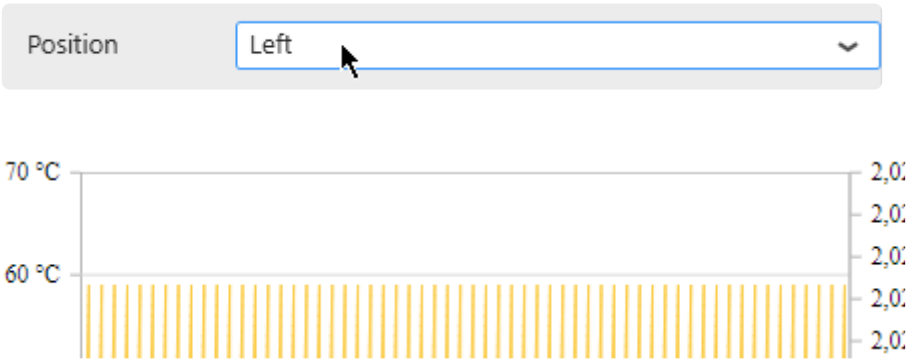
Custom Range

By default, the minimum and maximum value of the Y Axis is automatically calculated to scale to the data displayed.
Instead, you can force a *Custom Range* :



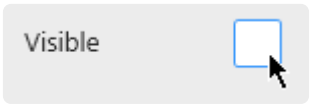
Position

Each Y Axis can be positionned to the *Left* or *Right* :



Hiding the Axis

All the Y Axes can be hidden by un-checking the *Visible* box.



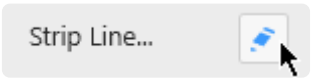
Last modified: Jan 02, 2020

2.8.1.6.2.1. Strip Line

Strip Lines allow to change the background color of the full chart between a minimum and maximum Y value.

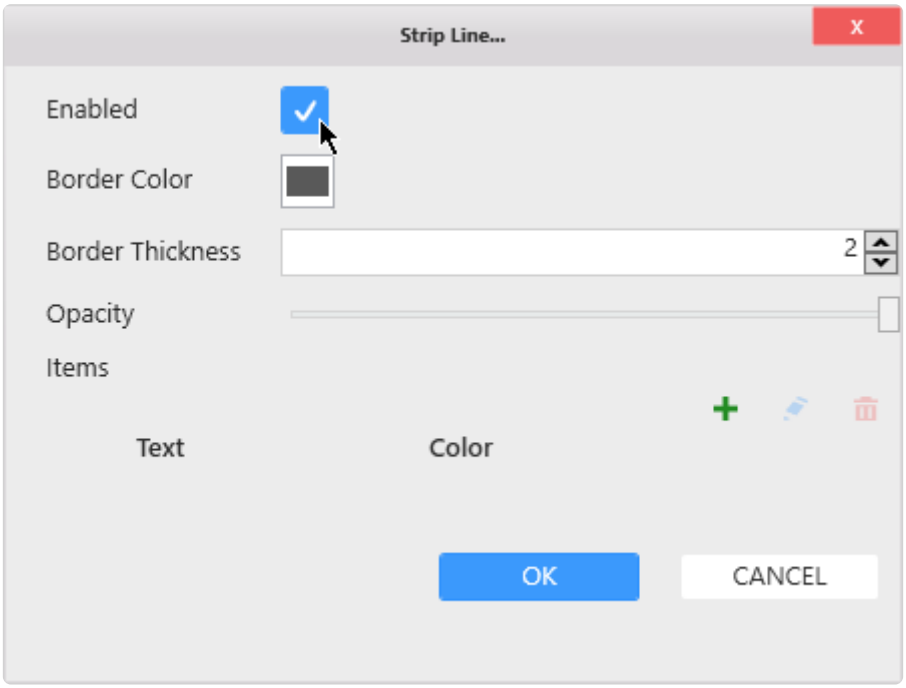
This can be useful for adding context information to a chart, for example the limit values of a measurement.

To open the Strip Line editor, click on *Strip Line…* :



Global Properties

In the *Strip Line…* pop up window, the top part lists some global properties of the Strip Line :



Enable

Check the *Enabled* box to enable to Strip Line and start configuring.

Border

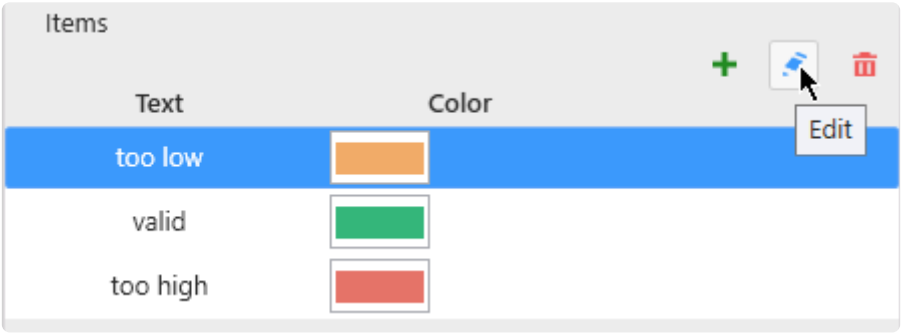
The Strip Line can have a border to better show where is starts and ends.
Use *Border Color* to set the color.
Use *Border Thickness* to set the thickness. A thickness of 0 means no border.

Opacity

The Strip Line can be made semi-transparent using the *Opacity* control.
This allows to integrate better with the background color of the theme and the rest of the chart.

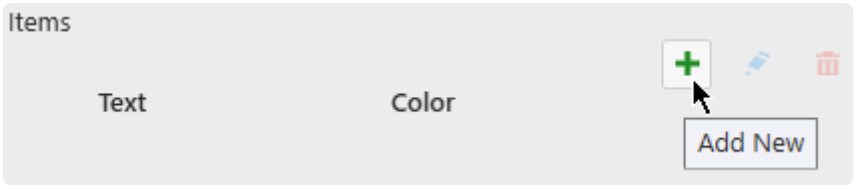
Items

There can be several Y Axis Strip Lines, and they are all listed under *Items*.



Adding a Strip Line

To create a first Strip Line, click the + icon :



This opens the *Items* configuration pop-up window to let you configure this Strip Line.
See below for *Configuring*.

Later, to add more, use the + button again.

Editing a Strip Line

When Strip Lines exist, you can edit one by selecting it in the list and clicking the pen icon.

This opens the *Items* configuration pop-up window to let you configure this Strip Line.
See below for *Configuring*.

Removing a Strip Line

When Strip Lines exist, you can remove one by selecting it in the list and clicking the trash icon.

Configuring

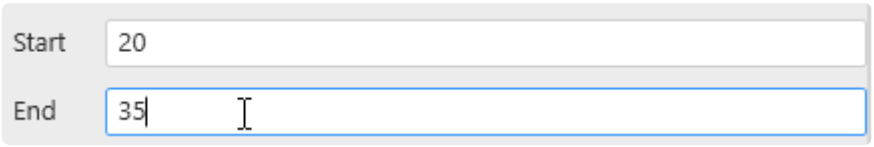
Style and Appearance

The Style following options are available for each Strip Line :

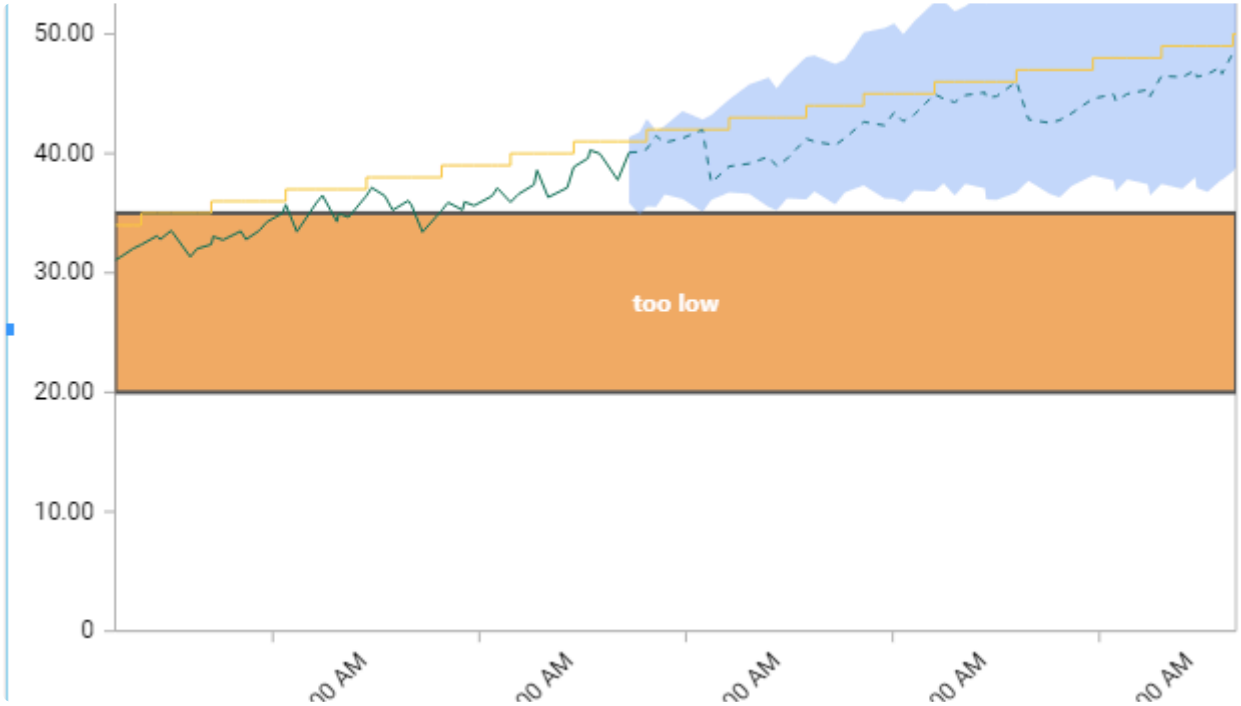
Color : Fill color of the Strip Line.
Text : Text label displayed on the Strip Line. Leave empty to hide text.
Foreground : Text color
Text Alignment : Text position : at the start, end or middle of the Strip Line.

Start / End

The *Start* and *End* value can be configured statically :



In this example, it means the Strip Line will start on `Y Value = 20` and end on `Y Value = 35` :



Last modified: May 24, 2019

2.8.1.7. Legend

Series Legend

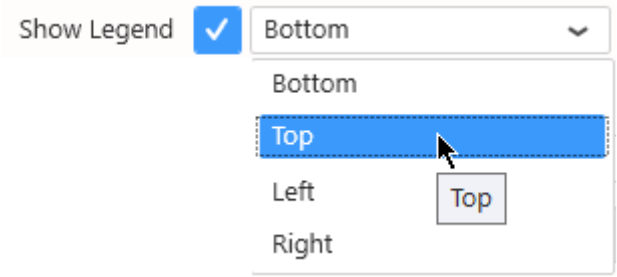
The Legend shows the meaning of each series :

Hide/Show Legend

The chart Legend can be be hidden or displayed using *Show Legend* under *Basic Settings* :

Legend position

The chart Legend can be positioned on any side of the widget :

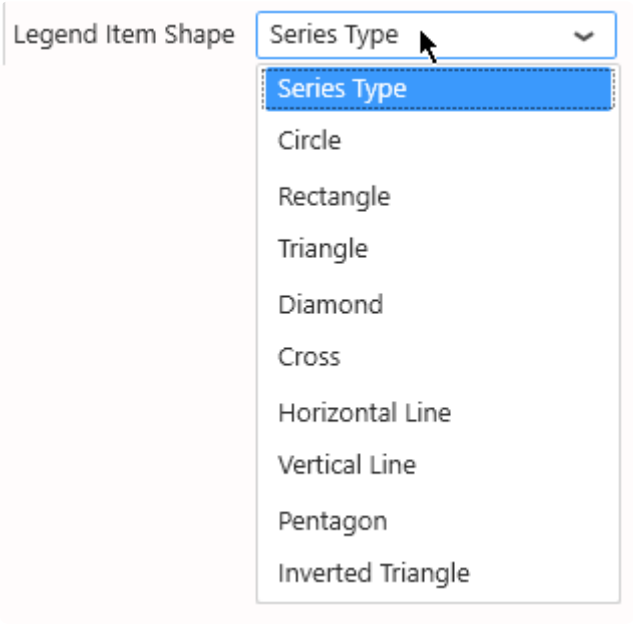


Custom Legend Item shape

For each item in the Legend, its name and icon is displayed, with the Series color displayed on the icon. By default, the icon shape depends on each Series' Chart type.

You can select a different shape that will apply to all Series in the Chart.

Select the desired shape from the *Legend Item Shape* list :

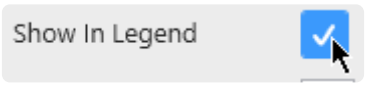


Note : *Series Type* means that the shape will depend on each Serie's Chart type (this is the default).
This is convenient for a Combo chart where you can combine different chart types.

Hide items from the Legend

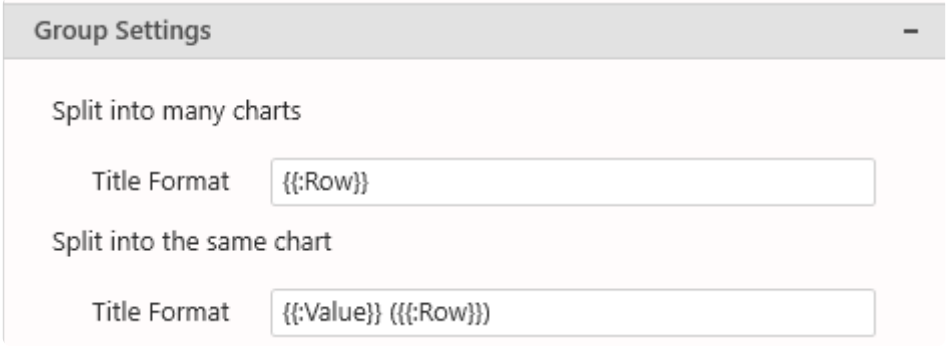
Any series can be hidden from the Legend.

Edit the Series Settings, in tab *Advanced*, un-check *Show In Legend* :

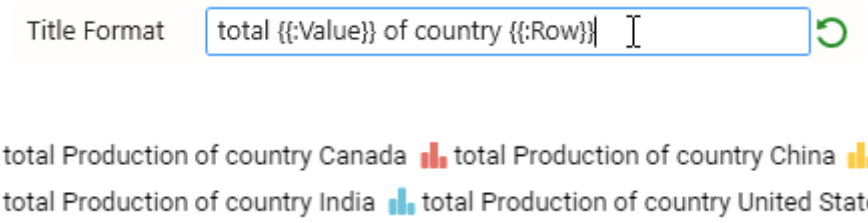


Customize Legend

Legend items are automatically generated from the field names and values, but you can customize how they are generated under *Group Settings* :



With *Split into the same chart*, you can use the placeholder `{{:Row}}` to be replaced by the field value :

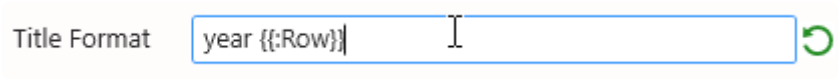


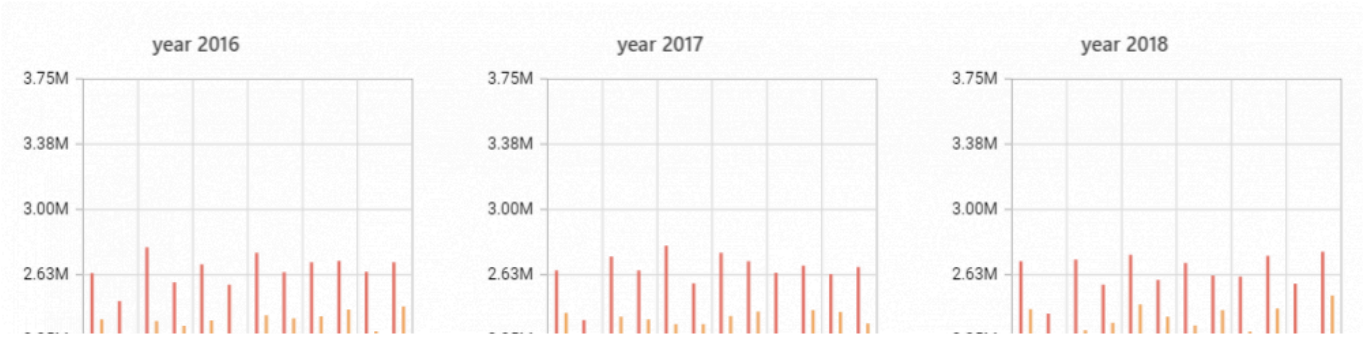
The `{{:Value}}` placeholder gets replaced by the Series Title.

Customize Titles of contained charts

Group Settings also allows to configure the series titles when the Chart has a field bound to *Split into many charts*.
Depending on your data configuration, not all configurations may be available because they wouldn't have meaning.

With *Split into many charts*, you can use the placeholder `{{:Row}}` to be replaced by the field value :





Last modified: Jan 02, 2020

2.8.1.8. Managing Series Color

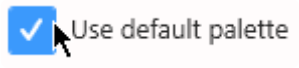
Indexed Colors

By default, Series are assigned a color based on their index in the ordered list in which they appear (usually alphabetical order).
Item1 is assigned Color1, Item2 is assigned Color2, etc…

The palette has 15 colors. When there are more series items than 15, the colors assigned loop back to Color 1.

Default Palette (default)

To enable the Default Palette (it is enabled by default), check *Use Default Palette* under *Series Palette* :

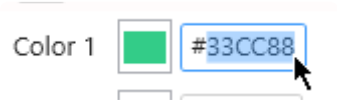


The list of colors in the default palette is below :

Color 1	#77A7FB
Color 2	#E57368
Color 3	#FBCB43
Color 4	#34B67A
Color 5	#F1AB68
Color 6	#6CC2D9
Color 7	#195962
Color 8	#E18878
Color 9	#1AA27D
Color 10	#BFAAC7
Color 11	#89695E
Color 12	#A5C77F
Color 13	#F48FB1
Color 14	#90CAF9
Color 15	#CC4452

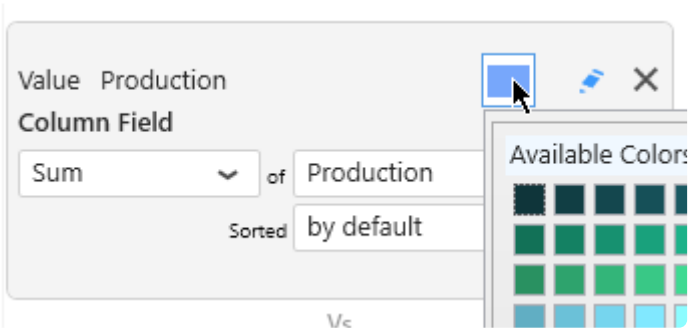
Custom Palette

To start customizing the Default Palette, un-check *Use Default Palette* under *Series Palette*.
Then you can customize colors either by entering their hexadecimal code (#RRGGBB) :



… or by clicking the color and using the graphical color picker :

When there is no data field bound to *Split into same chart*, there is also a shortcut to each serie's color next to the series settings :



If *Use Default Palette* is checked, then this setting is disabled.

Value-based Colors

What are Value-based colors ?

But what if your series changes over time ?

For example, when binding a field as *Split into the same chart*, the items are generated dynamically from values in that field. Their index cannot be known in advance.
When items are renamed, added, or deleted from the data, the index of existing items will change, and their color will change !
This could happen simply during filtering at runtime for example.

Instead of index-based, you may then choose **Value-based** colors : the same value in the Series field will always have the same color in the Chart.

For this, simply check *Use value-based colors* under *Series Palette* :

Customizing Value-based colors

Value-based colors are defined **at the Data Source level**.

This is in order to allow re-use of the colors everywhere the same data is used.

To configure value-based colors, see the corresponding chapter Value-based Settings, under Transforming Data.

Last modified: May 24, 2019

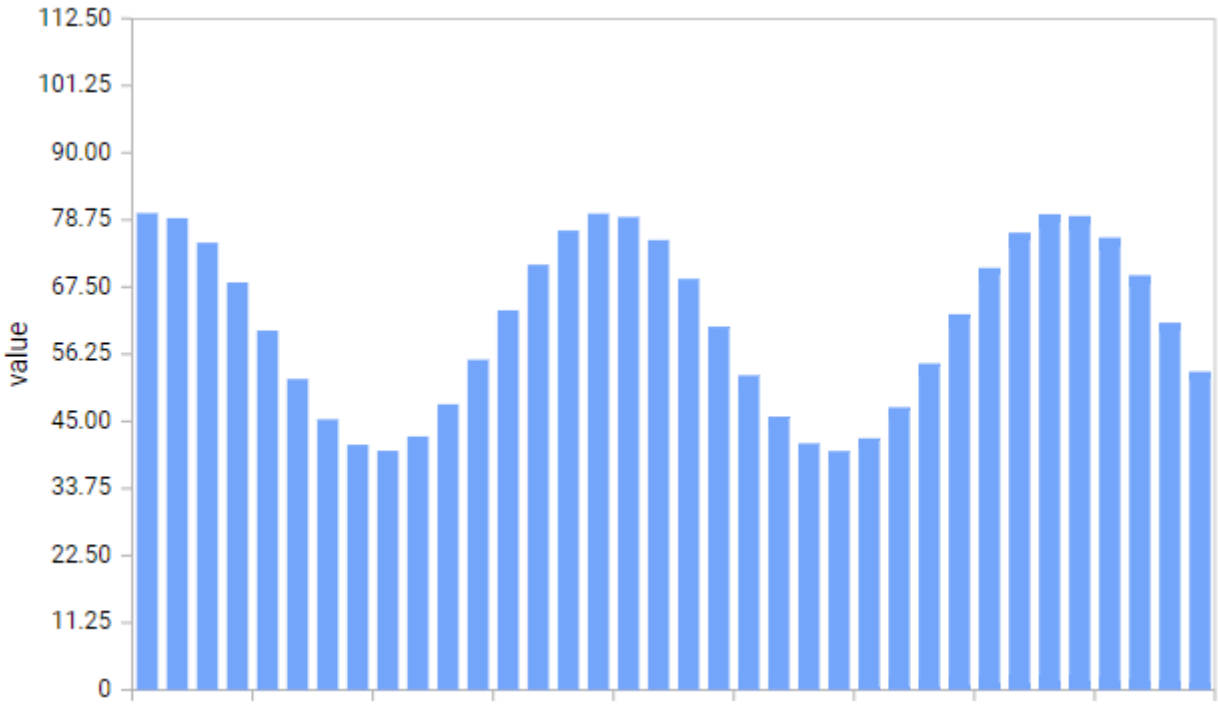
2.8.1.8.1. Chart value color binding

What is Color Binding

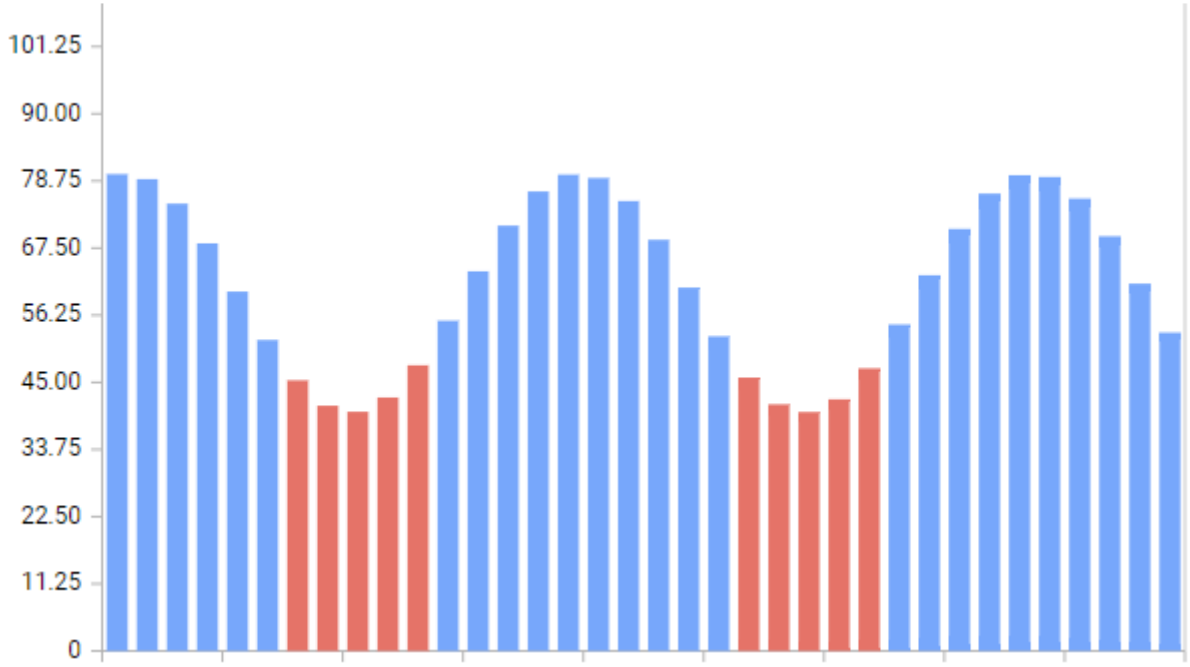
Instead of static colors, a series color can be calculated using a condition based on the data.
With *Color Bindings*, you configure a condition, so that when `Field = Value` then the color is applied.

Example

On a Column Chart with a blue series :



… you want the color to become red in some conditions :



Enabling Color Binding

Edit the Series Settings, in tab *Advanced*, check *Color Bindings* to enable the feature :

Selecting a field

Select a *Color Field* from the existing fields in the chart query, or click the **+** icon to add a new aggregated field to the query :

Select the desired aggregation and field, and **click the check icon to validate** :

Managing conditions

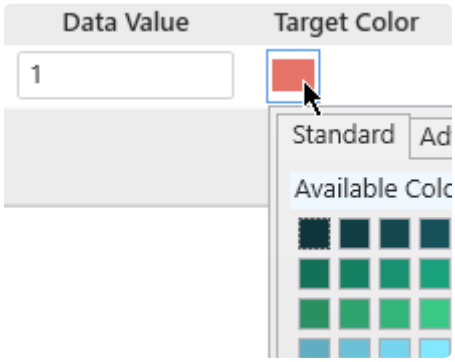
Based on the selected field, several conditions can be configured : `Field = Value1`, `Field = Value 2`, ...

Adding a condition

Click the + icon on top of the list to add a condition :

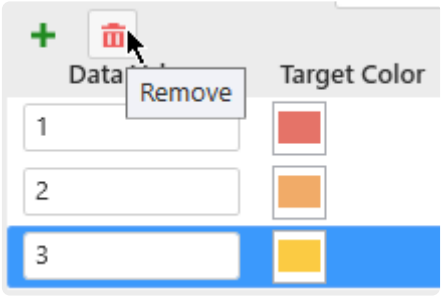
Configuring a condition

Configure a *Data Value* and a *Target Color* so that when `Field = DataValue`, then the color of the series turns into *Target Color* :



Removing a condition

If you wish to remove a condition from the list, select it by clicking it, then remove it using the trash icon :



Supported Chart Types

Some Chart Types like *Line* only show this value color when Show Marker is configured.

The following chart types don’t support value color binding :

- Spline
- Step Line
- Area, Stacked Area, 100% Stacked Area
- Spline Area
- Step Area
- Bubble
- Scatter
- Range Area
- Hi Lo
- Hi Lo Open CLose
- Candle
- Waterfall

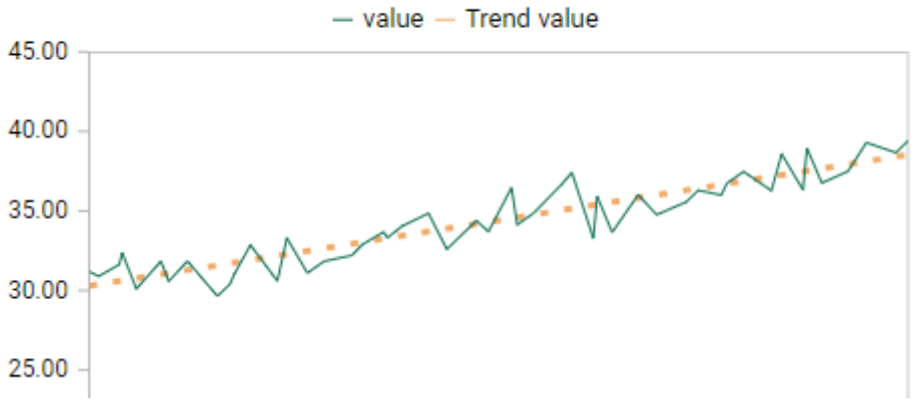
Last modified: May 24, 2019

2.8.1.9. Trend Lines

Introduction

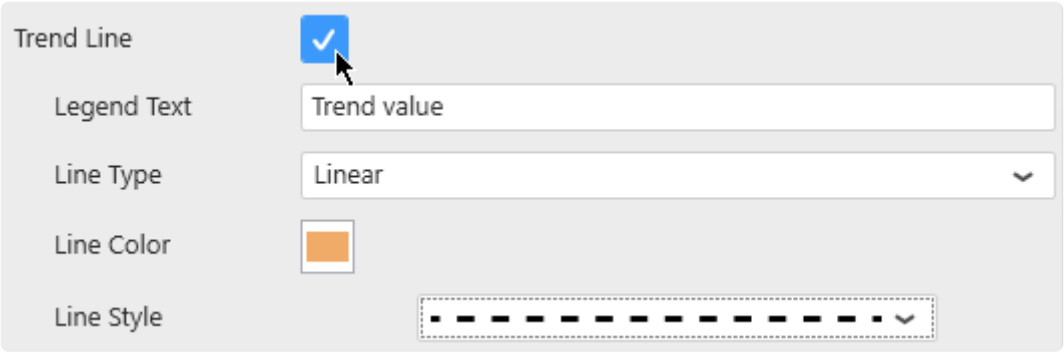
In addition to displaying the data itself, each series can be configured to display a *Trend Line*.
This can be useful for displaying a linear regression or an average on the data points.

Example



Enabling the Trend Line

To enable a Trend Line on a series, go to the Series Settings, in tab *Advanced*, check *Trend Line* :



Configuring the Trend Line

Legend Text : label of the Trend Line withing the Legend
Line Type : type of regression used to match the data values
Line Color : the Trend Line can have a different color from the series color
Line Style : the Trend Line can be displayed as a continuous line or as a variety of dashed lines

Last modified: May 24, 2019

2.8.2. Polar Chart

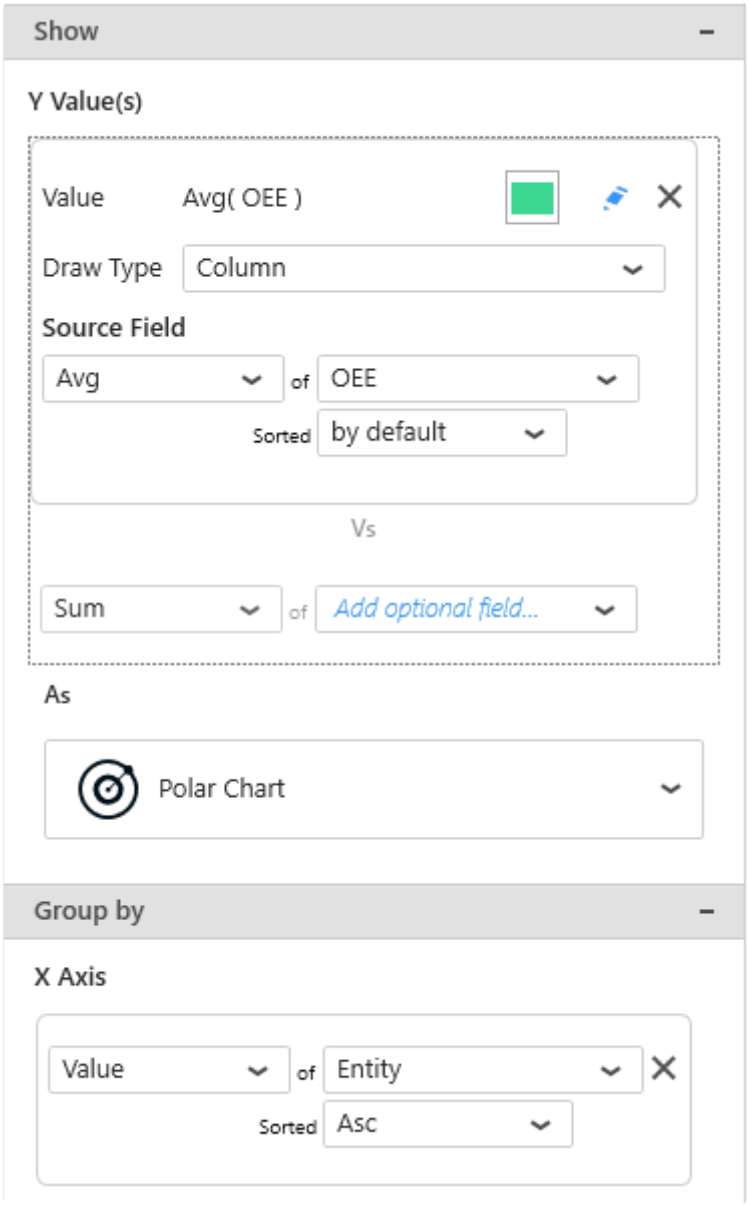
Polar Charts allow to view data on a 2D plane using polar coordinates : a center at 0 and values displayed as distance to that center.

Last modified: May 24, 2019

2.8.2.1. Binding Data

Introduction

The following fields are available for configuring data binding on a Polar Chart :



- *Y Value(s)* : the distance between the data point and the center of the chart
- *X Axis* : how we group values together at different angles to create points on the line

Show : *Y Value(s)*

This represents the *quantity* we want to display on the Chart, and will bind for example to the distance between the data point and the center of the chart.

These fields **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)

At least one field must be bound.

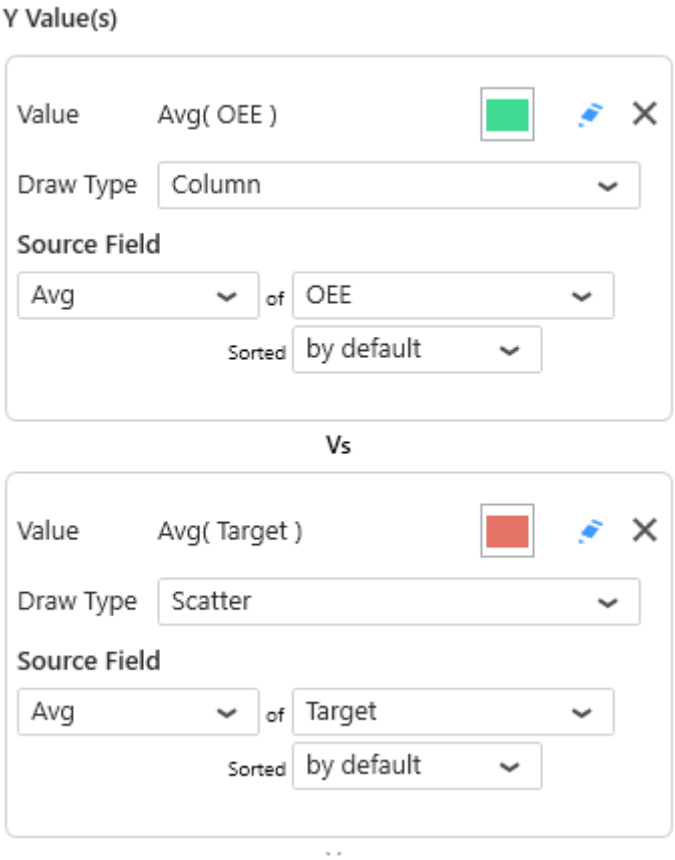
Multiple fields : multiple Series

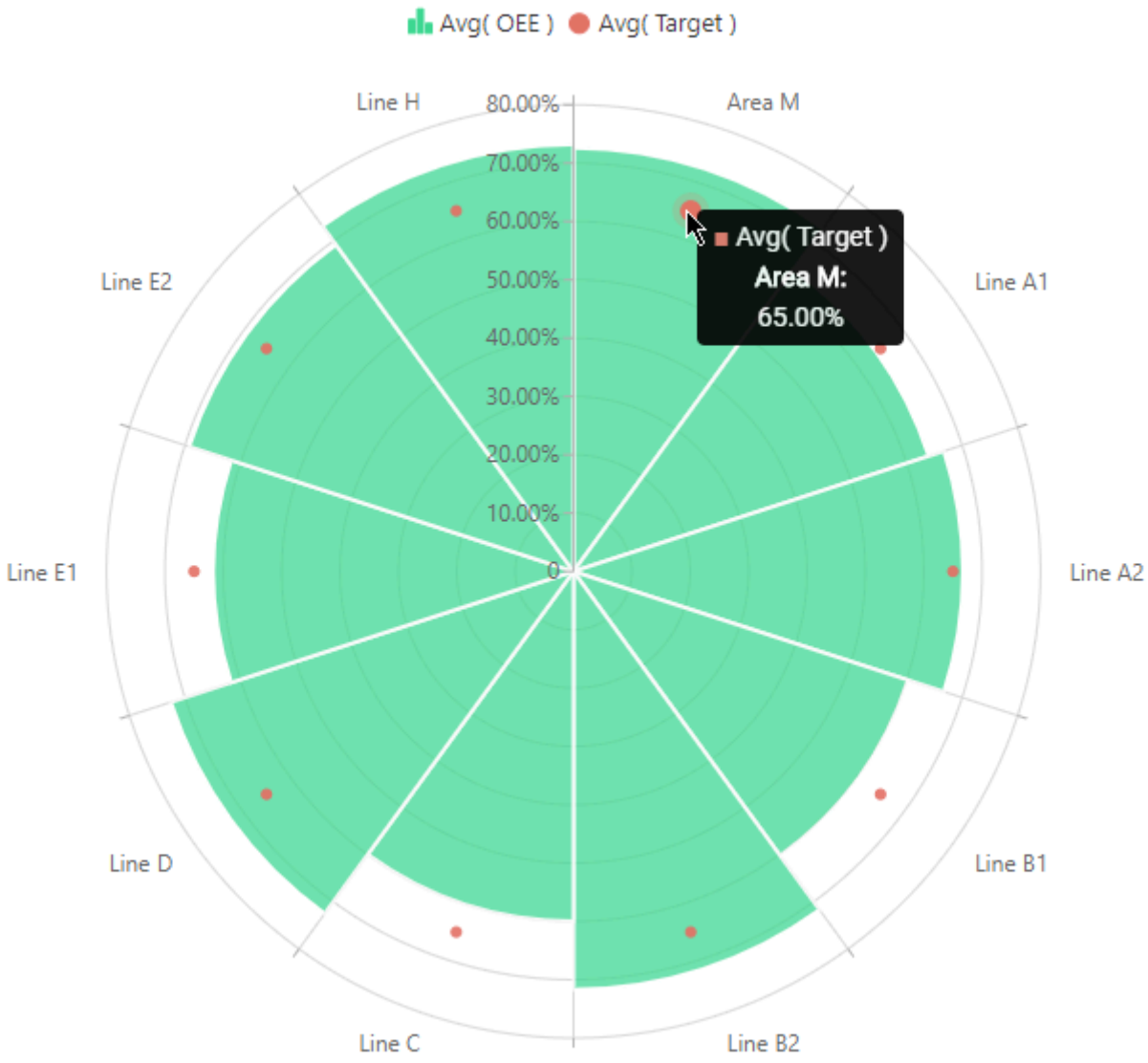
When binding multiple fields at the same time, each field will generate a separate Series.

This way, each Series can have its own graphical properties, its own Draw Type, etc…
To change the order of fields, see the corresponding chapter.

Example

With a field called “Price” and a field called “Target”, you will chart 2 separate series, one for each field.





Group by : *X Axis*

This represents the *category* we want to use to separate data points.

These fields **cannot be aggregated** : they must have several separate values in order to group the “Values” aggregation.
At least one field must be bound.

Last modified: May 24, 2019

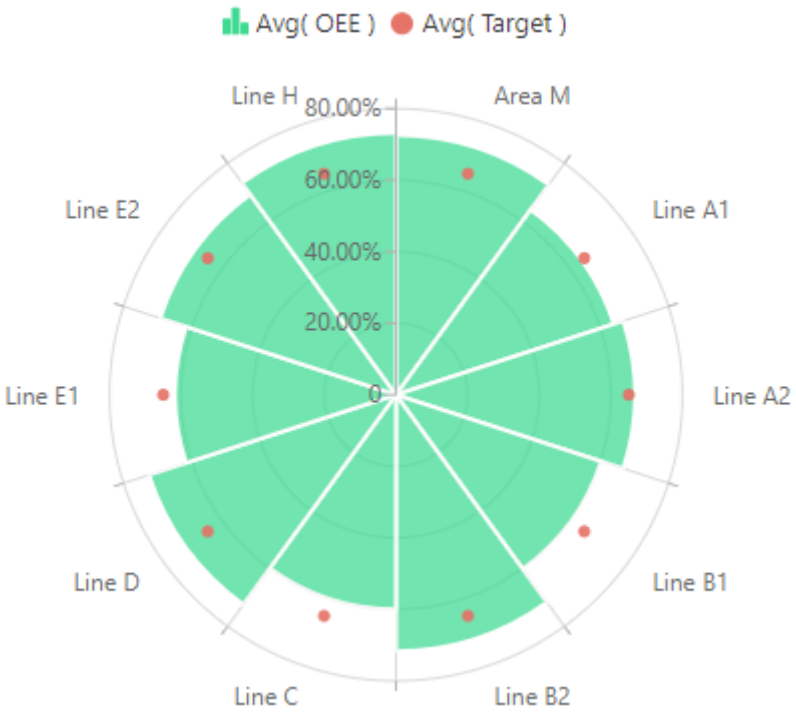
2.8.2.2. Chart Types / Draw Types

Chart Widget Types

There are 2 main types of polar charts :

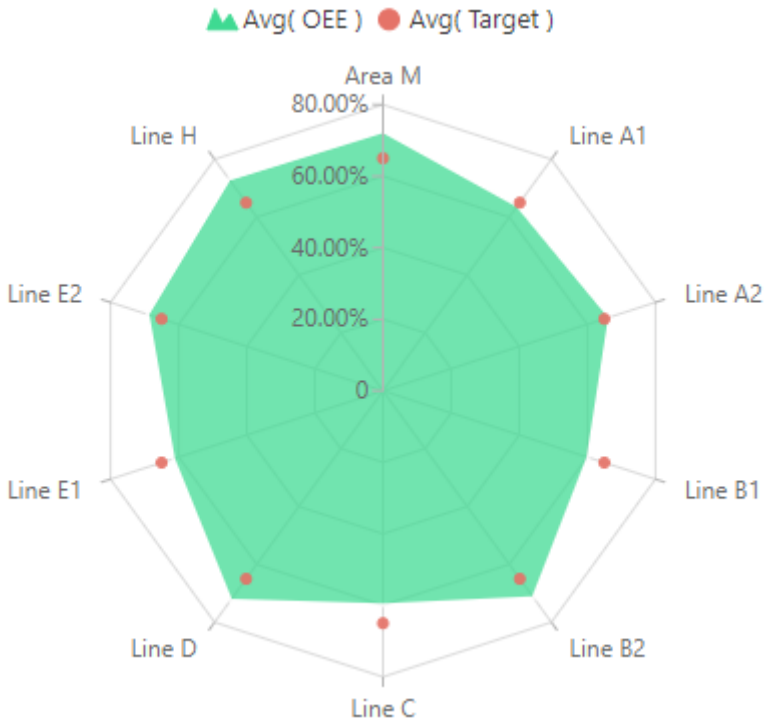
Polar Chart

Constant values are circles.
Each separate *X Axis* value is in the middle of a radial portion.



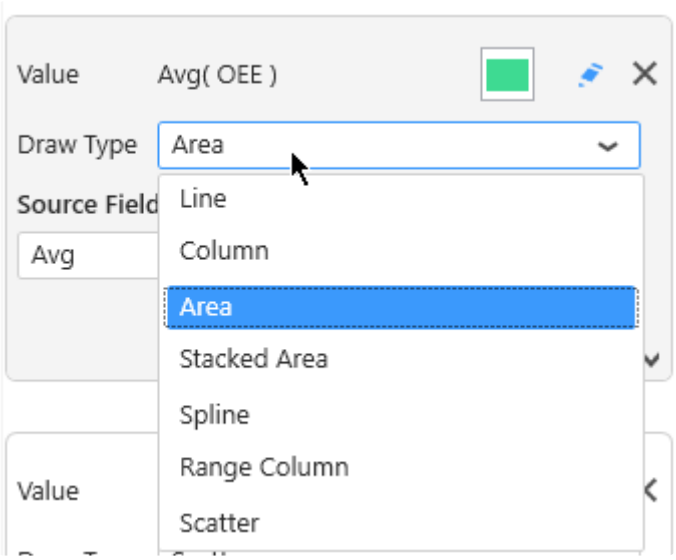
Radar Chart

Constant values are straight lines.
Each separate *X Axis* value is on the edge of a radial portion.



Draw Types

For each Chart Type (Polar or Radar), each series can have a variety of *Draw Types* :



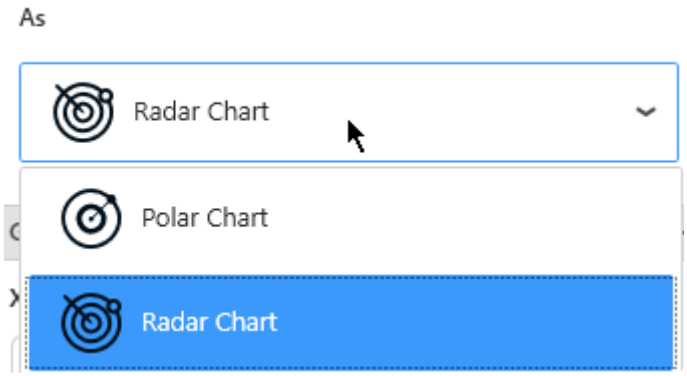
In the same chart, each separate series can have a different *Draw Type*.

Last modified: May 24, 2019

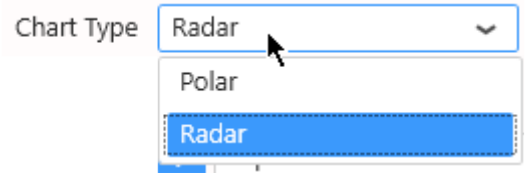
2.8.2.3. Switching Chart Types

When adding a Widget, the Chart will have a single type globally.
The Chart type can be changed on the fly while keeping the widget configuration.

To change the Chart type, use the selector under *As* in the data configuration :



There is also a shortcut under *Basic Settings* :

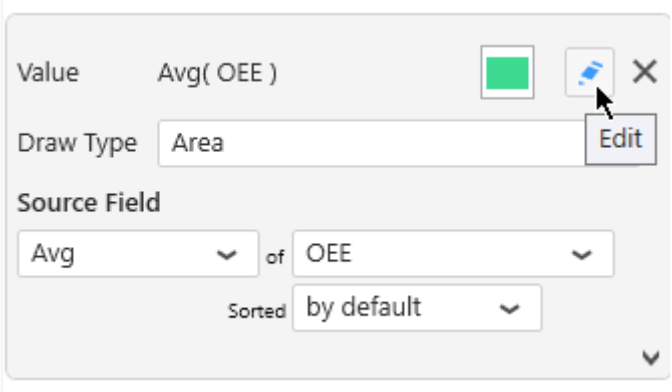


There is also a shortcut in the top bar :

Last modified: May 24, 2019

2.8.2.4. Series Settings

Each *Y Value* Series can be configured individually by clicking the pen icon next to its name :



This allows to fine-tune your chart by configuring exactly what you want on each series inside the widget.

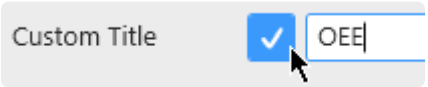
Keep in mind that some configuration options are not available for all chart types.

Last modified: May 24, 2019

2.8.2.4.1. Series Name

Each Series has a name displayed in various places.
By default, this name is automatically generated from the data field bound to the Series.

To configure this name, edit the Series Settings, check *Custom title* and write the desired name :

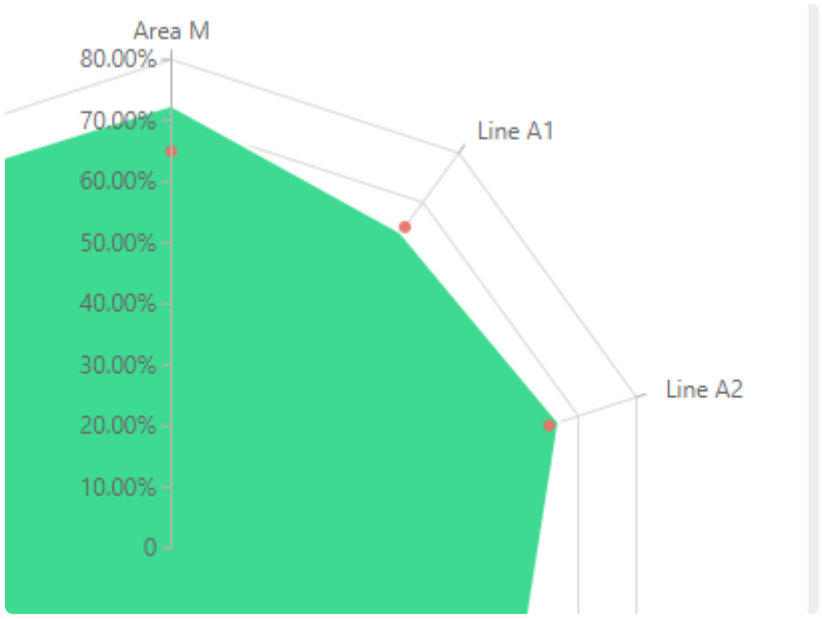


Last modified: May 24, 2019

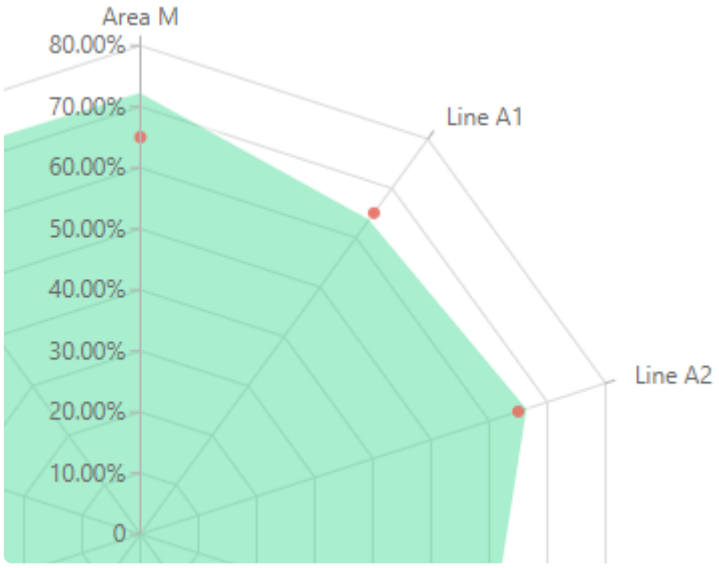
2.8.2.4.2. Style and Appearance

Opacity

Each chart series can be made semi-transparent to allow to see the other series behind it.
The *Opacity* of the Series can be configured :

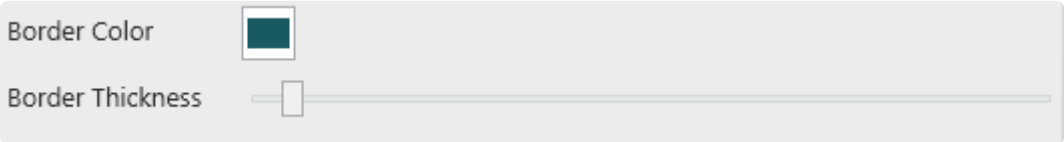


… will turn into :

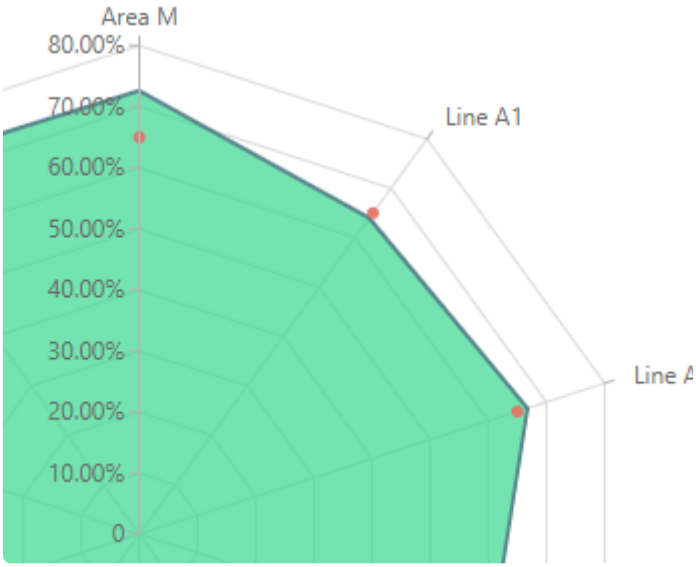


Border

Some Draw Types can have a border around the shapes :



Border Color : select a color from the color picker
Border Thickness : give a value greater than zero to see the border, increase to have a thicker border

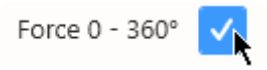


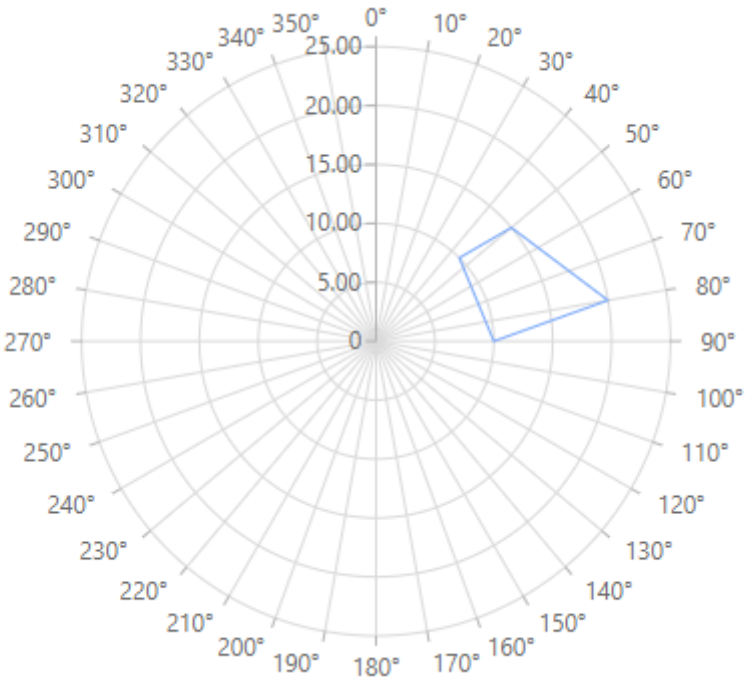
Last modified: May 24, 2019

2.8.2.5. X Axis

By default, the angular position of X Axis items is calculated automatically to divide the plane evenly : 360° / (number of items)
When the X Axis field is a Number, the minimum and maximum angles are calculated from the data.

Instead, if a Number type field is bound as X Axis, then you can force the X Axis scale to 0°-360°





Last modified: May 24, 2019

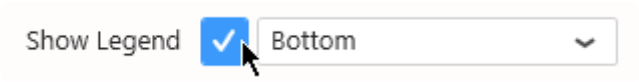
2.8.2.6. Legend

Series Legend

The Legend shows the meaning of each series :

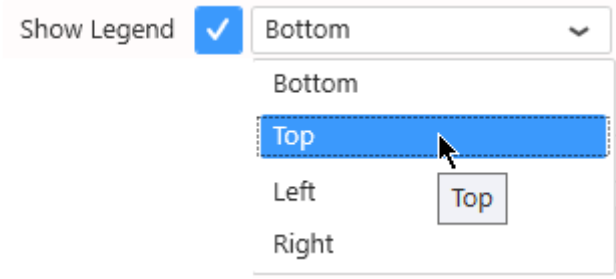
Hide/Show Legend

The chart Legend can be be hidden or displayed using *Show Legend* under *Basic Settings* :



Legend position

The chart Legend can be positioned on any side of the widget :

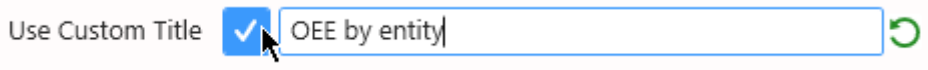


Customize Legend

Legend items are generated from the Series Title.

Chart Title

The chart has a Title.
By default, this is generated from the X Axis field name.
Instead, you can configure it under *Basic Settings* by checking *Use Custom Title* and writing the desired title :



You cna hide this Title by writing an empty Custom Title.

Last modified: May 24, 2019

2.8.3. Map

Alpana allows to display geographical data using 2 types of maps :

- Choropleth Maps show a color on the complete map areas
- Bubble Maps show a bubble centered on the map areas

Last modified: May 14, 2019

2.8.3.1. Maps format : GeoJSON

File format

Alpana supports a map format called GeoJSON

This is a structured text format similar to Javascript Objects and can be edited with any text editor.

Geographic or X-Y

The shapes represented by Alpana maps can be :

- geographic : coordinates are longitudes/latitudes and a Mercator projection is applied
- normal : coordinates are X/Y and are considered as on a flat surface

Features

The GeoJSON format allows to define shapes as “Features” (mostly polygons) which Alpana can draw in Widgets as vector graphics very efficiently.

GeoJSON “*Features*” supported by Alpana are :

- Polygon : any set of points define a closed shape as a polygon
- MultiPolygon : any set of polygons can be grouped into a single shape (example : the British Isles may be considered a single shape made of several separate polygons)
- Point : a single point can also be represented (point of interest, city, etc)

Properties

Each Feature can possess a set of *Properties* which can contain any metadata.

These Properties allow map makers to give unique names to each shape, and will allow Alpana to bind dashboard data to each shape depending on its name (“to each Feature depending on its Property”).

These Properties are a dictionary of key-value pairs.

Finding or Creating new Maps

Editing

Some GeoJSON files can be easily edited by hand using a text editor (Notepad), especially for modifying Properties.

This can be made easier using a text editor with syntax highlighting for Javascript (or JSON) like Visual

Studio or Notepad++.

Some easy-to-use graphical tools exist, including free web-based tools like http://geojson.io/

Creating

Creating maps from scratch is more difficult when using only a text editor.

One option is to use a graphical tool like http://geojson.io/

Another option is to generate the GeoJSON Features programatically (using Excel for example) by concatenating JSON strings and actual coordinates.

Finding GeoJSON Maps

It’s easy to find existing GeoJSON maps with relatively open licenses on the web.

Usually just type the name of your region + “geojson”, and you should find many results.

Last modified: Mar 08, 2019

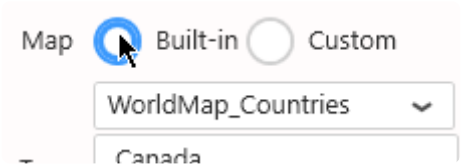
2.8.3.2. Choosing a map

Select a Map

Built-in Maps

Some geographical maps of the world are provided with Alpana.

They are available under Map Settings, by selecting Built-in :

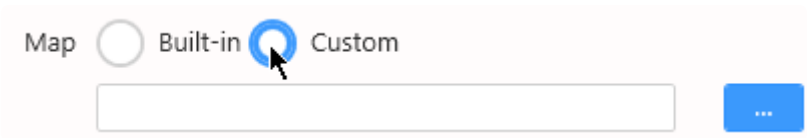


By default, the selected map shows the world's countries.

Custom Maps

You own custom GeoJSON files can be used instead.

Under Map Settings, select Custom :



Click the "..." to browse for a `*.json` file on your machine.

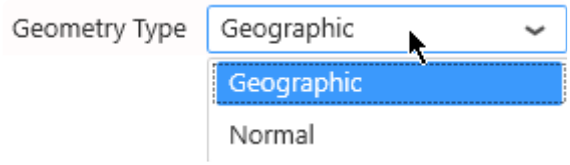
> Note : Performance VS Details
> This file will be embedded in the Dashboard itself.
> This may impact the file size of the web page generated by the dashboard.
> Any time a user will load the dashboard in their web browser, they will download the page in their browser. If the map is huge with many details, this may impact dashboard performance.

Geometry type

Select a Geometry Type :

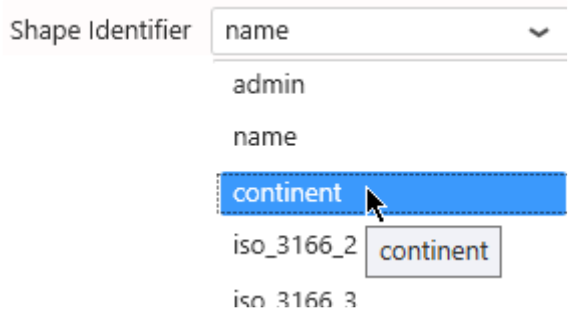
- *Geographic* : coordinates are longitude/latitude with a corresponding Mercator projection. This is useful for geographical data like country shapes. Not suitable for viewing very small values of latitude and longitude (eg: 0.0001 degree latitude or longitude)

- *Normal* : coordinates are X/Y with no projection. This is useful for representing a plan, like an equipment shape or a plant floor disposition.



Shape Identifier

When a map file is selected, the GeoJSON *Properties* key names are listed in the “Shape Identifier” combobox in Alpana Designer :



For example, the default map has Properties named “admin”, “name”, “continent”, …
And each Property has values like name=France, continent=Europe, …

Note :
This is specific to this map file and will be different in other files.

This setting is **very important** since it defines which GeoJSON Property will be bound to your data.

The values in the selected *Shape Identifier* **must match** the values in your Data Source.

For this, you have basically 2 solutions :

- edit the GeoJSON file so that there is a Property that possesses values that match your data

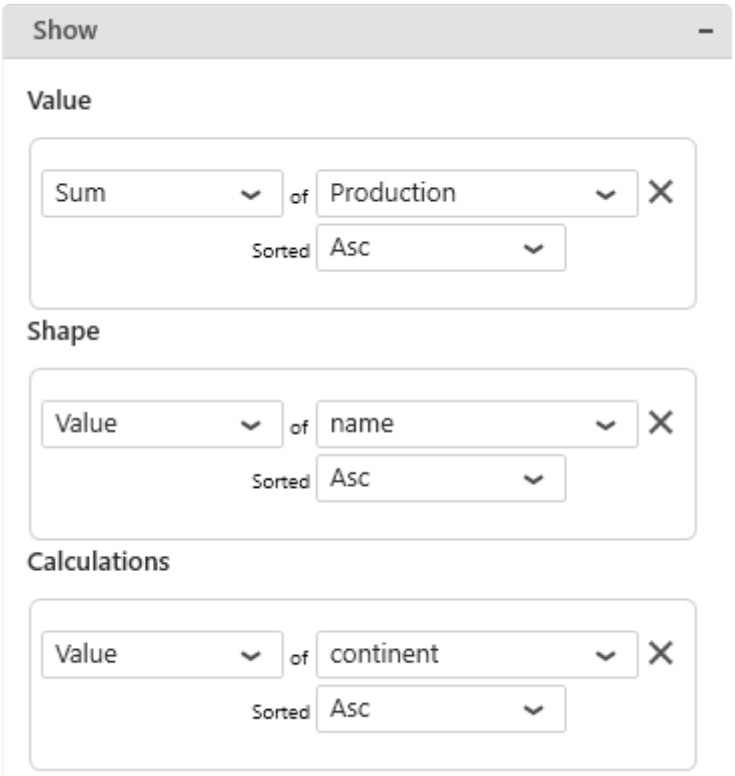
- edit your Data Source so that the values match the values of a map Property

Last modified: May 14, 2019

2.8.3.3. Binding Data

Introduction

The following field containers are available for configuring data binding on a Map :



- *Value* : the color intensity of the shape (Choropleth Map), or the size of the bubble (Bubble Map)
- *Shape* : what Data Source field will be bound to the selected GeoJSON Property/Shape Identifier
- *Calculations* : (Choropleth Map only) assign a single color to a shape or group of shapes

Value

This represents the quantity we want to display on the Map, and will bind for example to the color intensity of the shape (Choropleth Map), or the size of the bubble (Bubble Map).

The Value field **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)
This field is required.

Shape

This will link the values from the selected Data Source field to the values of the selected GeoJSON Property/Shape Identifier.

> ! This is very important : this is how Alpana knows which map shape is associated to

which data field.

The Shape field **cannot be aggregated** : it must have several separate values in order to group the “Value” aggregation.
This field is required.

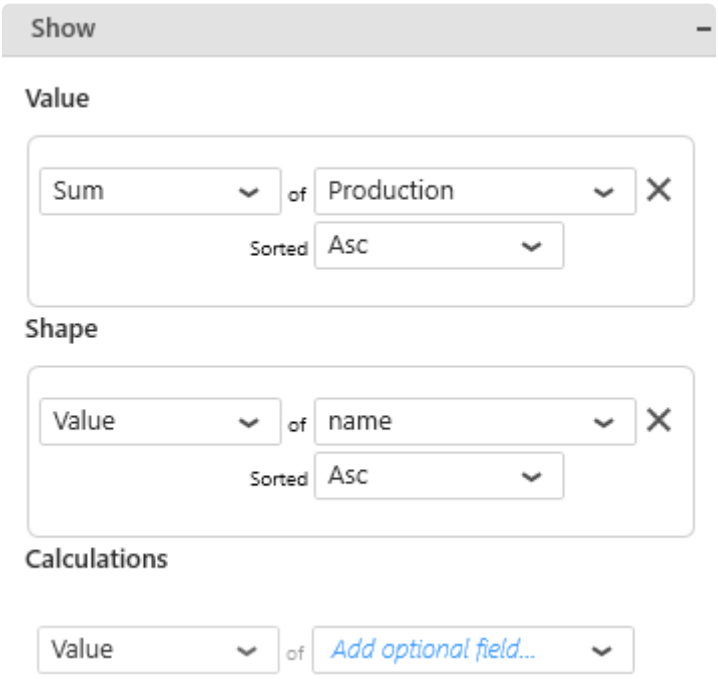
Calculations (Choropleth Map)

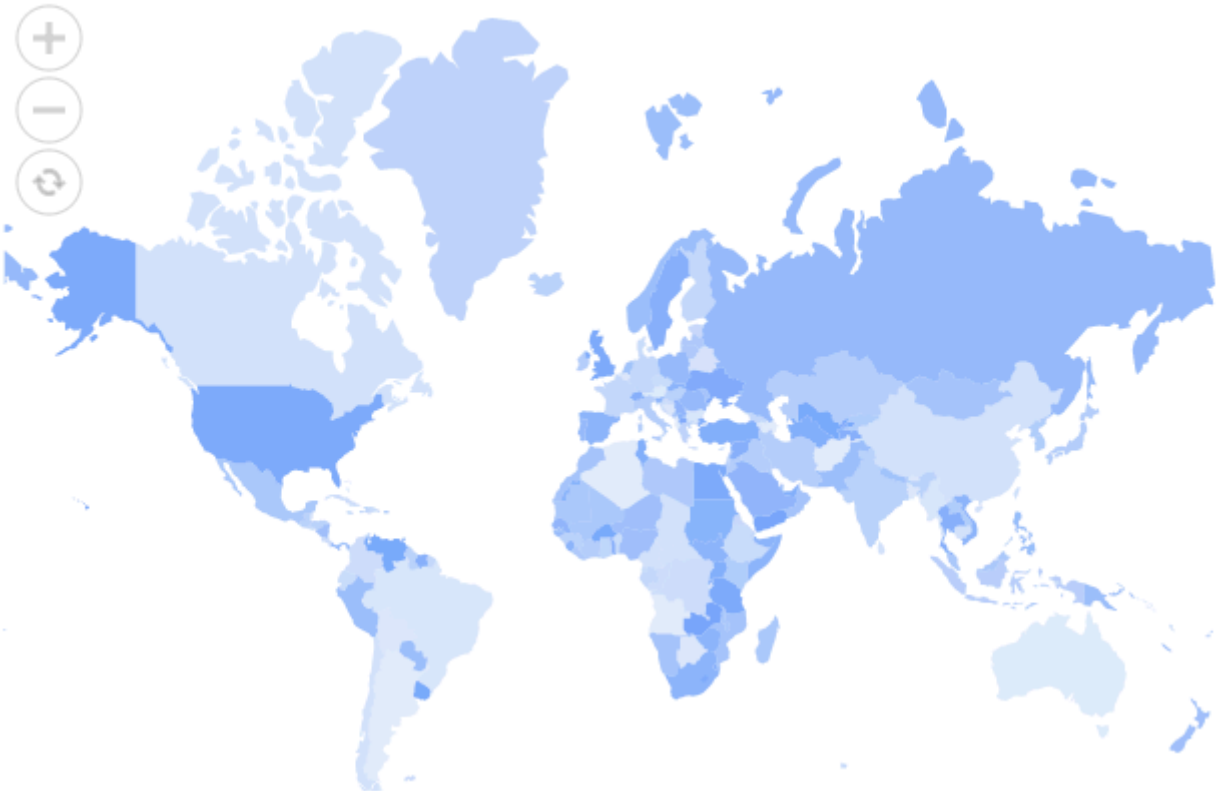
When this field is configured, the Choropleth Map colors no longer show gradients of a single color depending on the intensity of the Value.
Instead, each separate value of the Calculations field generates a single separate color, and all corresponding shapes are painted with that color.
Then the “*Value*” field is only displayed as indication in the tooltip.

The *Calculations* field **cannot be aggregated** : it must have several separate values in order to group the “Value” aggregation.
This field is optional.

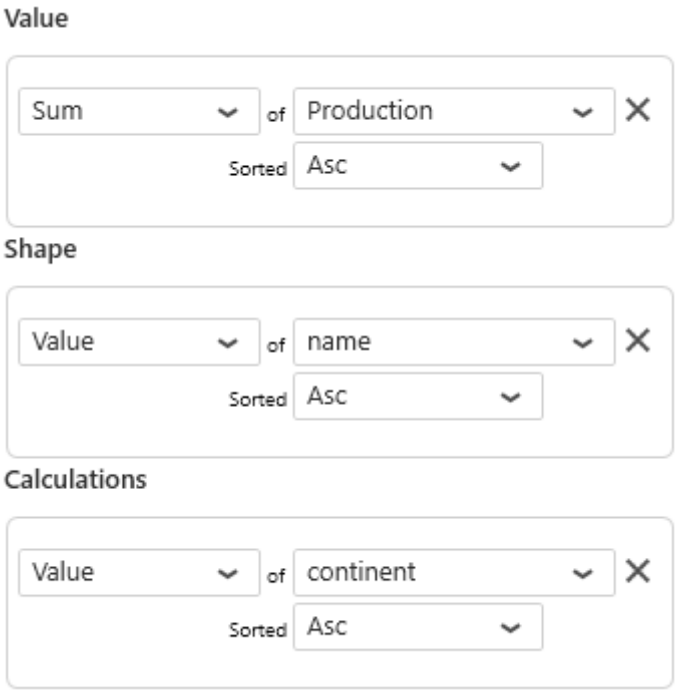
Example

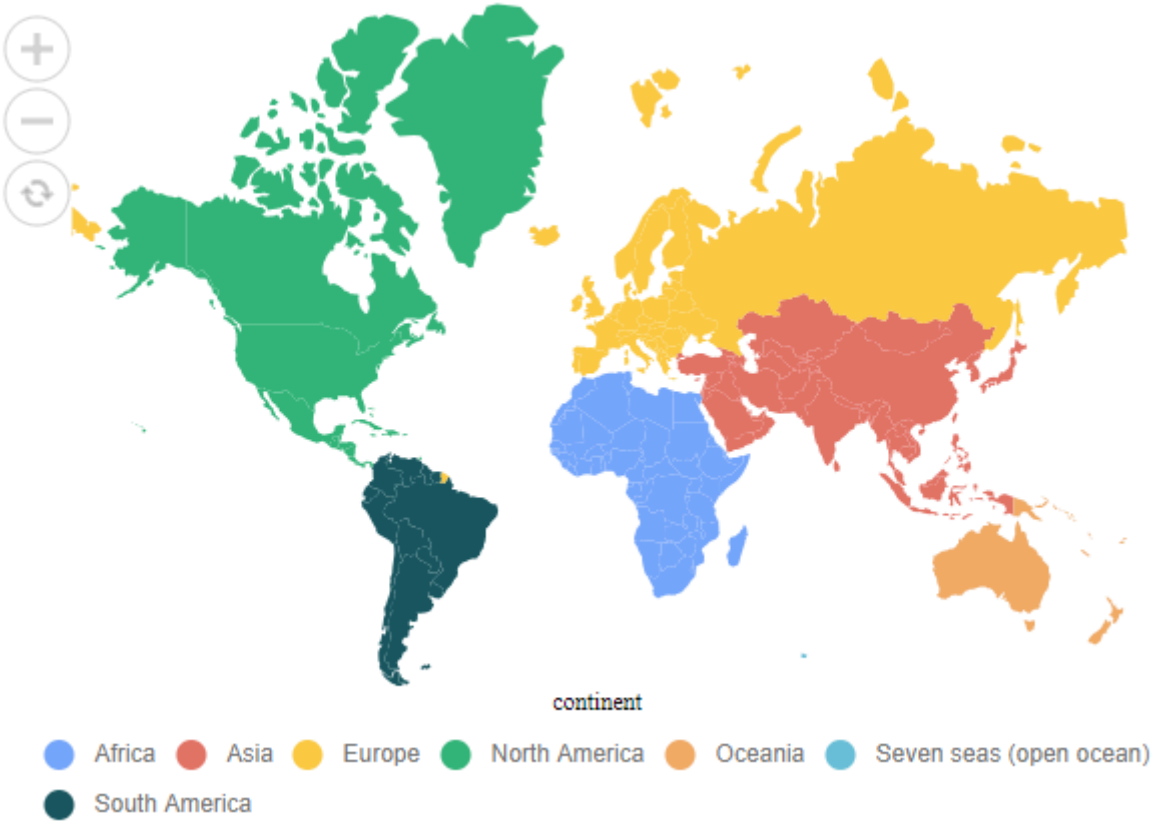
Before applying a calculation field, all countries have a different shade of blue :





After using “continent” as Calculations field, each country have the color of the corresponding continent :





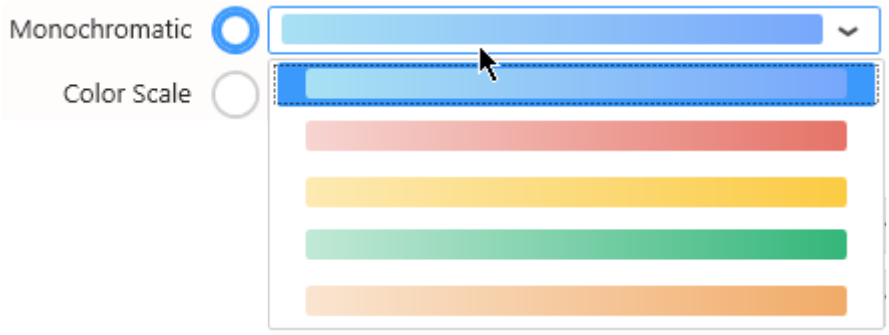
Last modified: May 14, 2019

2.8.3.4. Managing Colors

Monochromatic (Choropleth Map)

By default, Choropleth Map with **no Calculations** show value intensity as a gradient of a single color.

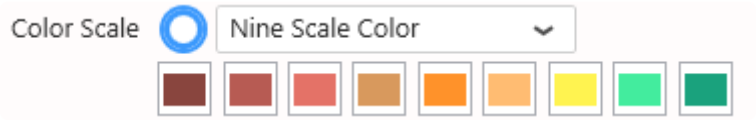
This *Monochromatic* Gradient can be set to a different color by using the dedicated combobox under *Map Settings* :



Color Scale (Choropleth Map)

Choropleth Map with no Calculations can be configured to show value intensity in ranges of single colors.

Select Color Scale and choose the number of colors and the colors of each interval :



The interval between the maximum of Value and the minimum of Value will be split in equal intervals and assigned a single color.

> This can be useful for setting a color code like “low is red, medium is orange, high is green”.

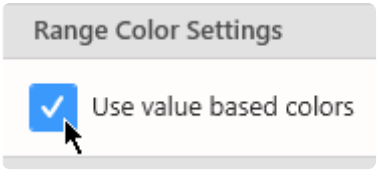
Value-based Colors (Choropleth Map)

Choropleth Map **with Calculations** show a single different color for each different value of the Calculations field.

By default, these colors are indexed from a default palette.

Instead of index-based, you may then choose **Value-based** colors : the same value in the Calculations field will always have the same color in the Map.

For this, simply check *Use value-based colors* under *Range Color Settings* :



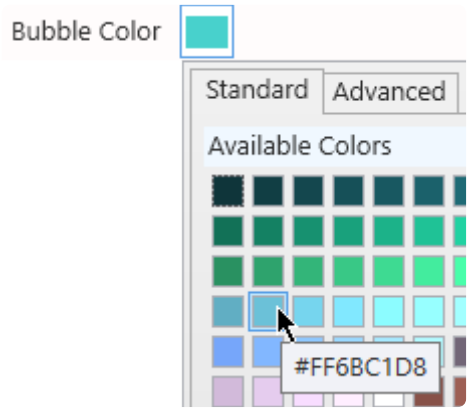
Customizing Value-based colors

Value-based colors are defined **at the Data Source level**.
This is in order to allow re-use of the colors everywhere the same data is used.

To configure value-based colors, see the corresponding chapter Value-based Settings under Transforming Data.

Bubble Color (Bubble Map)

Bubbles have a single color defined by *Bubble Color* under *Basic Settings* :



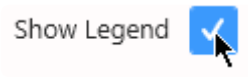
Last modified: May 14, 2019

2.8.3.5. Data Point information

Legend (Choropleth)

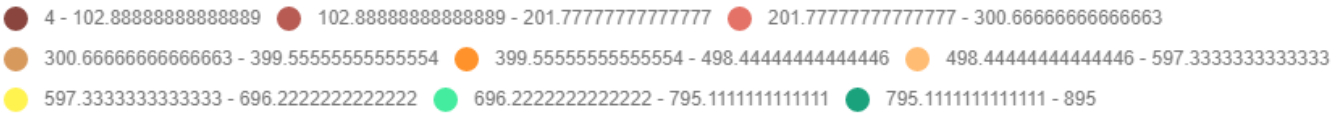
Depending on the color mode selected, the Choropleth Map can display a legend.

Click on *Show Legend* under *Basic Settings* :



In Monochromatic color, a gradient is displayed with the minimum and maximum values and an arrow for the current value :

In Color Scale mode, the scale is displayed with the minimum and maximum values and an arrow for the current value :



In Calculations mode, the legend displays the name and color of each item :

Last modified: May 14, 2019

2.8.3.6. Runtime navigation

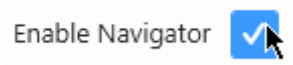
At runtime, the maps can be navigated by the user :

- zoom : mouse scroll allows to zoom in/out
- move : click and drag to move

Navigator

In addition, a Navigator can be displayed to help users navigate using mouse clicks.

Under *Basic Settings*, select *Enable Navigator* :



At runtime, 3 icons are displayed at the top left of the map :

: resets zoom level

: zoom in

: zoom out

Last modified: May 14, 2019

2.8.4. Grid

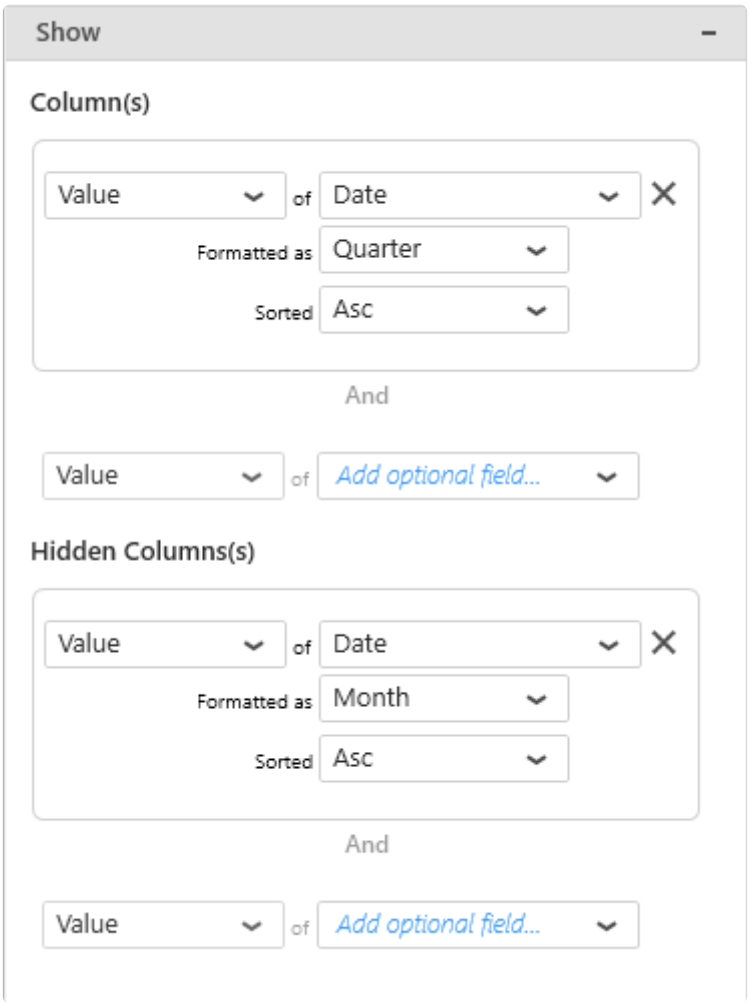
The Grid is a very powerful and simple widget.
It allows to display tabular data in many ways.

Last modified: May 07, 2019

2.8.4.1. Binding Data

Introduction

The following field containers are available for configuring data binding on a Grid :



- *Column(s)* : fields that will be Grid columns
- *Hidden Column(s)* : field that will be in the query but not visible to the user

Columns

Any number of fields of any type can be added as Grid Columns.

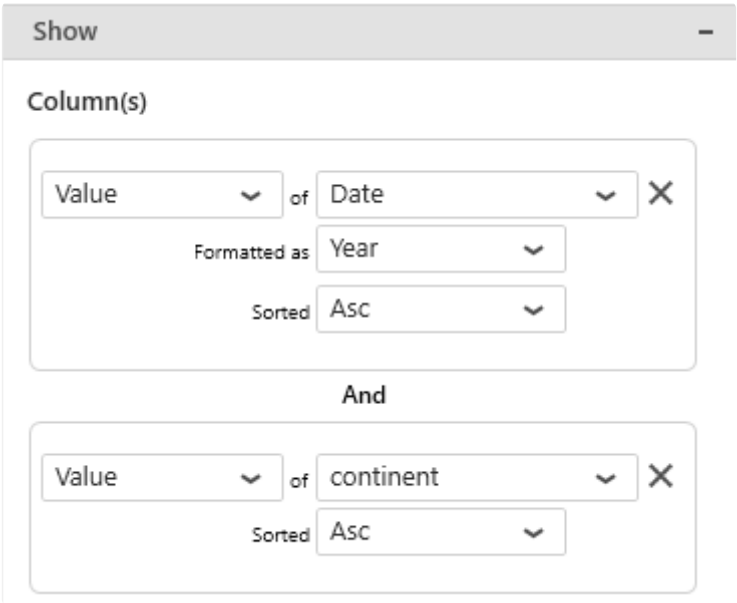
The order in the Columns container defines the order of Grid columns from left to right.

Value fields may or may not be aggregated.

Each field groups all the other fields on the right of it.
Changing the order of fields is documented here.

Example

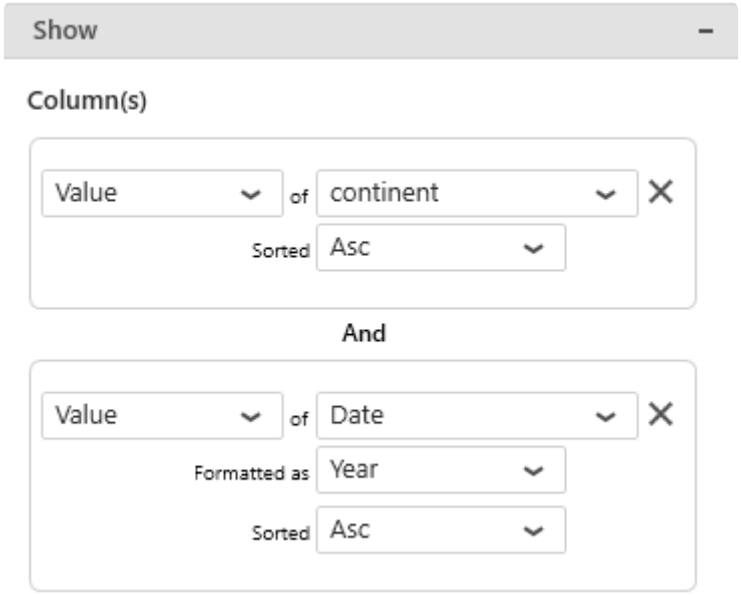
With fields Year, Continent :



The Grid shows the first Year (2016) and all its Continents,(Asia, Europe, North America), then the second Year (2017) and all its Continents, etc… :

Year(Date)	continent
2015	Asia
2015	Europe
2015	North America
2016	Asia
2016	Europe
2016	North America
2017	Asia
2017	Europe
2017	North America

With fields Continent, Year :



The Grid shows the first Continent (Asia) and all its Years (2015, 2016, 2017), then the second Continent (Europe) and all its Years, etc… :

continent	Year(Date)
Asia	2015
Asia	2016
Asia	2017
Europe	2015
Europe	2016
Europe	2017
North America	2015
North America	2016
North America	2017

Hidden Columns

Optionally, any number of fields of any type can be added as *Hidden Columns*.

Hints :
This can be useful for setting up conditional formatting based on calculations that the user doesn't need to see.
This can also be useful for grouping fields by a field where the values don't have to be

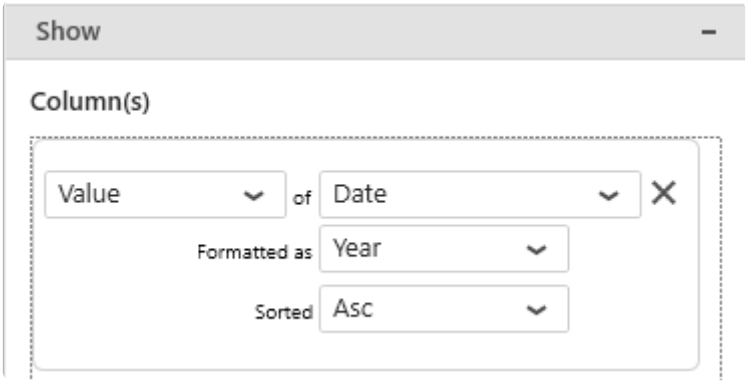
shown to the user.

The Hidden columns group the visible columns.

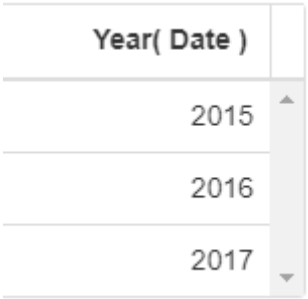
Value fields may or may not be aggregated.

Example

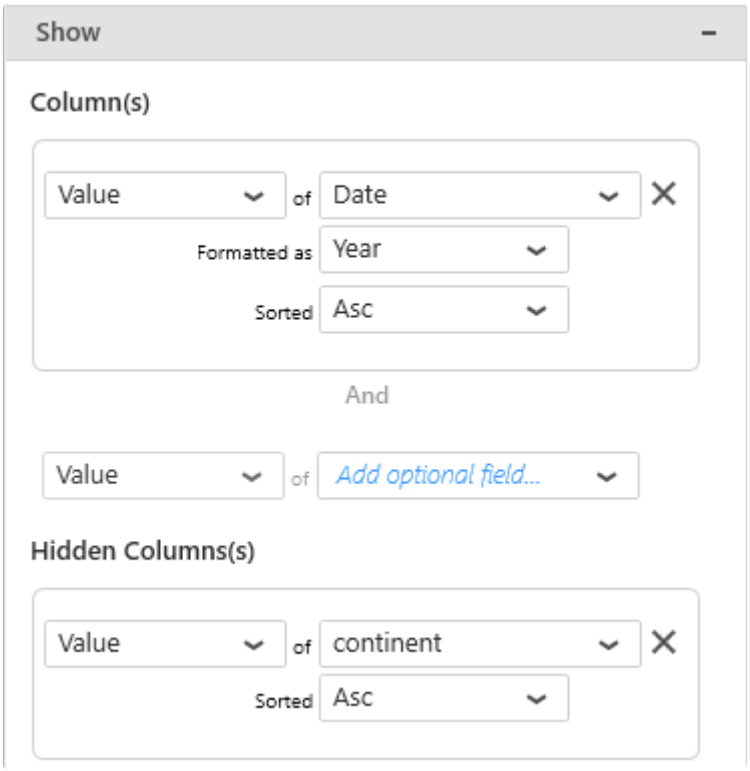
When configuring a single column called Year :



… the data is grouped, and only distinct values of Year (2015, 2016, 2017) are displayed, and only once :



When configuring additionally a hidden field Continent which has 3 distinct values :



… each Year will appear 3 times : once for each value in Continent :

Year(Date)
2015
2015
2015
2016
2016
2016
2017
2017
2017

Last modified: May 07, 2019

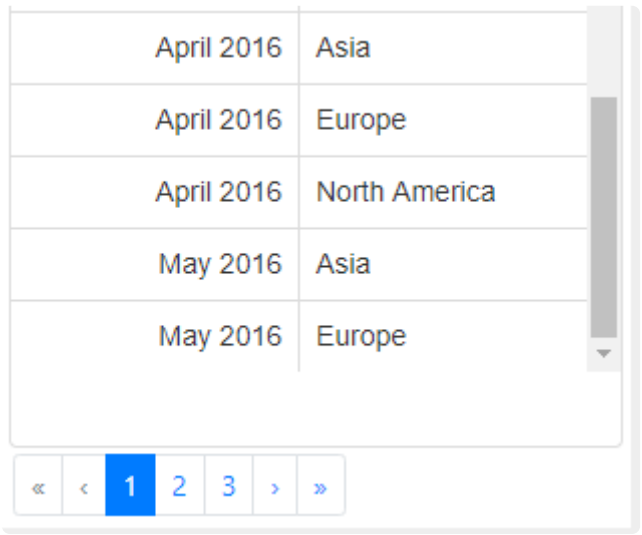
2.8.4.2. Layout and appearance

Paging

When loading a Grid Widget, all the rows of data are loaded.

The Grid can also be configured with *Paging* : it only loads pages of 50 rows (configurable), and a page navigator allows to see the rest of the data.

Check *Allow Paging* under *Basic Settings* and set a *Page Size* (default 50) :



The User can still scroll down each page (depending on the widget height), but not more than 50 rows at a time.

Then the navigator can be used to go to :

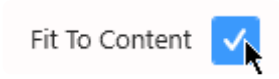
- « first page
- ‹ previous page
- 1 2 3 a specific page number
- › next page
- » last page

Column width

Grid column width is by default constant, but can be configured to adapt to the content width.

Year	Country_Full
2015	Asia - China
2015	Asia - India
2015	Europe - Denmark
2015	Europe - France
2015	North America - C…
2015	North America - U…

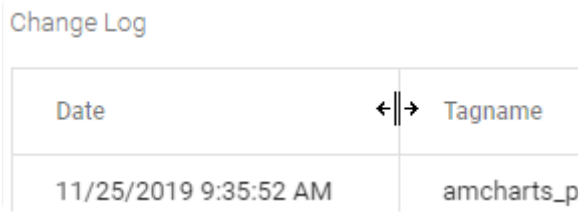
Check *Fit To Content* under *Basic Settings* :



Year	Country_Full
2015	Asia - China
2015	Asia - India
2015	Europe - Denmark
2015	Europe - France
2015	North America - Canada
2015	North America - United States

Manual resize

At runtime, the user can resize the columns.
Hover your mouse between the column headers to show a “resize” pointer :



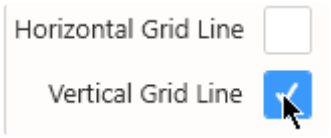
Then you can :

- drag to resize manually
- double-click to automatically resize the left column to the largest element

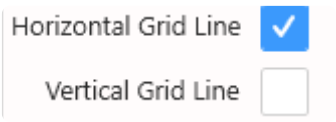
Grid Lines

Grid lines (columns/rows separators) can be displayed or hidden.

Check the corresponding *Horizontal_/_Vertical* option :



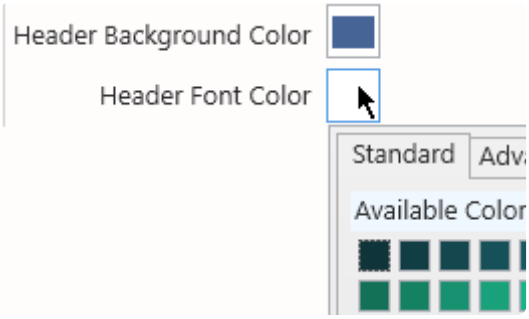
Month Year(Date)	continent
January 2015	Asia
January 2015	Europe
January 2015	North America



Month Year(Date)	continent
January 2015	Asia
January 2015	Europe
January 2015	North America

Column Headers style

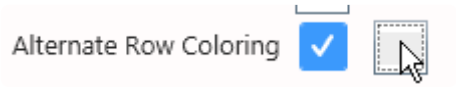
The visual style of widgets is inherited from the currently applied Theme.
You can override the column headers style by selecting a color for their background and font :



id	name
1.00	name 1
2.00	name 2

Alternate Row Coloring

For better readability, you can configure a secondary row color to display only on the odd-numbered rows.
Check *Alternate Row Coloring* and select a color :



Canada	54.00
China	87.00
Denmark	18.00
France	14.00

Last modified: Dec 23, 2019

2.8.4.3. User interaction

Note : Grid as Master Widget

As most other Widget types, the Grid can be configured as a Master Widget to filter other widgets.

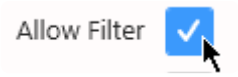
Additionally, the Grid allows you to make multiple selections : maintain the keyboard `Ctrl` key while clicking Grid rows to select multiple values.

Manual Filtering

Users can be allowed to filter column contents at runtime based on a text value.

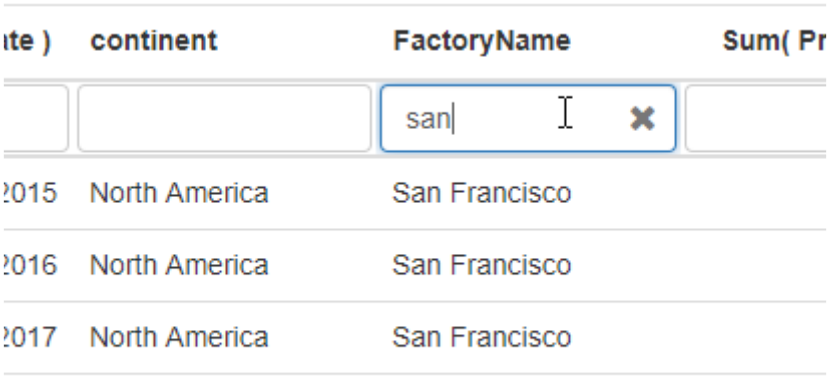
Configuring Manual Filter

In the *Basic Settings* section, select *Allow Filter* :



Applying Manual Filter

At runtime, a text box appears under each column name, allowing users to filter by this column content :



Only values that **start with** the input text is filtered.

> ! If paging is enabled, only the current page is filtered. To see the result across all pages, navigate to the corresponding page.

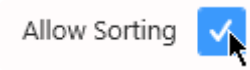
Clearing Manual Filter

To clear the filter, the User can click on the cross of that text box :

Manual Sorting

When loading a Grid widget, sorting order depends on data binding and sorting configuration.

However, when manual sorting is allowed, the User can click on column titles to manually sort the Grid on that column :
To enable this feature (enabled by default), un-check *Allow Sorting* under *Basic Settings* :



Single sorting

Clicking the column header will cycle between 3 states :

- not sorted
- sorted ascending
- sorted descending

Example

Here is the default sorting : Year ascending, continent ascending, (Production ascending) :

Year(Date)	continent	Sum(Production)
2015	Asia	59.86M
2015	Europe	11.75M
2015	North America	38.80M
2016	Asia	60.21M
2016	Europe	11.71M
2016	North America	38.72M
2017	Asia	59.99M
2017	Europe	11.70M
2017	North America	38.65M

When the User clicks on continent, the Grid is now sorted by continent in ascending alphabetical order :

Year(Date)	continent ↑	Sum(Production)
2015	Asia	59.86M
2016	Asia	60.21M
2017	Asia	59.99M
2015	Europe	11.75M
2016	Europe	11.71M
2017	Europe	11.70M
2015	North America	38.80M
2016	North America	38.72M
2017	North America	38.65M

When clicking a second time, the column is now sorted descending :

Year(Date)	continent ↓	Sum(Production)
2015	North America	38.80M
2016	North America	38.72M
2017	North America	38.65M
2015	Europe	11.75M
2016	Europe	11.71M
2017	Europe	11.70M
2015	Asia	59.86M
2016	Asia	60.21M
2017	Asia	59.99M

Number fields can also be sorted manually :

Year(Date)	continent	Sum(Production)
2016	Asia	60.21M
2017	Asia	59.99M
2015	Asia	59.86M
2015	North America	38.80M
2016	North America	38.72M
2017	North America	38.65M
2015	Europe	11.75M
2016	Europe	11.71M
2017	Europe	11.70M

Multiple sorting

Using the `Ctrl` keyboard key and clicking another column header (`Ctrl`+click) will combine the sorting clauses in the correct order.
A number icon indicates the order of the sorting clauses.

Example

1. Clicking on UP twice sorts it in descending order,
2. `Ctrl`+click on Count sorts by “UP descending” first, then “Count ascending”
3. `Ctrl`+click on Line twice sorts by “UP descending” first, then “Count ascending”, then “Line descending”

Last modified: Oct 30, 2019

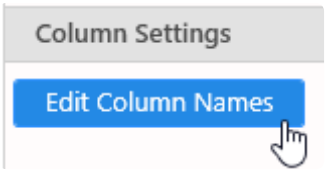
2.8.4.4. Column names

Edit Column Names

Columns are named automatically based on the corresponding field name and aggregation function.

These default names can be replaced by custom name for any column.

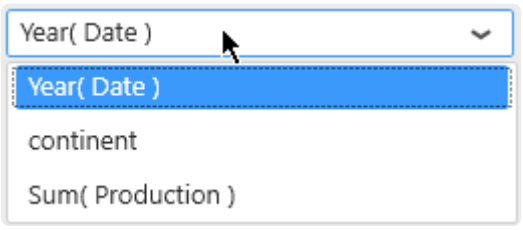
Under *Column Settings*, click *Edit Column Names* :



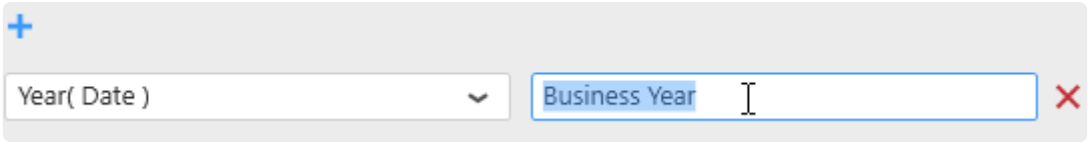
Rename column

In the Edit Column Names dialog, click the **+** icon to add a column name customization :

From the new customization item, select a column name to rename :



Write a new name for the column :



Reset column name

To reset a column name to its original (auto-generated) name, click the red cross on the right of the corresponding item :

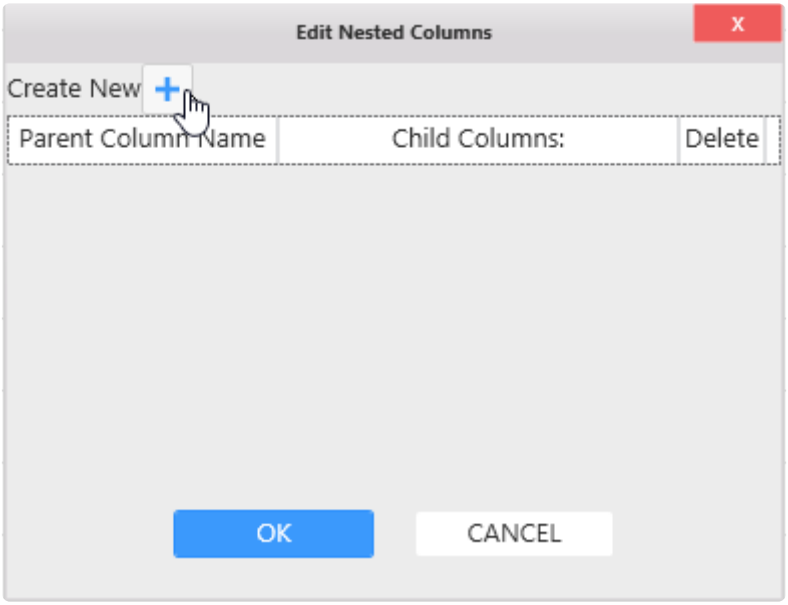
Nested Columns

You can group Grid columns and configure a common title on top of the columns titles.
The top title is called a “Parent Column”, and the regular Grid column titles are called “Child Columns” of this Parent Column.

Under *Column Settings*, click *Edit Nested Columns* to open the editor.

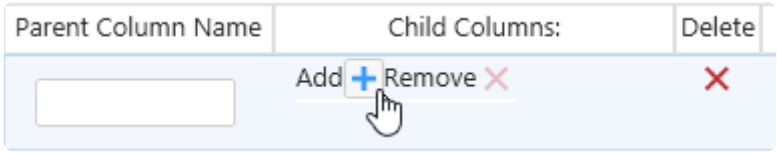
Create a Nested Column

Click *Create New* to start creating a new Nested Column :



A new un-configured row appears, and lets you configure :

- *Parent Column Name* : the text label you want to use on the Parent title
- *Child Columns* : the list of children columns that are nested under this Parent Column. This is where you can list, add, or remove children
- *Delete* : click to delete this Parent column.

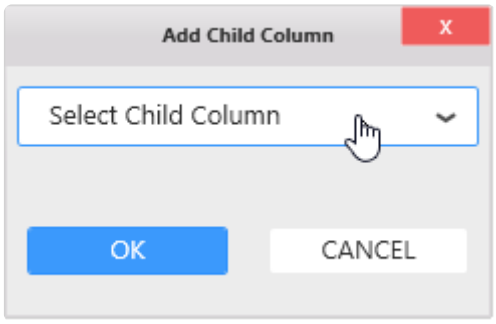


You can create several separate Nested Columns.

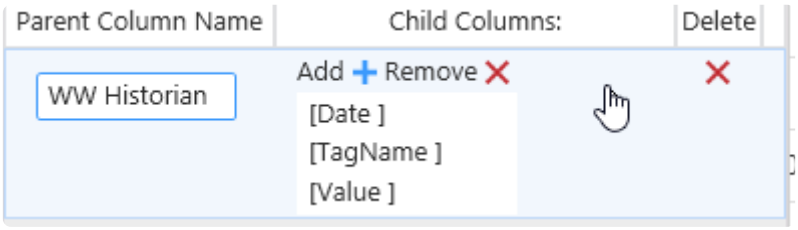
Add / Remove Children

Under *Child Columns*, click *Add* to select which current Grid columns you want to nest under this Parent Column.

The *Add Child Column* dialog appears, to let you select an available Column from the Grid :

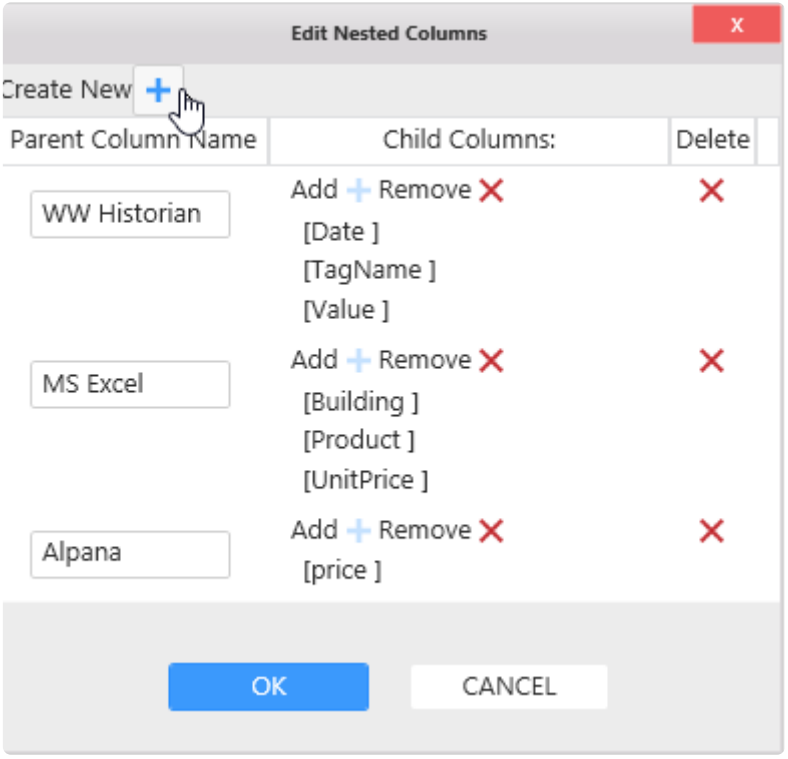


The column names are then listed in the *Edit Nested Columns* dialog :



Example

In this example, columns are grouped by data origin :



WW Historian			MS Excel			Alpana
Date	TagName	Value	Building	Product	UnitPrice	price
12/19/2019 3:29:54 PM	FT101.Value	24.69K	Building A	Product X	2.00	49.38K
12/19/2019 3:29:54 PM	FT102.Value	1.02K	Building A	Product Y	10.00	10.16K
12/19/2019 3:29:54 PM	FT201.Value	906.06	Building B	Product X	2.00	1.81K

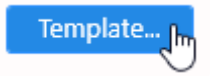
Last modified: Dec 19, 2019

2.8.4.5. Value display

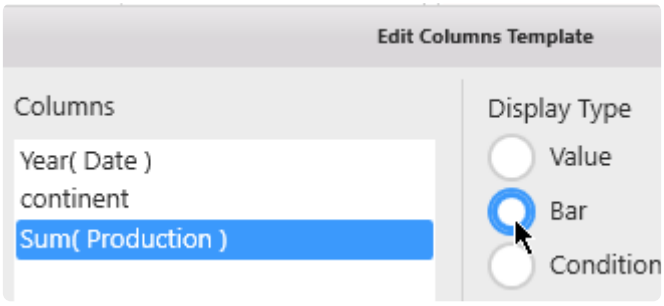
Bar graph

Value fields can be displayed as horizontal bar graphs.

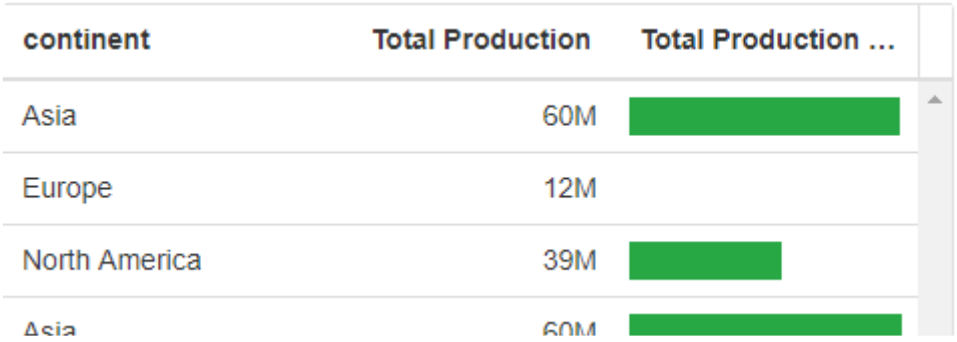
For this, click *Template* under *Column Settings* :



In the Edit Columns Template dialog, select a **numerical** column from the left, and change its display type from Value (default) to Bar :



The value is now displayed as a green horizontal bar in each cell of the column :



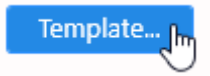
Last modified: May 07, 2019

2.8.4.6. Conditional Formatting

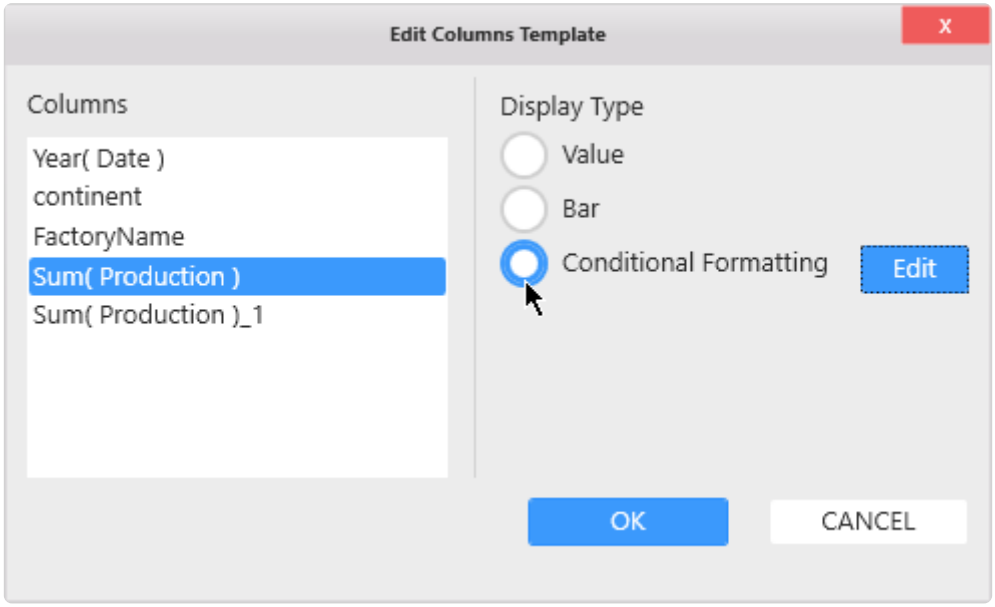
Grid cells can be formatted with color and text style based on data conditions.

Access *Conditional Formatting*

For this, click *Template* under *Column Settings* :

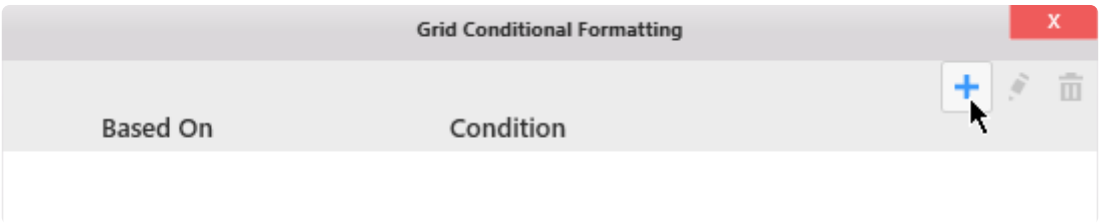


In the *Edit Columns Template* dialog, select on the left a a column where you want to apply a formatting, then select *Conditional Formatting* and click *Edit* :



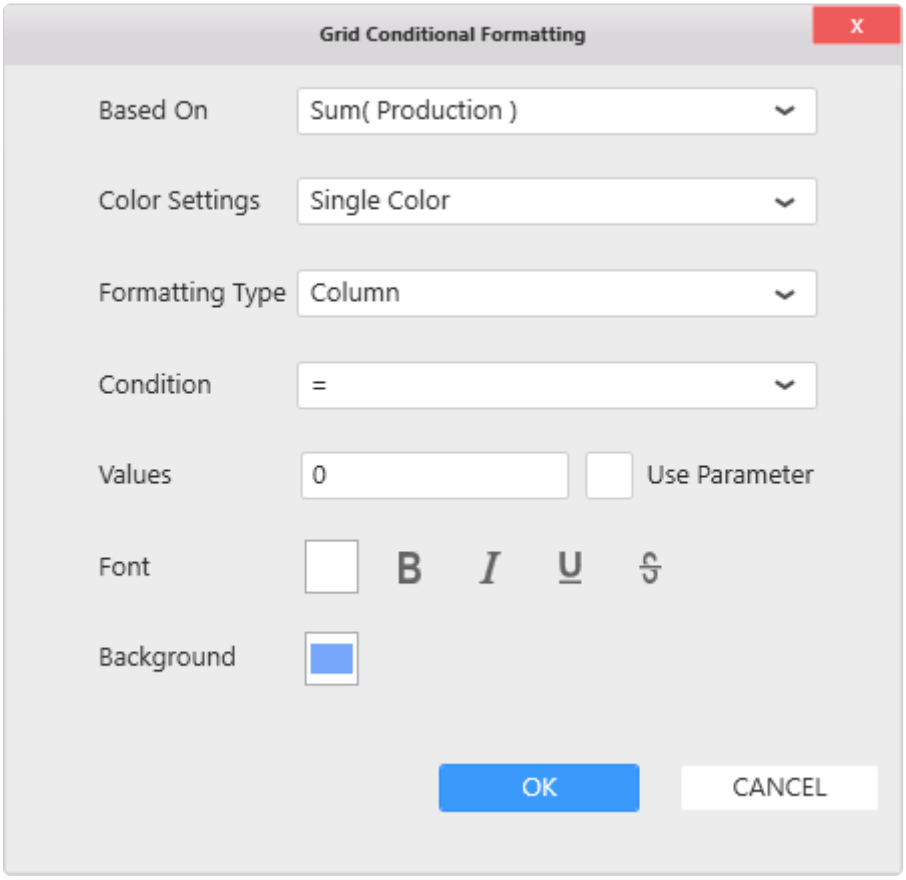
Add a Condition

In the *Grid Conditional Formatting* dialog, click **+** to add a first Condition :



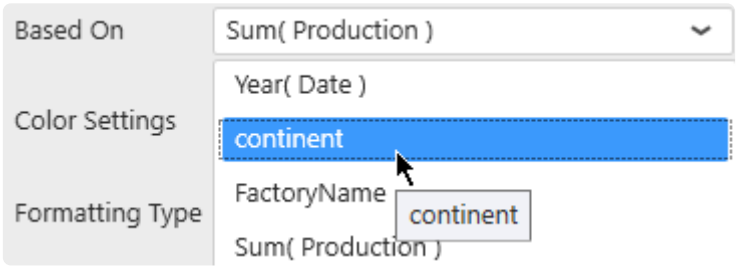
Configure a Condition

A new *Grid Conditional Formatting* dialog appears for this new Condition :



Based On column

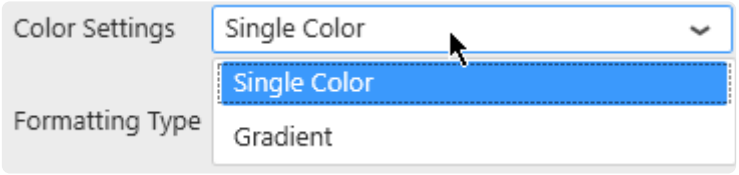
The Based On column selector allows to select which column will be used to calculate the condition :



> Hint :
> A Condition can be based on a different column than the one the formatting is applied to. It can also be based on a Hidden Column. This allows to base the condition on a complex Expression, but not show it to the end user.

Color Settings

Two color modes are possible :



- *Single Color* : based on a boolean condition, a single color is wholly applied or not, depending if the row meets the condition
- *Gradient* : based on a numerical field, a color gradient is applied to all cells, with color intensity depending on the magnitude of the number

Formatting Type

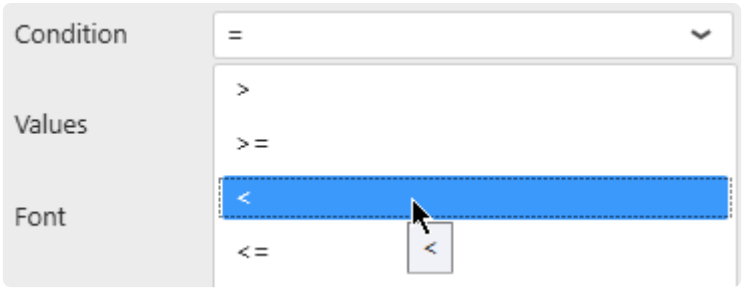
The formatting can be applied either :

- *Column* : on the cells of the selected column only
- *Row* : on all the cells of the same row

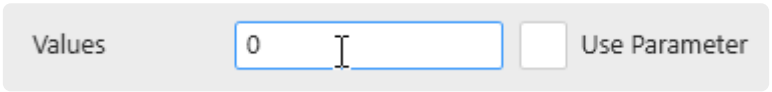
Single Color : *Condition*

For *Single Color* mode, the boolean condition is configured as : `Column Operator Value`
Example : `AlarmLevel >= 100`

The operator can be selected from the *Condition* combobox :

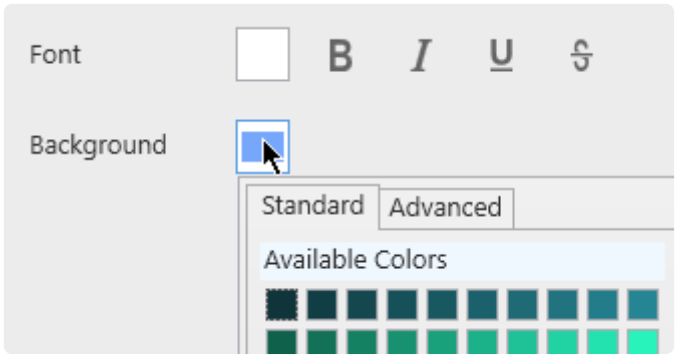


The value to compare to can be entered in the text field called *Values* or a Dashboard Parameter can be used instead :



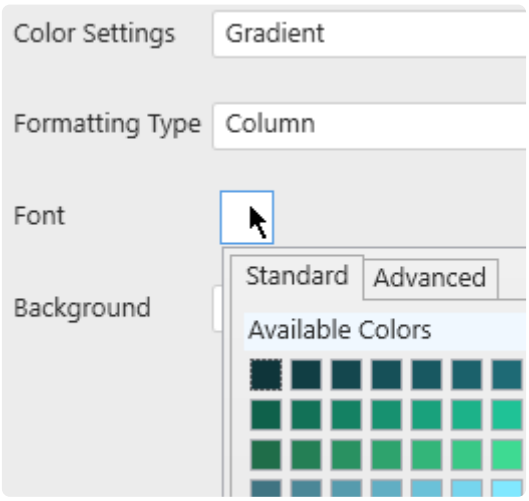
Single Color : Style

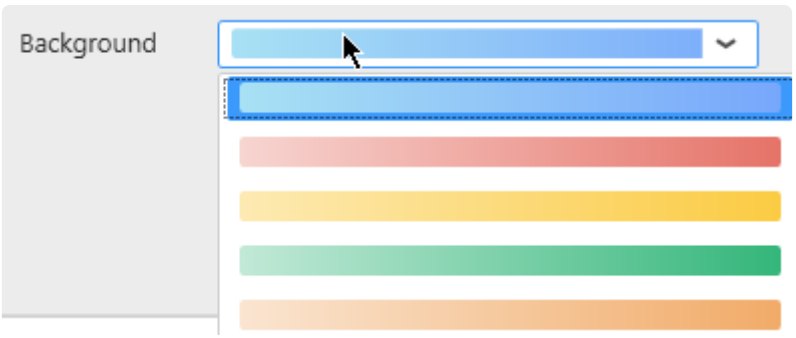
For *Single Color* mode, the style applied includes font color and style (**Bold**, *Italic*, Underlined, ~~Stricken~~) :



Gradient : Style

For *Gradient* color mode, the style applied includes font color and gradient color :





Condition Order

Conditions will be applied in the same order as the list of conditions.

Example

The following list of conditions will apply the widest condition last (>1M), which will override any formatting done in the previous narrower condition (>10M) :

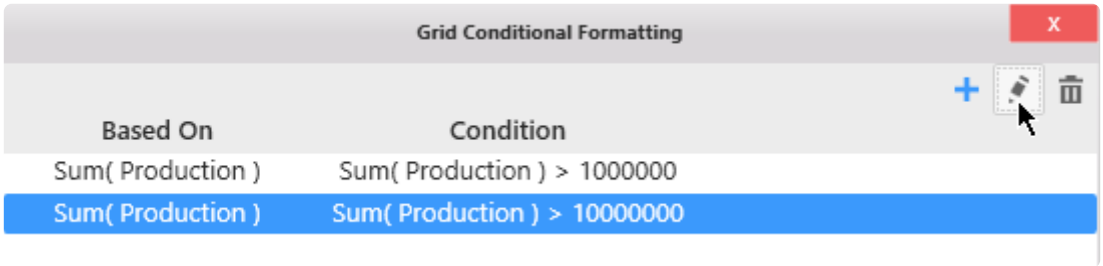
Based On	Condition
Sum(Production)	Sum(Production) > 10000000
Sum(Production)	Sum(Production) > 1000000

In order to apply a different color to the narrower condition (>10M), it needs to be listed last :

Based On	Condition
Sum(Production)	Sum(Production) > 1000000
Sum(Production)	Sum(Production) > 10000000

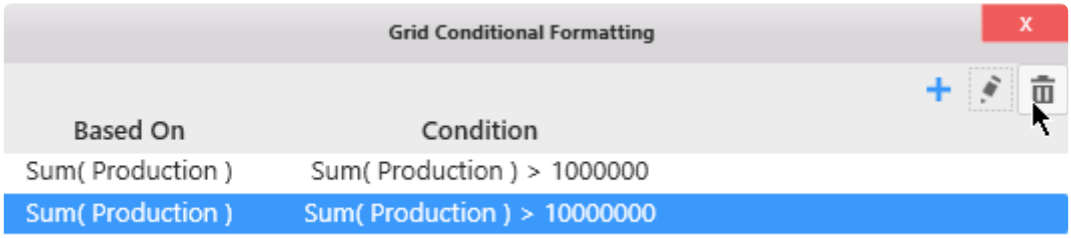
Editing a Condition

Click on a Condition and click on the pen icon to edit :



Deleting a Condition

Click on a Condition and click on the trash icon to delete :



Last modified: May 07, 2019

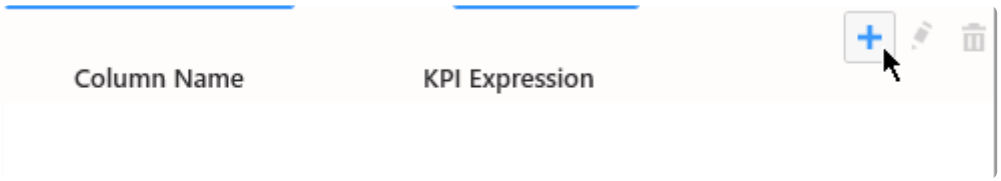
2.8.4.7. Grid KPI

Introduction

A **KPI Expression** (a new calculated column) can be generated to show how a value deviates from a target.

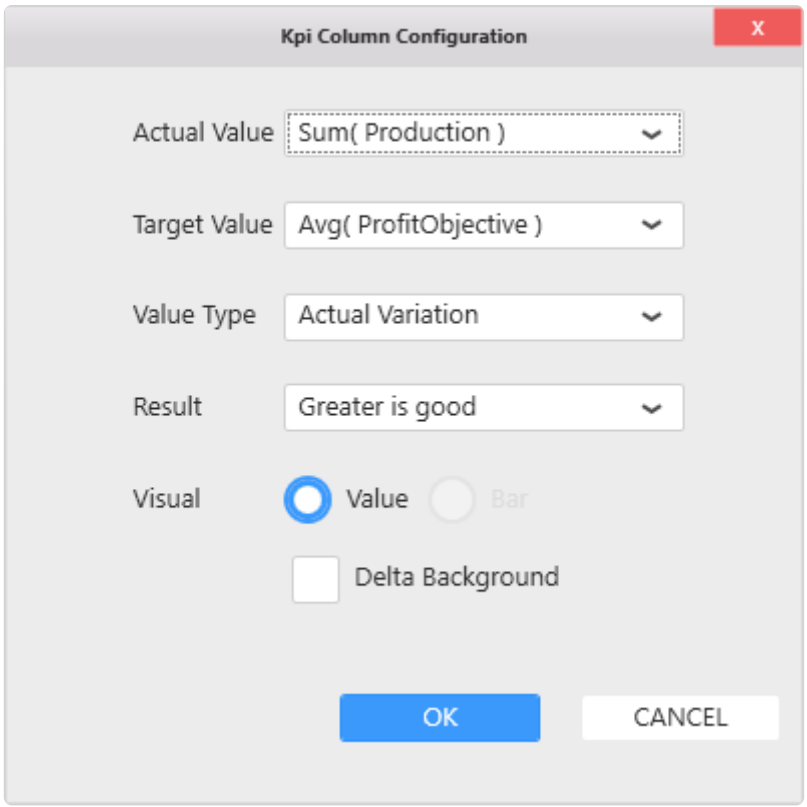
Creating a KPI

To create a KPI, navigate under section *Column Settings* near the list of KPI Expressions and click the **+** icon :



Configuring a KPI

In the *KPI Column Configuration* dialog that appears, you can configure the KPI :

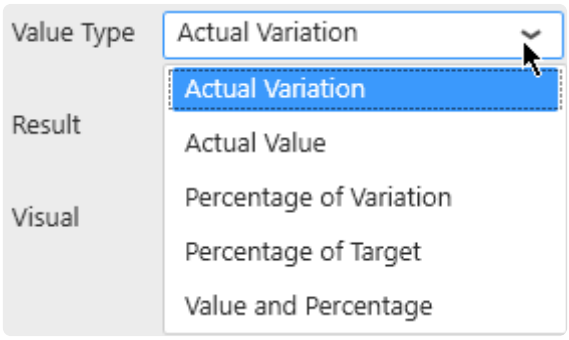


- *Actual Value* : the column that contains the value you want to track
- *Target Value* : the column to which you want to compare the Actual Value
- *Value Type* : what type of calculation is made to show the deviation

- *Result* : should we consider higher values as a good thing (green) or not (red)
- *Visual* : do we want to visualize the value as a number or a bar graph
- *Delta Background* : do we want to apply conditional color to the numbers or to the cell background

Value Types

Click to select a *Value Type* : what type of calculation is made to show the deviation :



Actual Variation

Shows as a single value the difference between the Value and the Target.

```
Calculation = Value - Target
```

Actual Value

Shows as a single value the Actual Value itself

```
Calculation = Value
```

Percentage of Variation

Shows as a single value the % of variation to target.

```
Calculation = 100 * (Value - Target) / Target
```

Percentage of Target

Shows as a single value the % of value to target.

```
Calculation = 100 * Value / Target
```

Value and Percentage

Shows 2 values in the cell :

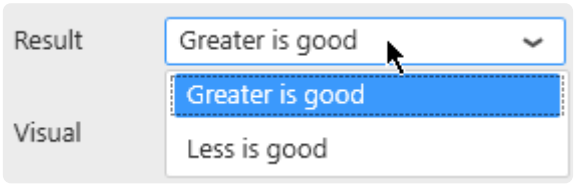
- *Actual Variation*
- *Percentage of Variation*

Result interpretation

The font and symbol of the KPI has a color :

- **green** for “good”
- **red** for “bad”

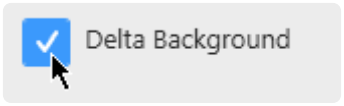
You can define high values as being “good” or “bad” using the Result configuration :



Style : background

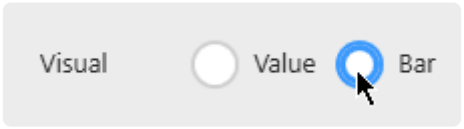
By default, the good/bad color is applied to the font, and the cell is white.

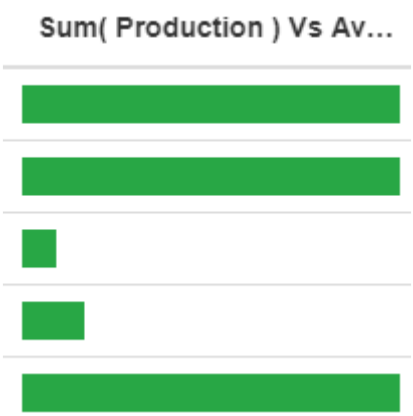
This can be configured as the opposite, for better visibility. Check *Delta Background* :



Style : Bar chart

Instead of showing the value as a number, the Actual Value calculation can be displayed as a Bar chart by selecting the appropriate Visual :





Last modified: Dec 19, 2019

2.8.4.8. Footer

Introduction

Footers are configured under *Footer Settings* and allow to create summary rows at the bottom of the Grid.

> Note : footers are disabled when Paging is enabled.

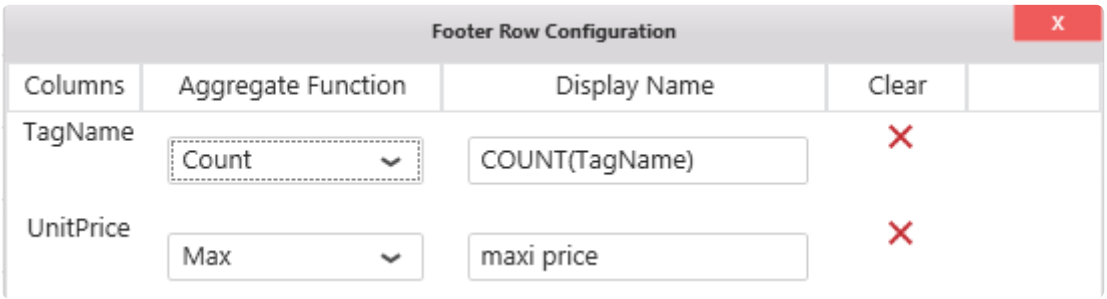
Creating a Footer

To create a Footer, navigate under *Footer Settings* and click the + icon :

Configuring a Footer

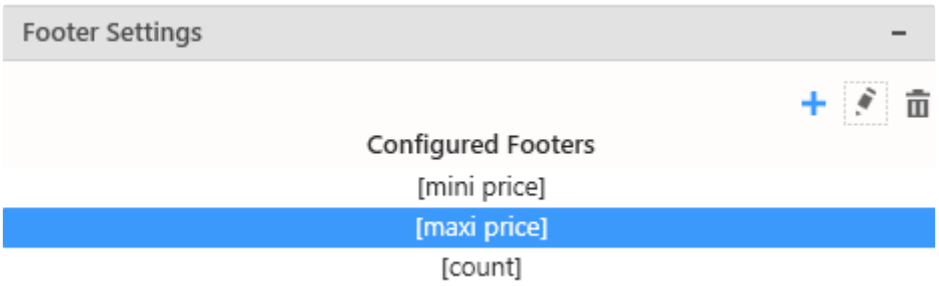
In the *Footer Row Configuration* dialog that appears, you can configure the Footer :

- *Columns* : the column names from the Grid
- *Aggregate Function* : choose a function to make the desired calculation
- *Display Name* : configure the text label
- *Clear* : use the red cross button to reset your configuration



Managing Footers

From the list of *Configured Footers*, you can Edit or Delte existing footers, by clicking the corresponding Footer name and action icon :



Result

The Footers are displayed at the bottom on the Grid :

TagName	UnitPrice
FT101.Value	2
FT102.Value	10
FT201.Value	2
FT202.Value	10
FT301.Value	2
FT302.Value	10
	mini price: 2
	maxi price: 10
	count: 6

Last modified: Dec 19, 2019

2.8.4.9. HTML in the Grid

Introduction

The Grid Widget will display any text content as HTML (provided there is no conflict with the page structure, styles, scripts, etc).

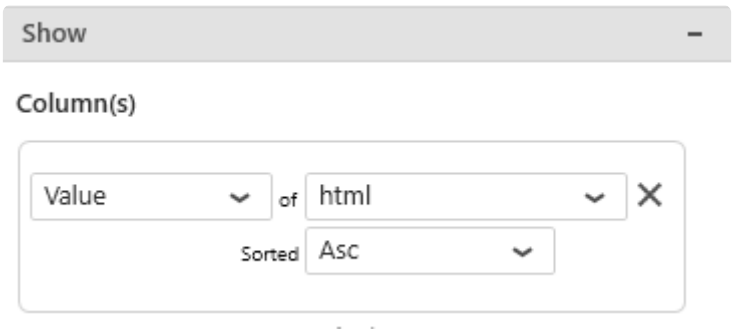
There is no clear boundaries of what is supported and what might break a dashboard, and you should make tests for the HTML features you want to use.

General example

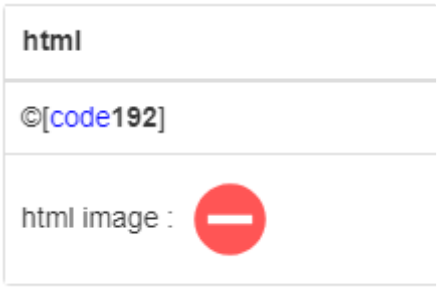
Your data contains a text field (called “html” here) with some HTML markup :

	A	B	C	D	E
1	html				
2	©[code192]				
3	html image : <img src="				
4					
5					

You can bind this field in the Grid :

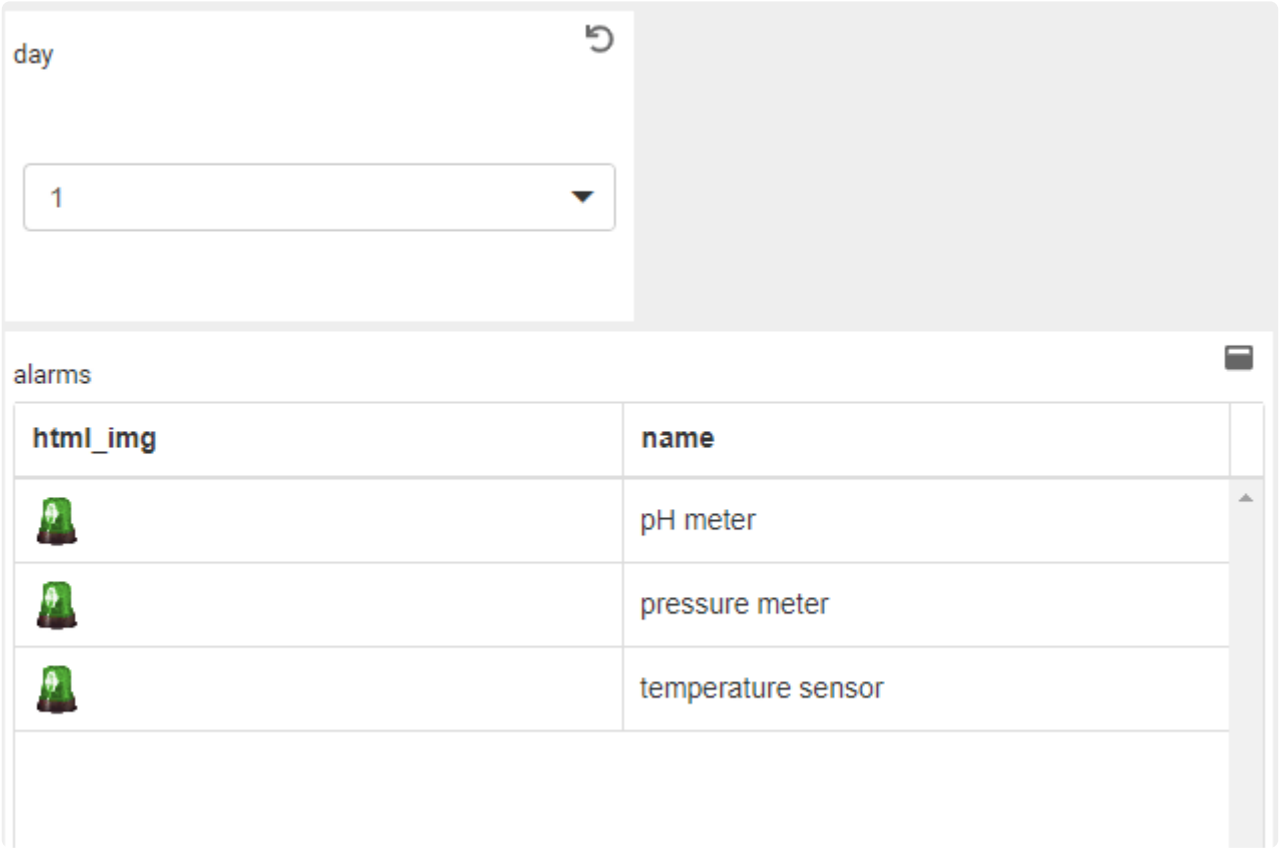


… and the HTML will be interpreted :



(this is from the same data as above)

This can be used for many customization cases, including displaying animated GIFs conditionally :



Last modified: May 07, 2019

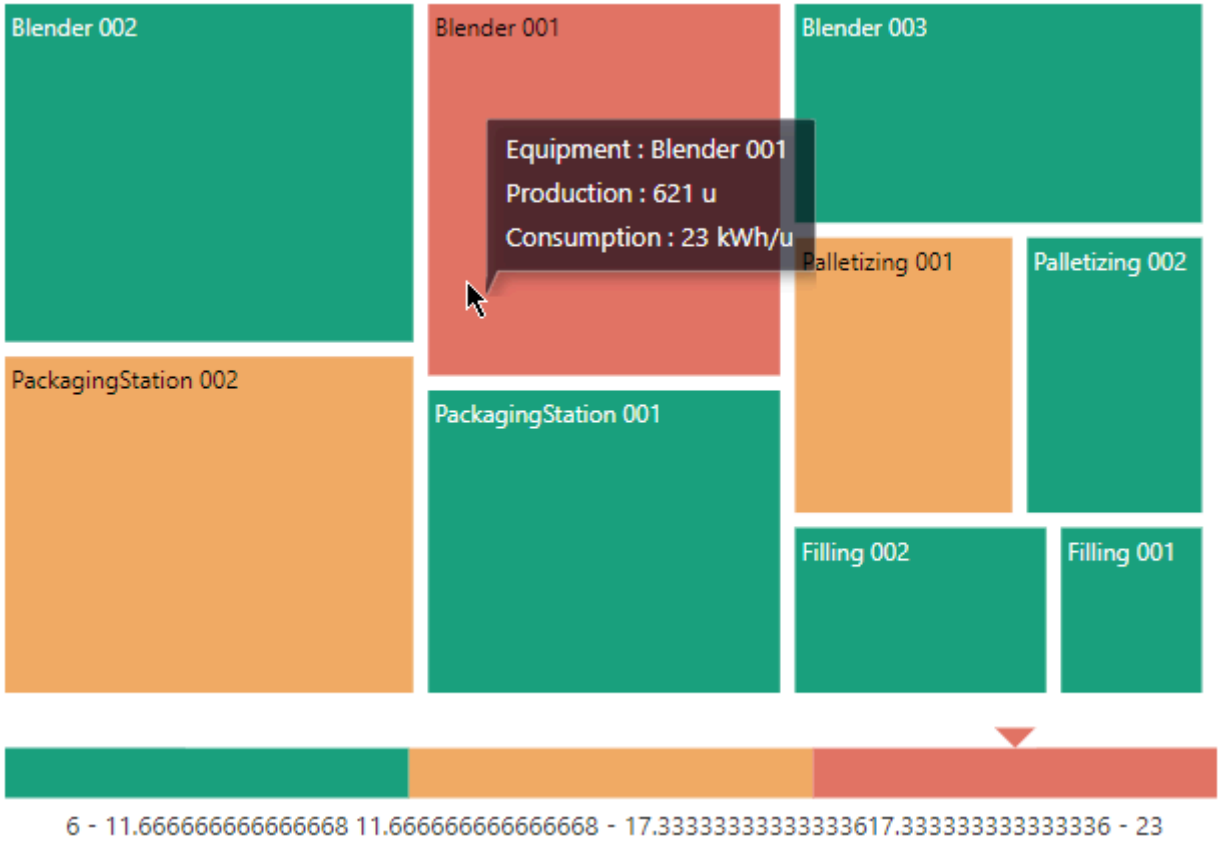
2.8.5. TreeMap

The TreeMap widget can be used as an extension of Proportion Charts : data is grouped in rectangles whose size and color can be bound to different data fields.

In a typical case, the size represents the importance, and the color represents the KPI itself.
This can be useful typically for showing the efficiency of something combined with the quantity.

Example :

Each rectangle is grouped by Equipment. The size of rectangles is bound to the Quantity produced, the color is bound to the energy Consumption.



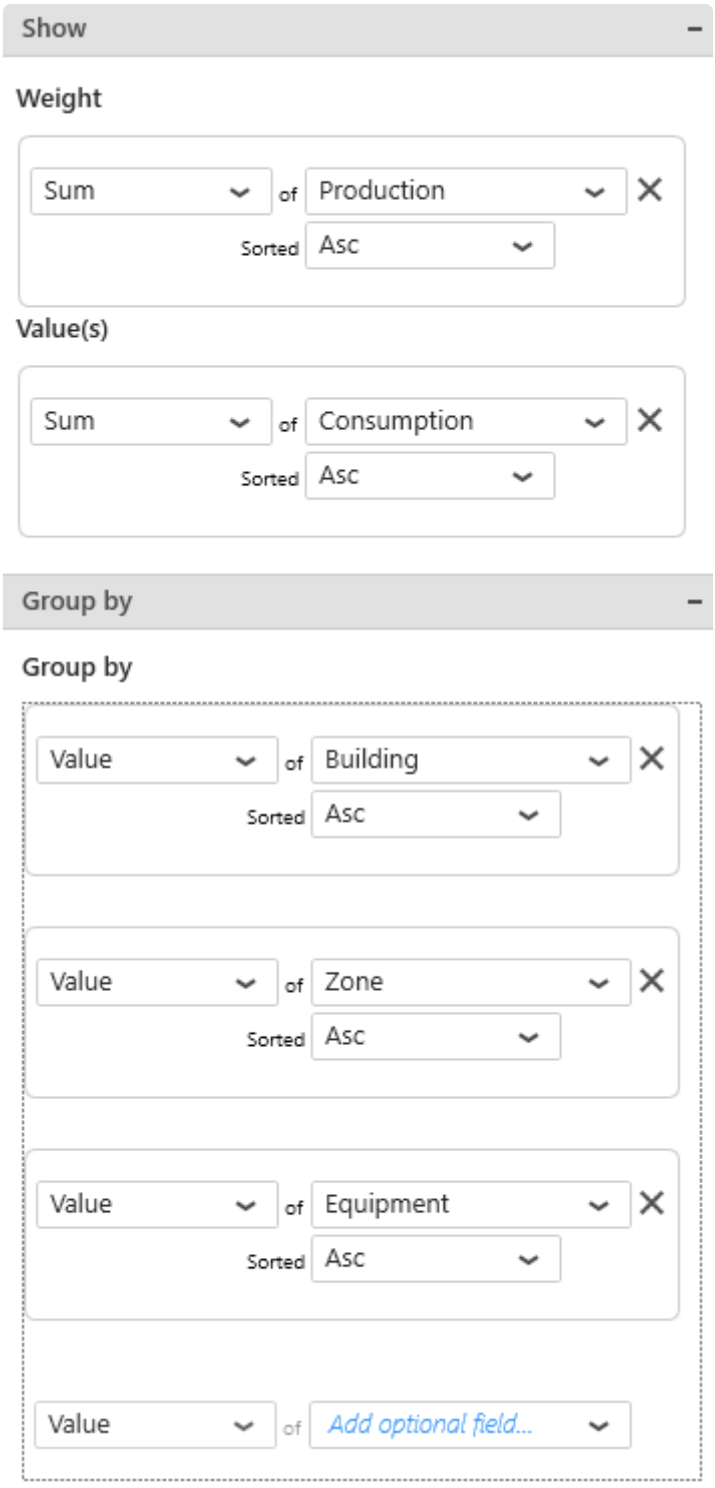
In the above TreeMap, we can see that the Blender_001 has very bad energy efficiency, but represents only a small fraction of the total production.

Last modified: May 14, 2019

2.8.5.1. Binding Data

Introduction

The following field containers are available for configuring data binding on a TreeMap :



- *Weight* : the size of the rectangles. If no “Value” is configured, this is also the color.
- *Value* : the color of the rectangles
- *Group By* : how we group values together to create separate rectangles

Weight

This represents the quantity we want to display on the TreeMap, and will bind for rectangle size.
If no "*Value*" field is configured, this field also binds to the color of the rectangles.

This field **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)
This field is required.

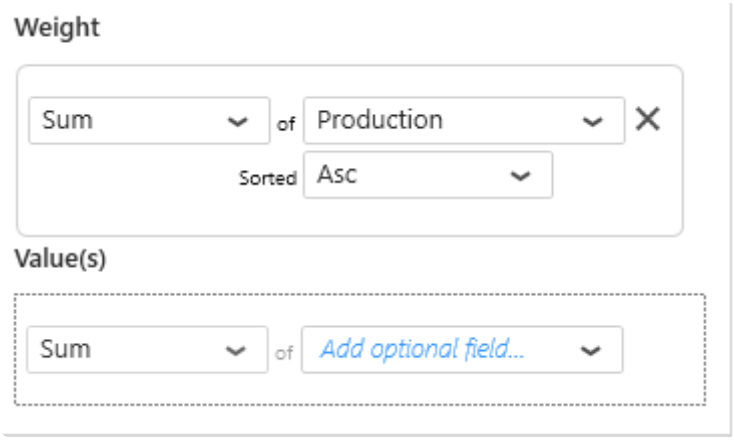
Values

When a *Values* field is bound, it defines the color of the rectangles, while the first field now only defines the size of rectangles.

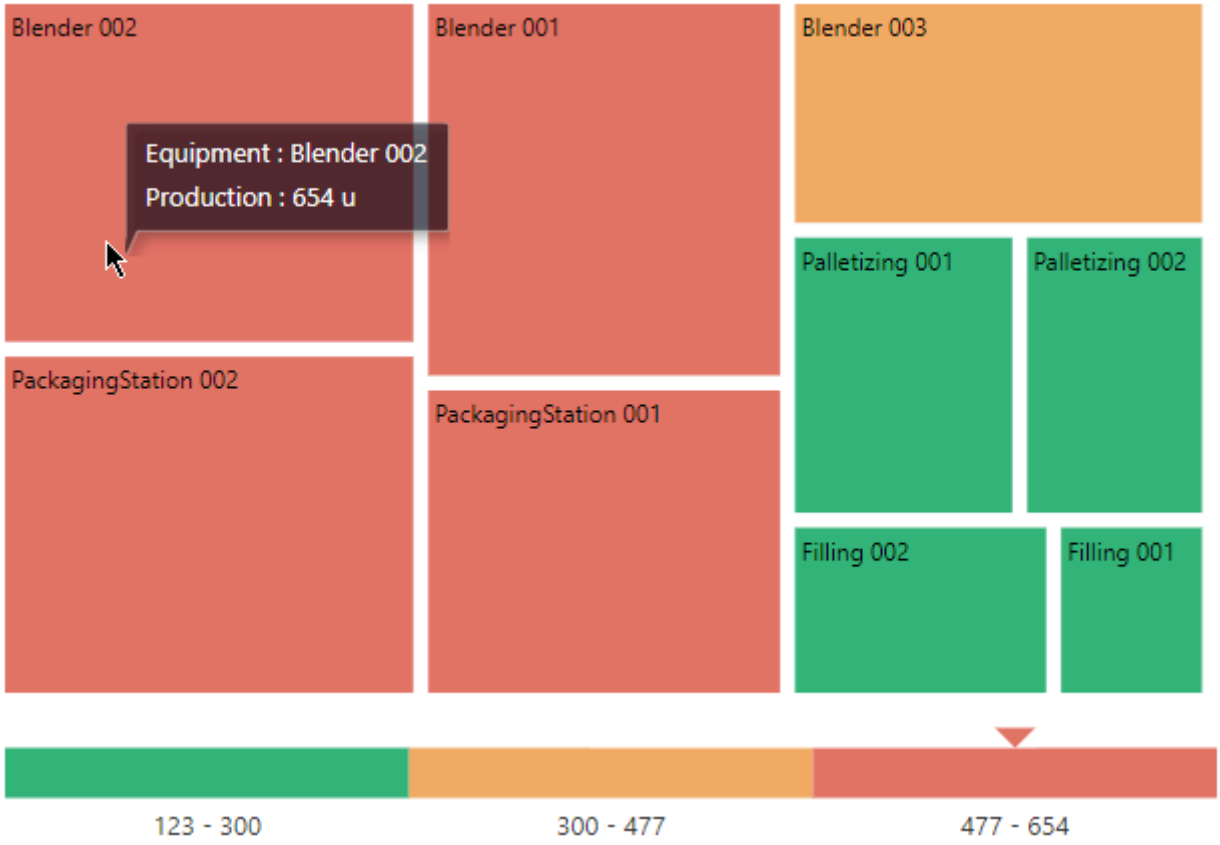
This field **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)
This field is optional.

Example

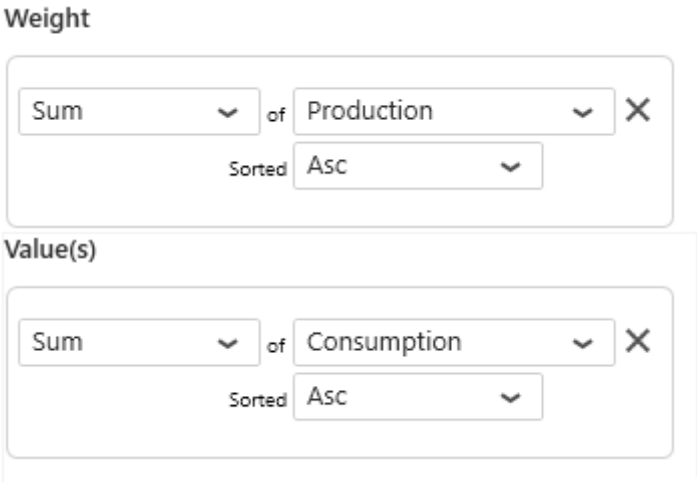
For example, you configure a single field "Production" :



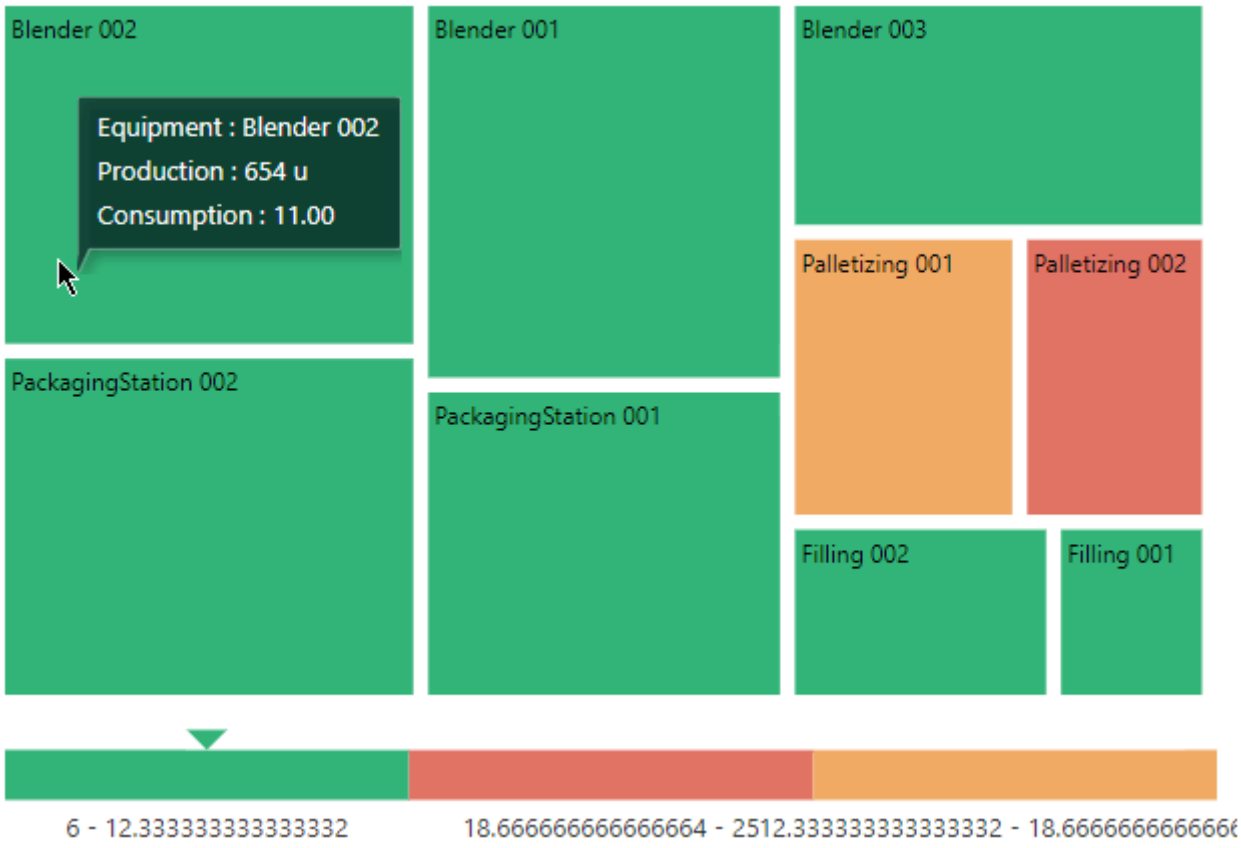
It defines both the size and color of rectangles.
So for example Equipment "Blender_002" has value 654 which is high (large and red) :



Now you configure as secondary field a “Consumption” :



Now the rectangles keep the same size (the *Weight* field is the same), but the color is different.
So for example Equipment “Blender_002” still has Quantity 654 which is high (large), but now the color is green because the secondary “Consumption” is low (11) :



Group By

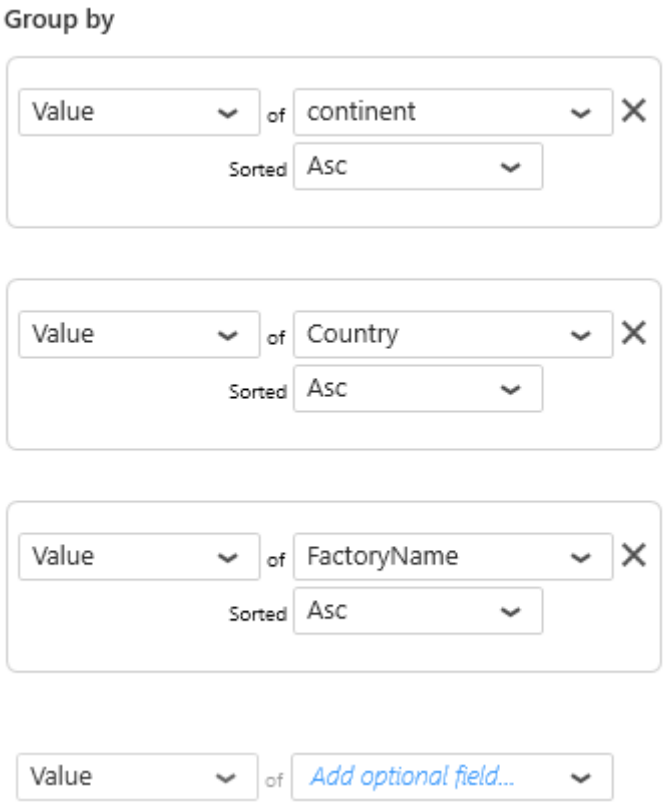
This represents the category we want to use to separate rectangles.

These fields **cannot be aggregated** : they must have several separate values in order to group the “Values” aggregation.

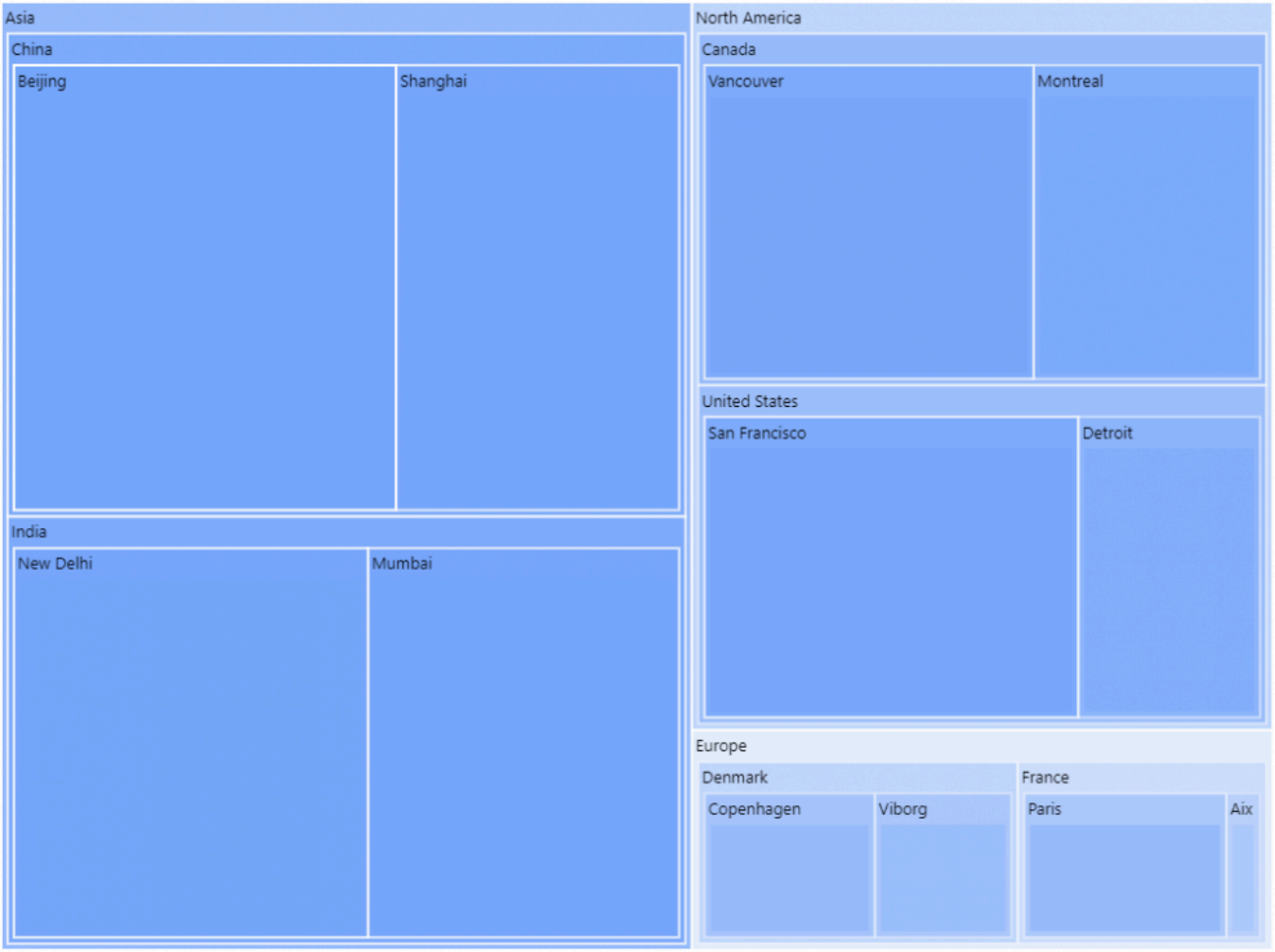
Multiple fields : Grouping

When binding multiple fields at the same time, the rectangles will be grouped by the defined hierarchy.

For example, binding fields Continent>Country>Factory :



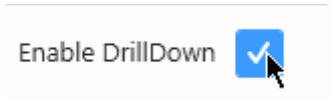
… will group factories by country and the countries by continent :



Multiple fields : Drill-Down

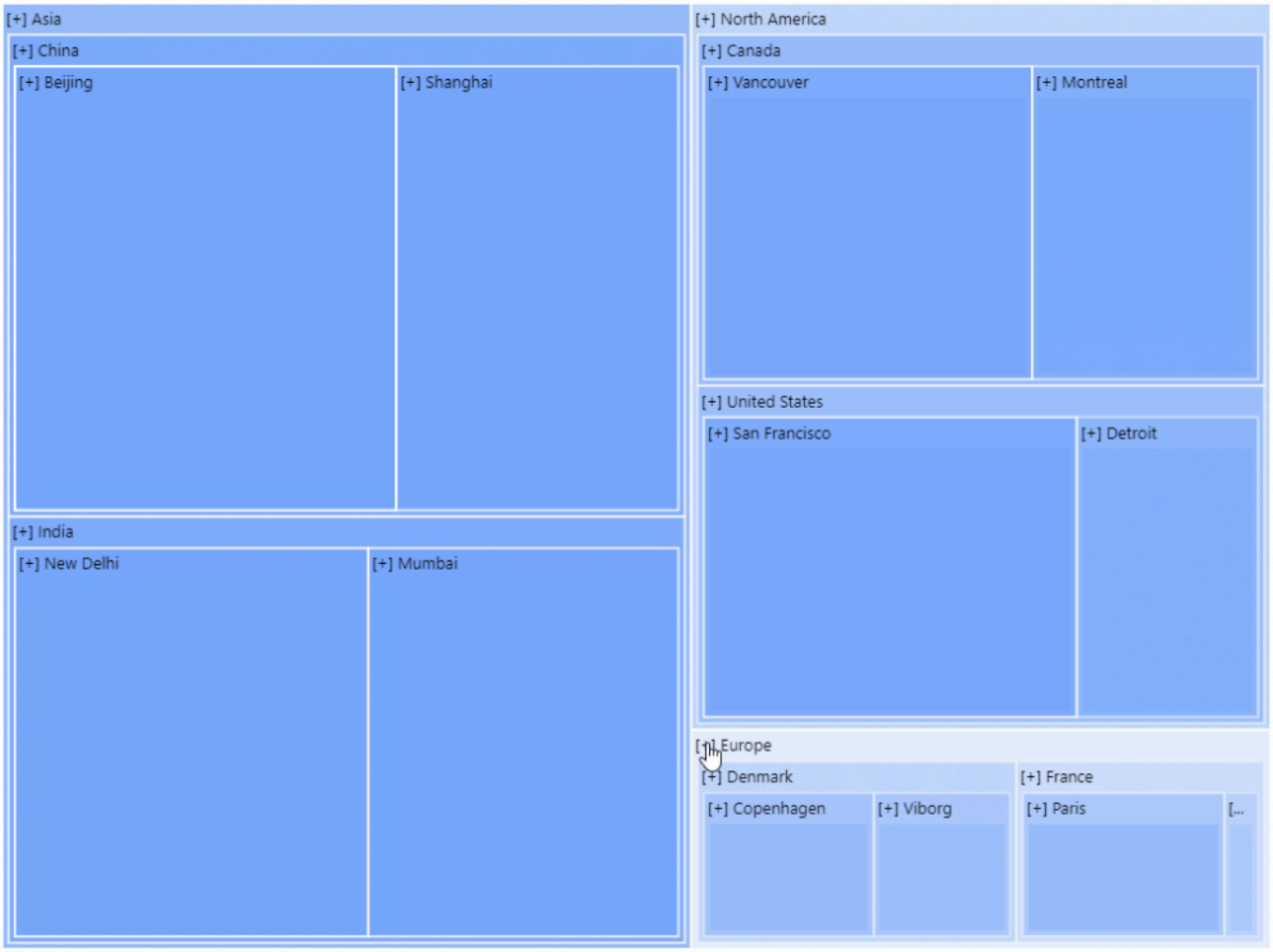
When binding multiple fields at the same time, it is possible instead to define Drill-Down : the user will navigate the hierarchy by clicking.

Under Basic Settings, select Enable DrillDown :

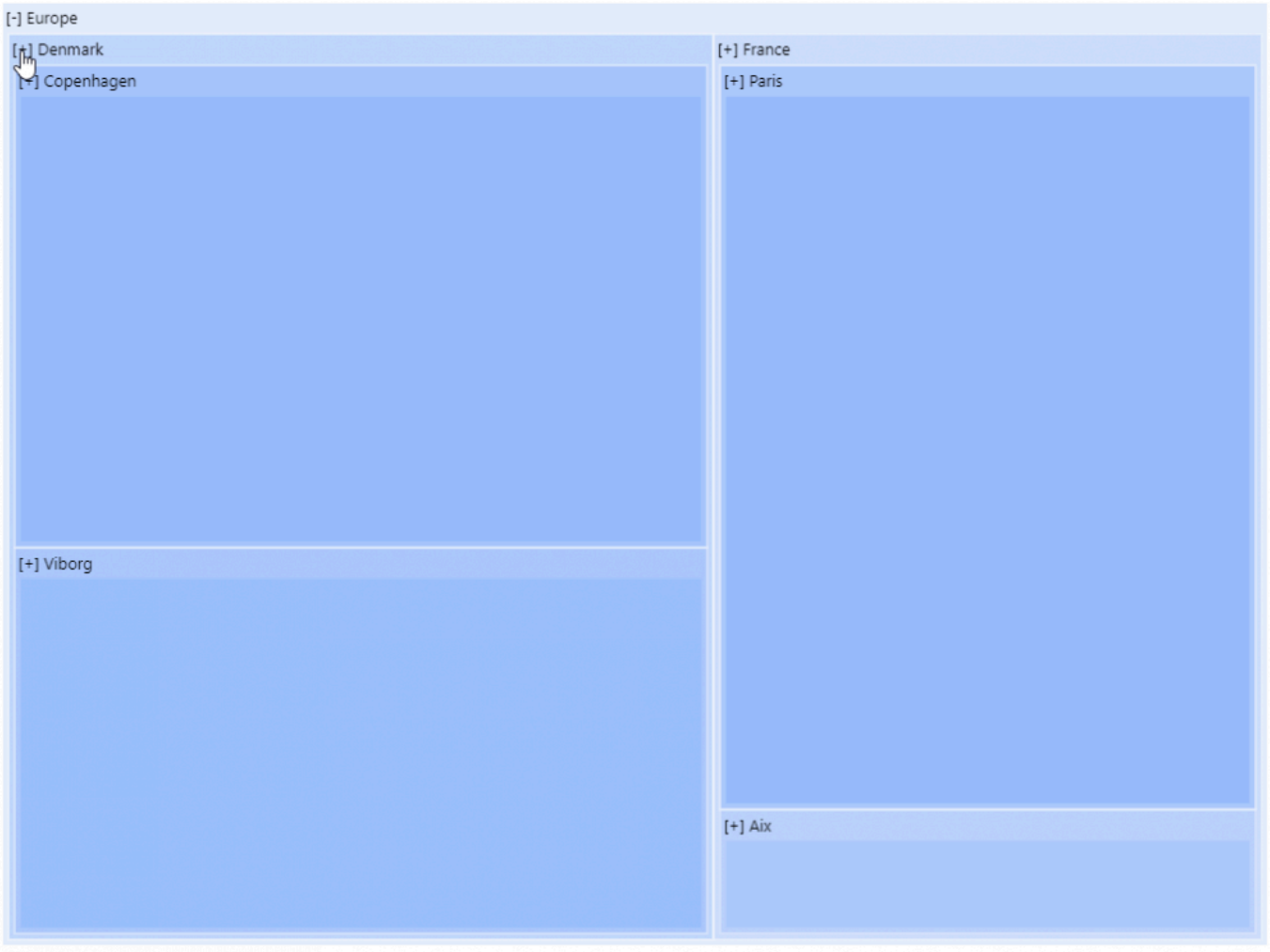


Now again the factories are grouped by country and the countries by continent.
But in addition, this can be navigated by clicking on the [+] or [-] icon on shapes :

The User can click a Continent to drill down. For example, here clicking [+] on “Europe” brings focus to this level :

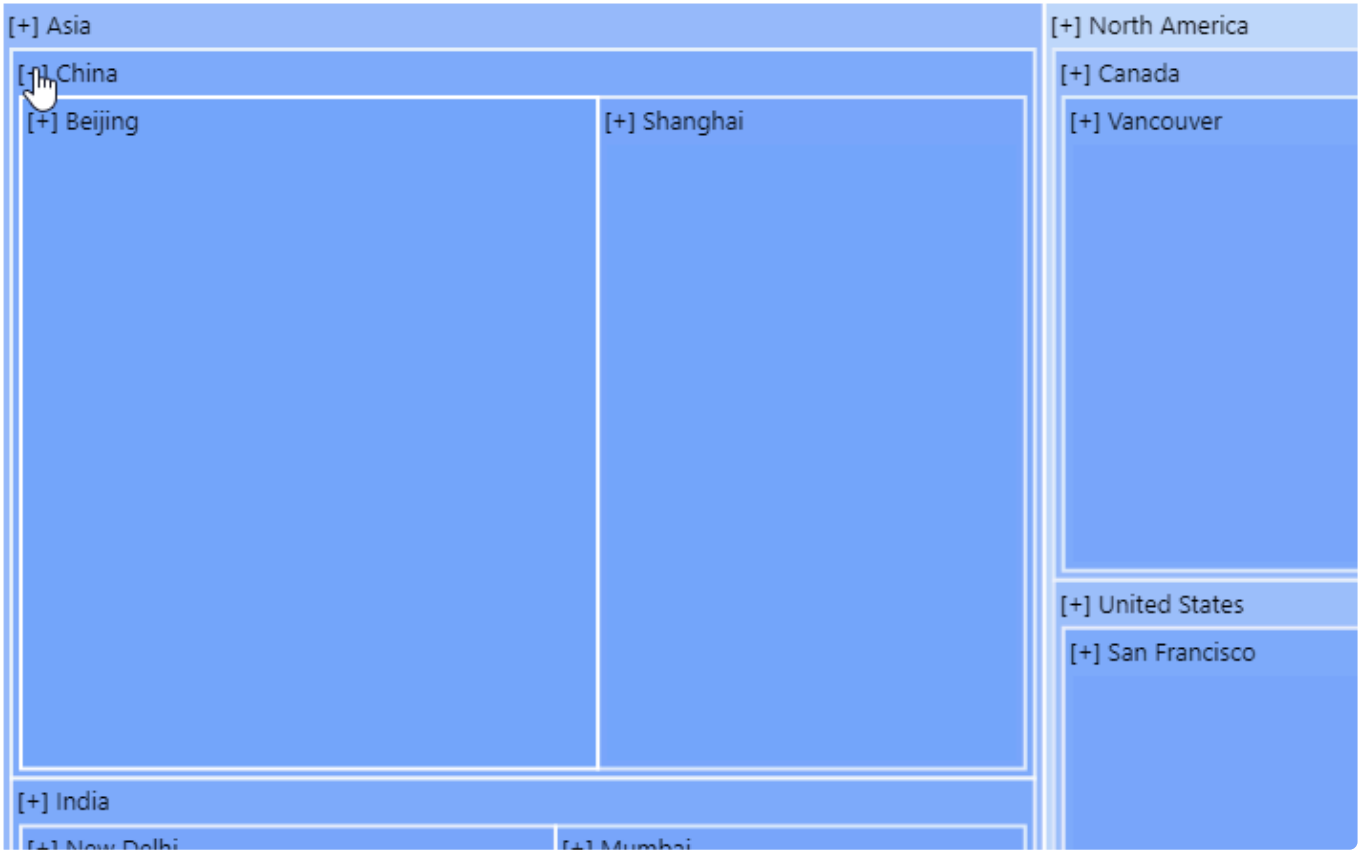


… and see a list of countries for that continent. Then clicking [+] on a Country brings focus on that Country :

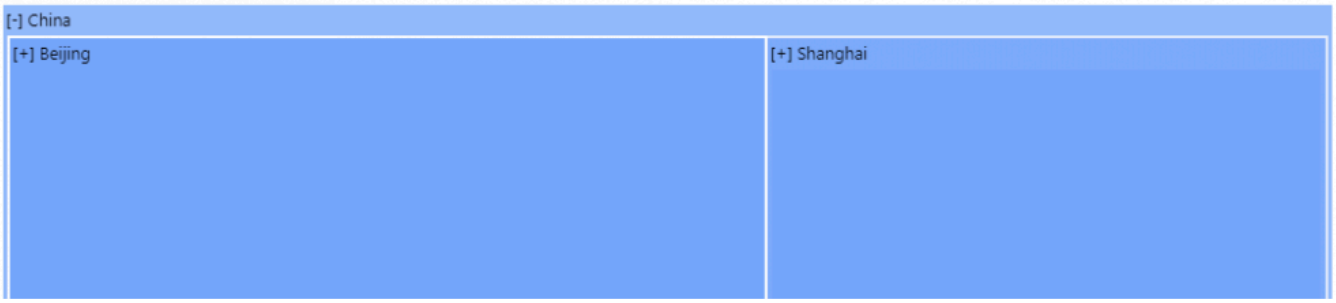


… and see a list of Factories for that Country :

It is possible to click the [-] to navigate up the DrillDown.



It is also possible to click on a lower level directly to navigate to it. For example, from top level, on can click directly a Country to drill down to this country and compare the Factories :

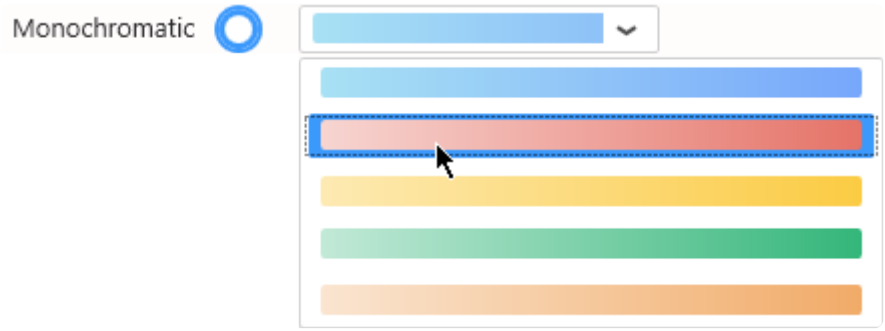


Last modified: May 14, 2019

2.8.5.2. Managing Colors

Monochromatic

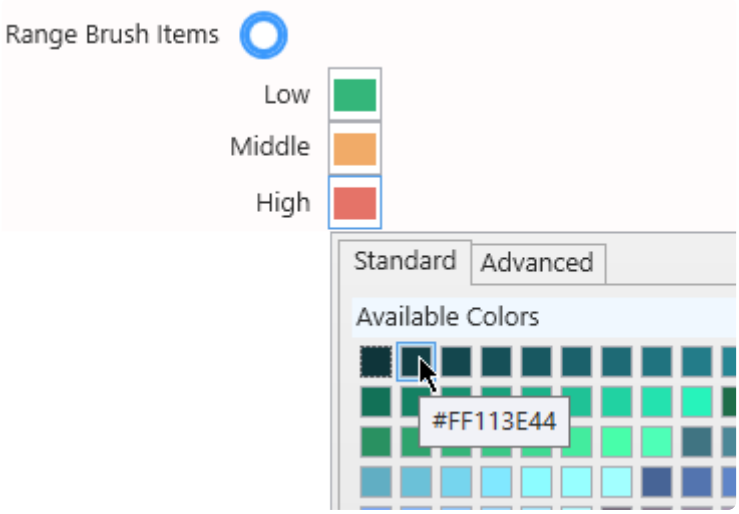
By default, the TreeMap shows value intensity as a gradient of a single color.
This *Monochromatic* Gradient can be set to a different color by using the dedicated combobox under *Range Color Settings* :



Range Brush

The TreeMap can be configured to show value intensity in ranges of single colors.

Select *Range Brush Items* and choose the colors of each interval :



The interval between the maximum of Value and the minimum of Value will be split in equal intervals and assigned a single color.

> This can be useful for setting a color code like “low is red, medium is orange, high is green”.

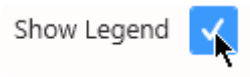
Last modified: May 14, 2019

2.8.5.3. Data Point information

Legend

If using **Range Brush** colors, the TreeMap can display a legend.

Click on Show Legend under Basic Settings :



The scale is displayed with the value interval for each color, and the value for the currently selected shape is highlighted with an arrow :

Last modified: May 14, 2019

2.8.6. Circular Gauge

The Circular Gauge Widget can be used to display a single value as a gauge filling and optionally compare it to a target.

It can also be turned into a container for an indeterminate number of Circular Gauges.

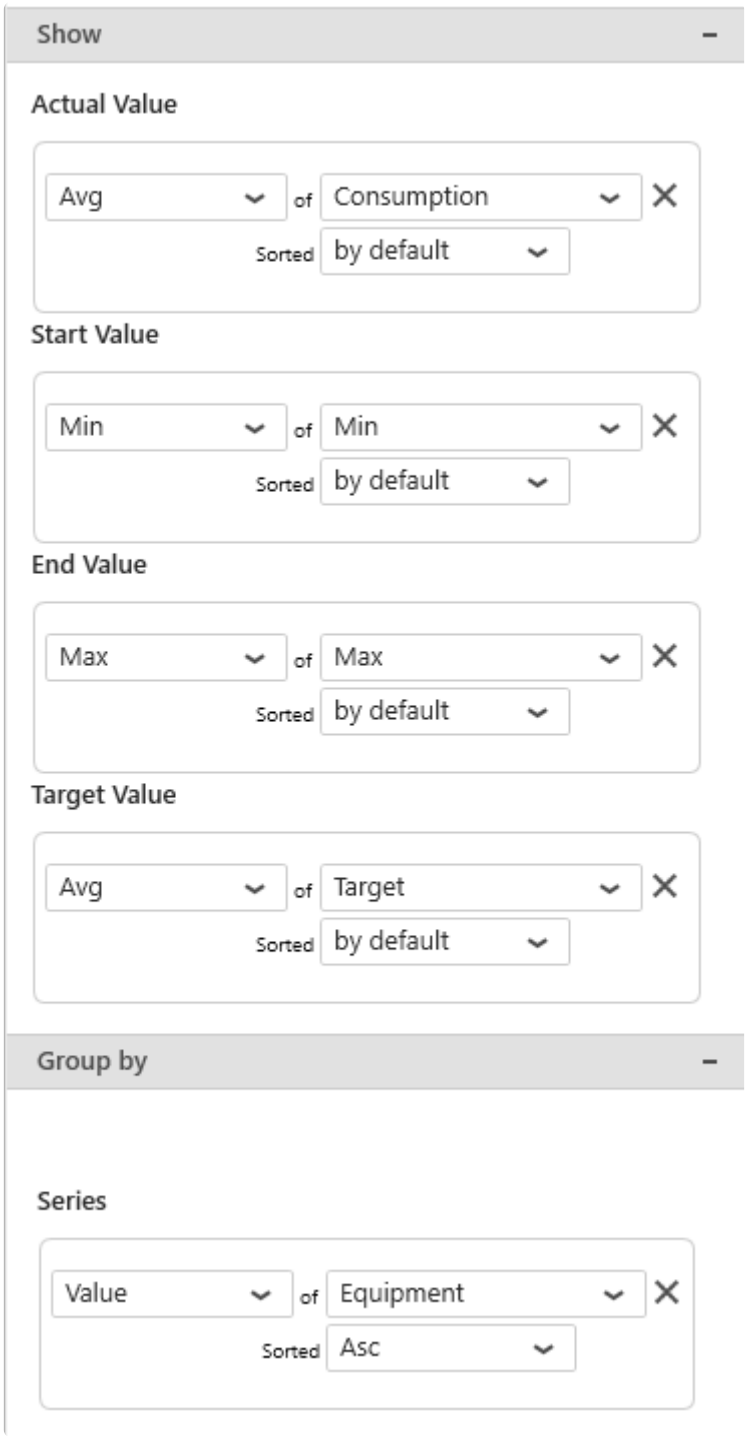
(new in Alpana 2020 R1)

Last modified: Apr 24, 2020

2.8.6.1. Binding Data

The Circular Gauge has 5 data fields available for binding :

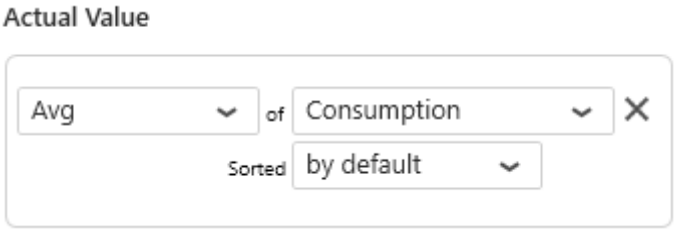
- *Actual Value* : value displayed as a filling of the Circular Gauge
- *Start Value* : dynamic minimum value
- *End Value* : dynamic maximum value
- *Target Value* : target displayed as an arrow on the Circular Gauge
- *Series* : how we split series of different gauges in the same container

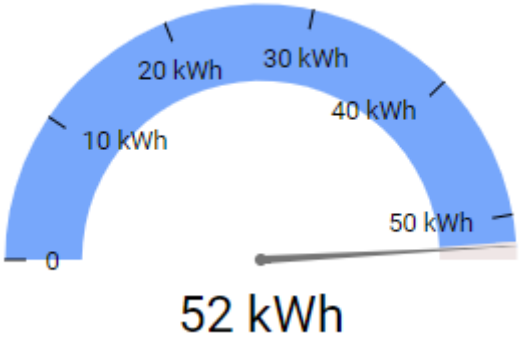


Actual Value

This represents the ***primary*** *quantity* we want to display on the Circular Gauge, and will bind to the needle position and filling of the Circular Gauge.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …)
This field is required.



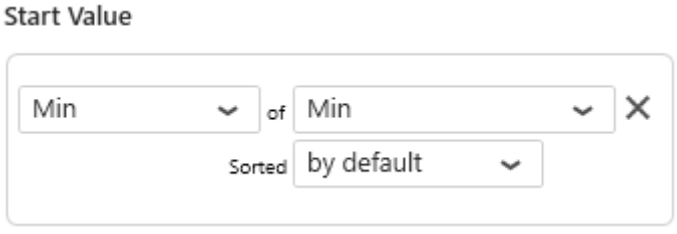


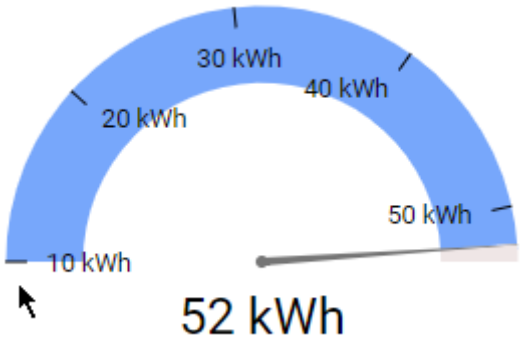
Start Value

This represents the Minimum we want to display on the Circular Gauge, for example the “MinEU” field from Wonderware Historian.
When this field is bound, it overrides manual configuration of the Custom Range Minimum.

If the Actual Value goes below this Start Value, the Circular Gauge will appear empty.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …).
This field is optional.



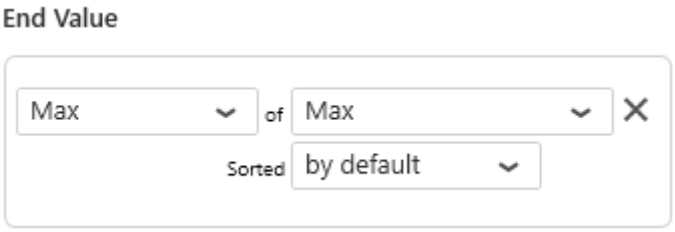


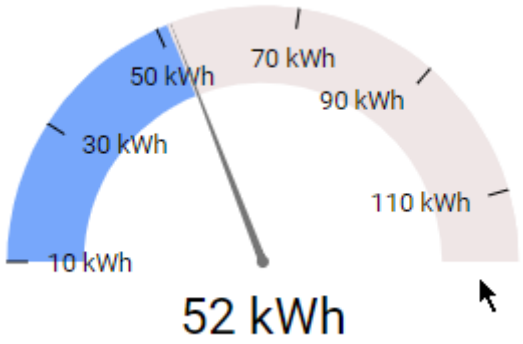
End Value

This represents the Maximum we want to display on the Circular Gauge, for example the “MaxEU” field from Wonderware Historian.
When this field is bound, it overrides manual configuration of the Custom Range Maximum.

If the Actual Value goes above this End Value, the Circular Gauge will appear empty.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …).
This field is optional.

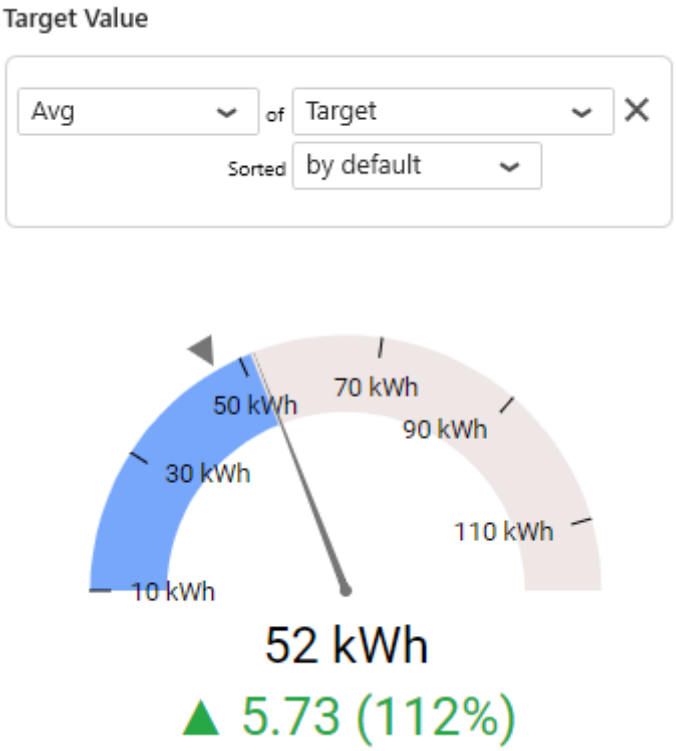




Target Value

This represents the Target we want to display on the Circular Gauge, and will bind to the position of a small arrow on the Circular Gauge.
When this field is bound, an additional information is displayed as text : the distance from Actual Value to Target Value in units and in percentage.

The Target Value field must be aggregated : you must choose an aggregation function (SUM, AVG, …). This field is optional.

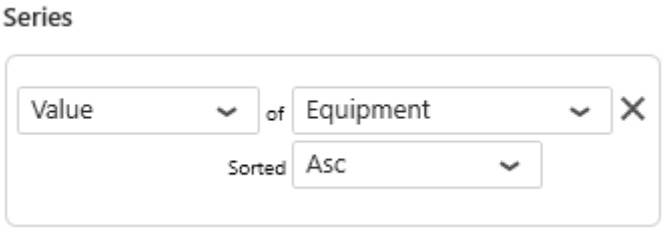


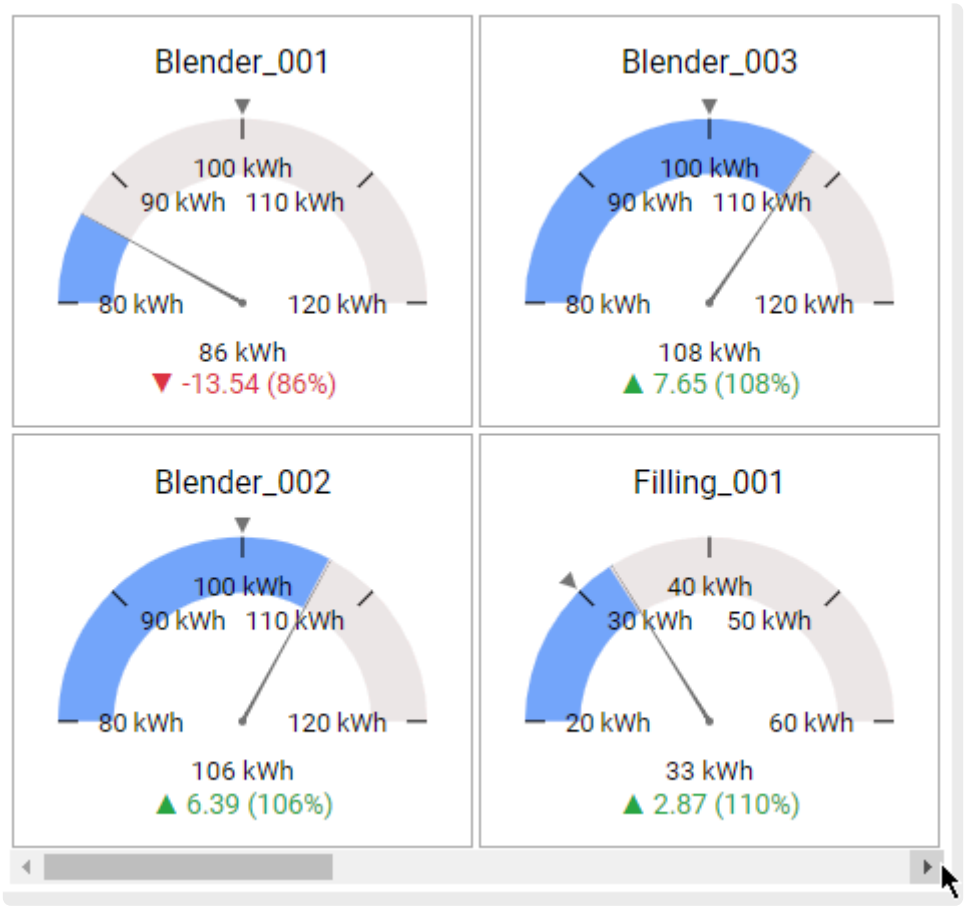
Series

This represents the series field to generate a different contained Circular Gauge for each different value in this Series field.
Each Circular Gauge will take a portion of size of the widget depending on the container shape. In order to access the other values, the user may need to scroll.
Please see the corresponding chapter to configure the Circular Gauge as Widget container.

> Note :
> This only works if your series are values into a single data field.

Example : If you have a field called “Equipment”, you may want to plot a completely separate Circular Gauge for each value of Equipment :





Last modified: Apr 24, 2020

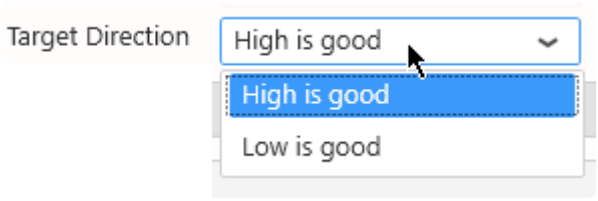
2.8.6.2. Value Display

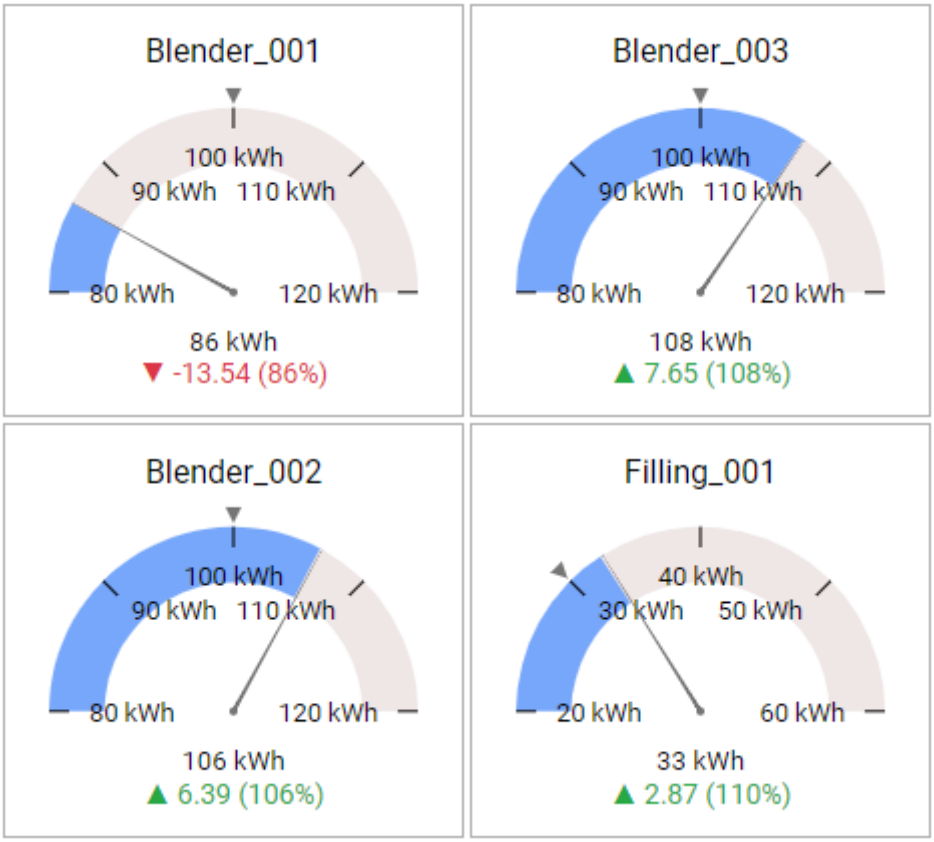
Result interpretation

The font and symbol of the distance to target has a color :

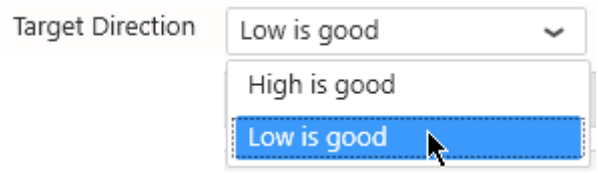
- **green** for “good”
- **red** for “bad”

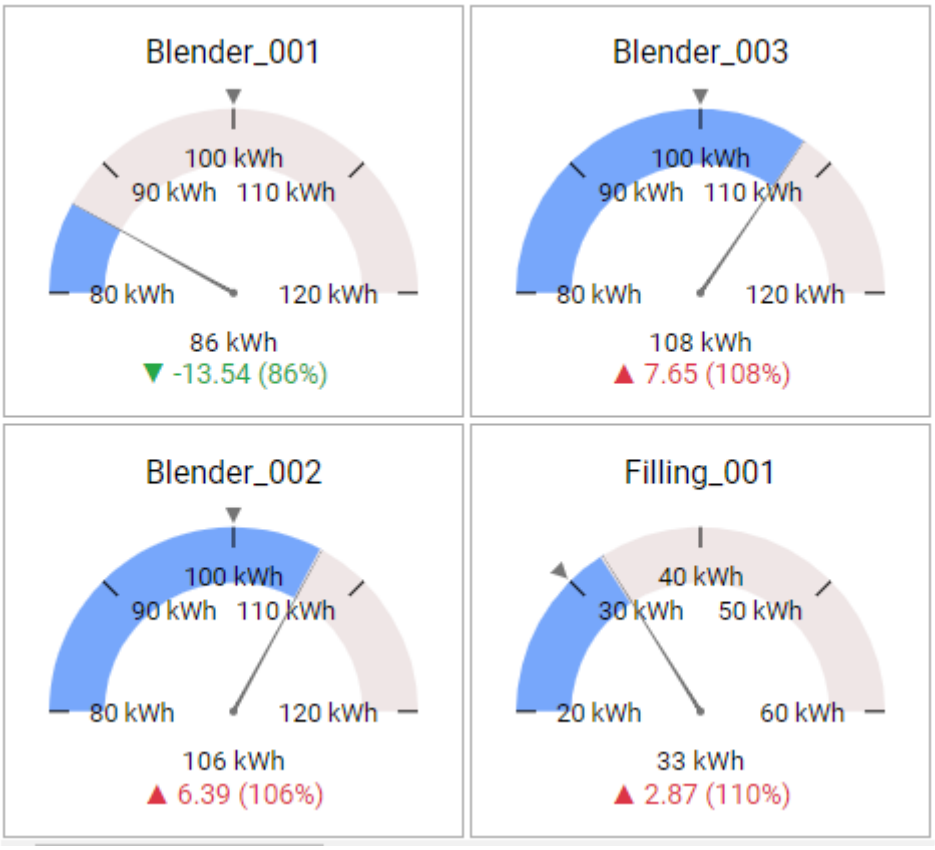
You can define high values as being “good” or “bad” using the *Target Direction* configuration in *Basic Settings* :





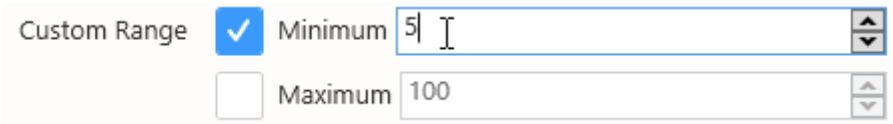
… reverts into :





Value Range

Dynamic value ranges can be defined by binding data fields to the Start Value and End Value configurations.
If instead you want static values for the boundaries of the value range, go under Basic Settings , check the desired part of the *Custom Range* and fill the number box :



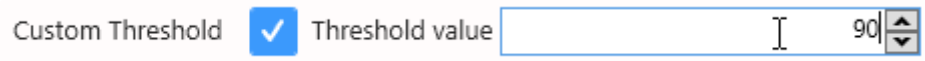
This will override the dynamic data binding from Start Value and End Value.

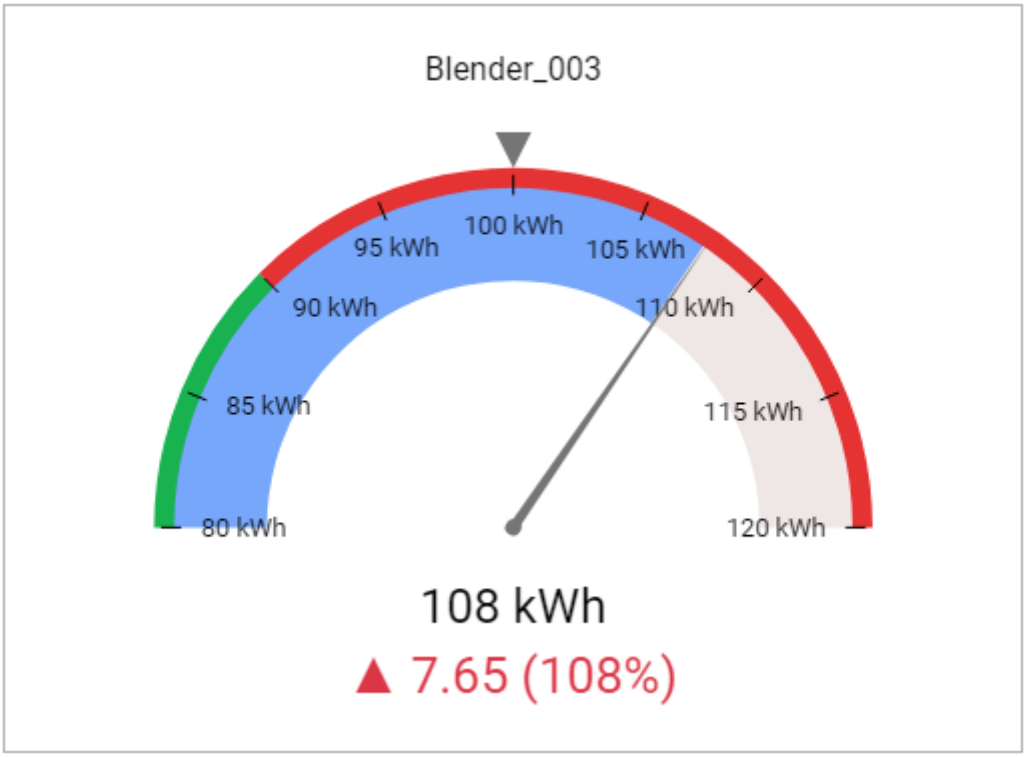
Value Threshold

Value of the Threshold

A value threshold can be displayed on the Circular Gauge to define a minimum or maximum bad value. This is displayed as a background color that goes from green (good) to red (bad).

To enable this feature, check *Custom Threshold* under *Basic Settings* and set a *Threshold Value* :

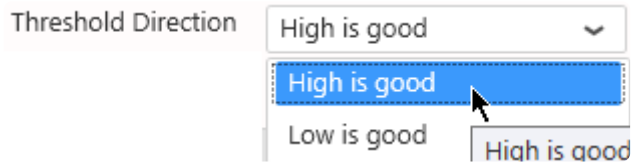


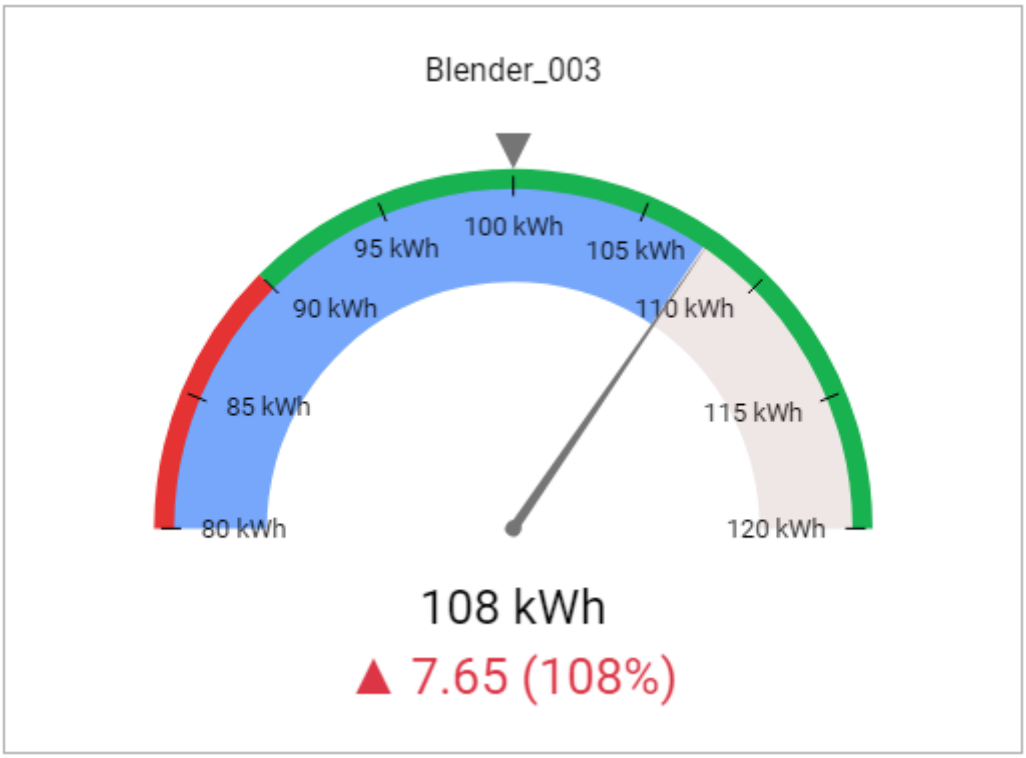


Note :
If a *Target Value* is configured (as in the screenshot above), the KPI still displays the distance to Target, and **not the distance to Threshold**.

Direction of the Threshold

By default, the low values are defines as good, so the green part is displayed on the low values.
If instead you want to define a low threshold and let low values be displayed as red, set the *Threshold Direction* as *High Is good* :



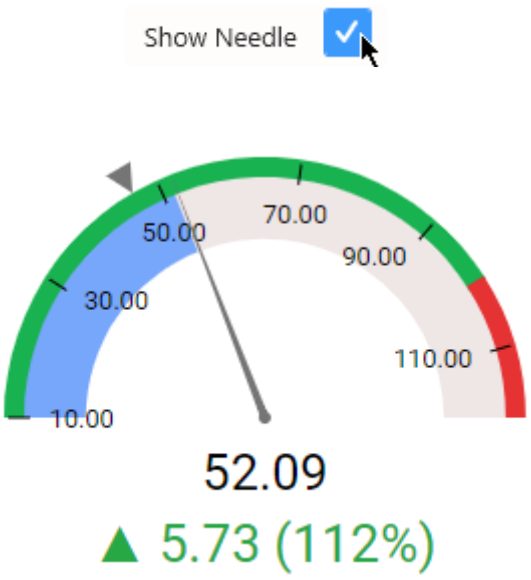


Last modified: Apr 24, 2020

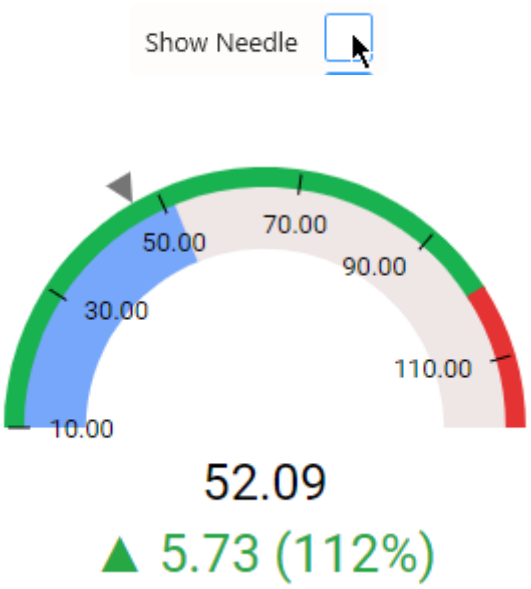
2.8.6.3. Style and Appearance

Needle

By default, the *Actual Value* is displayed using a gauge Needle.
This Needle can be shown or hiden by checking the option *Show Needle* under *Basic Settings* :

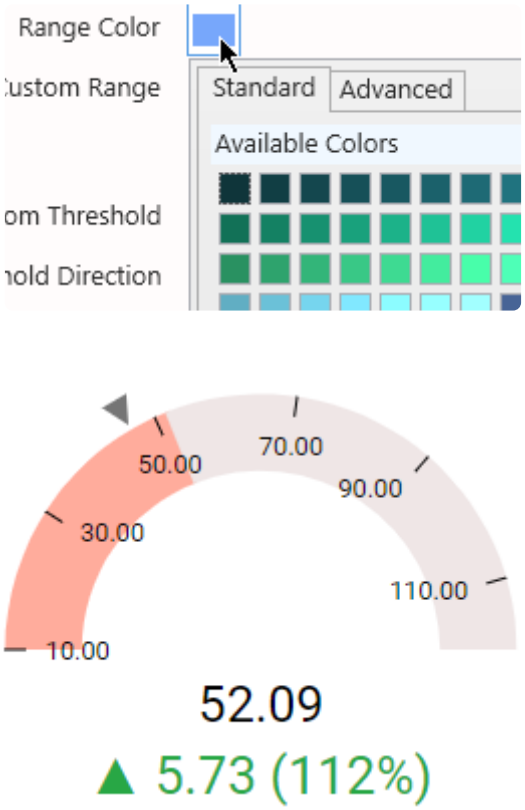


… turns into :



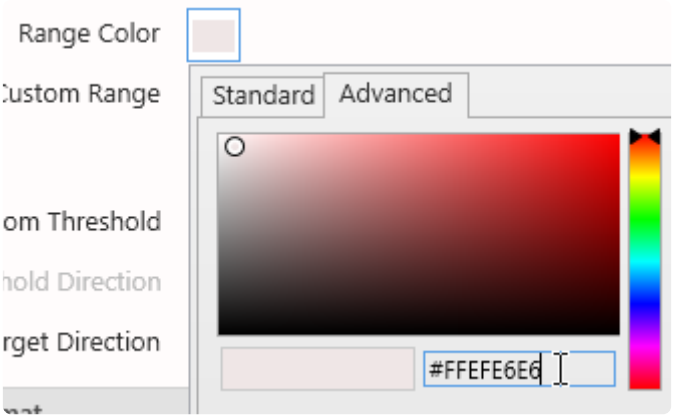
Circular Gauge filling : *Range Color*

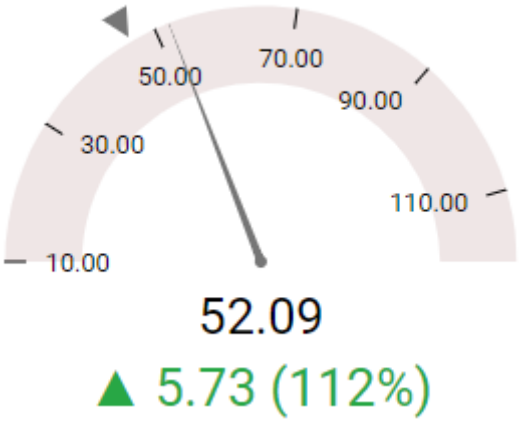
The Circular Gauge displays the *Actual Value* as a color filling.
This color can be configured from *Range Color* under *Basic Settings* :



tip : hiding color

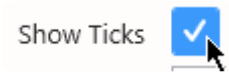
If you want to use only the Needle to display the *Actual Value*, you can hide the Range Color by setting it to the same color as the background color :

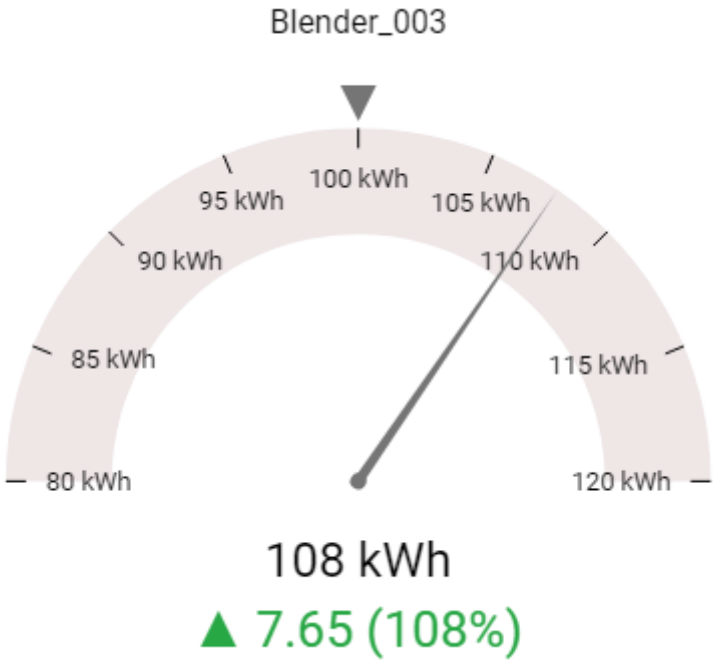




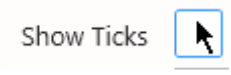
Range scale : *Ticks*

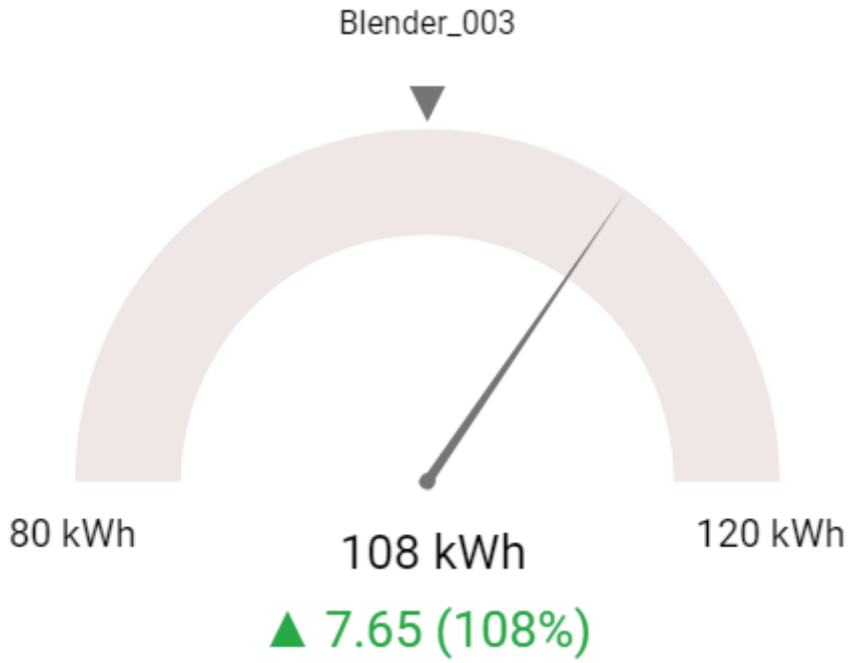
By default, the range of values is displayed as tick marks on a graduated scale.
When un-checking *Show Ticks* under the *Basic Settings*, these values are no longer displayed, and only the minimum and maximum are displayed :





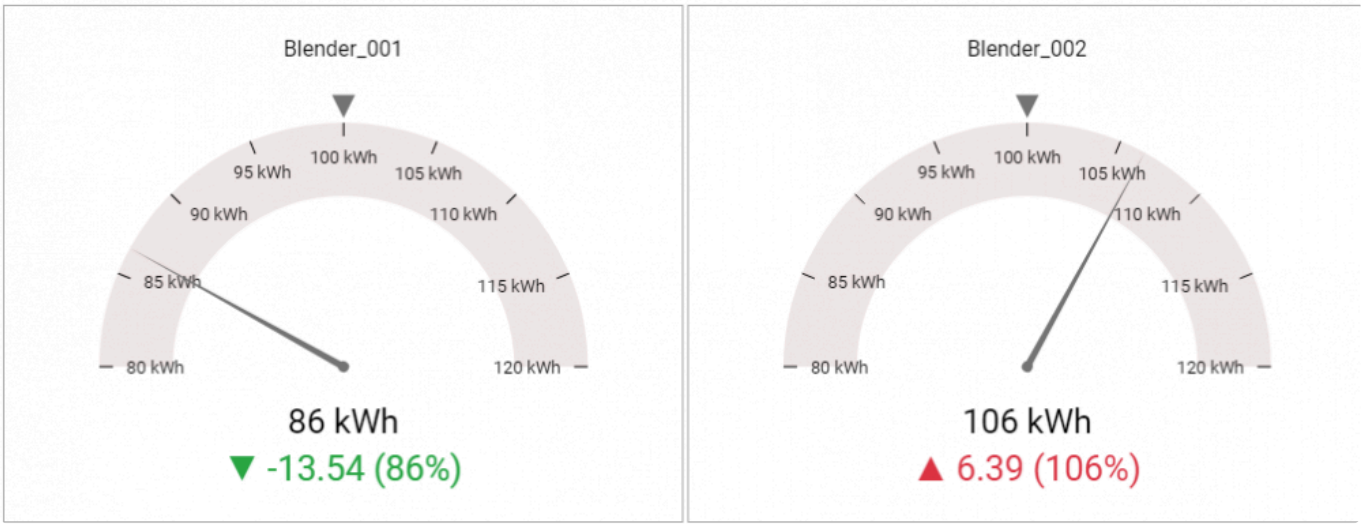
… turns into :



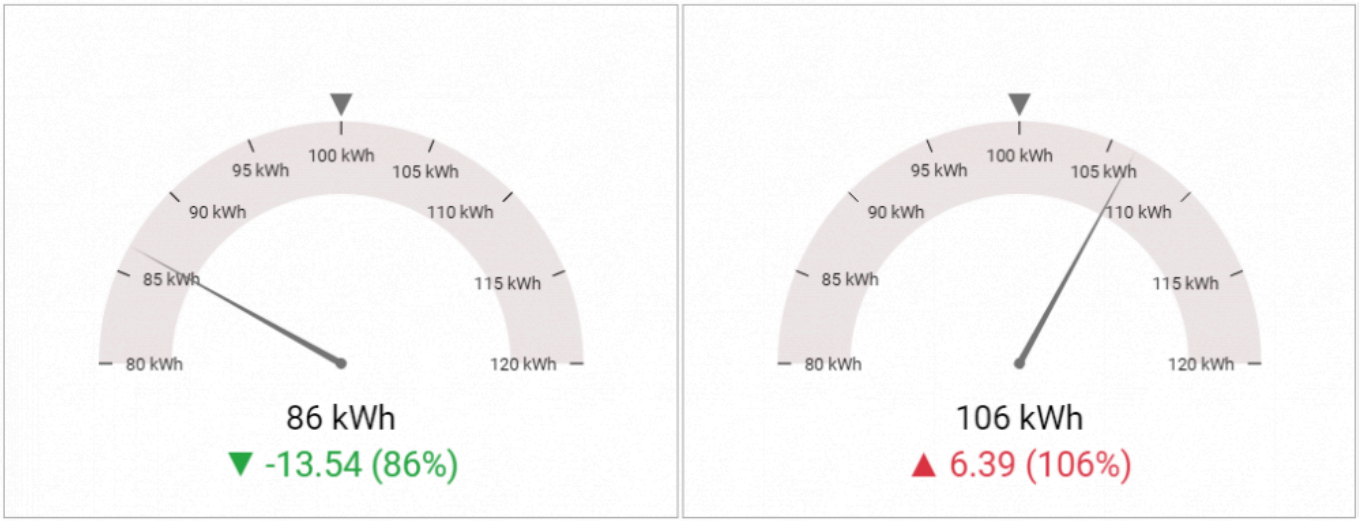


Series Titles

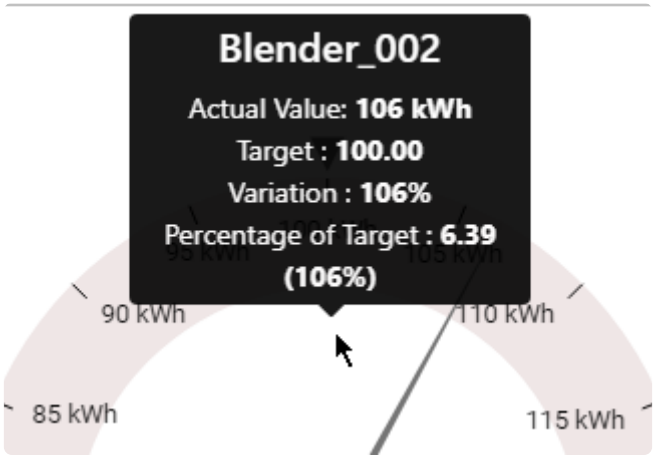
When the Circular Gauge is configured with a *Series*, each Circular Gauge in the series displays a Title :



… turns into :



Hint : when the title is hidden, you can still identify each Circular Gauge by hovering it to display the tooltip :



Last modified: Apr 24, 2020

2.8.7. Linear Gauge

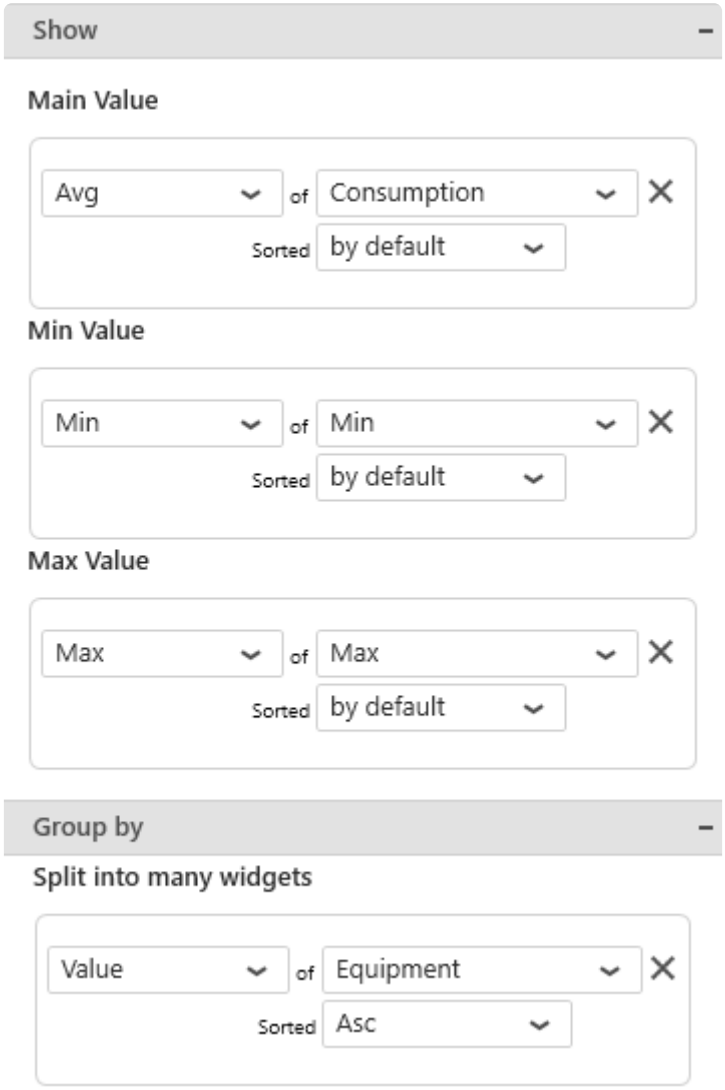
The Linear Gauge widget allows to display a single value as a cursor on a linear gauge. Value intervals can be assigned a color to represent limits or a gradual value meaning.

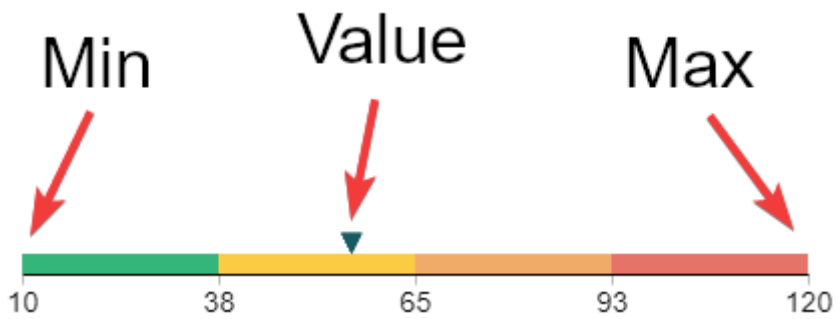
Last modified: Apr 24, 2020

2.8.7.1. Data Binding

The Linear Gauge has 4 data fields available for binding :

- *Main Value* : main value displayed on the Linear Gauge, as a cursor position
- *Min Value* : minimum value for the Linear Gauge
- *Max Value* : maximum value for the Linear Gauge
- *Split into many widgets* : how we split series of different Linear Gauge widgets in the same container





Main Value

This represents the ***primary*** *quantity* we want to display on the Linear Gauge, and will bind to the cursor position of the Linear Gauge.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …)
This field is required.

Min Value

This represents the Minimum we want to display on the Linear Gauge, for example the “MinEU” field from Wonderware Historian.
If the *Main Value* goes out of bound and becomes less than this *Min Value*, then the cursor points to the minimum, and doesn’t go further.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …).
This field is required.

Max Value

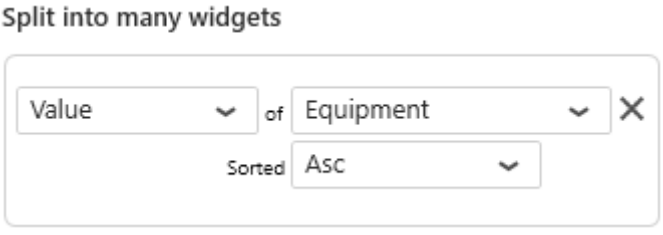
This represents the Maximum we want to display on the Linear Gauge, for example the “MaxEU” field from Wonderware Historian.
If the *Main Value* goes out of bound and becomes less than this *Max Value*, then the cursor points to the maximum, and doesn’t go further.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …).
This field is required.

Group By : *Split into many widgets*

This represents the series field to generate a different contained Linear Gauge for each different value in this *Split into many widgets* field.
Each Linear Gauge will take a portion of size of the widget depending on the container shape. In order to access the other values, the user may need to scroll.
Please see the corresponding chapter to configure the Linear Gauge as Widget container.

Example : If you have a field called “Equipment”, you may want to plot a completely separate Linear Gauge for each value of Equipment :



Linear Gauge

Last modified: Apr 24, 2020

2.8.7.2. Value display and Limits configuration

Linear Gauge colors

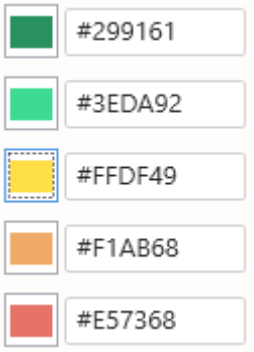
The Linear Gauge cursor is constrained between the minimum and maximum bounds.
The interval between the minimum and the maximum can be split into intervals.
Those intervals are calculated linearly by splitting the interval into a scale of `N` equal parts.
Those intervals can be assigned a color to represent limits or a gradual value meaning.

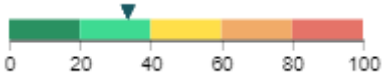
The number of intervals `N` is defined by the number of color items in the list of *Scale Colors*.

Add colors with the `+` button.
Remove colors with the `X` button.
Configure colors with the color picker or the text input (color codes).

Example : gradual value

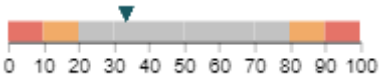
Min Value = `0`
Max Value = `100`
Colors (`N` = 5) : dark green, light green, yellow, orange, red
The scale will represent 5 intervals : `0-20`, `20-40`, `40-60`, `60-80`, `80-100`, corresponding to the configured colors.
When the cursor will move, it can represent a temperature going warmer and warmer.





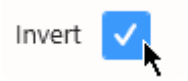
Example : limits

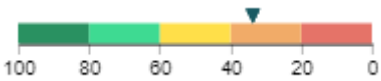
Min Value = `0`
Max Value = `100`
Colors (`N` = 10) : red, orange, grey, grey, grey, grey, grey, grey, orange, red
The scale will represent 10 intervals : `0-10`, `10-20`, etc, corresponding to the configured colors.
When the cursor will move, it can represent a KPI with a value which must not go near limits.



Invert

Checking the *Invert* option will display values from right-to-left instead of the default left-to-right.



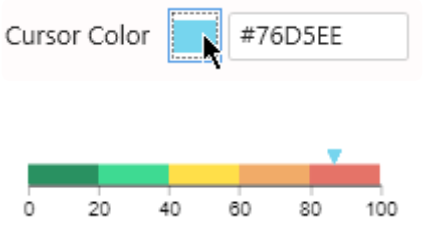


Last modified: May 05, 2020

2.8.7.3. Style and Appearance

Cursor Color

This setting allows to configure the color of the cursor.



Last modified: May 05, 2020

2.8.8. Card

The Card Widget can be used to display a single value as text and optionally compare it to a target.

(new in Alpana 2020 R1) The Card can display the KPI history on a small sparkline.

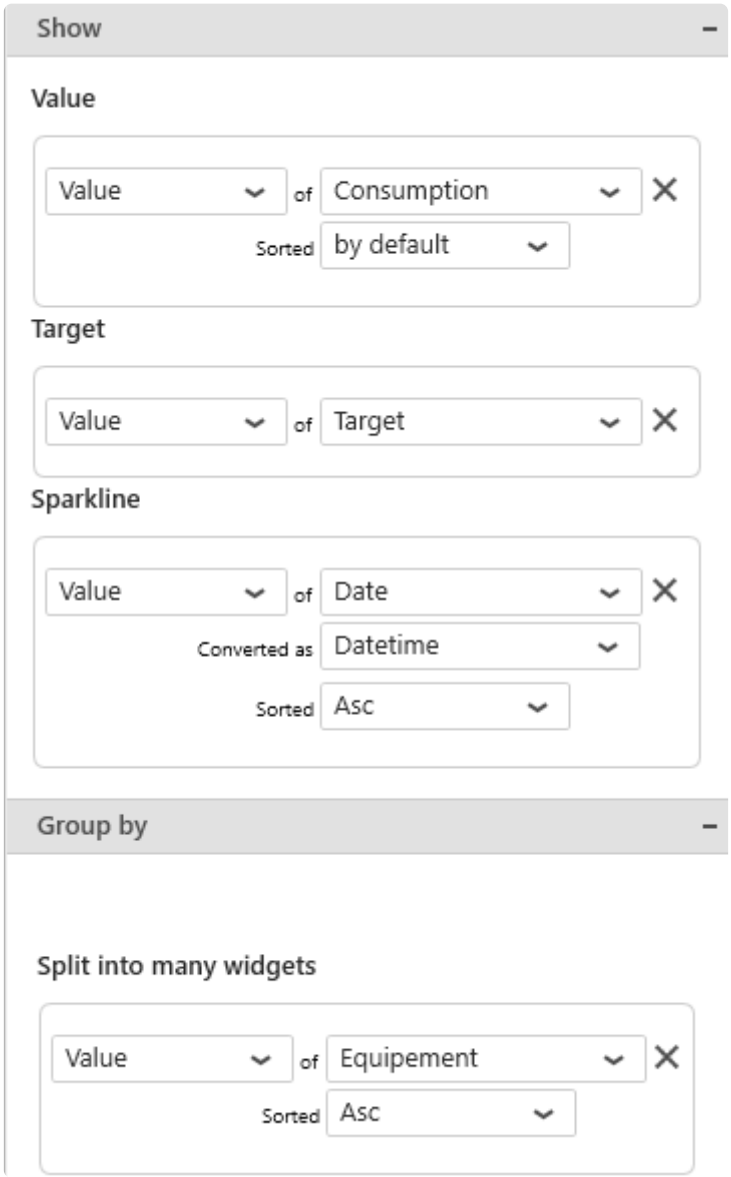
Last modified: Apr 27, 2020

2.8.8.1. Binding Data

Data fields

The Card has 3 data fields available for binding :

- *Value* : main value displayed on the Card
- *Target* : target displayed in the Card, compared to the Value
- *Sparkline* : series field (usually a date) to use as X Axis values on the sparkline
- Group By : *Split into many widgets* : how we split series of different Cards in the same container

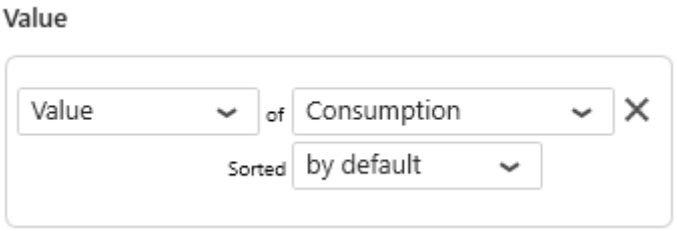


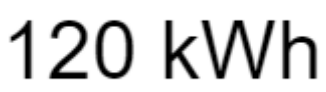
Value

This represents the ***primary*** *quantity* we want to display as text on the Card.

See below about Value Aggregation.

This field is required.

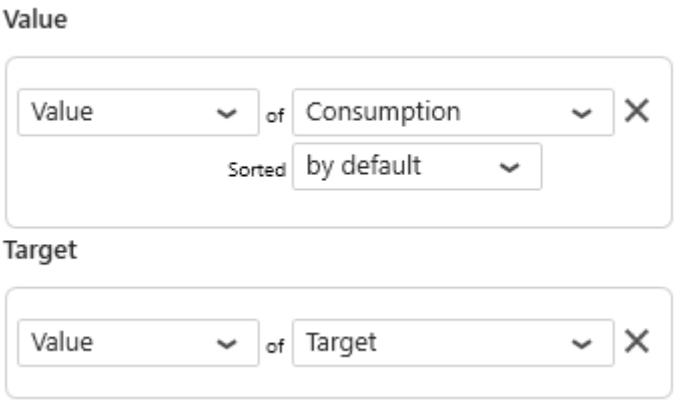




Target Value

This represents the Target we want to display as text on the Card, and will be compared to the *Value*. When this field is bound, an additional information is displayed as text : a *KPI* indicating how the *Value* compares to the *Target* value, which can be configured.

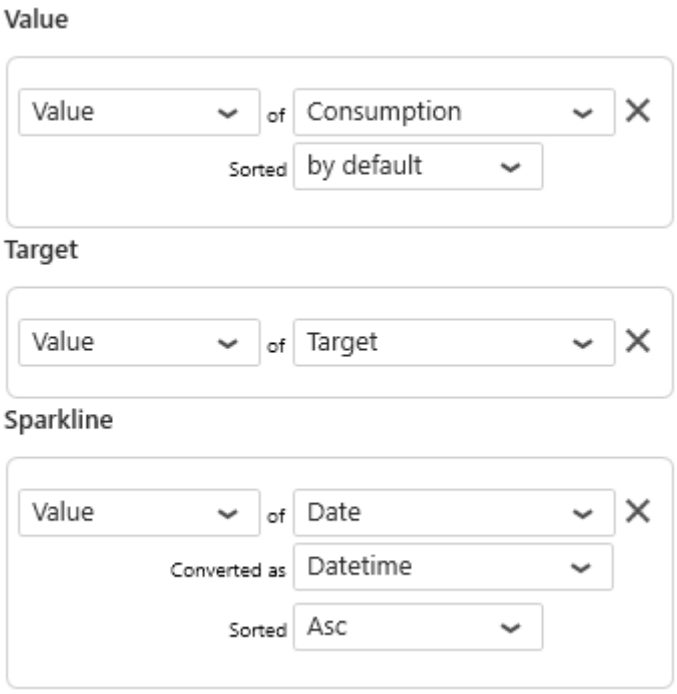
See below about Value Aggregation.
This field is optional.

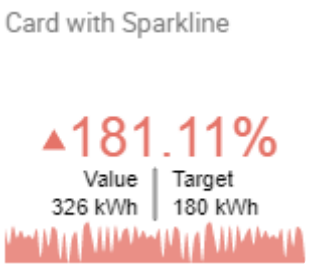


Sparkline

This is the field (usually a date/time) used for the X Axis of the sparkline chart.

This field is optional if you don't want to display a sparkline on the Card.



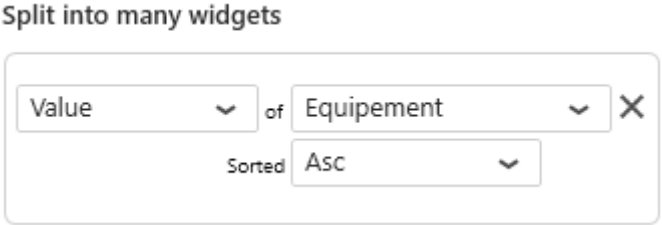


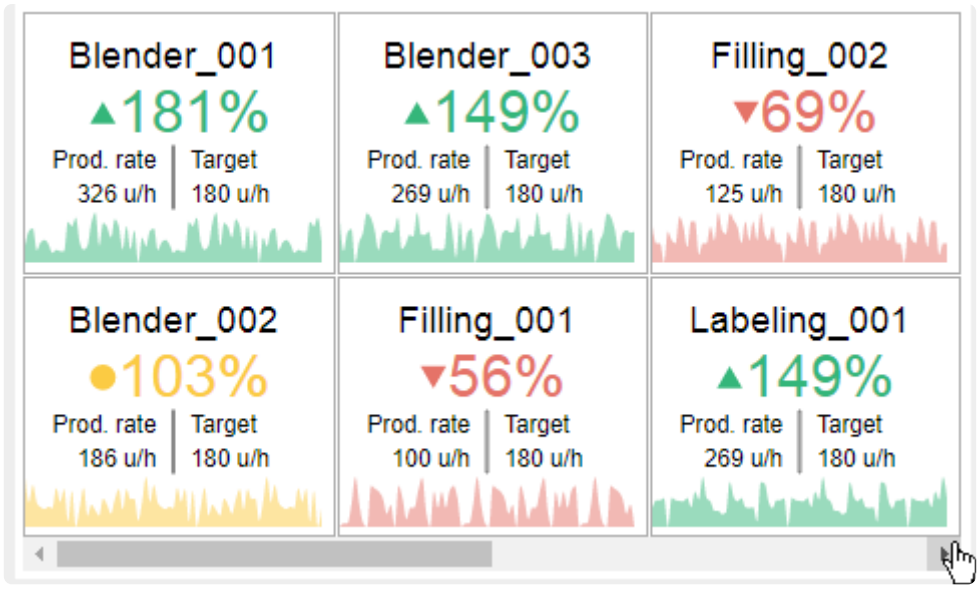
Group By : *Split into many widgets*

This represents the series field to generate a different contained Cards for each different value in this Series field.
Each Card will take a portion of size of the widget depending on the container shape. In order to access the other values, the user may need to scroll.
Please see the corresponding chapter to configure the Card as Widget container.

> Note :
> This only works if your series are values into a single data field.

Example : If you have a field called “Equipment”, you may want to plot a completely separate Card for each value of Equipment :





Value aggregation

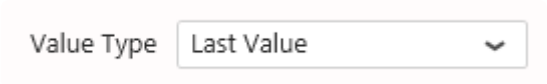
This Widget displays two types of values :

- the Card “summary value” displayed as text labels : Value, Target, KPI
- the sparkline chart’s Y value

You may configure the summary value in two different ways, under the *Value* category :

Value Type = Last Value

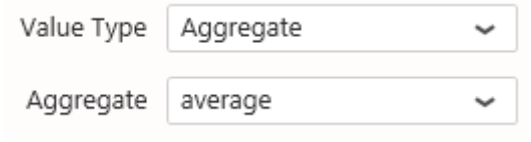
Displaying the *Last Value* means the last value on the sparkline chart, according to the sorting by the *Sparkline* field.



Note : in this case, the *Aggregate* selection is ignored.

Value Type = Aggregate

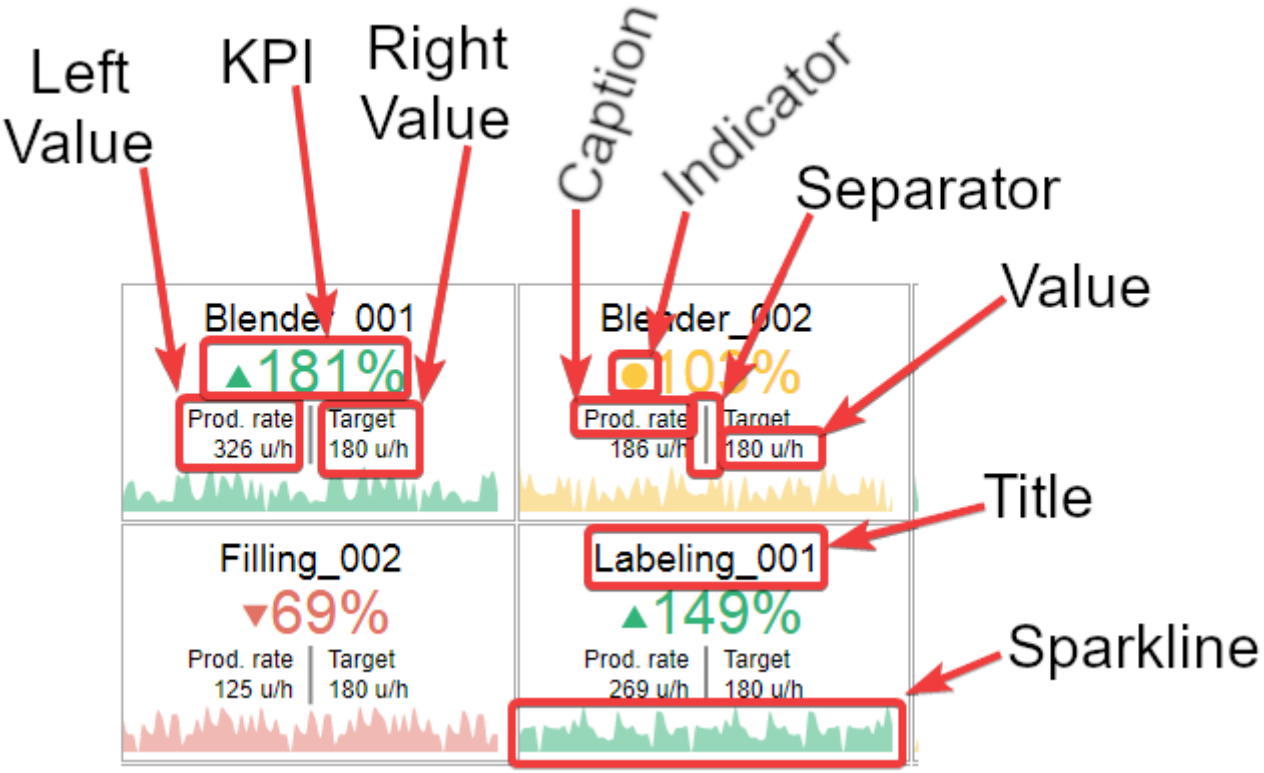
Instead of displaying the last value from the sparkline chart, the *Aggregate* setting allows to calculate a value based on all values in the chart.
The aggregation function used to aggregate all the values in the chart is defined as the *Aggregate* option.



Last modified: Apr 27, 2020

2.8.8.2. Value Display

Depending on how it's configured, the Card displays several informations :



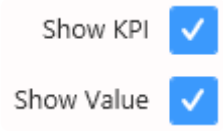
Each part can be configured, as detailed below.

Configuring KPI, Left Value, Right Value

For each of *KPI*, *Left Value*, *Right Value*, the following can be configured :

Show/hide

Each can be displayed or hidden by checking *Show Value* (or *Show KPI*) :



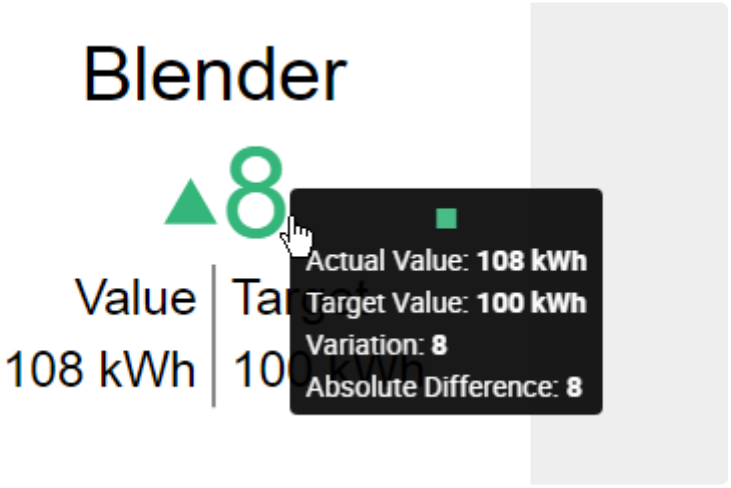
Type

The value displayed at each position can be one of several types :

Absolute Difference

Shows the difference between the Value and the Target.

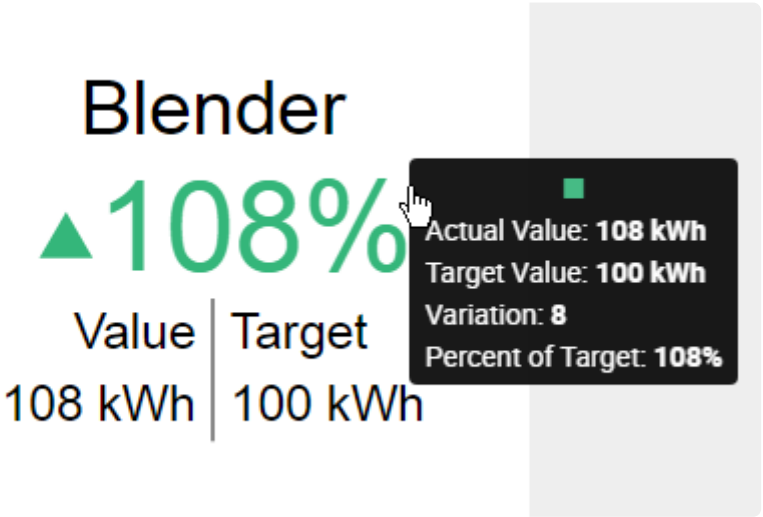
```
Calculation = Value - Target
```



Percent of Target

Shows the % of value to target

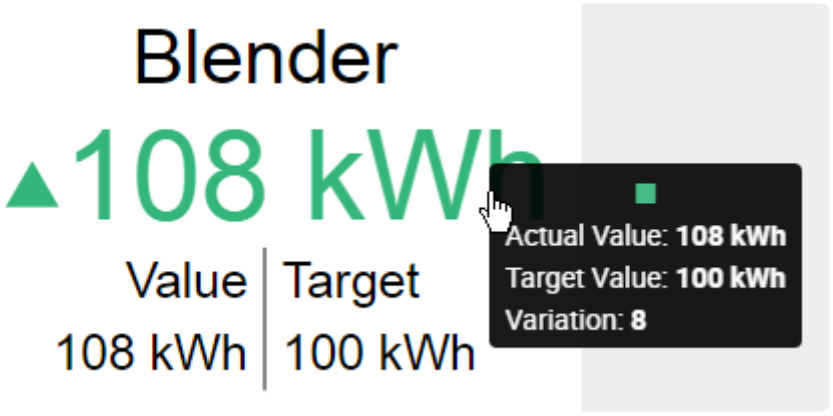
```
Calculation = 100 * Value / Target
```



Actual Value

Shows the value of the Value field itself.

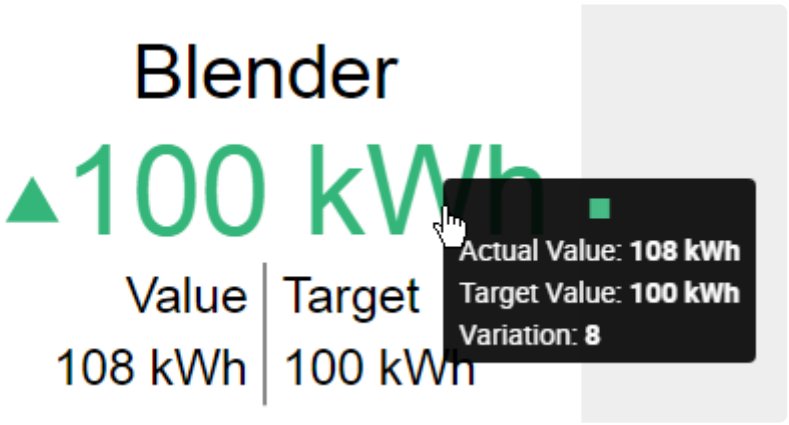
```
Calculation = Value
```



Target Value

Shows the value of the Target field itself.

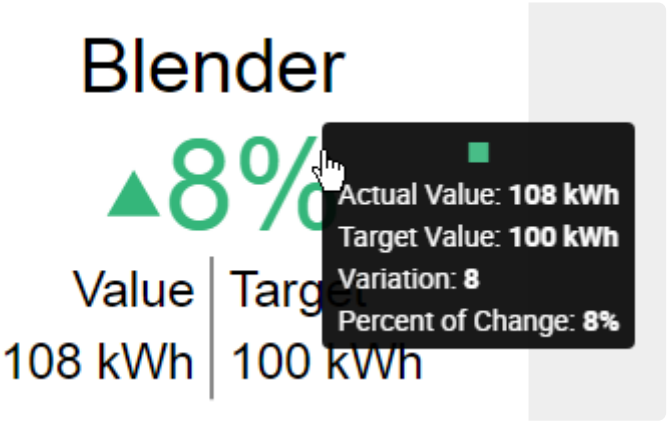
```
Calculation = Target
```



Percent of Change

Shows the % of variation to target.

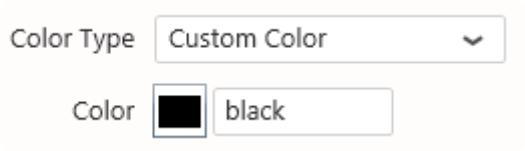
```
Calculation = 100 * (Value - Target) / Target
```



Color

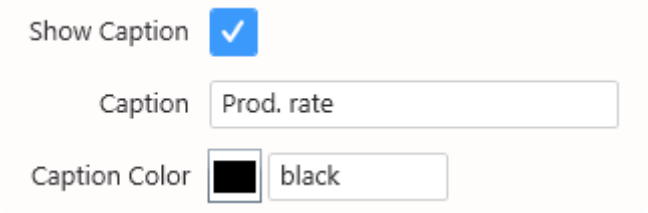
The value displayed at each position can have :

- the *Direction Color* that depends on the distance between the Value and the Target, and whether we consider that “good” or “bad” (see below “Colors and result interpretation”)
- or a *Custom Color* that is fixed



Caption

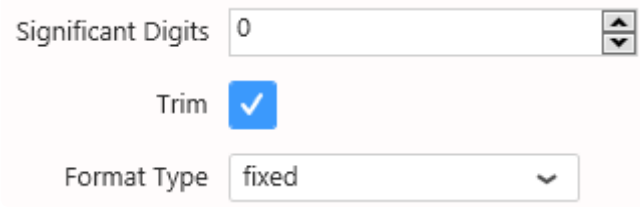
The Caption can be displayed or hidden by checking *Show Caption*.
The Caption’s text and *Caption Color* can be configured :



> Note : the top part (dedicated to the KPI) doesn’t have its own Caption

KPI Value formatting

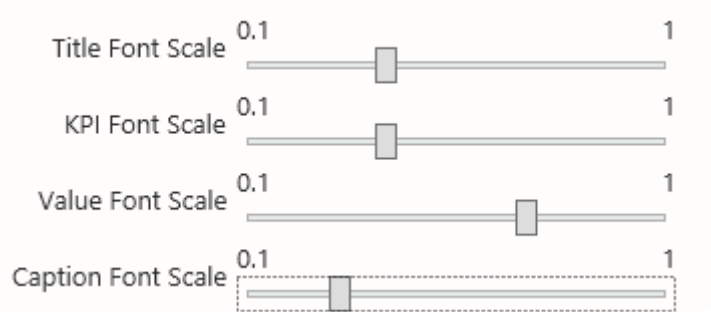
The calculated KPI has its own value formatting, because it is a calculated value and isn’t listed for Value Formatting.

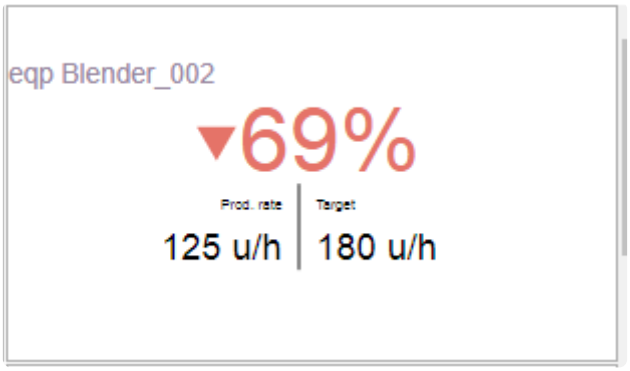


> Tip : If the *Significant Digits* has value `-1`, then the full value is displayed.

Element sizes

The relative font sizes of elements can be configured :



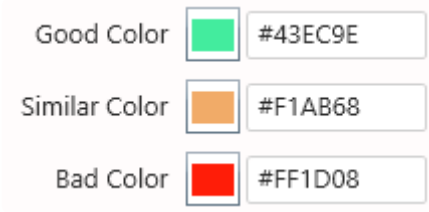


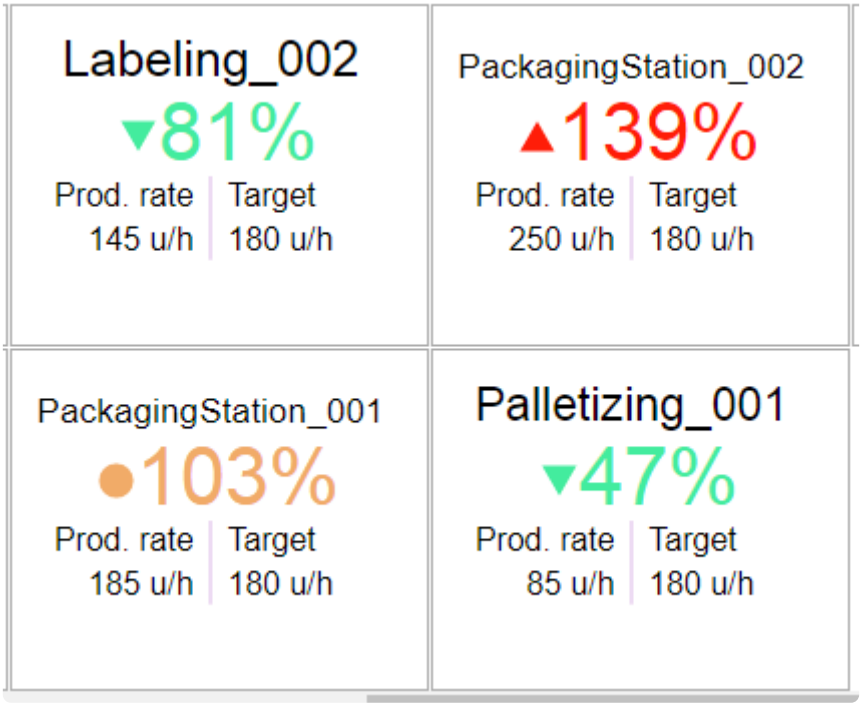
This *Font Scale* is relative, so that it remains responsive when the widget is viewed on different devices.

Colors and result interpretation

The font and symbol of the distance to target has a color :

- **green** for “good”
- **yellow** for “near target” or “similar”
- **red** for “bad”





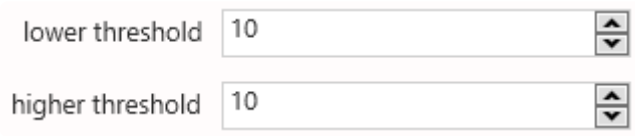
High/Low Is Good

You can define high values as being “good” or “bad” by selecting the *Direction* configuration in *Colors* :

- with *High is Good*, values that go above target will be displayed with the *Good Color*, and values that go below target will be displayed with the *Bad Color*
- with *Low is Good*, values that go above target will be displayed with the *Bad Color*, and values that go below target will be displayed with the *Good Color*

Thresholds for the “similar” state

Instead of displaying the “similar” state only when the value equals the target exactly, you can configure thresholds to accept an interval of values that are similar to the target.
These *lower threshold* and *higher threshold* are configured under *KPI thresholds* :



Other colors

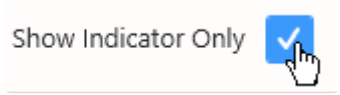
When no Target is configured, only the Value is displayed, with a constant color.
This color can be configured under *Default Color* :

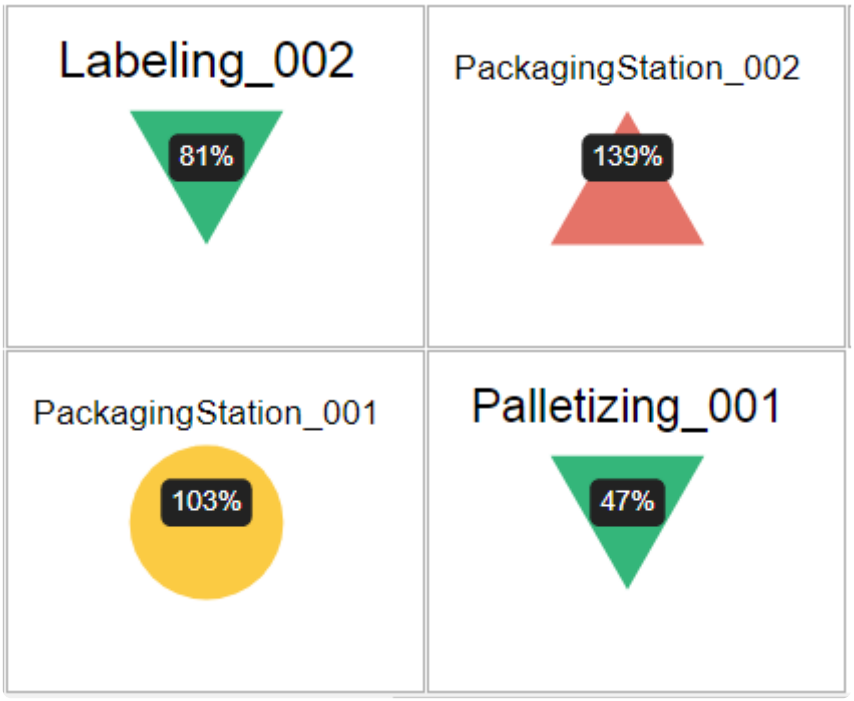
The *Separator Color* can also be configured :

Full detail or KPI only

By default, when the Card has a Target Value, it displays both values as text, as well as a KPI with its value as text.
It is also possible to instead simplify this display and have only the KPI in a very readable way.

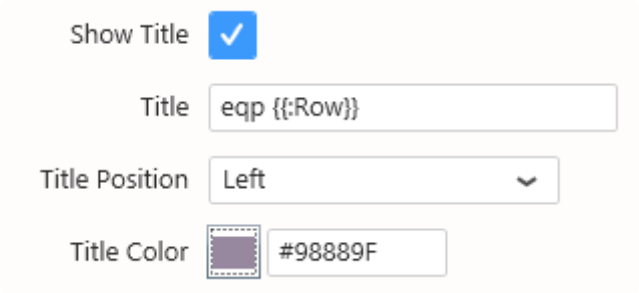
For this, select *Show Indicator Only* :

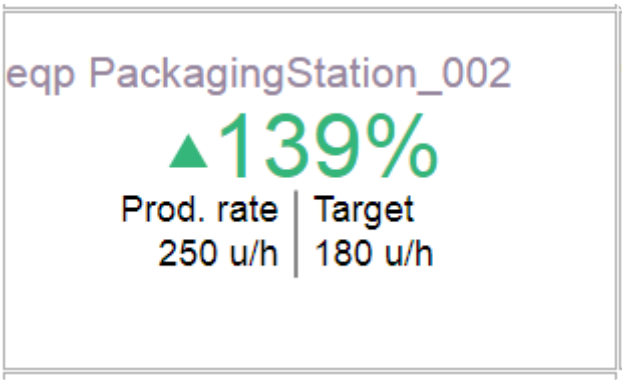




Title style

The Title is especially useful when a “Split into many Widgets” field is bound.
It can be hidden or shown, the text, position and color can be configured :





Last modified: Apr 28, 2020

2.8.8.3. Card Sparkline

The Card displays a sparkline which is a mini-chart that gives an idea of the trend of the KPI.

Sparkline Scale

The vertical *Scale* of the sparkline chart can be configured, in % of the height of the row :

- `1` means to take the full height of the row
- `0.9` means almost the full height
- `0.1` means a very thin display

Sparkline Color

The *Color Type* of the sparkline chart can be :

- the *Direction Color* that depends on the distance between the Value and the Target, and whether we consider that “good” or “bad” (see “Colors and result interpretation”)
- or a *Custom Color* that is fixed

The *Opacity* of the color can also be configured.

Sparkline Interpolation

The sparkline chart is interpolated.
Available interpolation methods are :

basis

A cubic basis spline
The result is a smooth curve which doesn't exactly pass through the data points themselves.

linear

A series of interconnected lines.
The result is that of a line chart.

monotone

A cubic spline that preserves monotonicity in y, assuming monotonicity in x.
The result is a smooth curve which goes through all data points and doesn't overshoot.

step

A step line centered on the data point.
The result is a step line which is constant around the data points.

step-after

A step line starting on the data point.
The result is a step line which is constant after the data points.

step-before

A step line ending on the data point.
The result is a step line which is constant before the data points.

Last modified: Apr 28, 2020

2.8.9. Sparkline

The Sparkline is a good way to summarize the state of many variables : you can view them side-by-side, with their current value compared to a mini-trend.
Optionally, you can improve this summary by displaying limits and a target.

(new in Alpana 2020 R1)

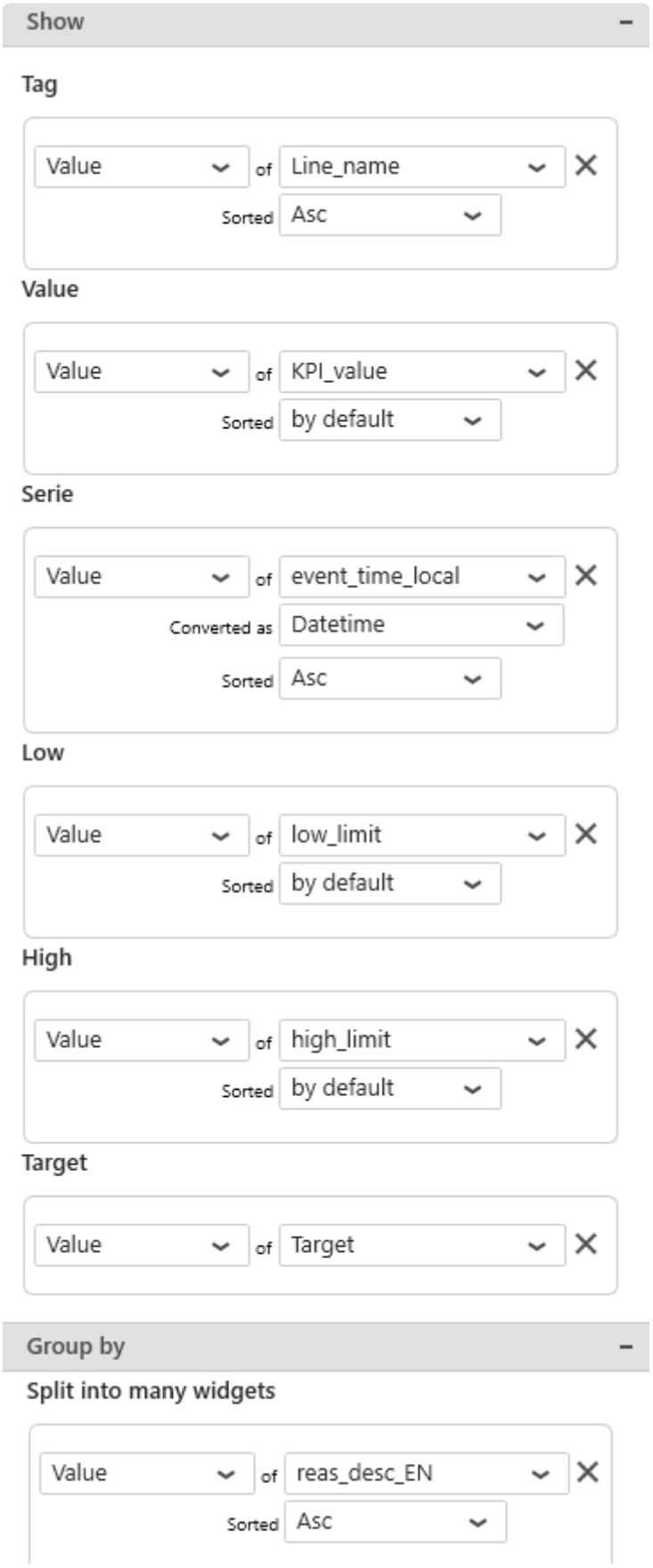
Last modified: Apr 24, 2020

2.8.9.1. Binding Data

Data fields

The Sparkline has 7 fields available for binding :

- *Tag* : the name of the variable
- *Value* : the value of the variable
- *Serie* : the field used as X Axis of the chart
- *Low* : the low threshold
- *High* : the high threshold
- *Target* : the target value
- *Split into many widgets* : a field to group and split variables

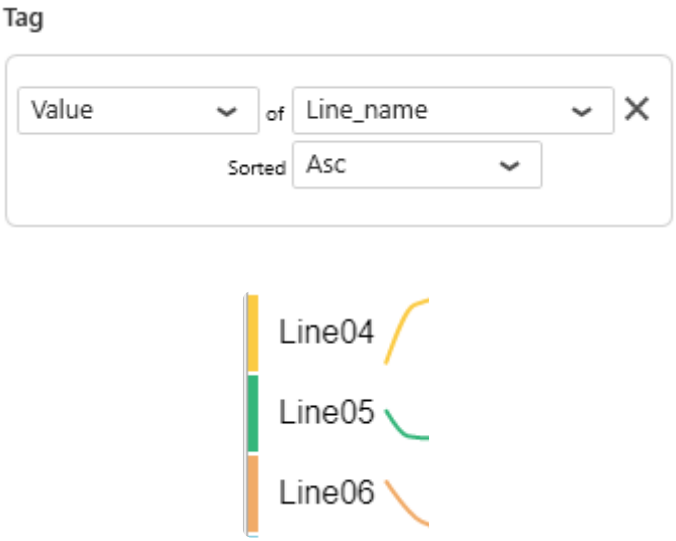


Tag

This represents the variable name, used to show separate indicators side-by-side.
The indicators are laid out in consecutive rows. Each indicator will take a given vertical size, and the full

width of the container.
In order to access the other indicators, the user may need to scroll down.

This field is required.

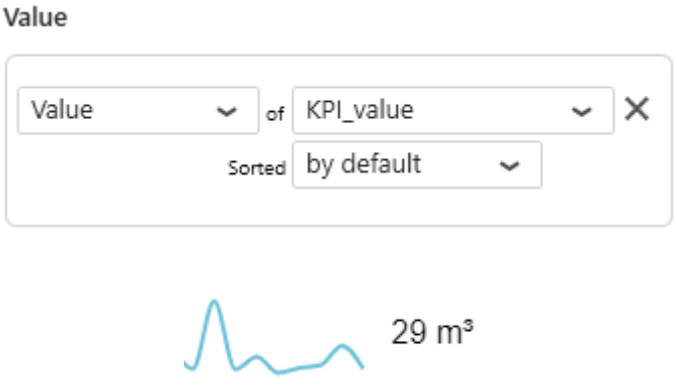


Value

This is the main value of the variable.
It will be displayed both :

- as Y value on the sparkline chart
- as main value on the summary

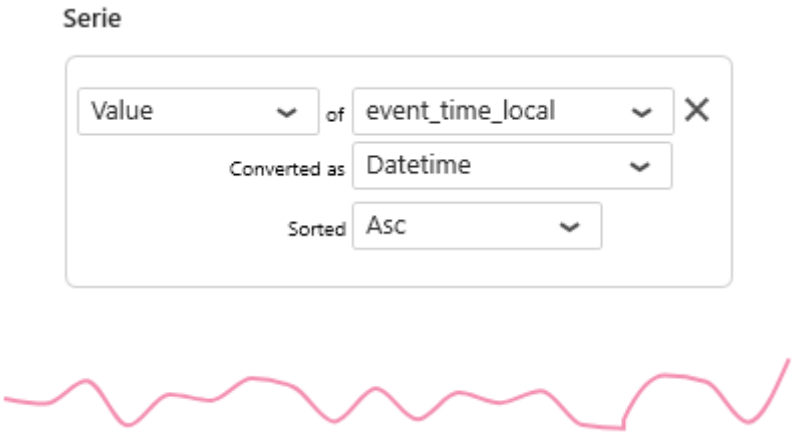
This field is required.



Serie

This is the field used for the X Axis of the sparkline chart.

This field is required.



Low

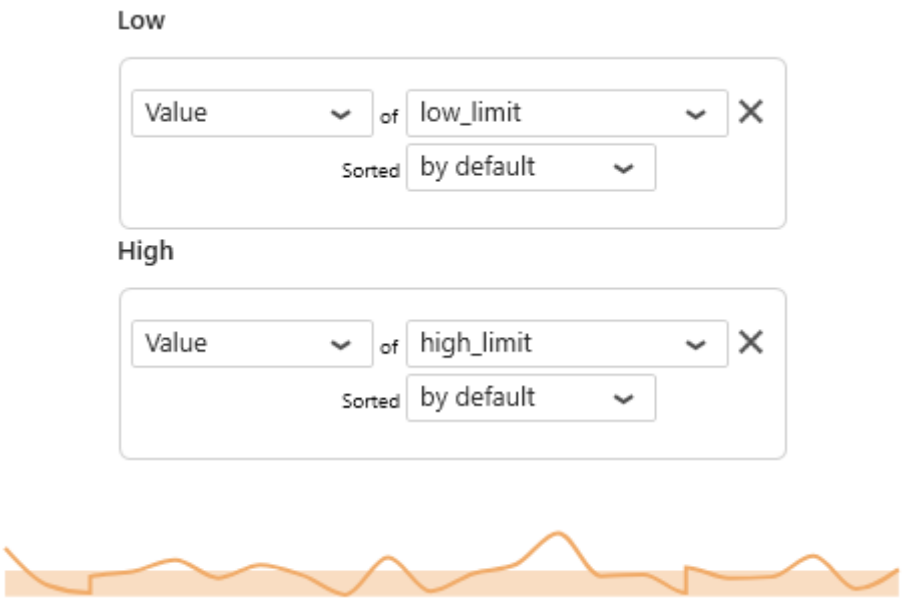
This is the lower bound of the limit.
The limit is displayed as a range area on the sparkline chart.

This field is optional, and only useful when a *High* field is also configured.

High

This is the higher bound of the limit.
The limit is displayed as a range area on the sparkline chart.

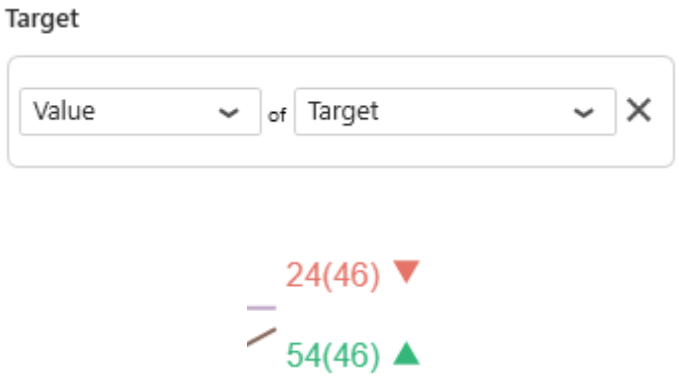
This field is optional, and only useful when a *Low* field is also configured.



Target

This is the target value for the variable.
The target is displayed as numerical value in the summary.

This field is optional.



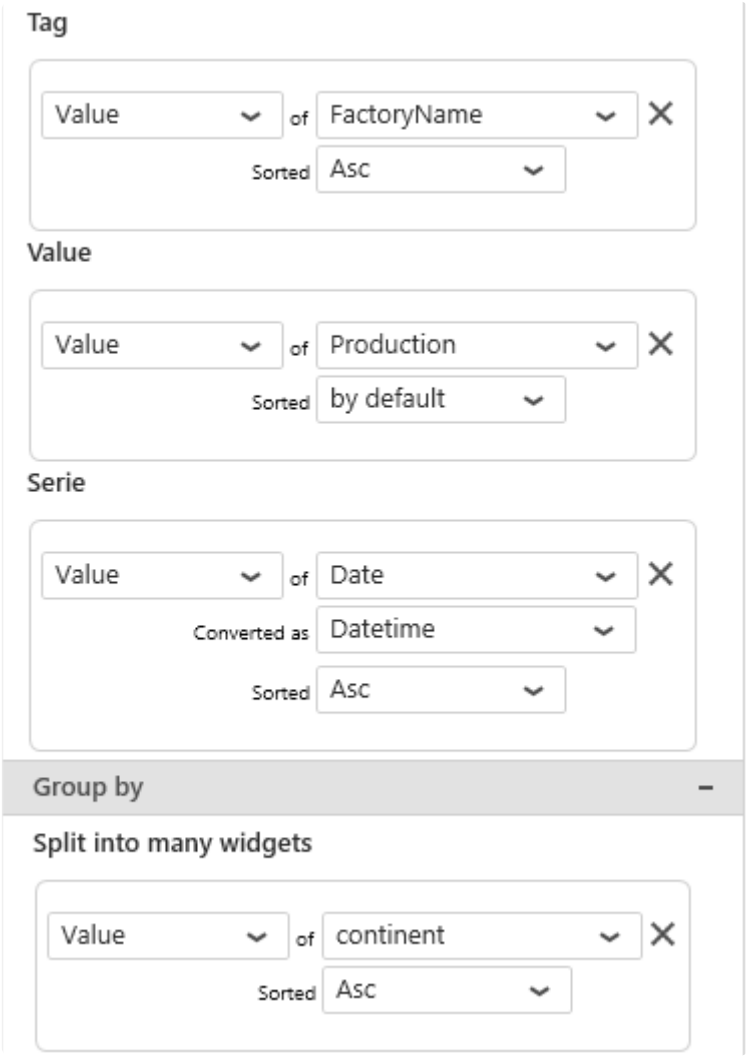
Group by : *Split into many widgets*

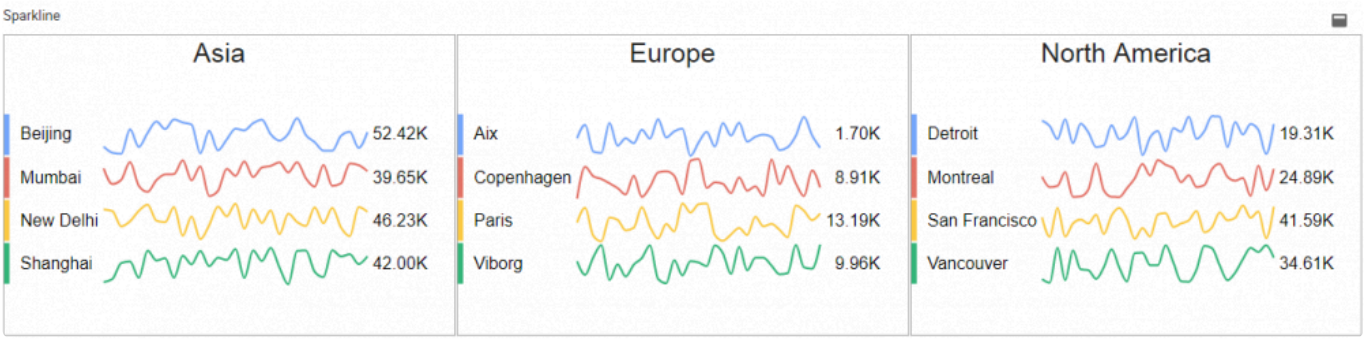
This represents the series field used to generate a different contained Sparkline Widget for each different value in this field.
This allows to create many sub-widgets dynamically.

This will turn the widget into a container, and generate a complete different chart for each value in the series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

This field is optional.

Example : If you have a field called “Continent”, you may want to plot a completely separate Sparkline for each value of “Continent”, dynamically :





Value aggregation

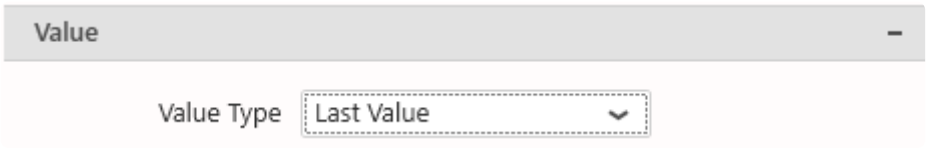
This Widget displays two values :

- the sparkline chart's Y value
- the summary value

You may configure the summary value in two different ways, under the *Value* category :

Value Type = Last Value

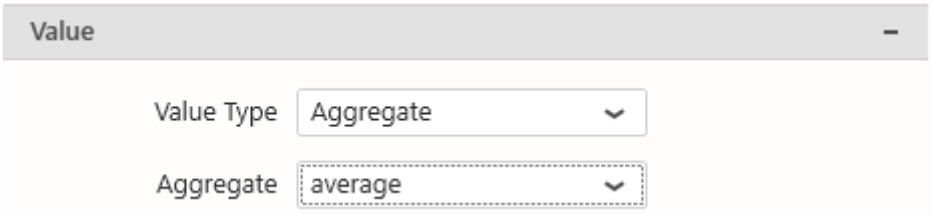
Displaying the *Last Value* means the last value on the sparkline chart, according to the sorting by the *Series* field.



Note : in this case, the *Aggregate* selection is ignored.

Value Type = Aggregate

Instead of displaying the last value from the sparkline chart, the *Aggregate* setting allows to calculate a value based on all values in the chart.
The aggregation function used to aggregate all the values in the chart is defined as the *Aggregate* option.

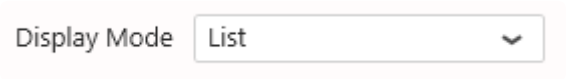


Last modified: Apr 22, 2020

2.8.9.2. Configuring Layout

Display Mode

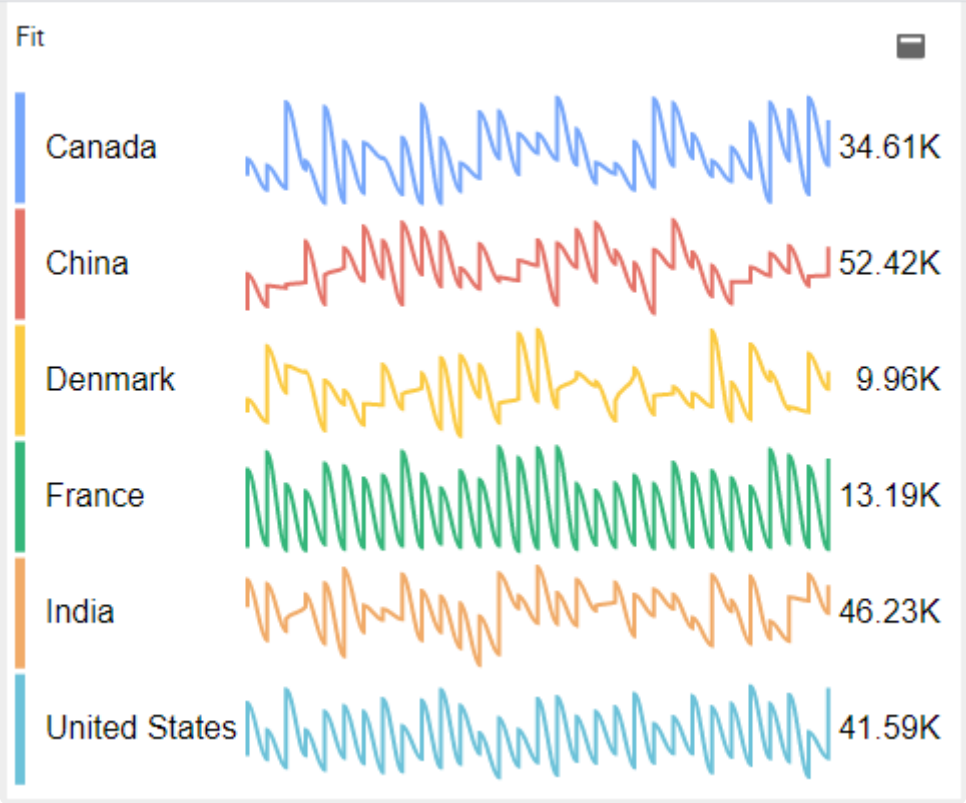
There are 3 ways to lay out the list of indicators vertically :



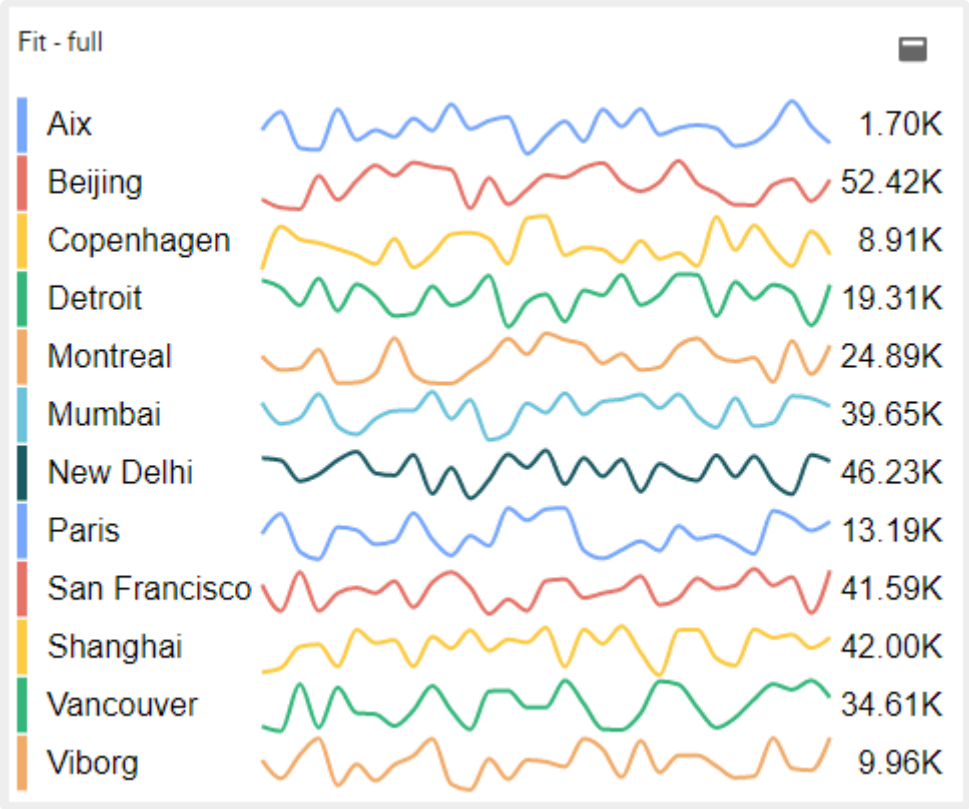
Fit

All indicators will fit the height allocated to the Widget, and take all the vertical space.

If there are few indicators and a lot of vertical space, they will be stretched vertically, and an lot of detail can be seen on the sparkline chart's variations.



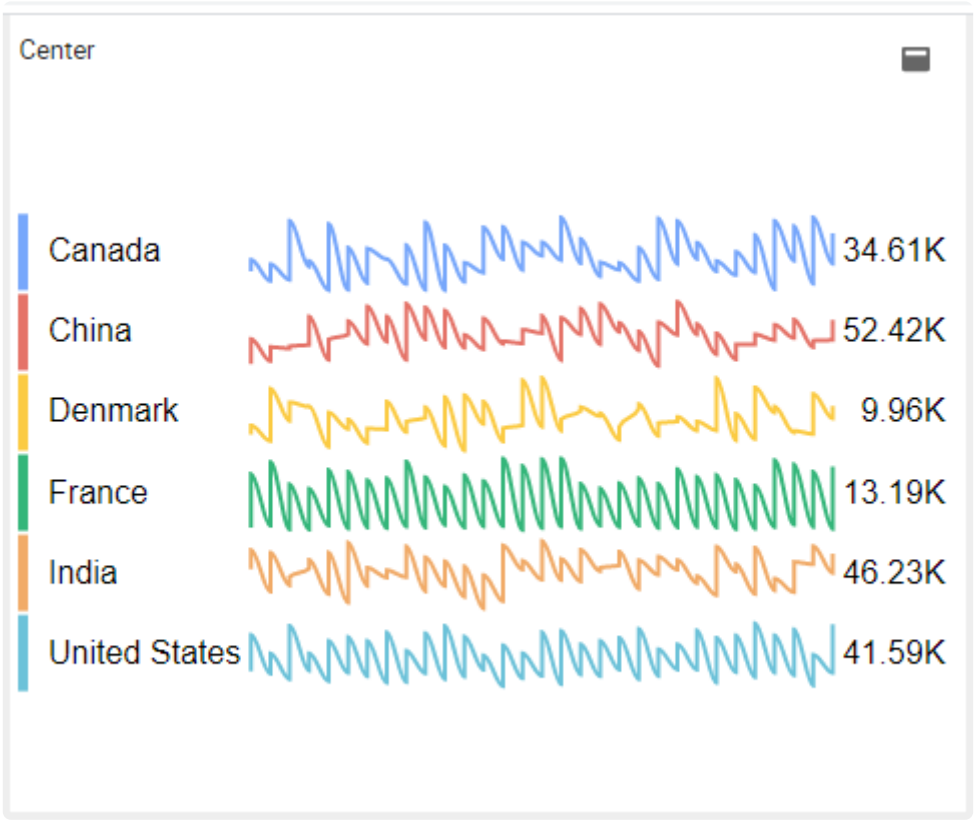
If there are many indicators and little vertical space, they will be squeezed vertically, so that all of them will fit.



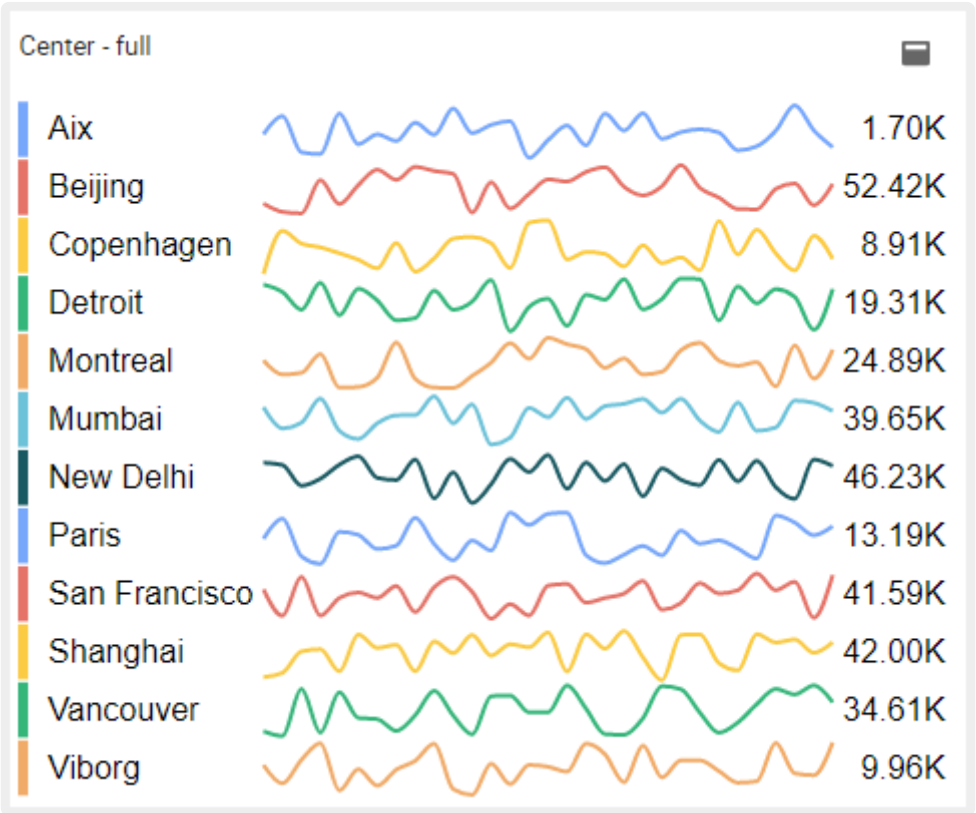
Center

The indicators will only attempt to fit if there are too many. Otherwise, they will be displayed with a fixed vertical size and centered vertically :

If there are few indicators and a lot of vertical space, they will not be stretched vertically, and there will be margins at the top and bottom.



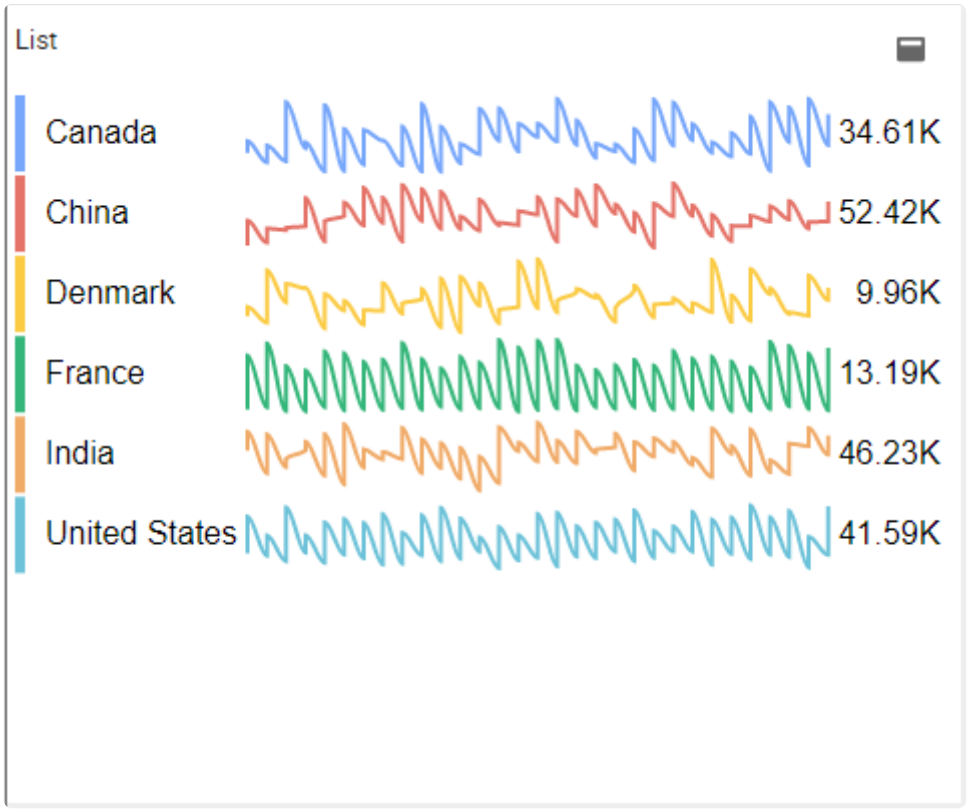
If there are many indicators and little vertical space, they will be squeezed vertically, so that all of them will fit (same as the *Fit* layout).



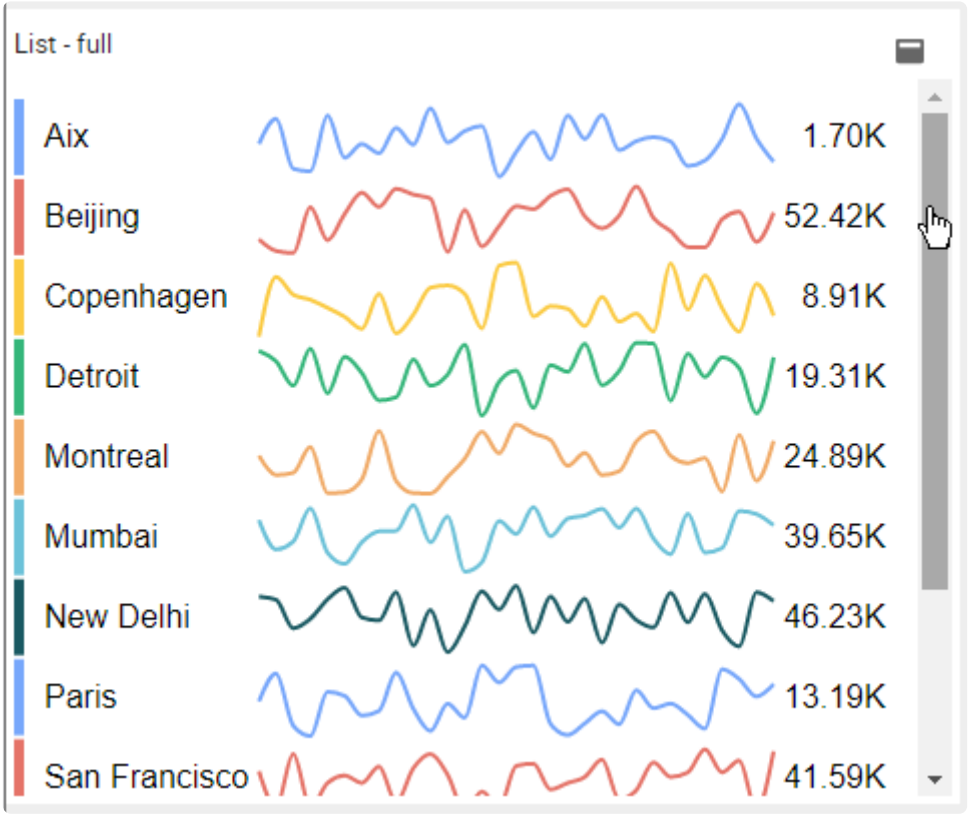
List

The indicators will not attempt to fit, they will be displayed with a fixed vertical size and aligned at the top :

If there are few indicators and a lot of vertical space, they will not be stretched vertically, and there will be margins at the bottom.



If there are many indicators and little vertical space, they will be displayed with a scroll bar.



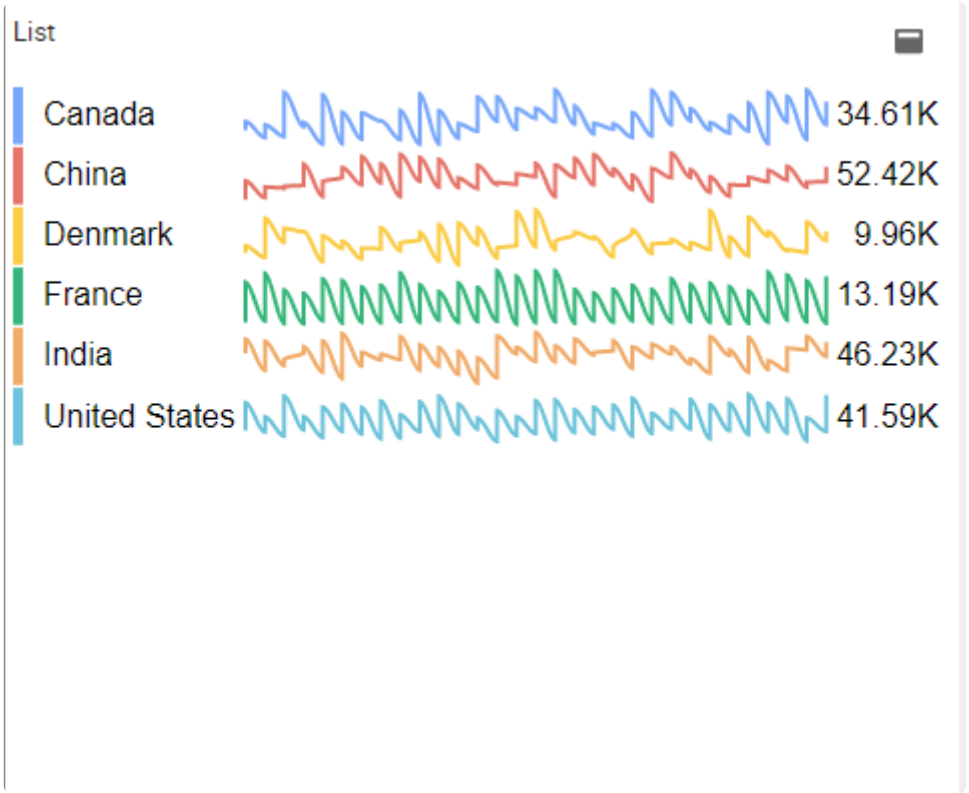
Row Size

This is the height of the indicators.

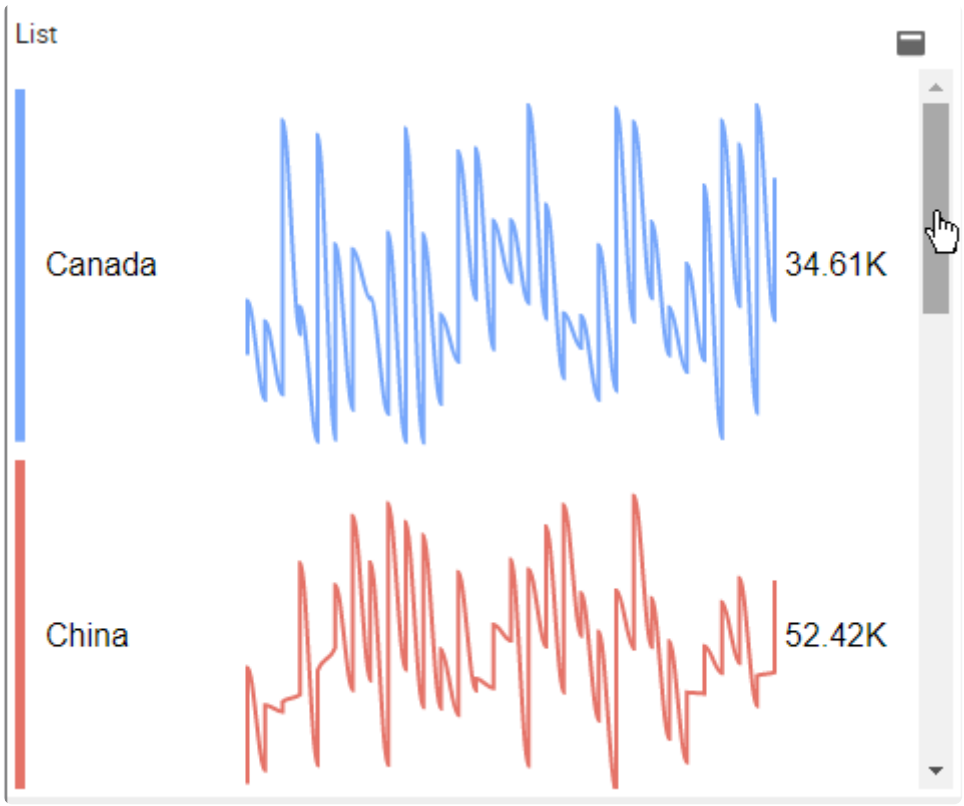
The effect depends on the *Display Mode* :

- *Fit* : *Row Size* is not applicable. The row height is calculated as the height of the Widget divided by the number of rows (number of *Tags*).
- *Center* : *Row Size* will only apply while there is some free vertical space
- *List* : *Row Size* will be the vertical size of the indicators

Example of small *Row Size* in List mode :



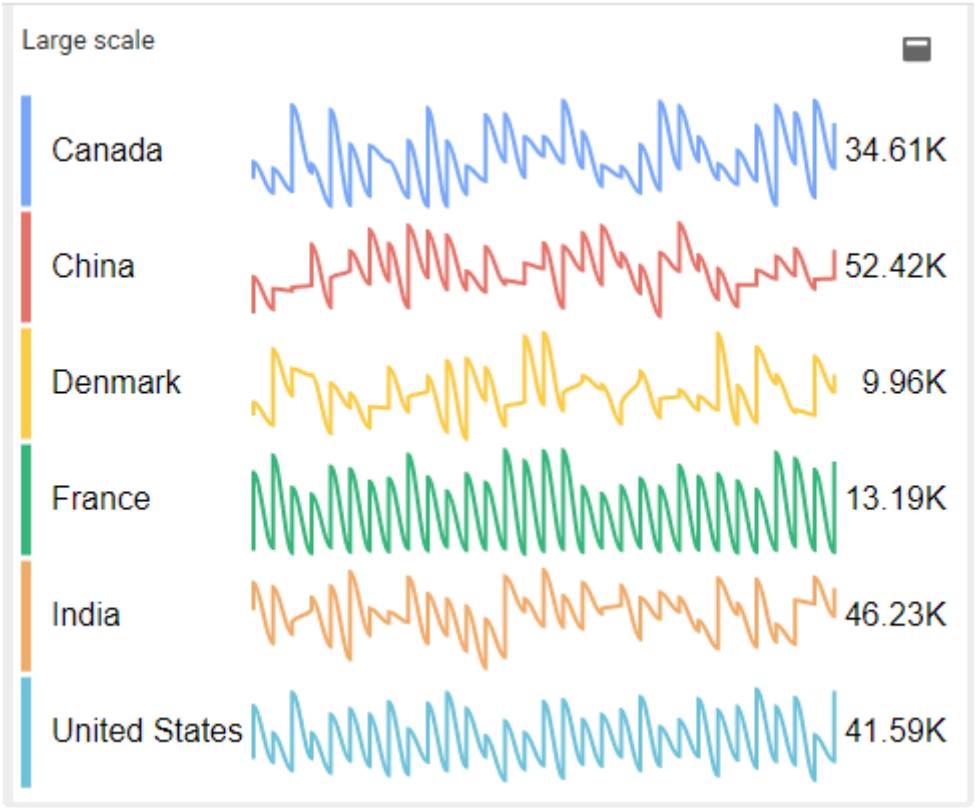
Example of large *Row Size* in List mode :

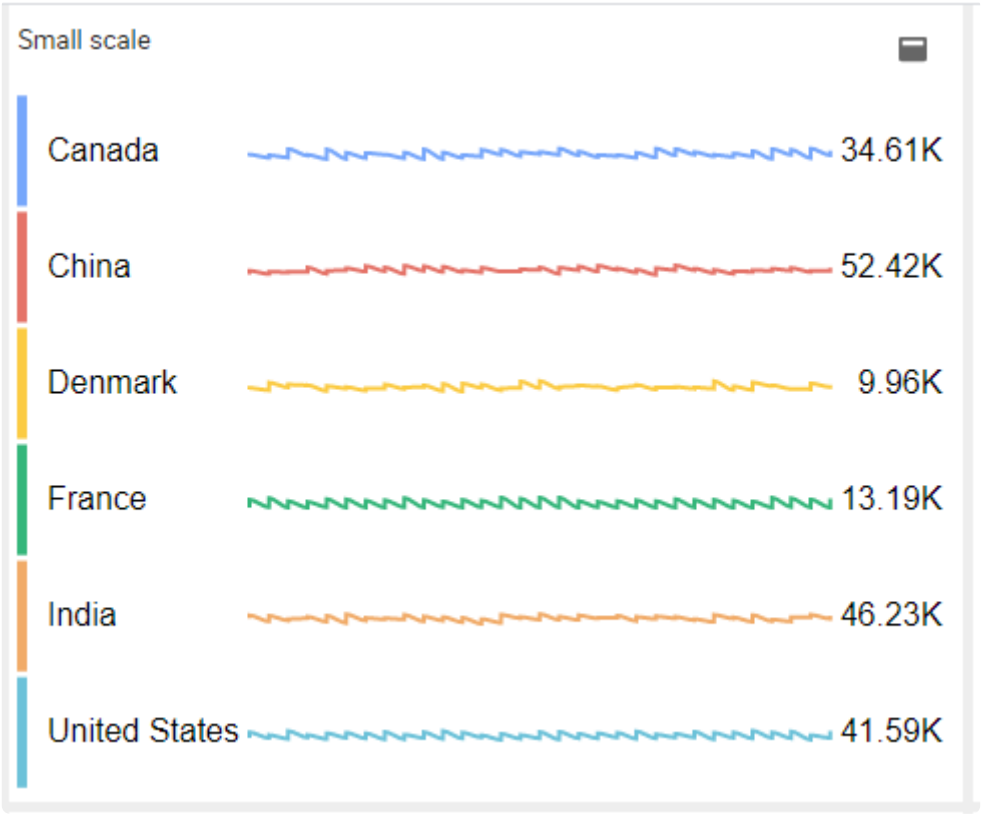


Sparkline Row Scale

Configure the relative vertical size of the sparkline chart, in % of the height of the row :

- `1` means to take the full height of the row
- `0.95` means almost the full height
- `0.05` means a very thin display

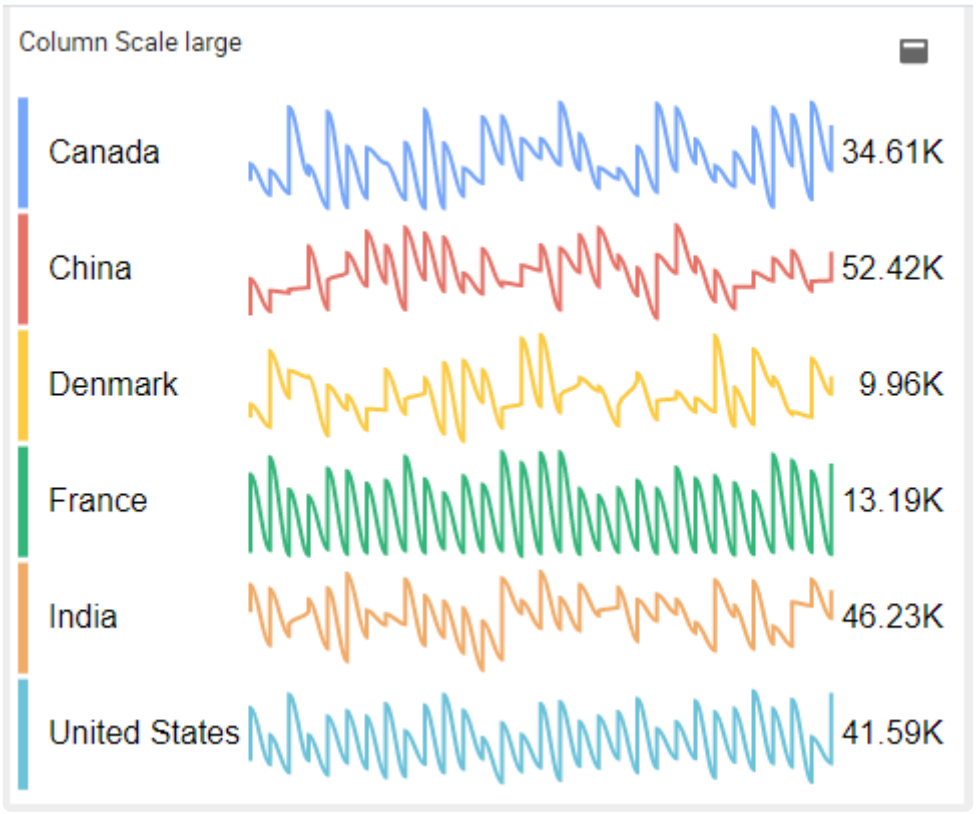


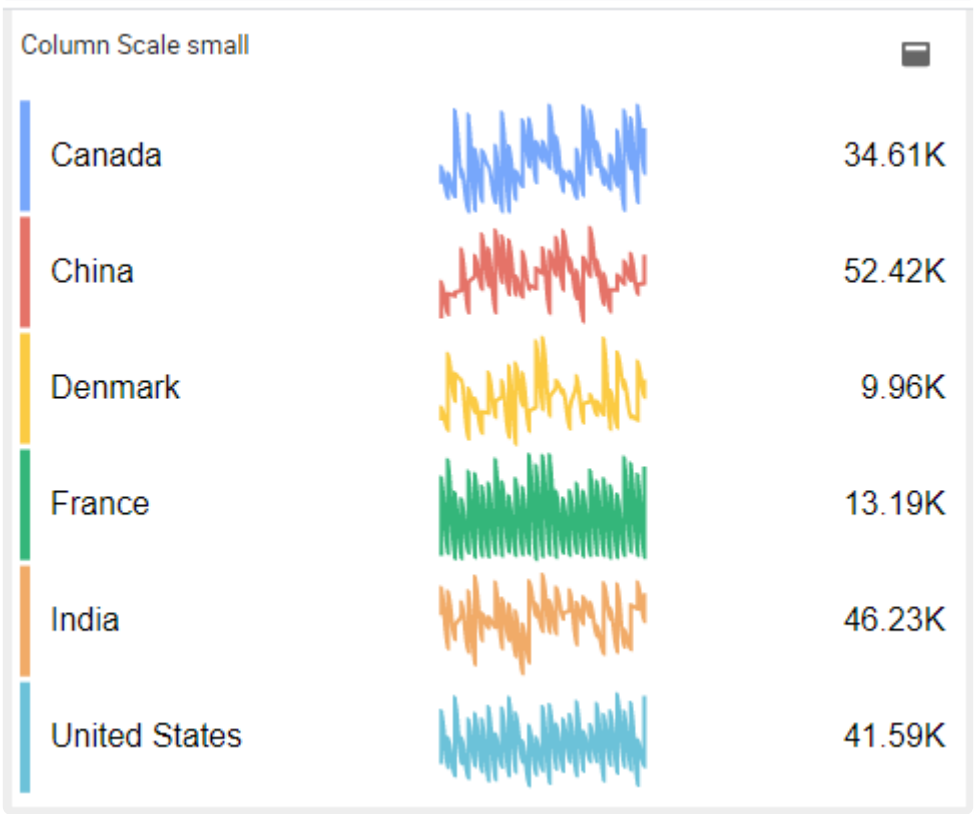


Sparkline Column Scale

Configure the relative horizontal size of the sparkline chart, in % of available space between the *Tag* name on the left and the summary value on the right :

- `1` means to take the full width
- `0.95` means almost the full width
- `0.05` means a very narrow display





Last modified: Apr 22, 2020

2.8.9.3. Sparkline chart style

Limits Opacity

This controls the opacity/transparency of the range area chart used to display the limit value, using fields *Low* and *High* as thresholds.

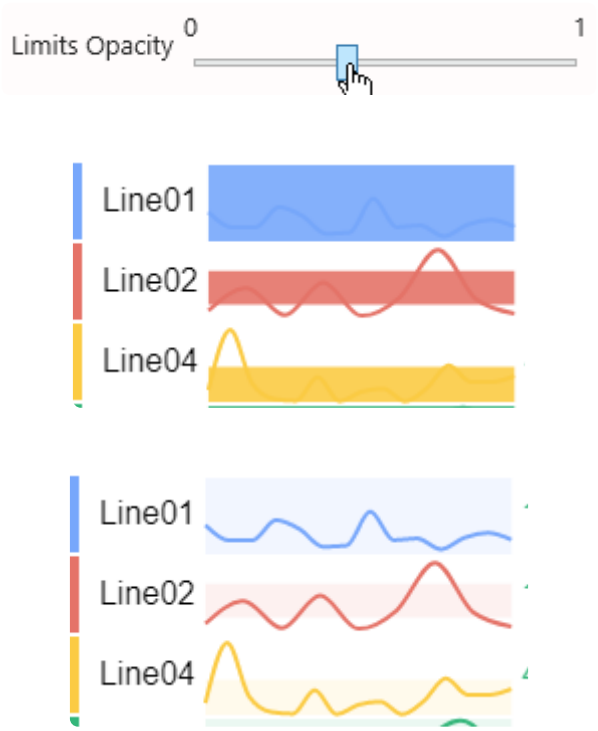


Chart interpolation

The main sparkline chart and the limit chart are each interpolated :

- *Line interpolate* : interpolation method used for the main sparkline chart
- *Limits interpolate* : interpolation method used for the limits range chart

Available interpolation methods are :

basis

A cubic basis spline
The result is a smooth curve which doesn't exactly pass through the data points themselves.

linear

A series of interconnected lines.
The result is that of a line chart.

monotone

A cubic spline that preserves monotonicity in y, assuming monotonicity in x.
The result is a smooth curve which goes through all data points and doesn't overshoot.

step

A step line centered on the data point.
The result is a step line which is constant around the data points.

step-after

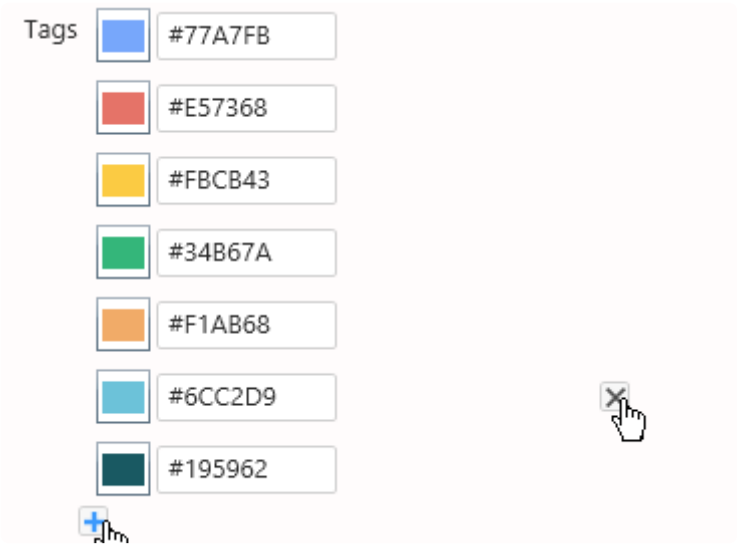
A step line starting on the data point.
The result is a step line which is constant after the data points.

step-before

A step line ending on the data point.
The result is a step line which is constant before the data points.

Chart colors

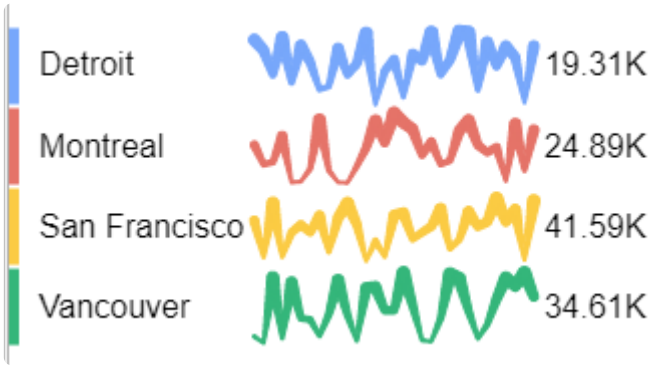
The color of each sparkline chart is picked from an indexed palette defined under *Color* > *Tags*.
You can add/remove colors and customize them.
If there is more Tags than colors, the colors are cycled.



Trail

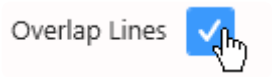
The sparkline charts can be configured to be drawn with a variable line thickness depending on the value.
Check *Use Trail* and define :

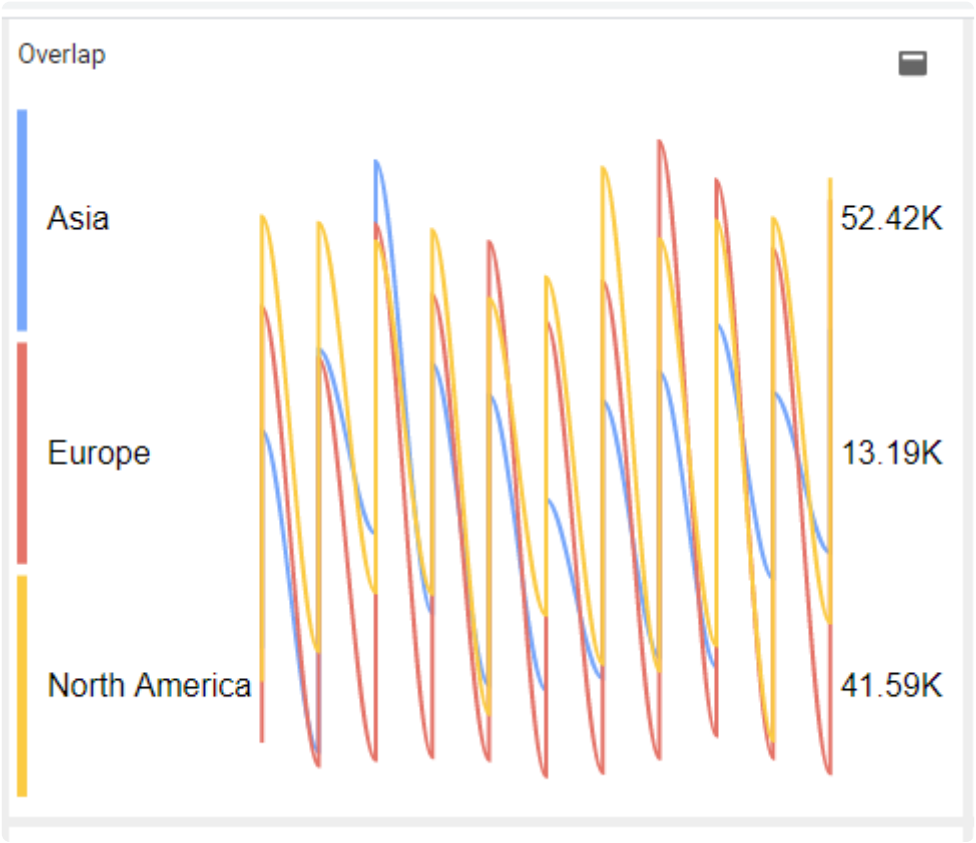
- *Trail Size Low* : the line thickness when the value is low
- *Trail Size High* : the line thickness when the value is high



Overlap Lines

This completely changes the display : instead of showing separate sparkline charts side-by-side, they are all superimposed on the same chart.





Last modified: Apr 28, 2020

2.8.9.4. Style and Appearance

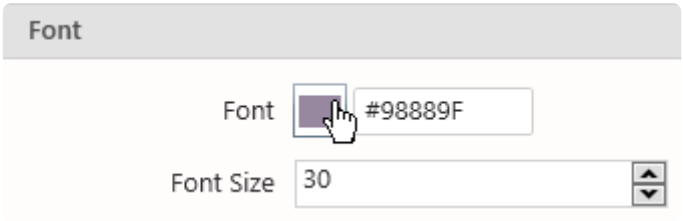
The summary value label displays :

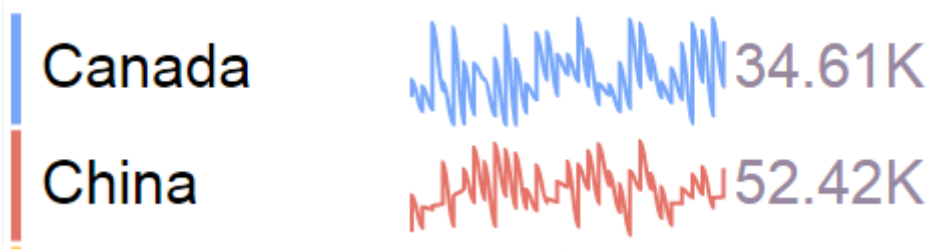
`Value (Target)`

The Target is omitted if Value = Target, or if the *Target* field is unbound.

Summary style

Font





Color

This defines the color of the text used to display the *Value* (the color is overridden if a *Target* is defined, see below).

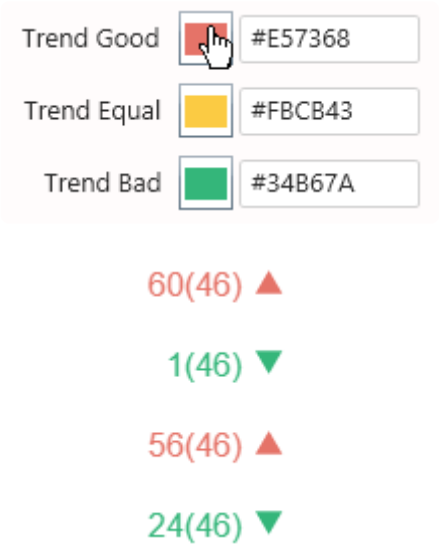
The color of the *Tag* name depends on the dashboard theme.

Font Size

This defines the font size and color of the text used to display :

- the *Tag* name
- the *Value*

Color



Trend Good

The color used to display the value summary when it is higher than the Target.

Trend Equal

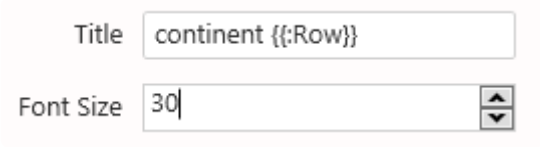
The color used to display the value summary when it is “Equal” to the Target.

Trend Bad

The color used to display the value summary when it is lower than the Target.

Title

The title and its font size.
It is especially useful when configured with the *Split into many widgets* configuration.



Last modified: Apr 22, 2020

2.8.10. Banner

The Banner Widget allows to display a stream of many KPI’s inside an animated scrolling container. Users can click on the widget to pause the scrolling, and click again to resume.

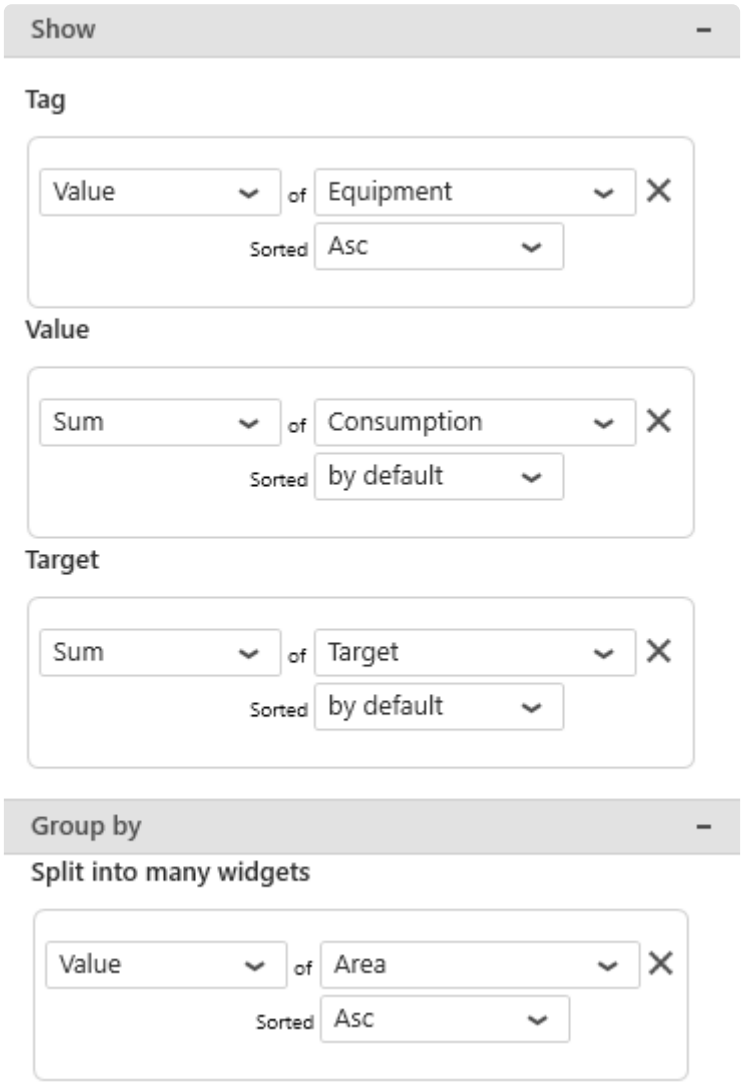
(new in Alpana 2020 R1)

Last modified: Apr 24, 2020

2.8.10.1. Binding Data

The Banner has 4 data fields available for binding :

- *Tag* : how we split series of different items in the same stream
- *Value* : main value displayed
- *Target* : target displayed, compared to the Actual Value
- *Split into many widgets* : how we split series of different Banners in the same container



Tag

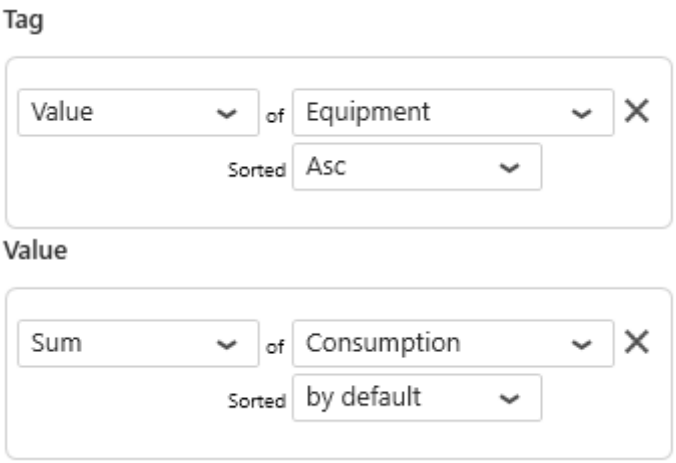
This represents the series field to generate different items in the stream of displayed values.

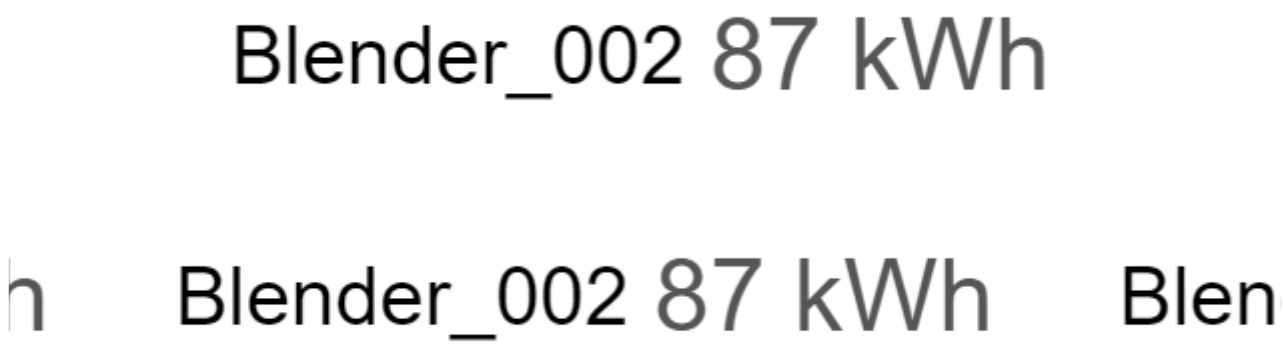
This field **cannot be aggregated** : it must have several separate values in order to create separate items and group the “Values” aggregation.
This field is required.

Value

This represents the main number value displayed for the items.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …)
This field is required.

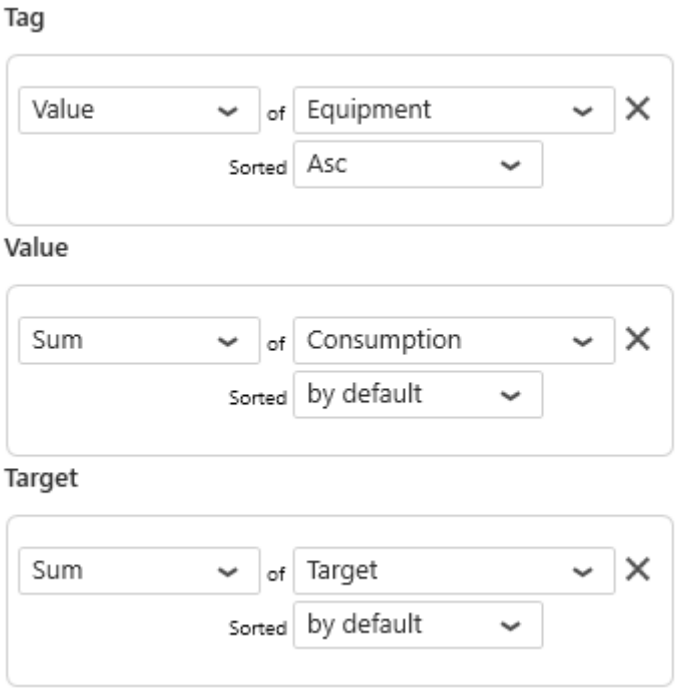


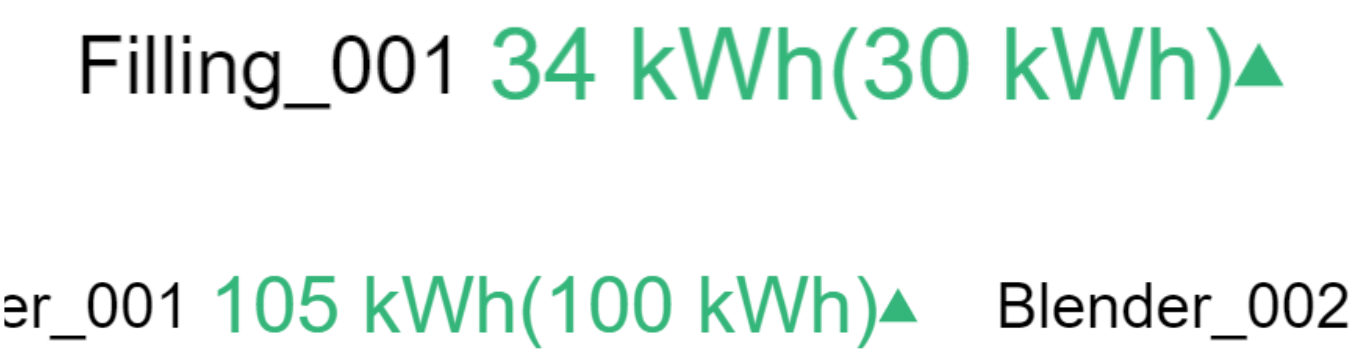


Target

This represents a target value to compare to the main *Value*.

This field must be aggregated : you must choose an aggregation function (SUM, AVG, …)
This field is optional.





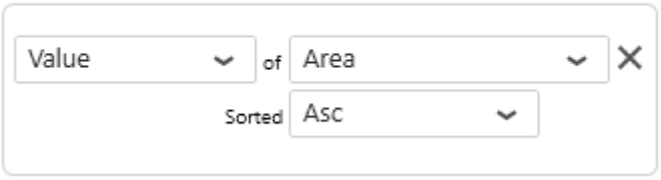
Group By : *Split into many widgets*

This represents the series field used to generate a different contained Banner Widget for each different value in this field.
This allows to create many sub-widgets dynamically.

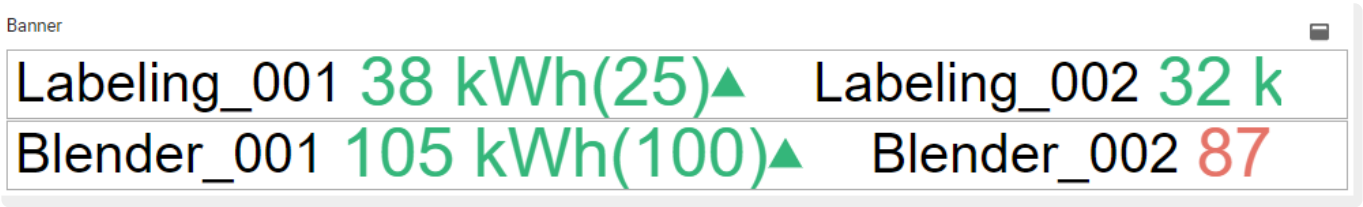
This will turn the widget into a container, and generate a complete different chart for each value in the series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

This field is optional.

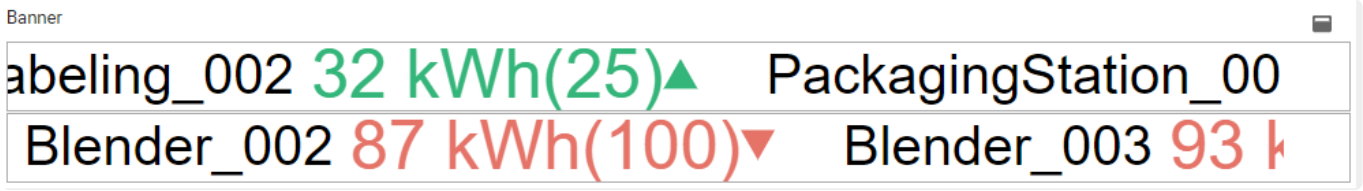
Split into many widgets



Banner



Banner

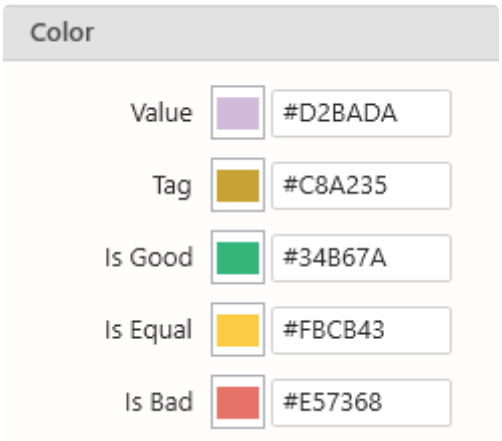


Last modified: Apr 23, 2020

2.8.10.2. Style and Appearance

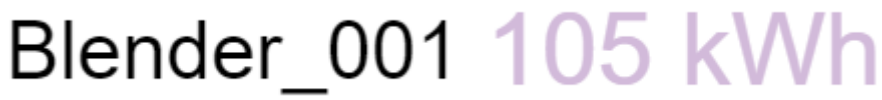
Colors

The colors of items can be configured :



Value

This is the color of the numerical value (the value from the *Value* field).



If a *Target* field is configured, then this color is ignored.
Instead the Good/Bad color is displayed (see below).

Tag

This is the color of the item's label (the value from the *Tag* field).

Is Good

This is the color of the numerical values when the main *Value* is considered "good" compared to the *Target* value.

Is Equal

This is the color of the numerical values when the main *Value* is considered "equal" compared to the

Target value.

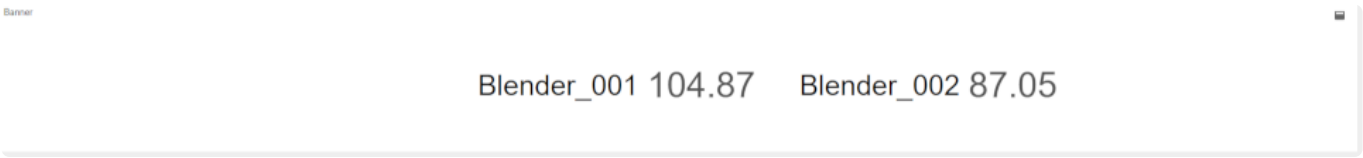
Is Bad

This is the color of the numerical values when the main *Value* is considered “bad” compared to the *Target* value.

Layout

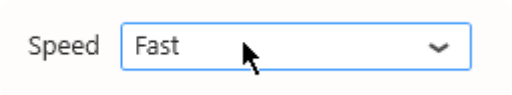
Item Group Size

The stream of items is grouped in packets.
This setting controls the size of the group.



Speed

This setting controls the scrolling speed of the stream of values.



Last modified: Apr 23, 2020

2.8.11. Proportion Charts

Proportion Charts allow to see the relative importance of a quantity in different context values.

Proportion Charts are split into several Widget types, but it’s possible to change type after configuring the widget :

- Pie
- Doughnut (Donut)
- Pyramid
- Funnel

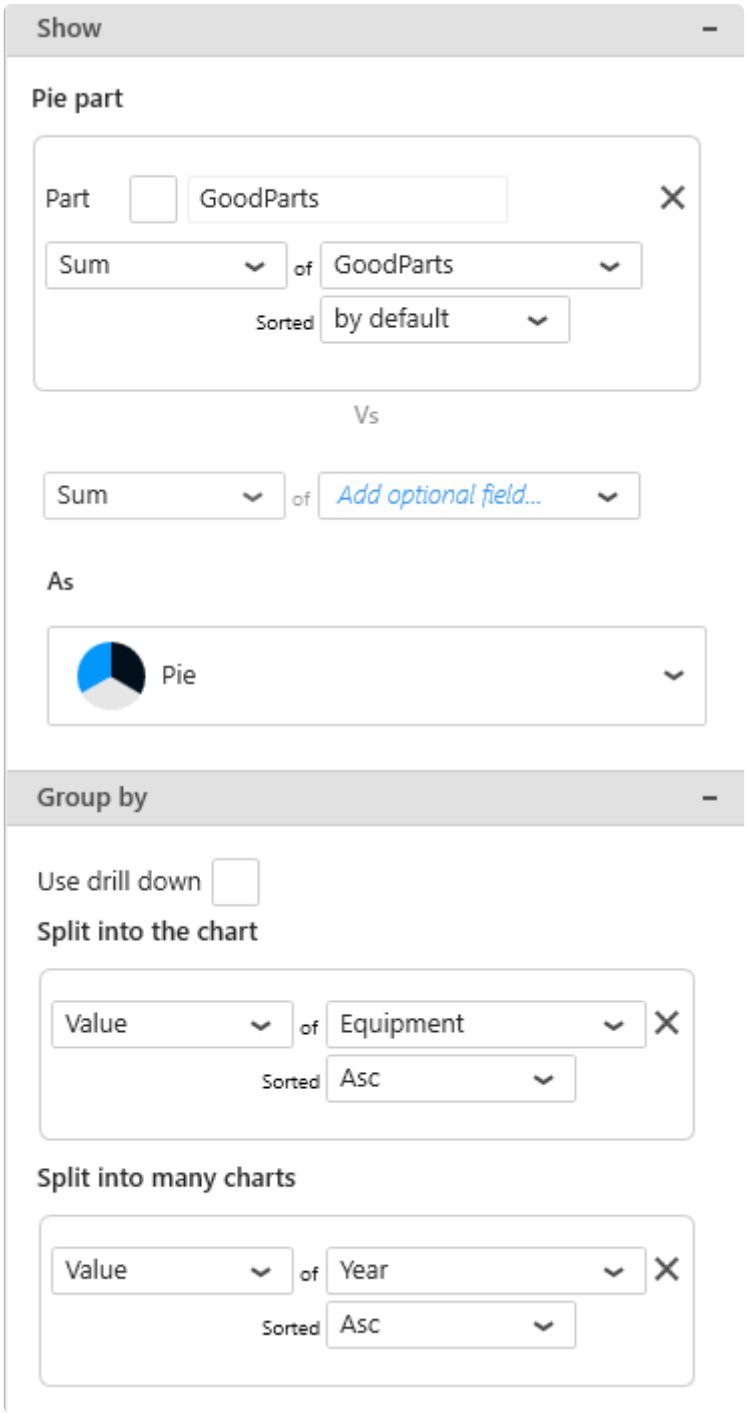
They all have the same properties documented in this Chapter.

Last modified: May 15, 2019

2.8.11.1. Binding Data

Introduction

The following fields are available for configuring data binding on a Proportion Chart :



- *Part* : the size of the portion (pie slice, doughnut slice, etc)
- *Split into the chart* : how we group values together to create separate portions
- *Split into many charts* : how we split series of different charts in the same container

Parts

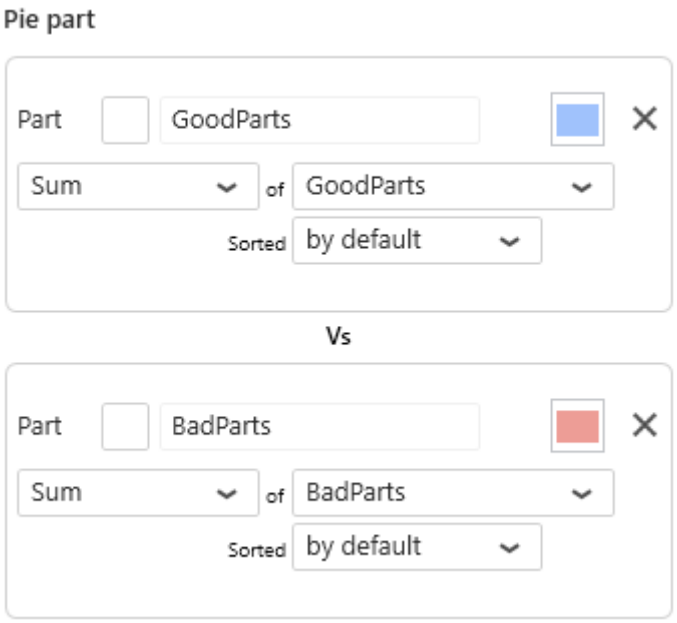
This represents the *quantity* we want to display on the Proportion Chart, and will bind to the size of the portion (pie slice, doughnut slice, etc).
These fields **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)

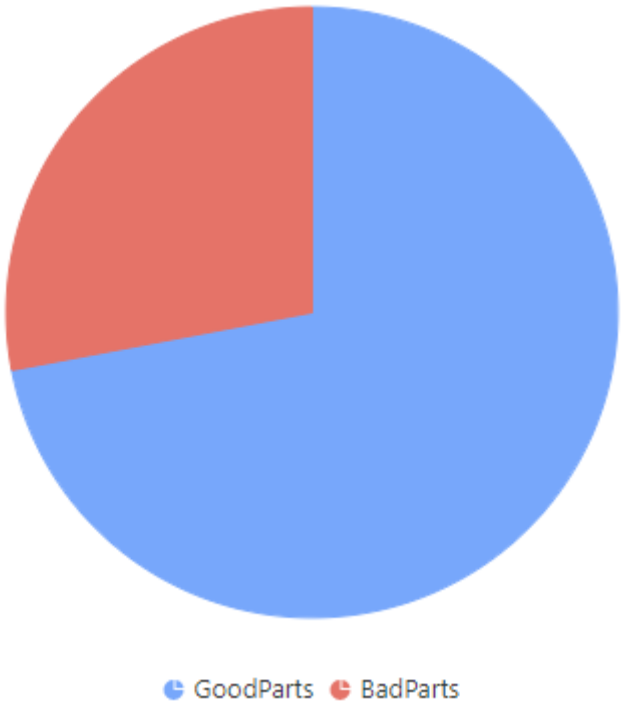
At least one *Part* field must be configured, but it's also possible to configure several.
When a single *Part* field is used, the portions are separated by the values in the *Split into the chart* fields (see below).

When binding multiple *Part* fields at the same time, 2 behaviors are possible :

Multiple *Part* fields : multiple portions

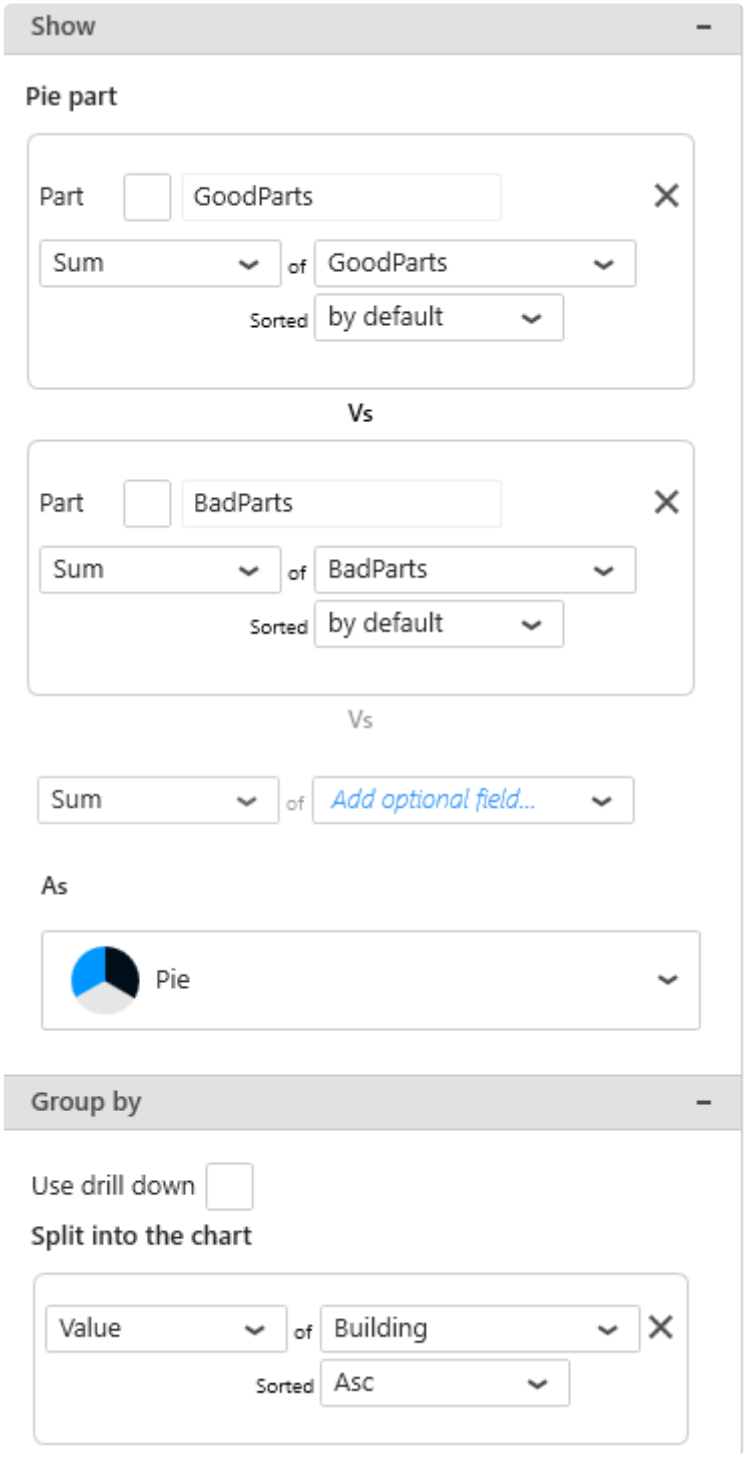
If there is no *Split into the chart* field, each Value field will generate a portion :

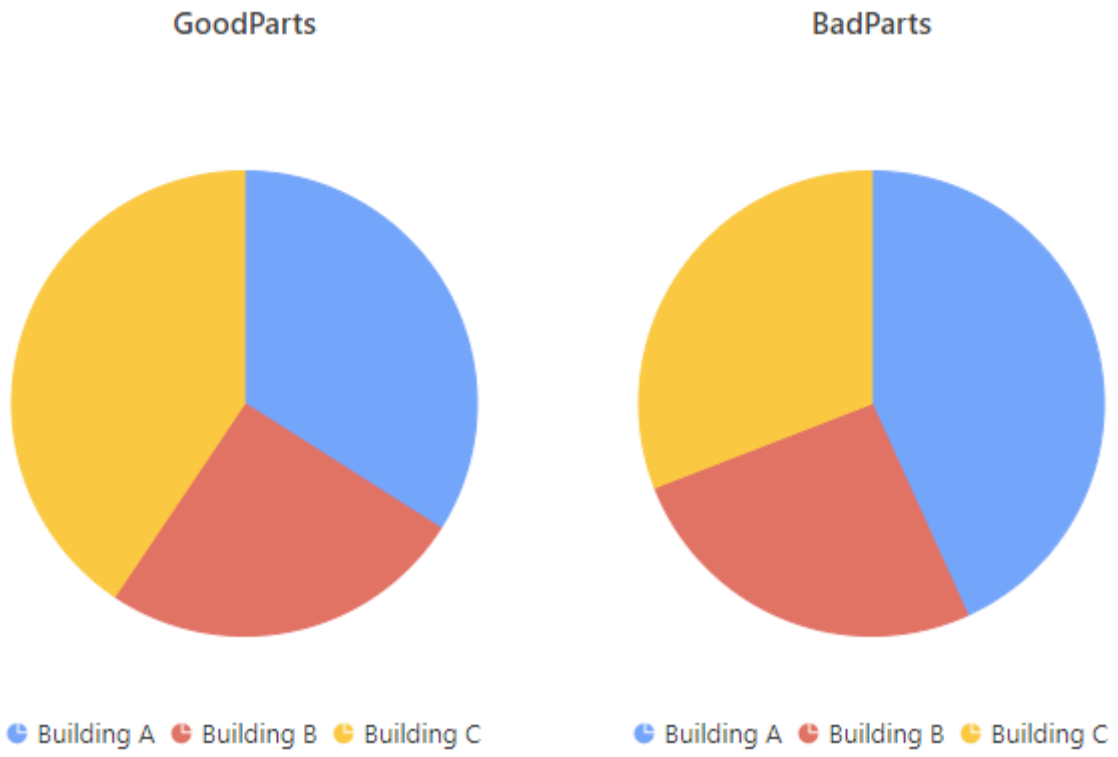




Multiple *Part* fields with *Split into the chart* : multiple series

If a *Split into the chart* field is bound, each *Part* field will generate a separate Proportion Chart :


Show
Pie part
Part
GoodParts
Sum
of
GoodParts
Sorted
by default
Vs
Part
BadParts
Sum
of
BadParts
Sorted
by default
Vs
Sum
of
Add optional field...
As
Pie
Group by
Use drill down
Split into the chart
Value
of
Building
Sorted
Asc



Split into the chart

This represents the *category* we want to use to dynamically separate portions on the Pie/Doughnut/etc.

These fields **cannot be aggregated** : they must have several separate values in order to group the “Values” aggregation.

Show

Pie part

Part GoodParts

Sum of GoodParts

Sorted by default

Vs

Sum of Add optional field...

As

Pie

Group by

Use drill down

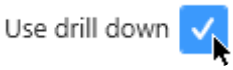
Split into the chart

Value of Building

Sorted Asc

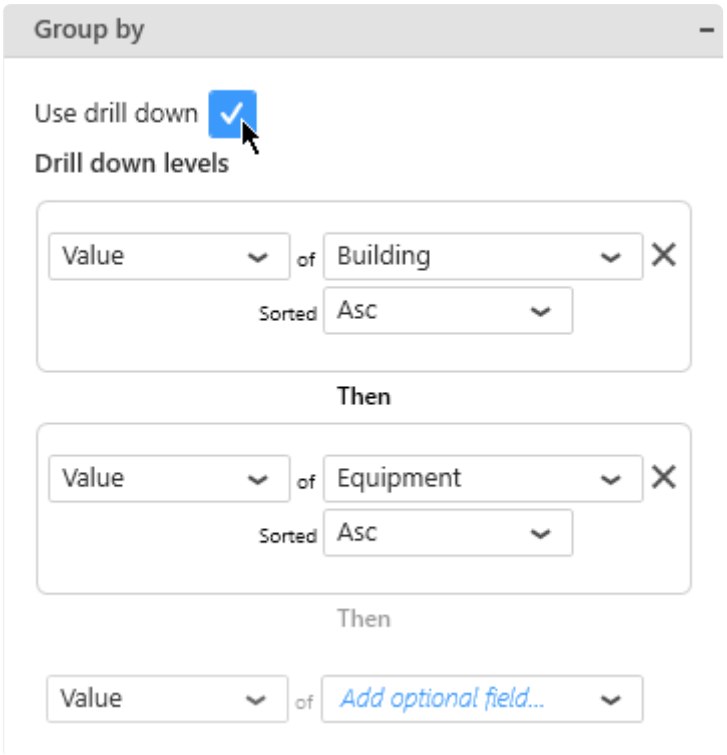
Multiple *Split into the chart* fields : Drill-Down

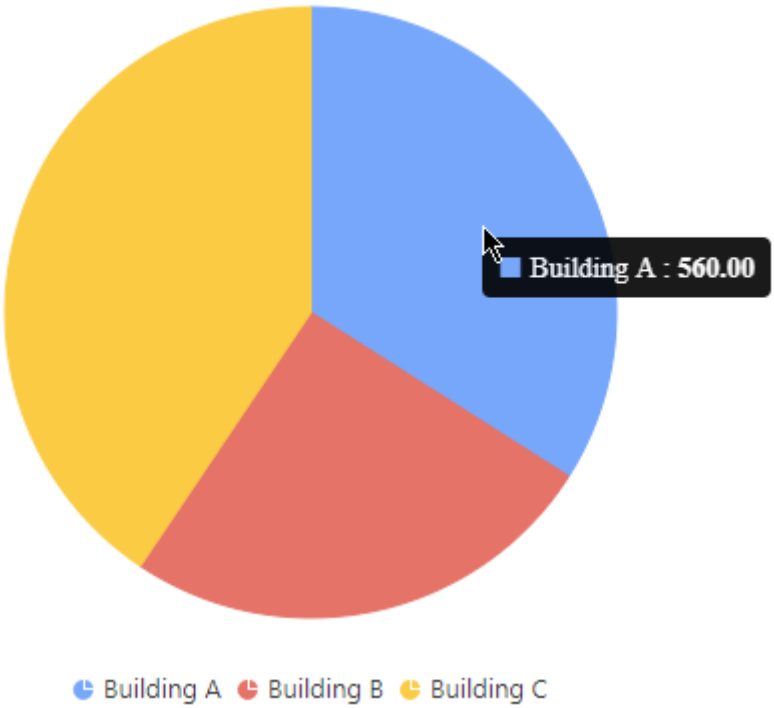
To enable Drill-Down, you can check the box *Use drill down* :



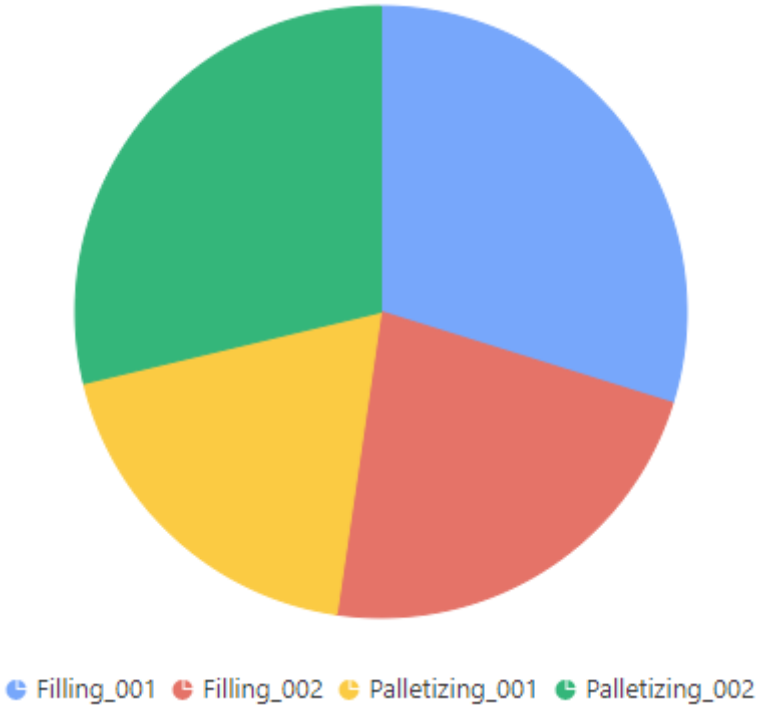
This allows to bind any number of fields to drill-down levels, in the same order as they are in the list. To change the order of fields, see the corresponding chapter.

Example : With fields called “Building” and “Equipment”, the following configuration will configure the corresponding drill-down :





… drills down into :



Split into many charts

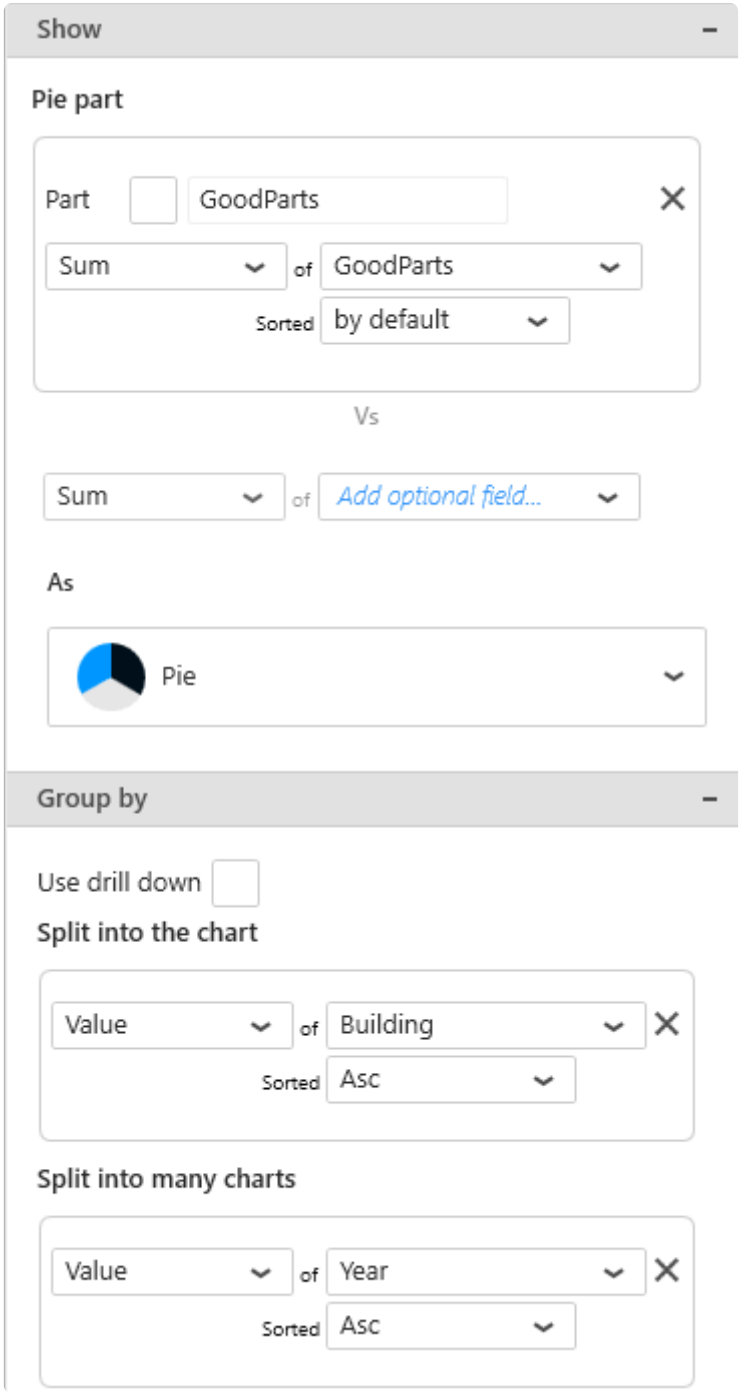
This represents the series field to generate a different contained Proportion Chart for each different value in this *Split into many charts* field.

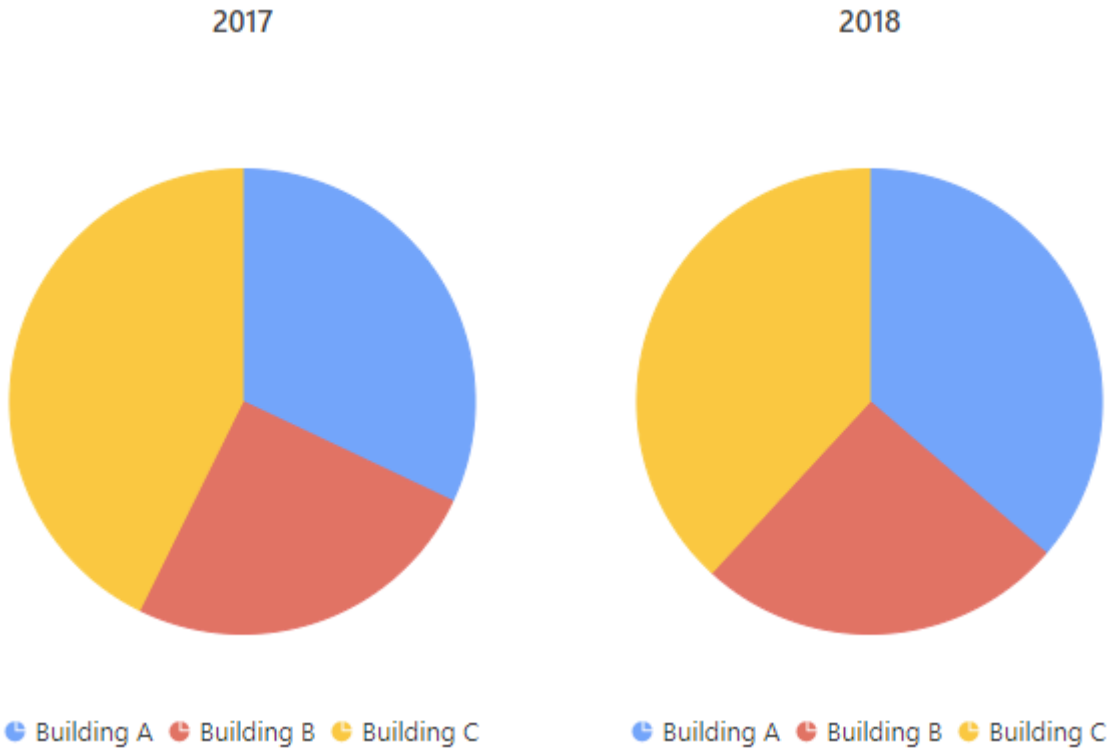
This will turn the widget into a container, and generate a complete different chart for each value in the

series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

Split into many charts Series with *Split into the chart* as portions

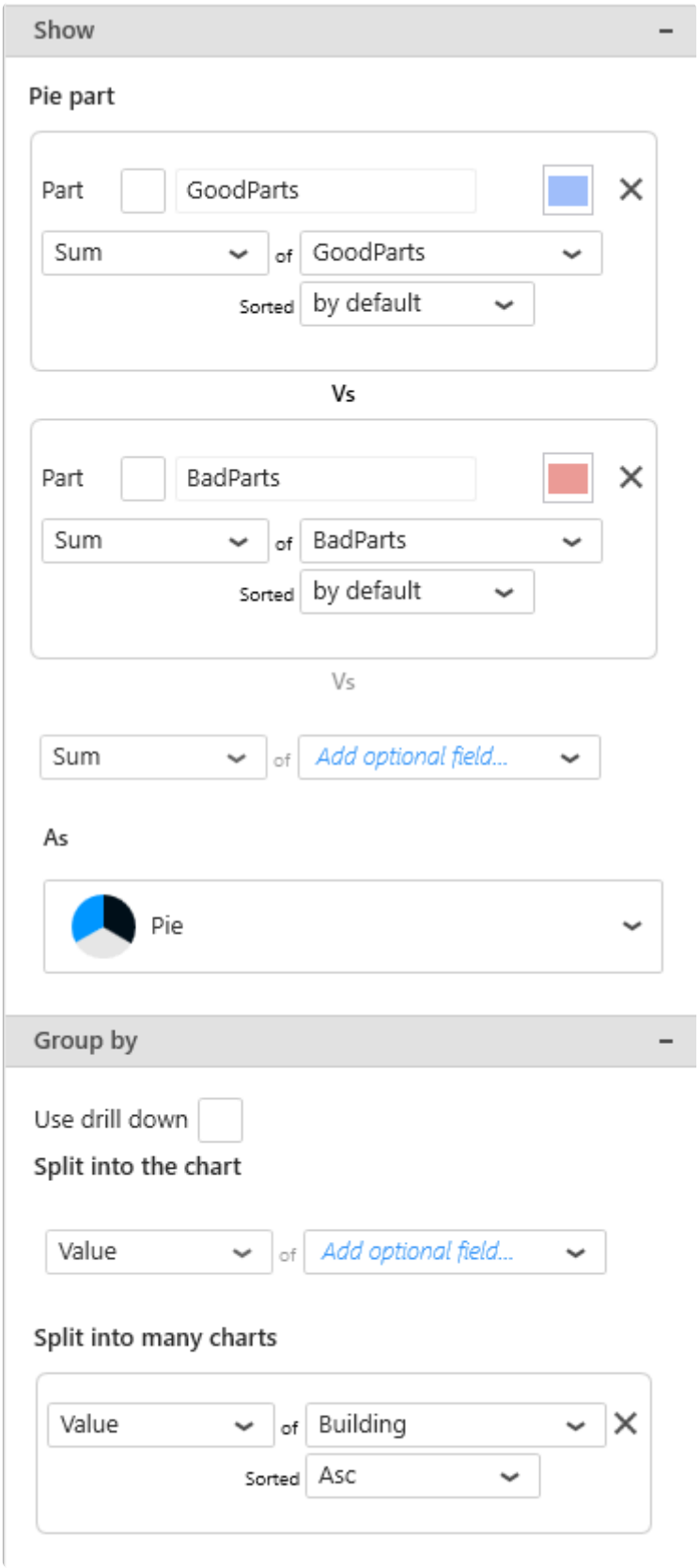
Example : Your GoodParts are compared by Building. If you have a field called “Year”, you may want to plot a completely separate chart for each value of Year :

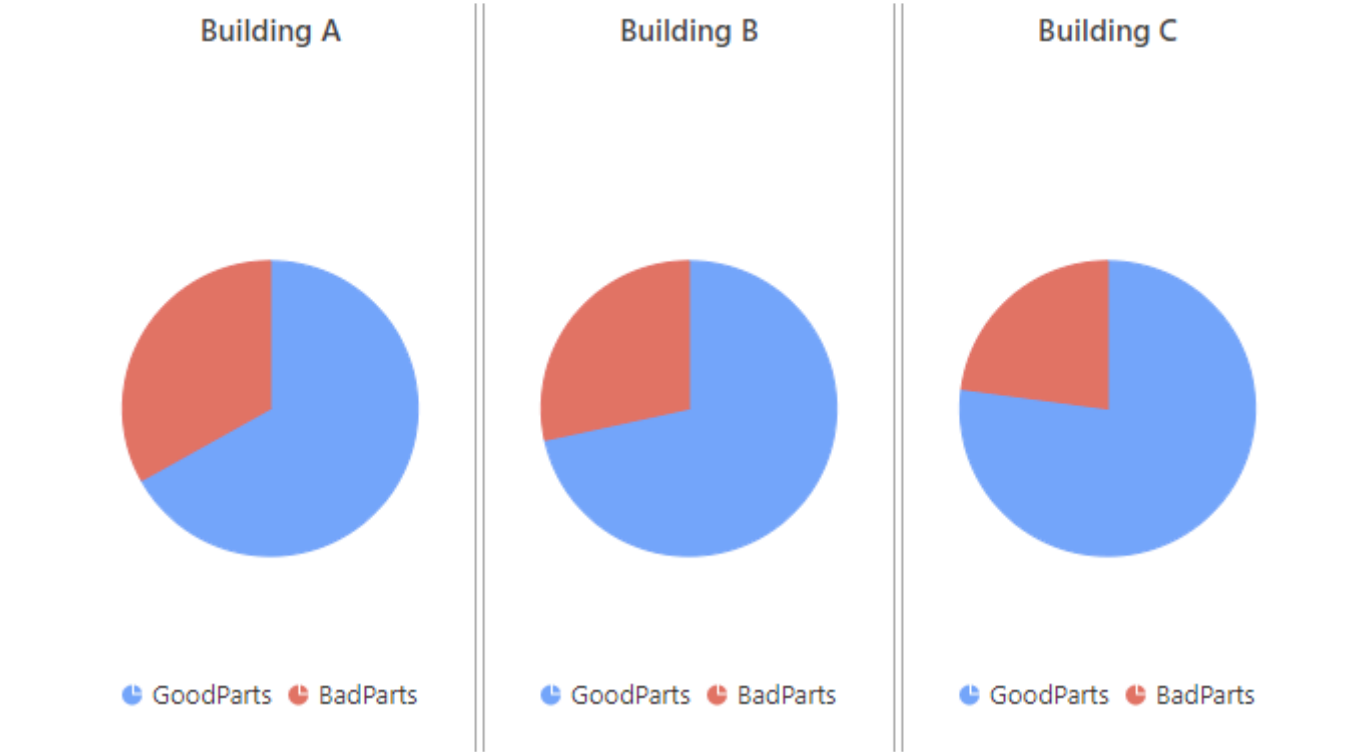




Split into many charts Series with multiple *Parts* as portions

Example : Your GoodParts are displayed versus the BadParts. If you have a field called “Building”, you may want to plot a completely separate chart for each value of Building :

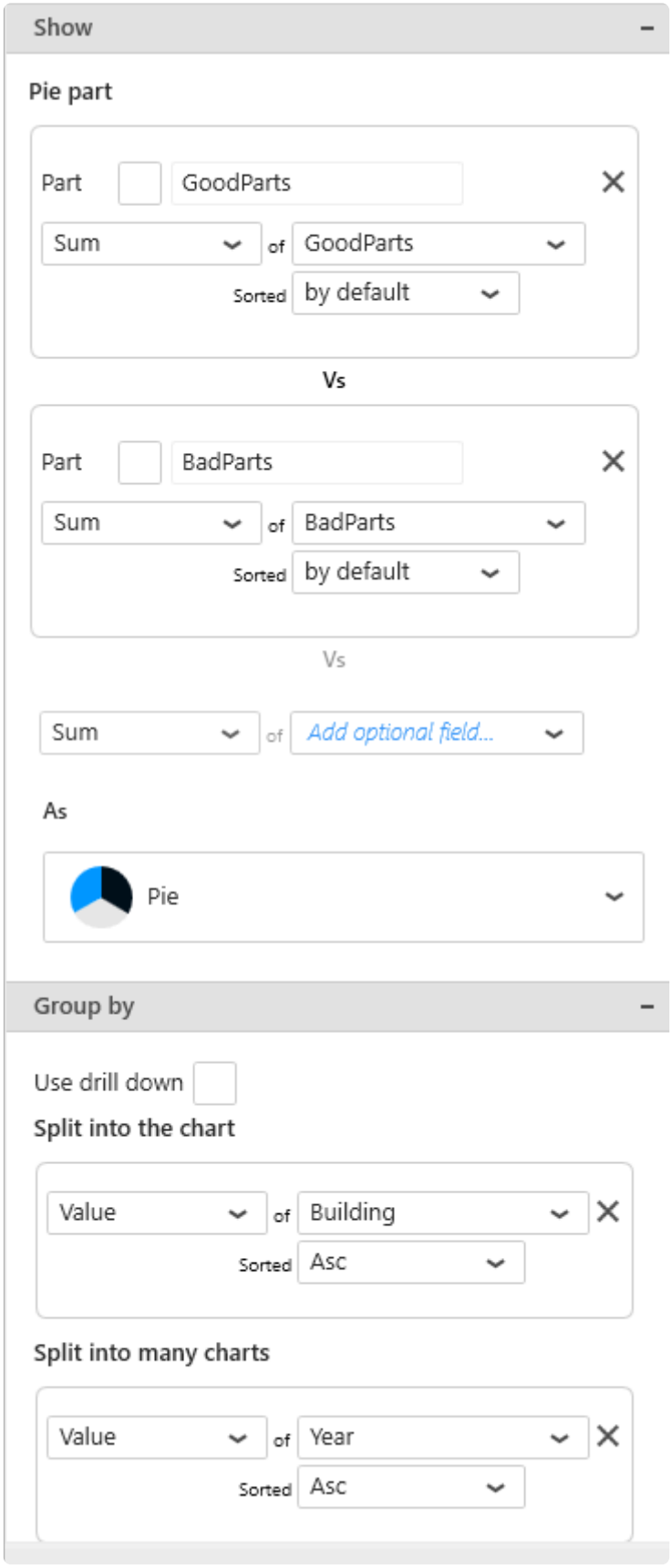

Show
Pie part
Part
GoodParts
Sum
of
GoodParts
Sorted
by default
Vs
Part
BadParts
Sum
of
BadParts
Sorted
by default
Vs
Sum
of
Add optional field...
As
Pie
Group by
Use drill down
Split into the chart
Value
of
Add optional field...
Split into many charts
Value
of
Building
Sorted
Asc

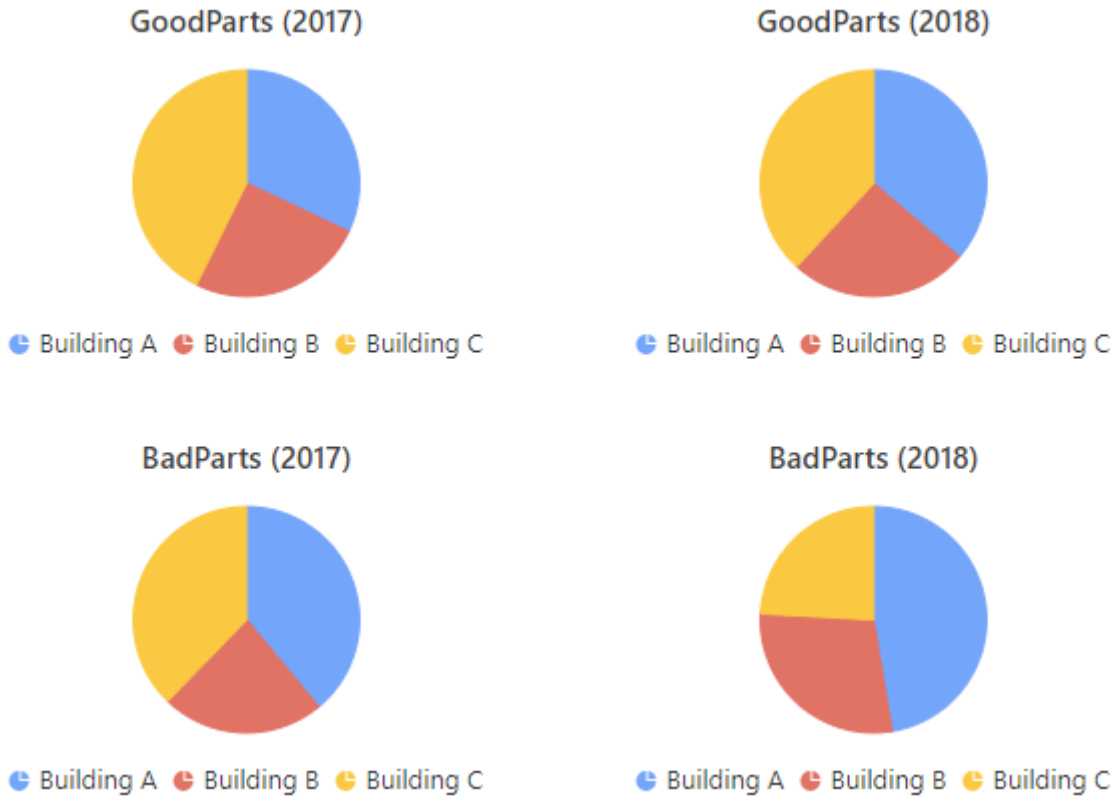


Split into many charts Series multiple *Parts* with *Split into the chart* as portions

All three types of fields can also be configured together : each *Part* field will generate a separate chart for each separate value of the *Split into many charts* field, and the portions will be defined by the *Split into the chart* field.

Example : You want a separate chart for the GoodParts of each Year, and a separate chart for the BadParts of each Year, and on each chart you want to compare the Buildings values :


Show
Pie part
Part
GoodParts
Sum
of
GoodParts
Sorted
by default
Vs
Part
BadParts
Sum
of
BadParts
Sorted
by default
Vs
Sum
of
Add optional field...
As
Pie
Group by
Use drill down
Split into the chart
Value
of
Building
Sorted
Asc
Split into many charts
Value
of
Year
Sorted
Asc



Split into many charts Series with synchronized Drill-Down

If Drill-Down is configured, it becomes synchronized between each chart in the series.

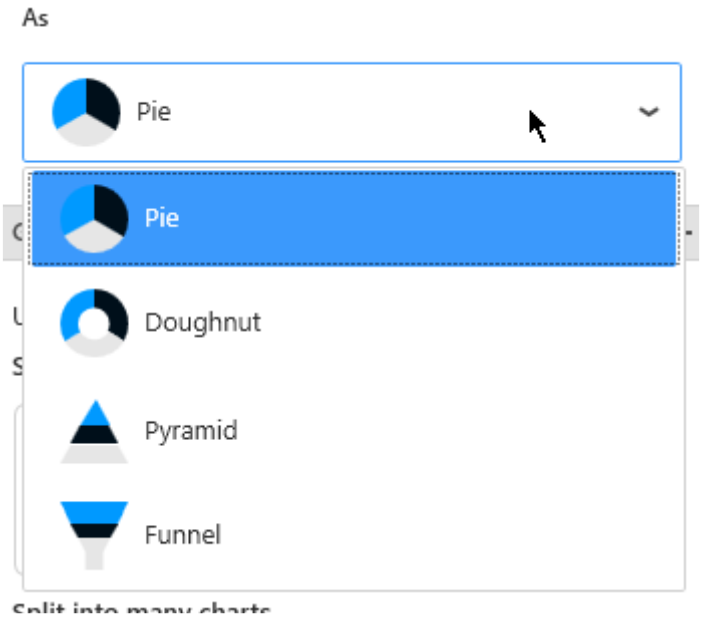
Last modified: May 16, 2019

2.8.11.2. Switching Chart type

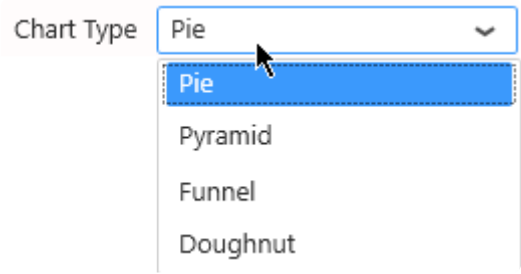
Proportion Chart type

Chart type can be changed on the fly while keeping the widget configuration.

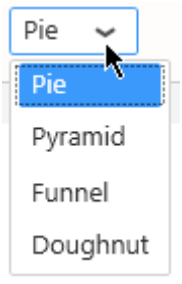
To change the Chart type, use the selector under *As* in the data configuration :



Or use the setting *Chart Type* under *Basic Settings* :

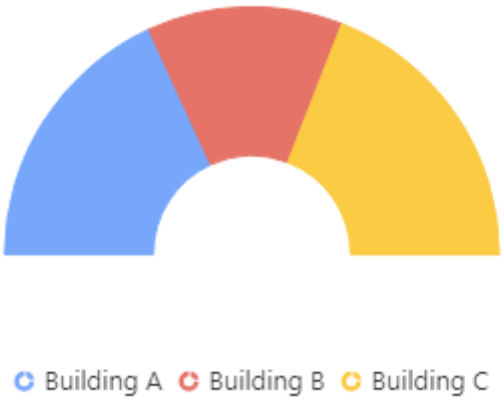


Or use the corresponding shortcut in the top toolbar :



Semi

Pie and Doughnut types also can be displayed as Half-Pie or Half-Doughnut :



For this, check *Semi* under *Basic Settings* :

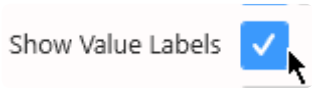
Last modified: May 15, 2019

2.8.11.3. Data Point information

Value Labels

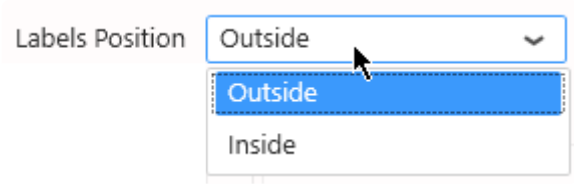
Text labels can be used to display the value next to each portion.

Under *Basic Settings*, check *Show Value Labels* :



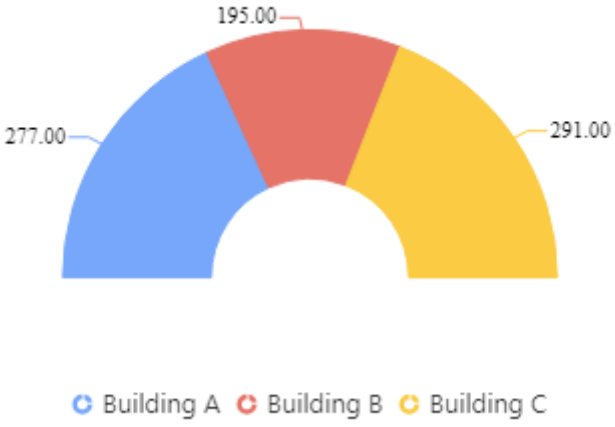
Position

The position of labels can be configured with the *Labels position* selector :



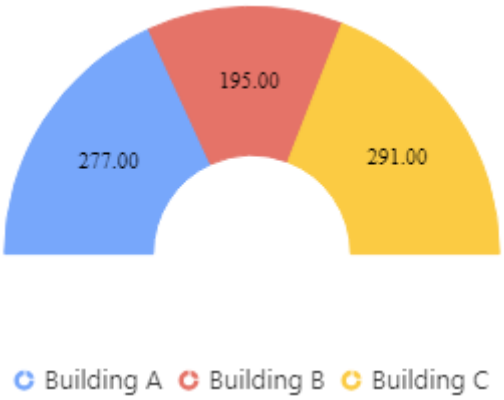
Outside (default)

By default, labels are displayed outside of the portions :



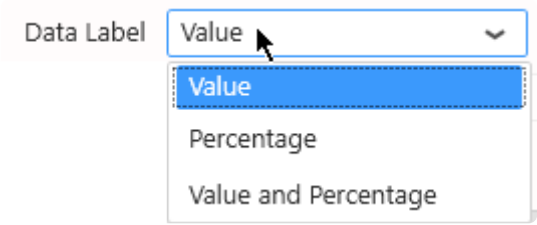
Inside

Labels can also be displayed inside the portion :



Type

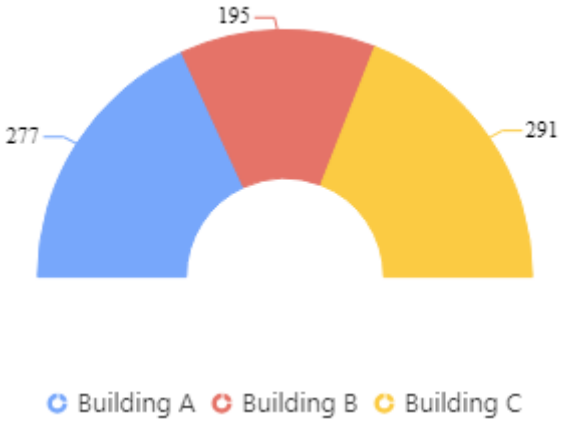
The type of labels can be configured with the *Data Label* selector :



- Value
- Percentage
- Value and Percentage

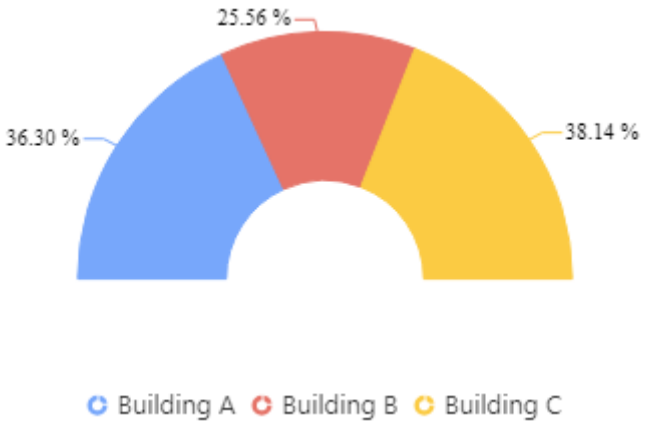
Value

Value labels show the aggregated Value field for the portion :



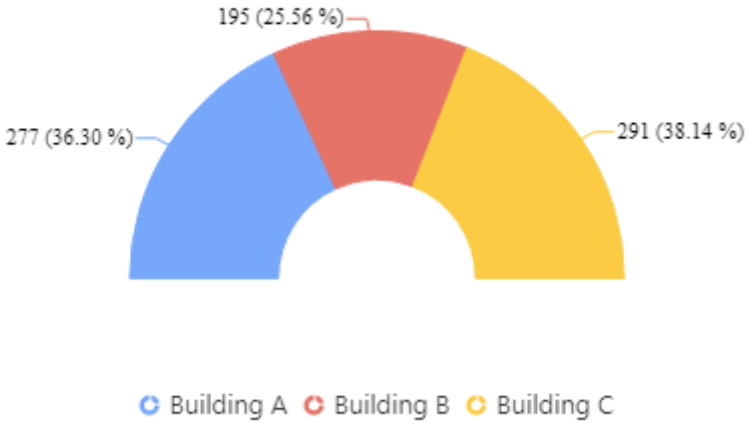
Percentage

Percentage labels show what percentage of the total the portion represents :



Value and Percentage

Value and Percentage can be combined in a single label :



Last modified: May 15, 2019

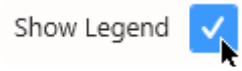
2.8.11.4. Legend

Portion Legend

The Legend shows the meaning of each portion :

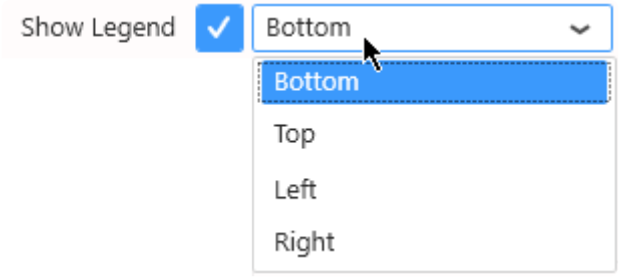
Hide/Show Legend

The chart Legend can be be hidden or displayed using the corresponding checkbox :



Legend position

The chart Legend can be positioned on any side of the widget :

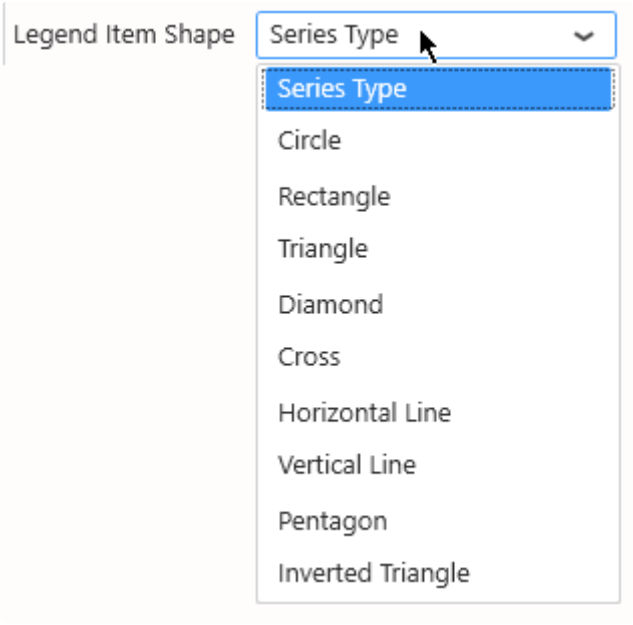


Custom Legend Item shape

For each item in the Legend, its name and icon is displayed, with the Series color displayed on the icon. By default, the icon shape depends on the Chart type.

You can select a different shape that will apply to all Series in the Chart.

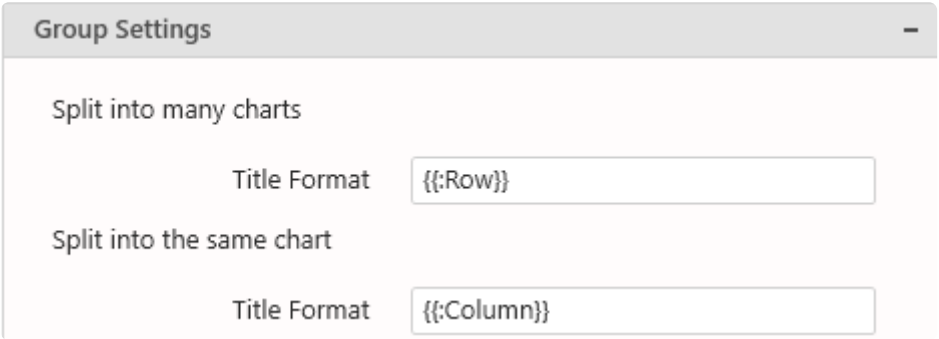
Select the desired shape from the *Legend Item Shape* list :



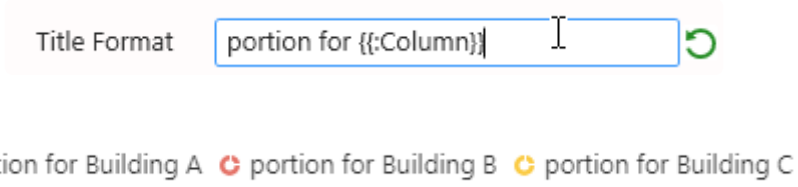
Note : *Series Type* means that the shape will depend on the Chart type (this is the default).

Customize Legend of portions

Legend items are automatically generated from the field names and values, but you can customize how they are generated under *Group Settings* :



With *Split into the same chart*, you can use the placeholder `{{:Column}}` to be replaced by the field value :

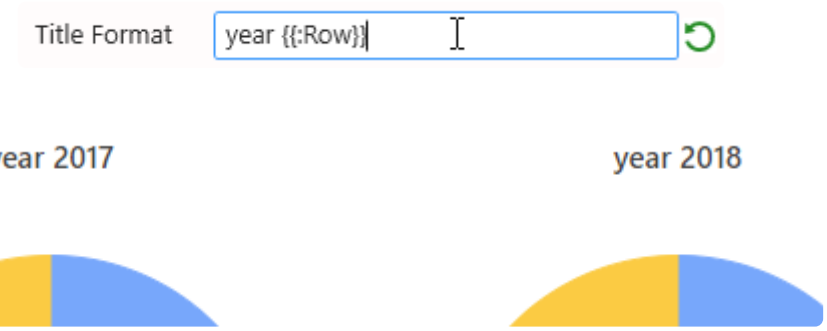


Customize Titles of contained charts

Group Settings also allows to configure the series titles when the Proportion Chart has a field bound to *Split into many charts*.
Depending on your data configuration, not all configurations may be available because they wouldn't

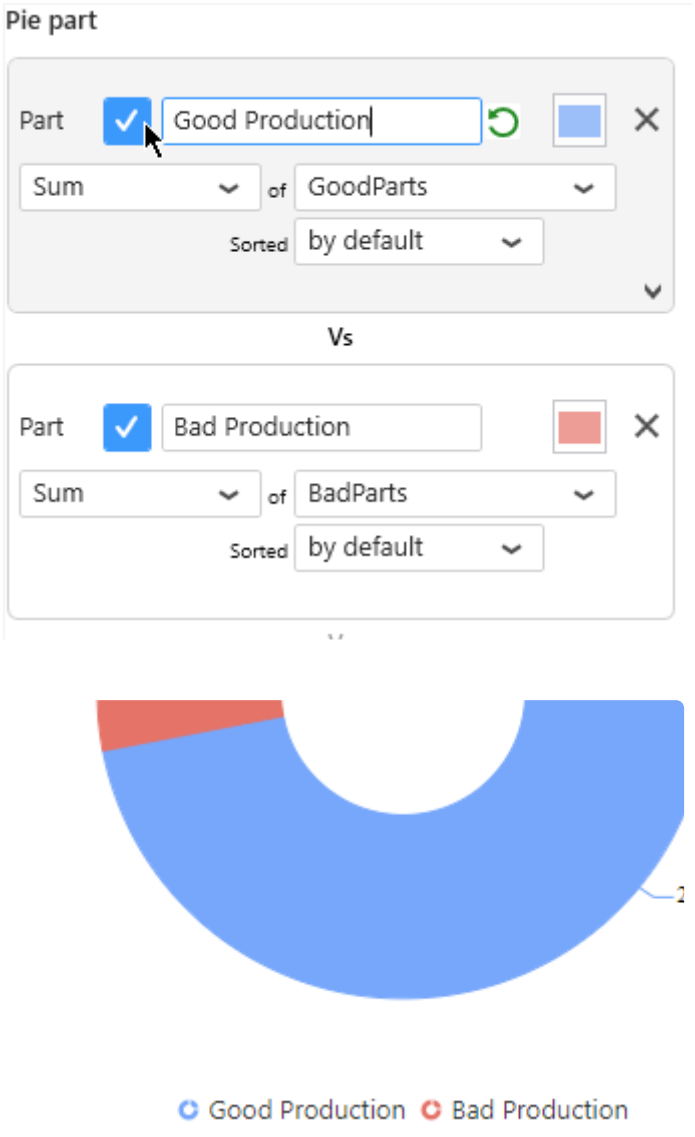
have meaning.

With *Split into many charts*, you can use the placeholder `{{:Row}}` to be replaced by the field value :



Customize the name of *Part* fields

The legend and titles generated by the *Part* fields can be renamed by checking the text box next to the data field configuration :



Last modified: Oct 30, 2019

2.8.11.5. Managing Series Color

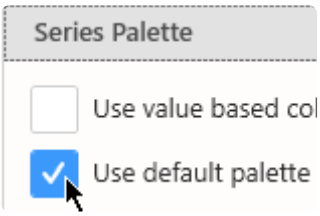
Indexed Colors (default)

By default, Proportion Chart portions are assigned a color based on their index in the ordered list in which they appear (usually alphabetical order) :
Item1 in the values of the series (see Legend order) is assigned Color1, Item2 is assigned Color2, etc…

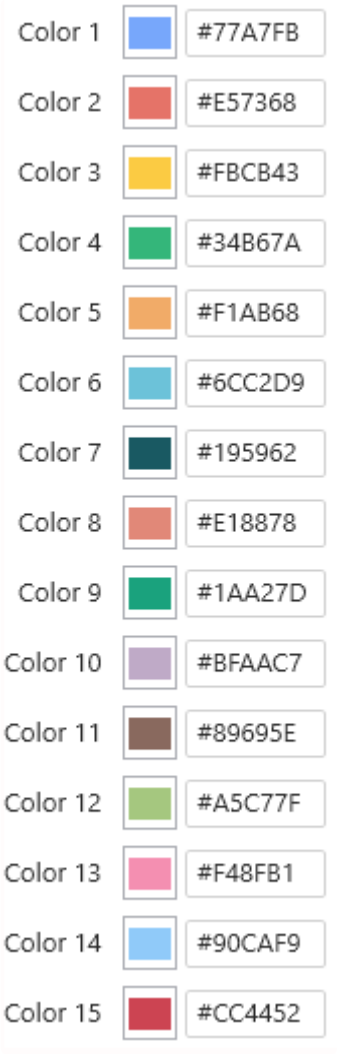
The palette has 15 colors. When there are more series items than 15, the colors assigned loop back to Color 1.

Default Palette (default)

To enable the Default Palette (it is enabled by default), check *Use Default Palette* under *Series Palette* :

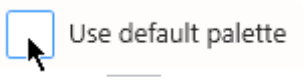


The list of colors in the default palette is below :



Custom Palette

To start customizing the Default Palette, un-check *Use Default Palette* under *Series Palette* :



Then you can customize colors either by entering their hexadecimal code (#RRGGBB) :

… or by clicking the color and using the graphical color picker :

Value-based Colors

What are Value-based colors ?

But what if your series changes over time ?

When items are renamed, added, or deleted from the data, the index of existing items will change, and their color will change !
This could happen simply during filtering at runtime for example.

Instead of index-based, you may then choose **Value-based** colors : the same value in the Series field will always have the same color in the Chart.

For this, simply check *Use value-based colors* under *Series Palette* :

Customizing Value-based colors

Value-based colors are defined **at the Data Source level**.

This is in order to allow re-use of the colors everywhere the same data is used.

To configure value-based colors, see the corresponding chapter Value-based Settings, under

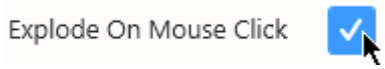
Transforming Data.

Last modified: May 15, 2019

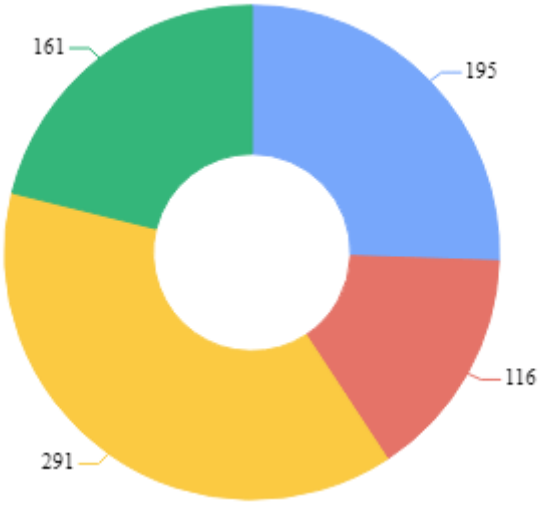
2.8.11.6. Interaction

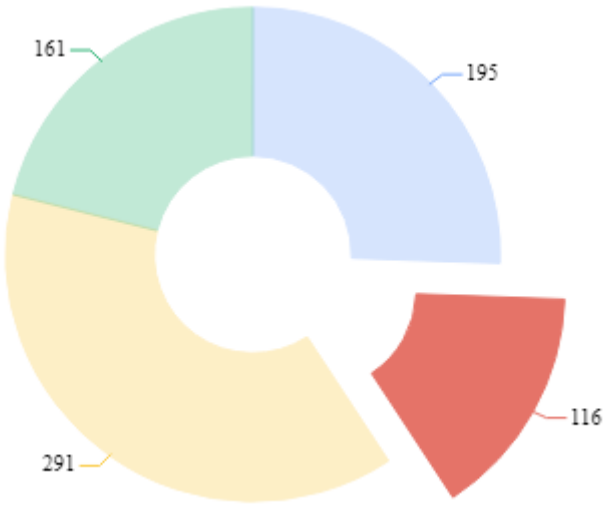
Explode On Mouse Click

Check *Explode On Mouse Click* under *Series* :



At runtime, the user will be able to click on a portion to highlight it by slightly removing it from the chart :



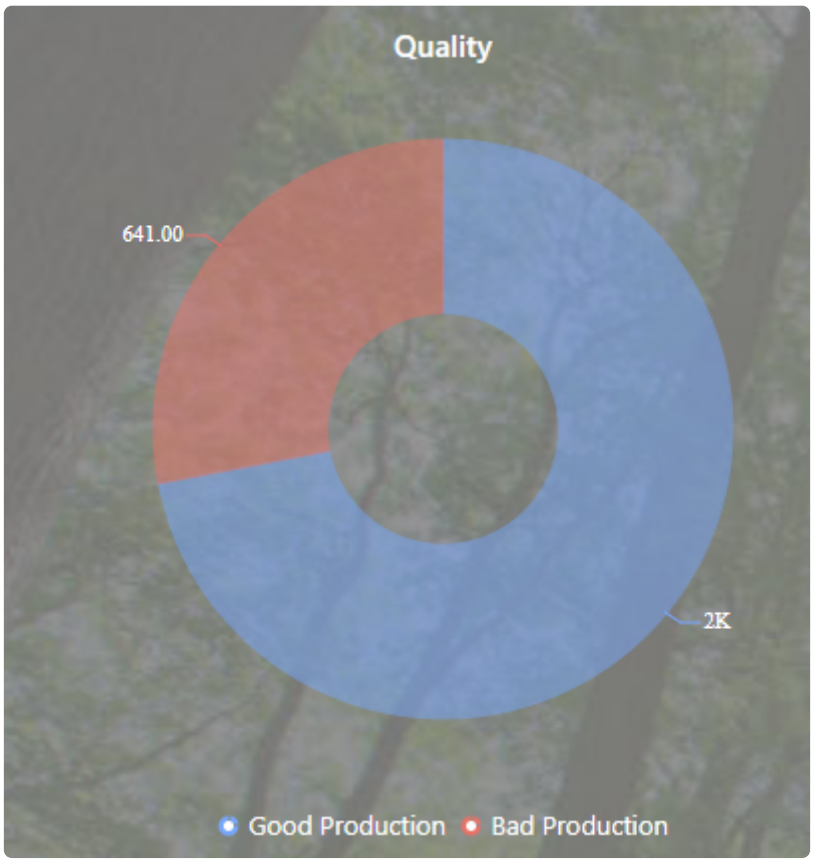


Last modified: May 15, 2019

2.8.11.7. Style and Appearance

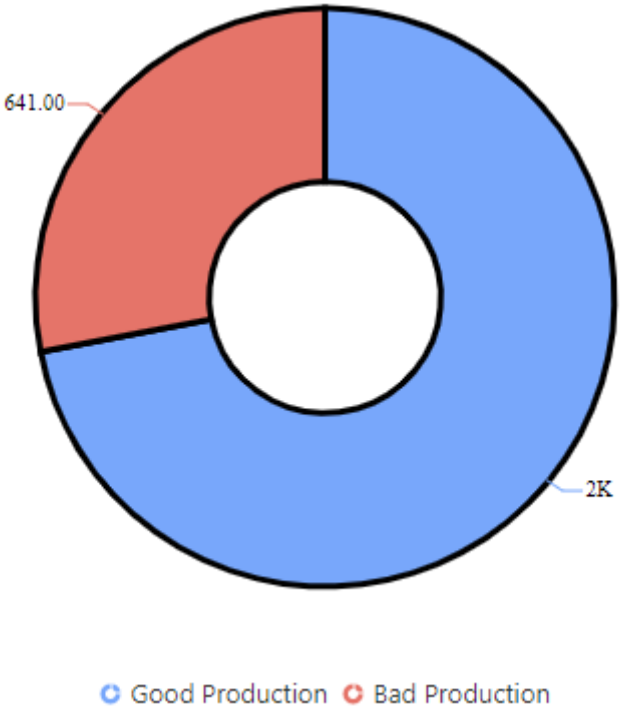
Opacity

The chart can be made semi-transparent to better integrate with a background.
Under *Series*, configure the *Opacity* :



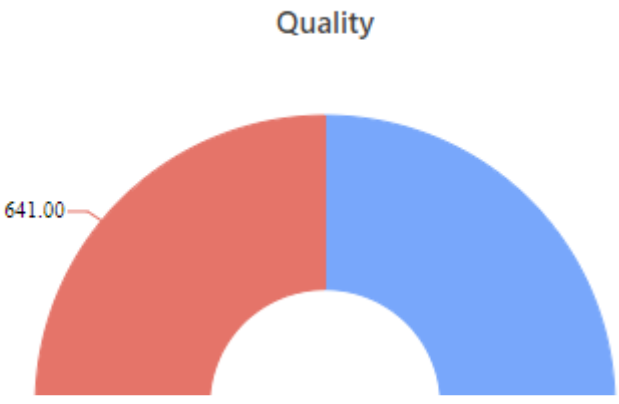
Border

Under *Series*, a *Border Thickness* and *Border Color* can be configured :



Custom Title

When the chart is not configured to show a series of several charts, a *Custom Title* can be configured under *Basic Settings* :



Last modified: May 15, 2019

2.8.12. Gantt

The Gantt Widget allows to display scheduling data : work orders time line, equipment states history, teams schedules, …

(new in Alpana 2020 R1)

Last modified: Apr 24, 2020

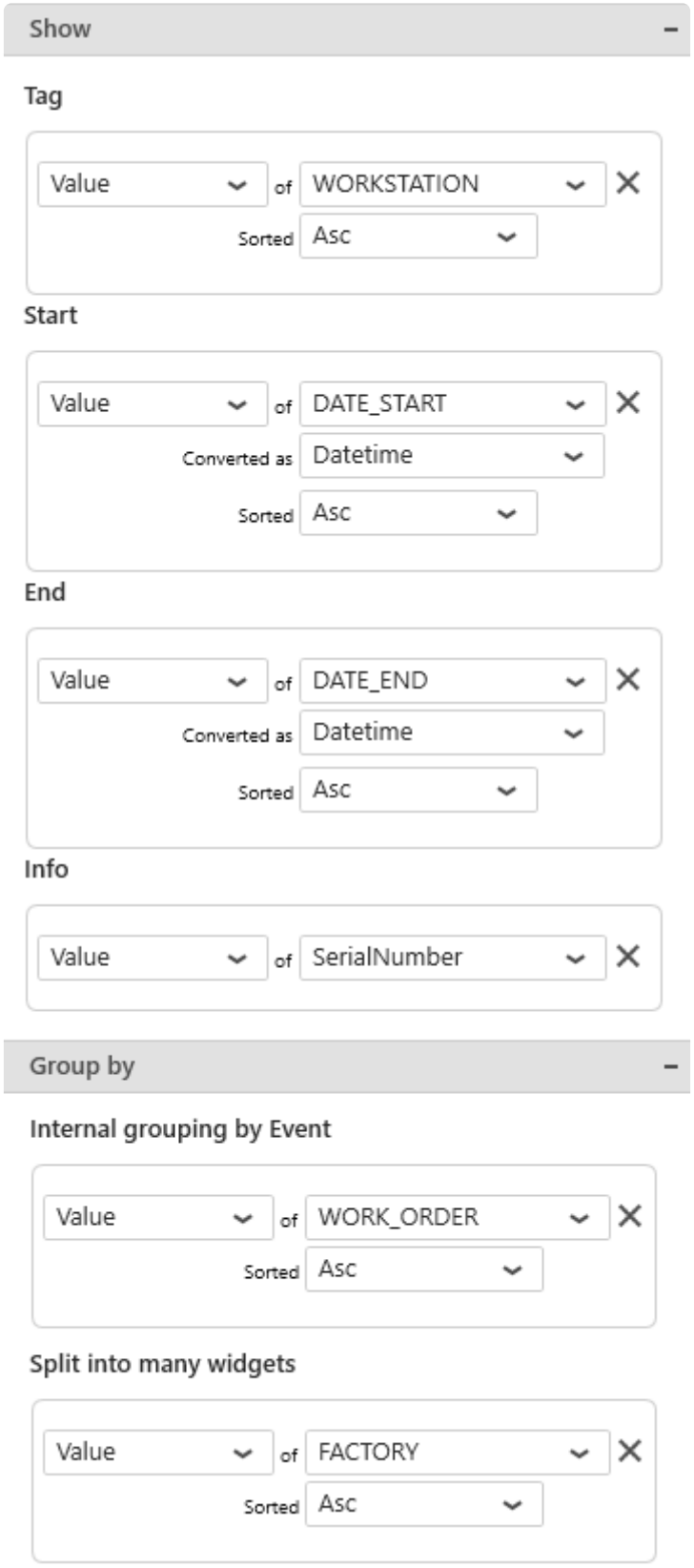
2.8.12.1. Binding Data

Introduction

The Gantt Widget represents the time span of a series of events (work orders, shifts, machine states, …) happening in parallel on a series of “Tags” (equipments, production lines, …).
Time is represented on a X axis, and the events’ time span is represented as shapes spanning the desired duration.
Tags are separated by different labels on the Y axis, each one corresponding to a row on the visualization.

The following fields are available for configuring data binding on a Gantt :

- *Tag* : represent separation between events happening in parallel. *Tags* are represented side-by-side in different rows.
- *Start* : the start date of the events
- *End* : the end date of the events
- *Info* : additional information to display in the tooltip
- *Event* : allows to group several events with the same color
- *Split into many widgets* : allows to create many sub-widgets dynamically

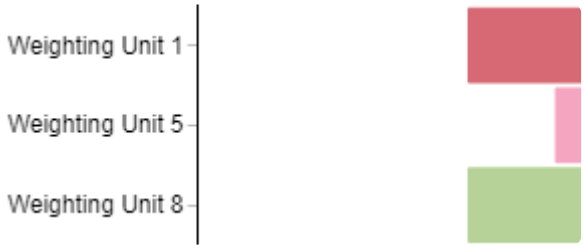


Tag

This is basically a series used for separating events happening in parallel, and displaying them side-by-side.

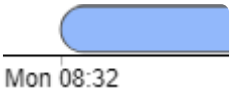
This field **cannot be aggregated** : it must have several separate values in order to create several

separate series.
This field is required.



Start

This must be a date/time value representing the start of the events.

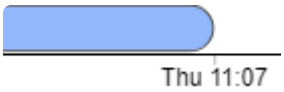


If the value is `NULL`, then it means we don't know when the event started, and the event starts at the beginning of the row :

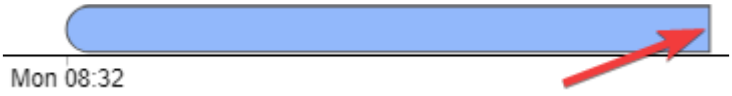
This field **cannot be aggregated** : it must have several separate values in order to create several separate events.
This field is required.

End

This must be a date/time value representing the end of the events.



If the value is `NULL`, then it means we don't when the event will end, and the event ends at the end of the row :



This field **cannot be aggregated** : it must have several separate values in order to create several separate events.
This field is required.

Info

This allows to add additional text information inside the tooltip.

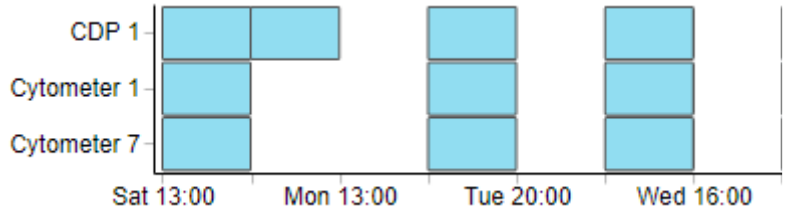
This field is optional.

Group By : *Event*

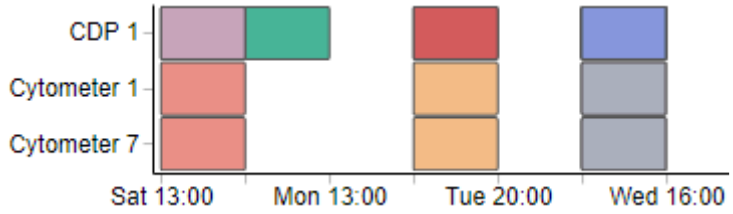
This allows to group several events in the same row or across rows, by allocating a color and displaying it in a legend.

This field **cannot be aggregated** : it must have several separate values in order to create several separate groups.
This field is optional.

Without grouping by Event :



With Events grouped :



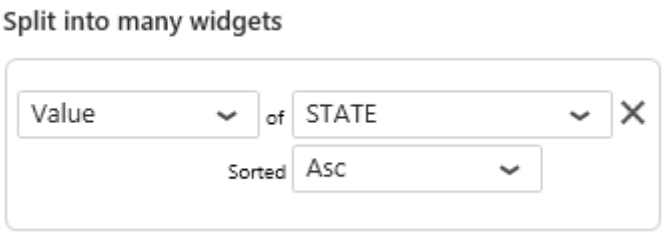
Group By : *Split into many widgets*

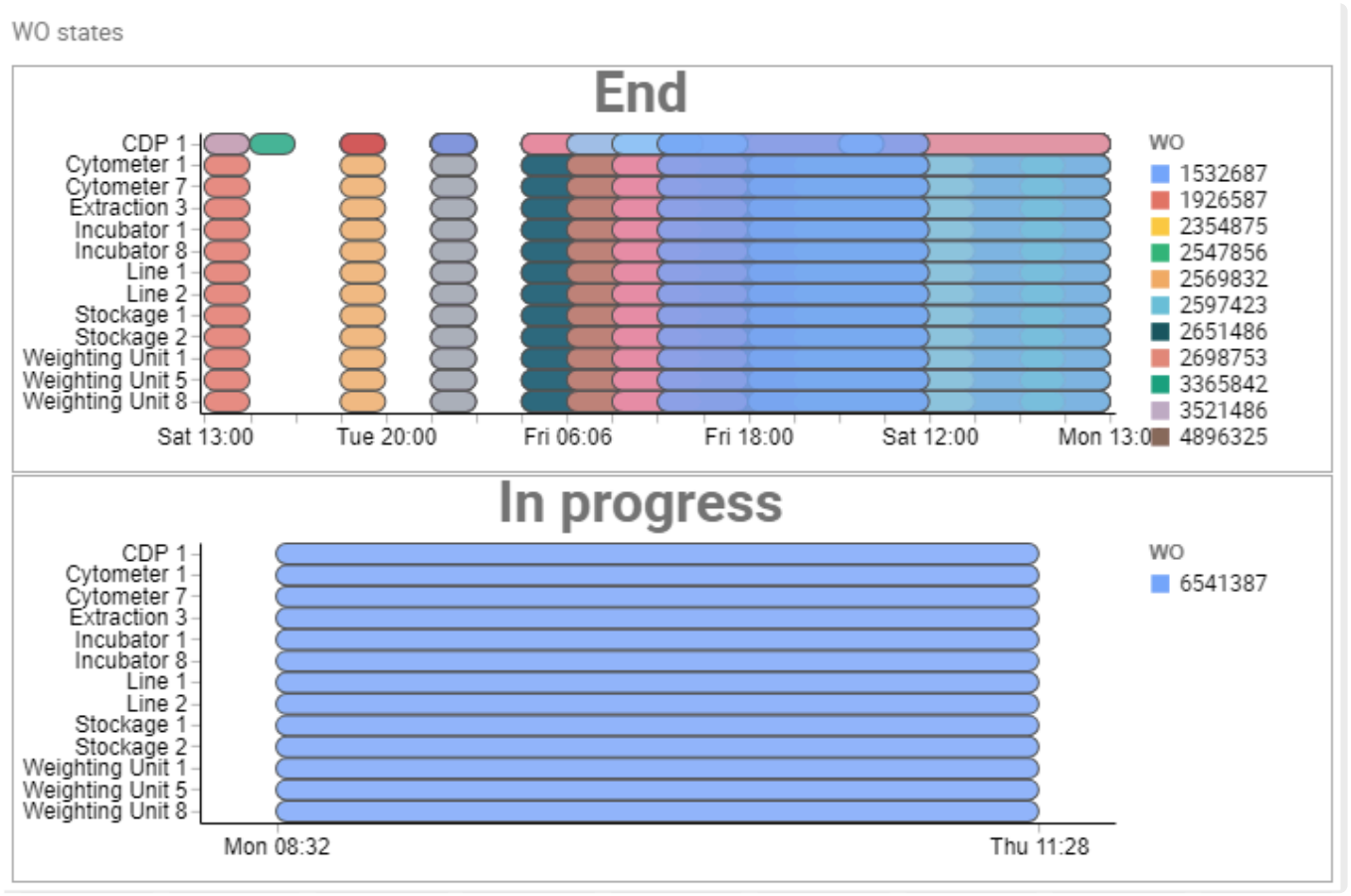
This represents the series field used to generate a different contained Gantt for each different value in this field.
This allows to create many sub-widgets dynamically.

This will turn the widget into a container, and generate a complete different chart for each value in the series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

This field **cannot be aggregated** : it must have several separate values in order to create several separate series.

This field is optional.





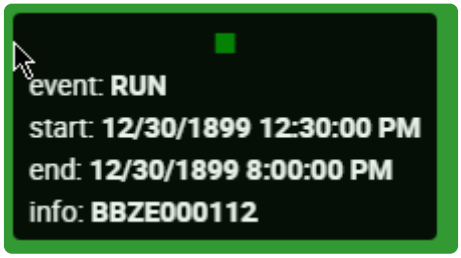
Last modified: Aug 26, 2020

2.8.12.2. Data Point Information

Tooltip

The tooltip displays the detailed information about the event :

- Event name
- Start Date
- End Date
- Info

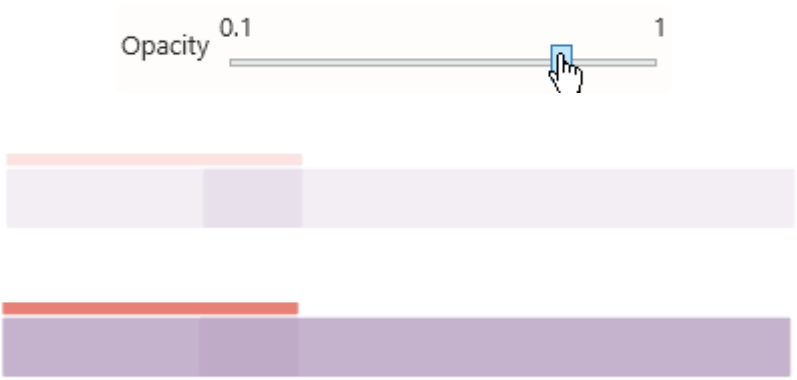


Last modified: Aug 26, 2020

2.8.12.3. Events Configuration

Opacity

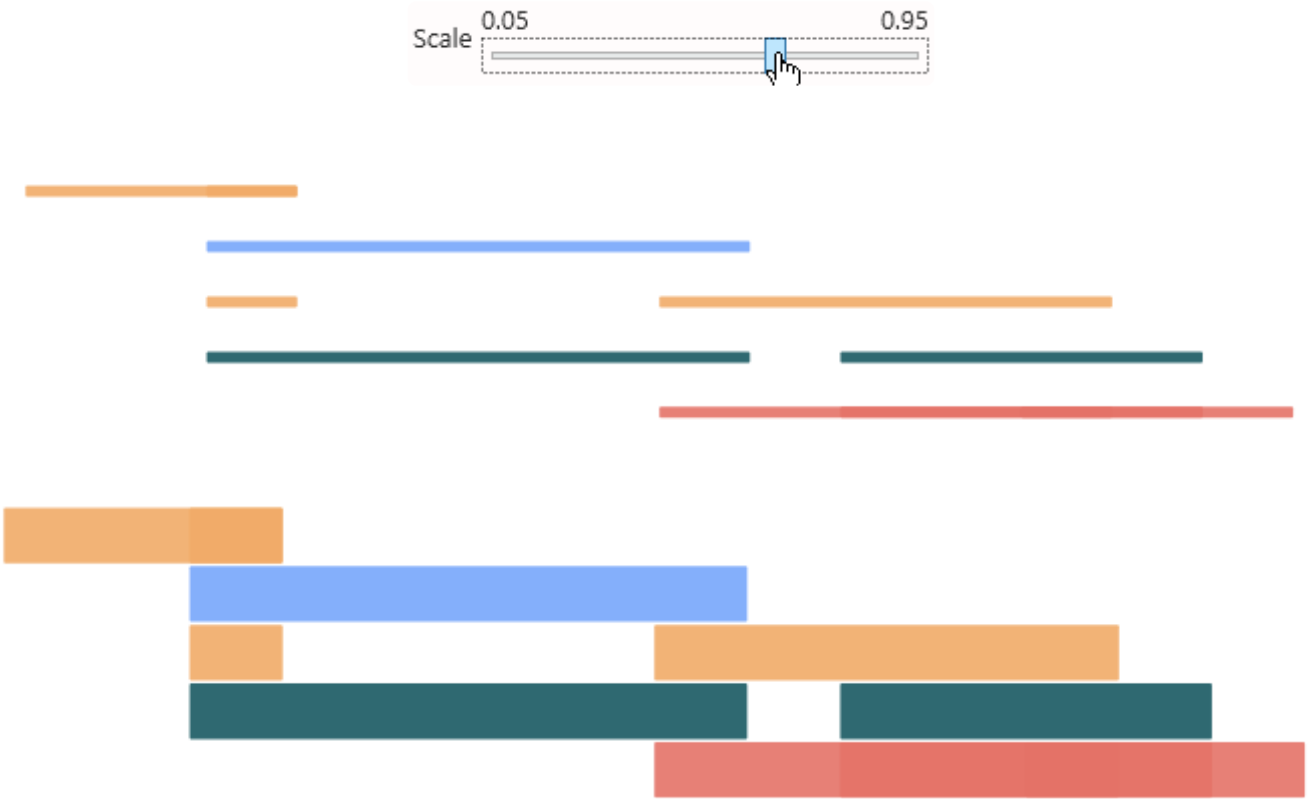
Configure the opacity of each event.
Lower/intermediate values (more transparent) allow to more easily visualize overlapping events.



Scale

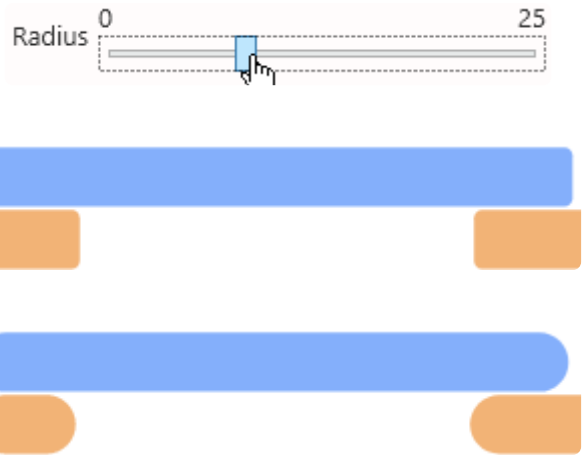
Configure the relative vertical size of all the events :

- `1` means to take the full height of the row
- `0.95` means almost the full height
- `0.05` means a very thin display



Radius

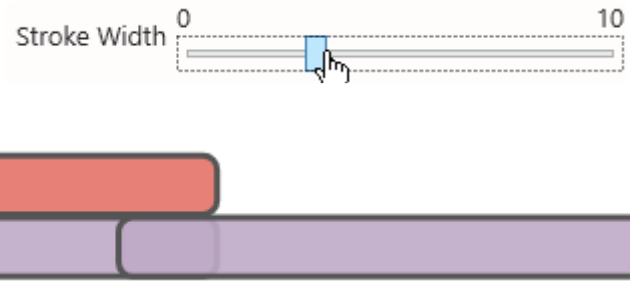
The radius of the rounded corners of all the events.
A *Radius* of `0` means square corners.
More round corners can be useful for distinguishing start/end dates of overlapping events.



> ✻ Note : the rounded corners are only displayed on the actual Start and End of events.
> If your event doesn't have an End Date (the value is `NULL`), then the corners are squared.
> This useful for making the visual difference between finished and un-finished events.

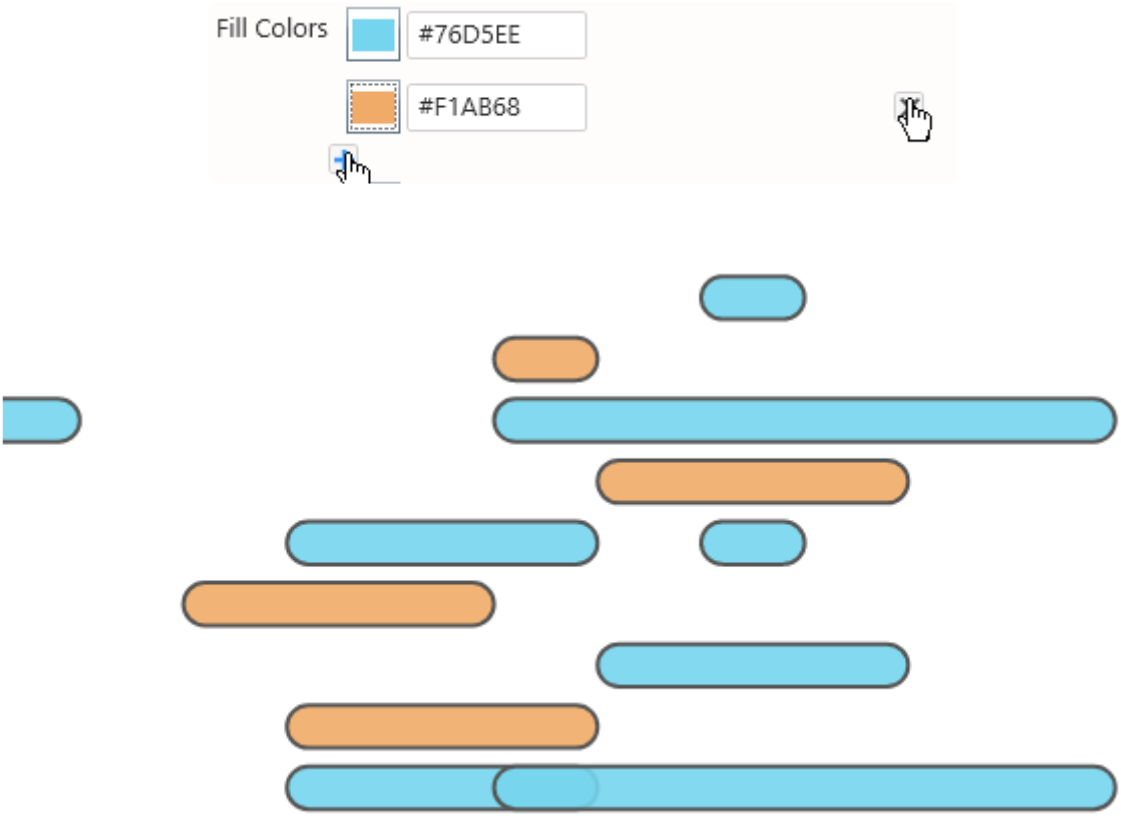
Stroke Width

The width (in pixels) of the border of each event.
A *Stroke Width* of `0` means no border.
Displaying the border can allow to more easily visualize overlapping events.



Fill colors

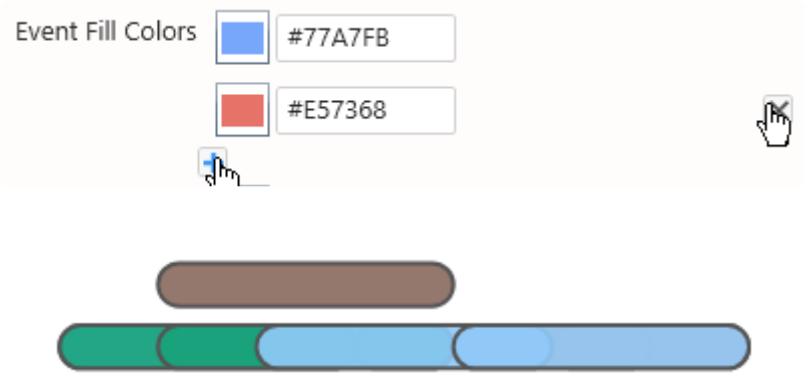
This is the color that all the events take, **when there is no *Events* field bound**.
If you create several colors, they will be cycled for each row.



If there is an *Events* field bound, this configuration is ignored, and the *Event Fill colors* is used instead.

Event Fill colors

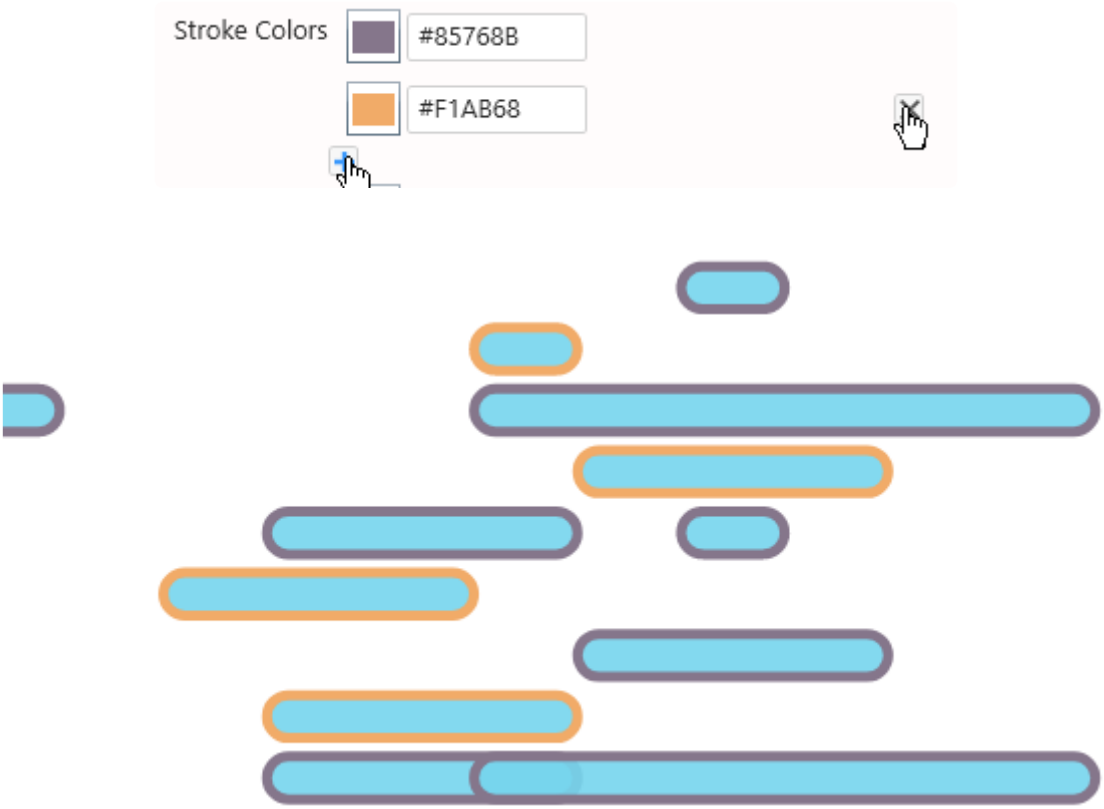
This is the color that each of the events takes, in order, **when there is an *Events* field bound**.
This is an indexed palette.
If there is more events than colors, the colors are cycled.
When several events have the same value of the *Events* field, they have the same color.



If there is no *Events* field bound, this configuration is ignored, and the *Fill colors* is used instead.

Stroke colors

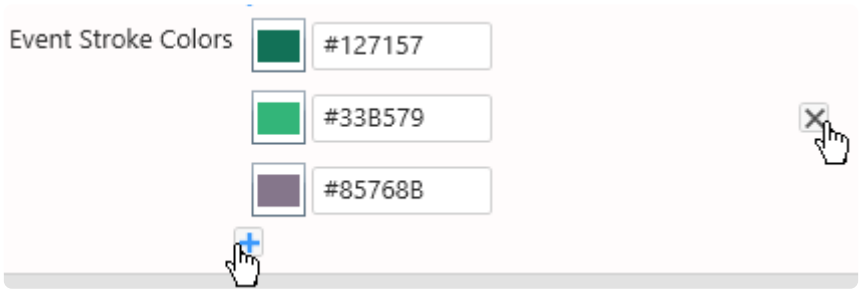
This is the color that the border of each of the events takes, **when there is no *Events* field bound**.
If you create several colors, they will be cycled for each row.



If there is an *Events* field bound, this configuration is ignored, and the *Event stroke colors* is used instead.

Event Stroke colors

This is the border color that all the events takes, in order, **when there is an *Events* field bound**.
This is an indexed palette.
If there is more events than colors, the colors are cycled.
When several events have the same value of the *Events* field, they have the same color.



If there is no *Events* field bound, this configuration is ignored, and the *Stroke colors* is used instead.

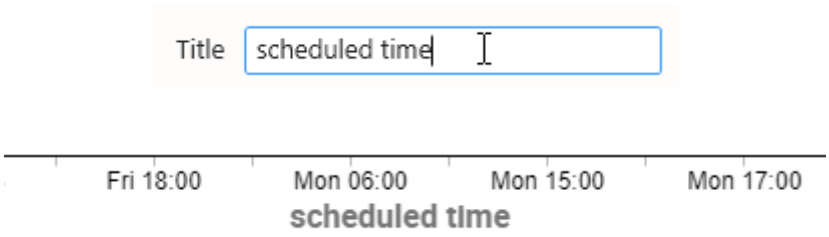
Last modified: Jun 19, 2020

2.8.12.4. Axis Configuration

X Axis configuration

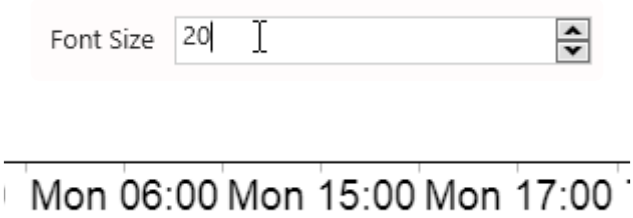
Title

This is the Axis title.
If no title is configured, then it is the same as hiding the title (it doesn't take visual space).



Font size

This is the font size of the Axis Labels.



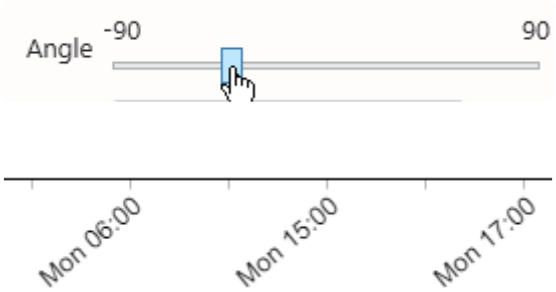
Format

This defines how dates should be formatted to display the X Axis labels :

- *Time* : `HH:MM`, example : `23:59`
- *Day (long) / Time* (default setting) : `ddd HH:MM`, example : `Thu 23:59`
- *Day / Time* : `dd HH:MM`, example : `31 23:59`
- *Month / Day / Time* : `mm-dd HH:MM`, example : `12-31 23:59`
- *Date / Time* : `yyyy-mm-dd HH:MM`, example : `2020-12-31 23:59`
- *Month / Day* : `mm-dd`, example : `12-31`
- *Day / Month (long)* : `dd mmm`, example : `31 Dec`
- *Month (long) / Year* : `mmm yyyy`, example : `Dec 2020`
- *Date* : `yyyy-mm-dd`, example : `2020-12-31`

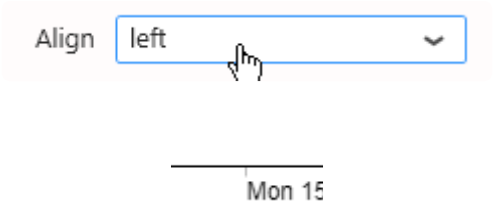
Angle

This is the orientation of the Axis Labels.
Positive values rotate clockwise, negative values rotate counter-clockwise.



Align

This is the text alignment of the Axis Labels compared to the tick mark : left/center/right.

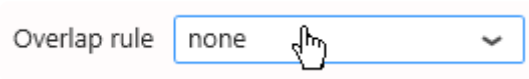


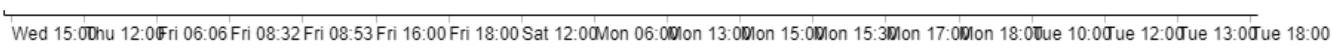
Overlap Rule

This represents how the Axis Labels are displayed when they overlap :
This defines the strategy to avoid overlap of the axis labels :

none

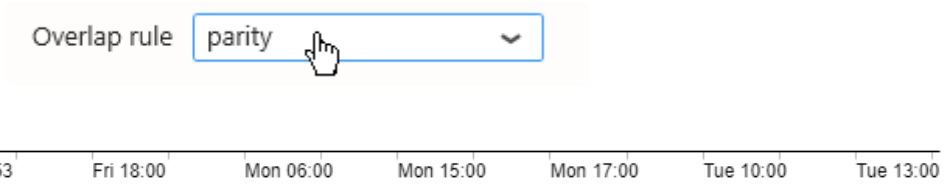
No strategy. If the labels overlap, they will be displayed as such.





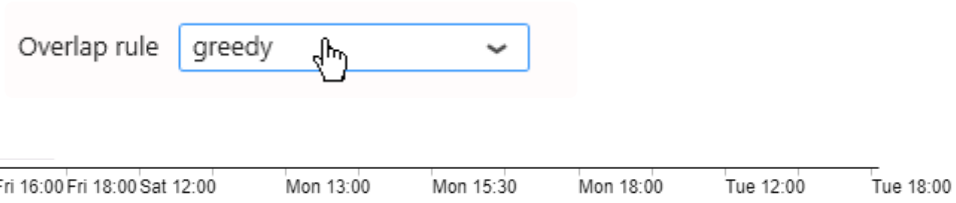
parity

Removing every other label. This is the recommended default.



greedy

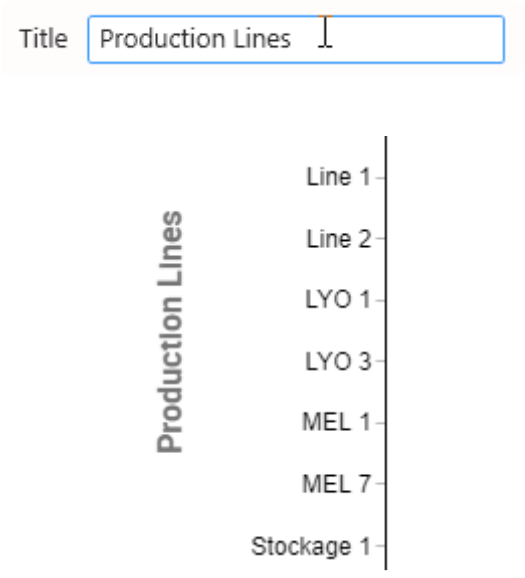
A linear scan of the labels is performed, removing any label that overlaps with the last visible label. This works better when the axis is not linear.



Y Axis configuration

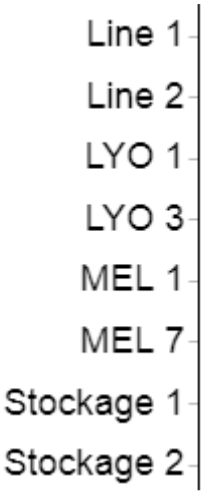
Title

This is the Axis title.
If no title is configured, then it is the same as hiding the title (it doesn't take visual space).



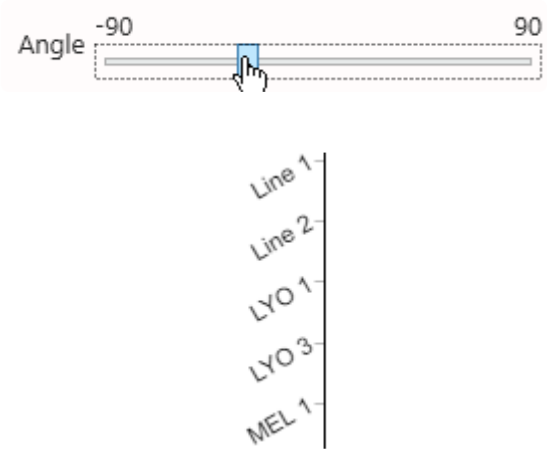
Font size

This is the font size of the Axis Labels.



Angle

This is the orientation of the Axis Labels.
Positive values rotate clockwise, negative values rotate counter-clockwise.



Align

This is the text alignment of the Axis Labels compared to the tick mark : left/center/right.

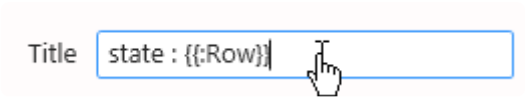
Last modified: May 05, 2020

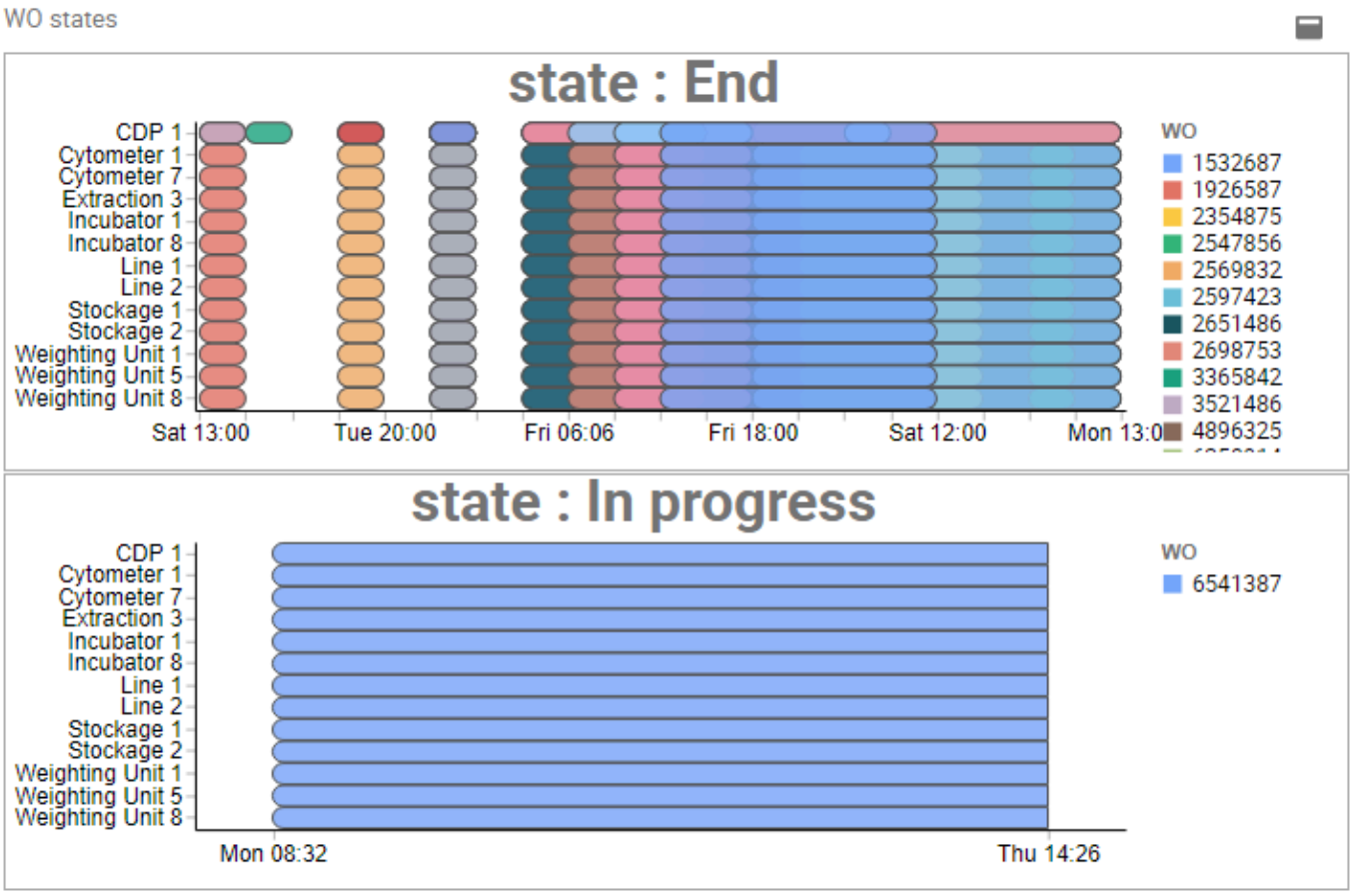
2.8.12.5. Titles Configuration

Optionally a title can be displayed on top of the Gantt.
This is useful especially when configuring a *Split into many widgets* field : then configuring the Title as `{{:Row}}` will give you the field value for each sub-widget.

Title

This is the title itself.
If no title is configured, then it is the same as hiding the title (it doesn't take visual space).





Title Font Size

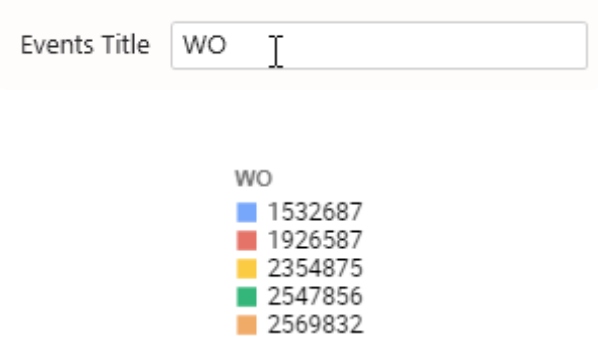
This is the font size of the Title.

Axis Title Font Size

This is the font size of both the Axis Titles.

Events Title

This is the title of the legend when an *Event* field is configured.



Events Font Size

This is the font size of the legend items when an *Event* field is configured.

Last modified: May 05, 2020

2.8.13. Sankey

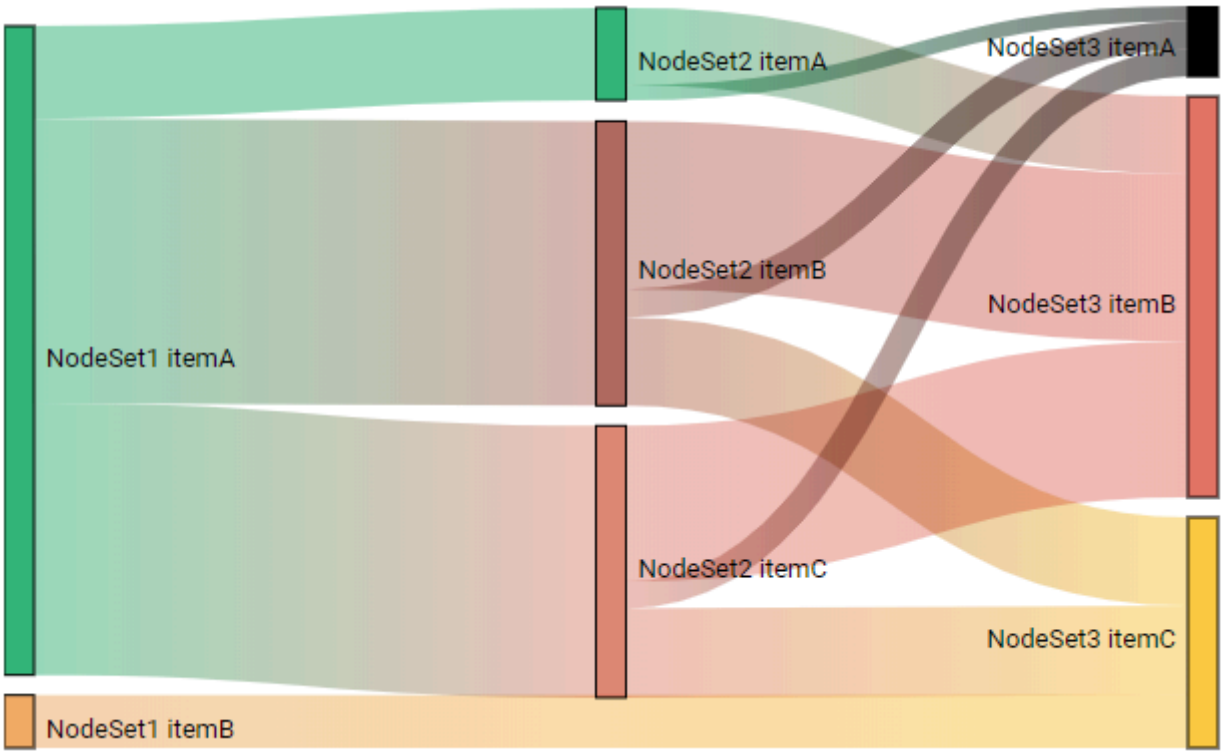
The Sankey widget allows to see and quantify topological connections between items.
For example, you can analyze your energy flows, identify losses, find water leaks, etc…

Last modified: Oct 17, 2019

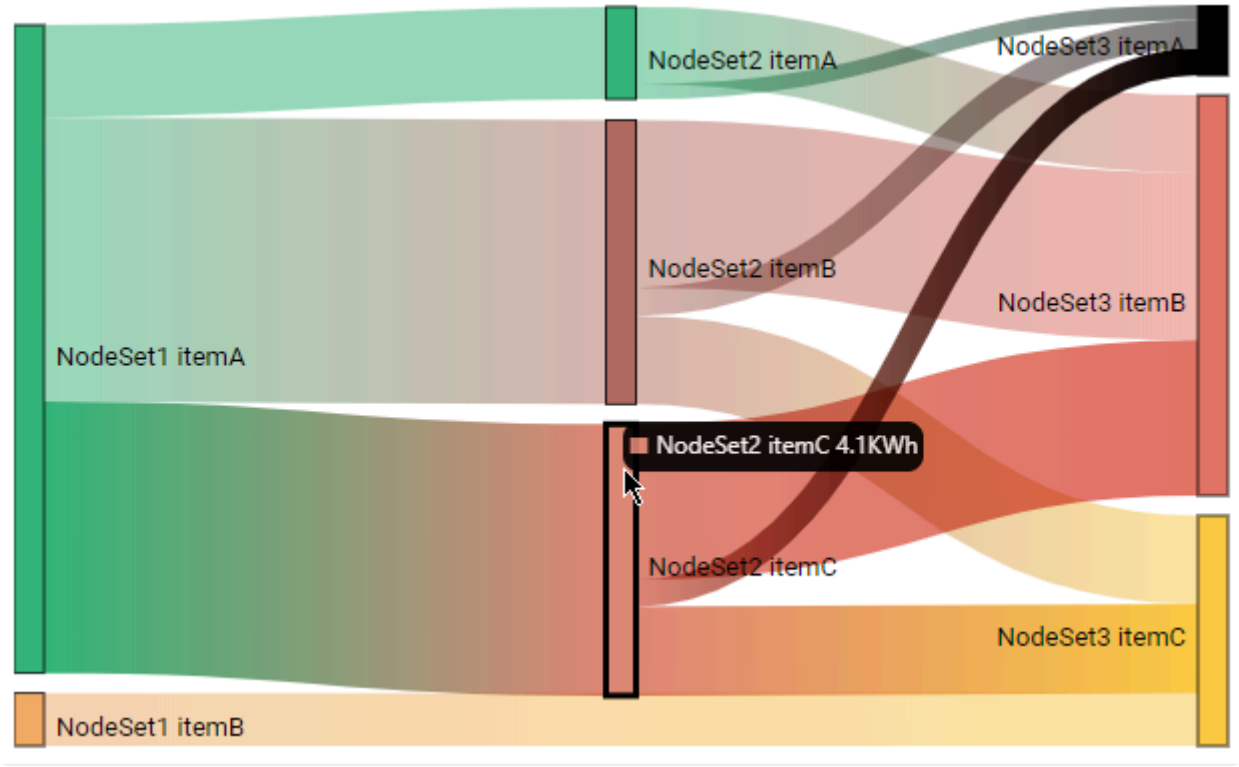
2.8.13.1. Binding Data

Introduction

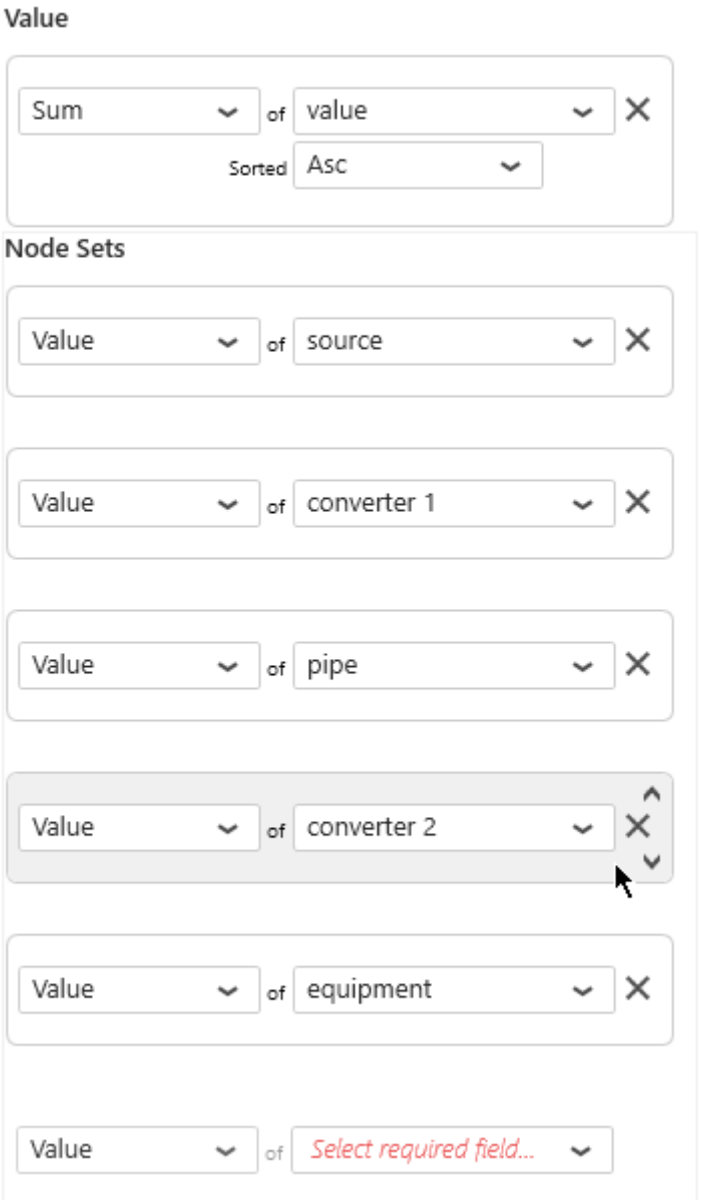
The Sankey diagram is composed of Nodes representing the items that will be interconnected.
Nodes are grouped in Sets of columns. this represents items that are at the same hierarchy level and which never interconnect.
Each Node from each Node Set can connect to any Node from any **other** Node Set, using a visual Link.



At runtime, clicking a Node allows to highlight all paths that go through this Node :



The following fields are available for configuring data binding on a Sankey :



- *Value* : The visual size of the Nodes and their Links is determined by the *Value* field.
- *Node Set 1, 2, 3, …* : the name of Nodes inside Node Set 1 is all the unique values in the *Node Set 1* field.

Value

This represents the *quantity* we want to display on the Sankey diagram, and will bind to the size of the Nodes and their Links.
This field **must be aggregated** : you must choose an aggregation function (SUM, AVG, …)

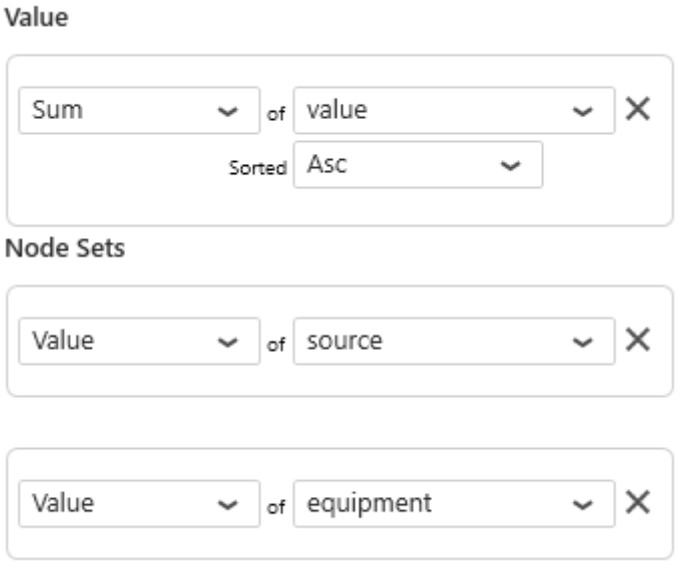
Node Set 1

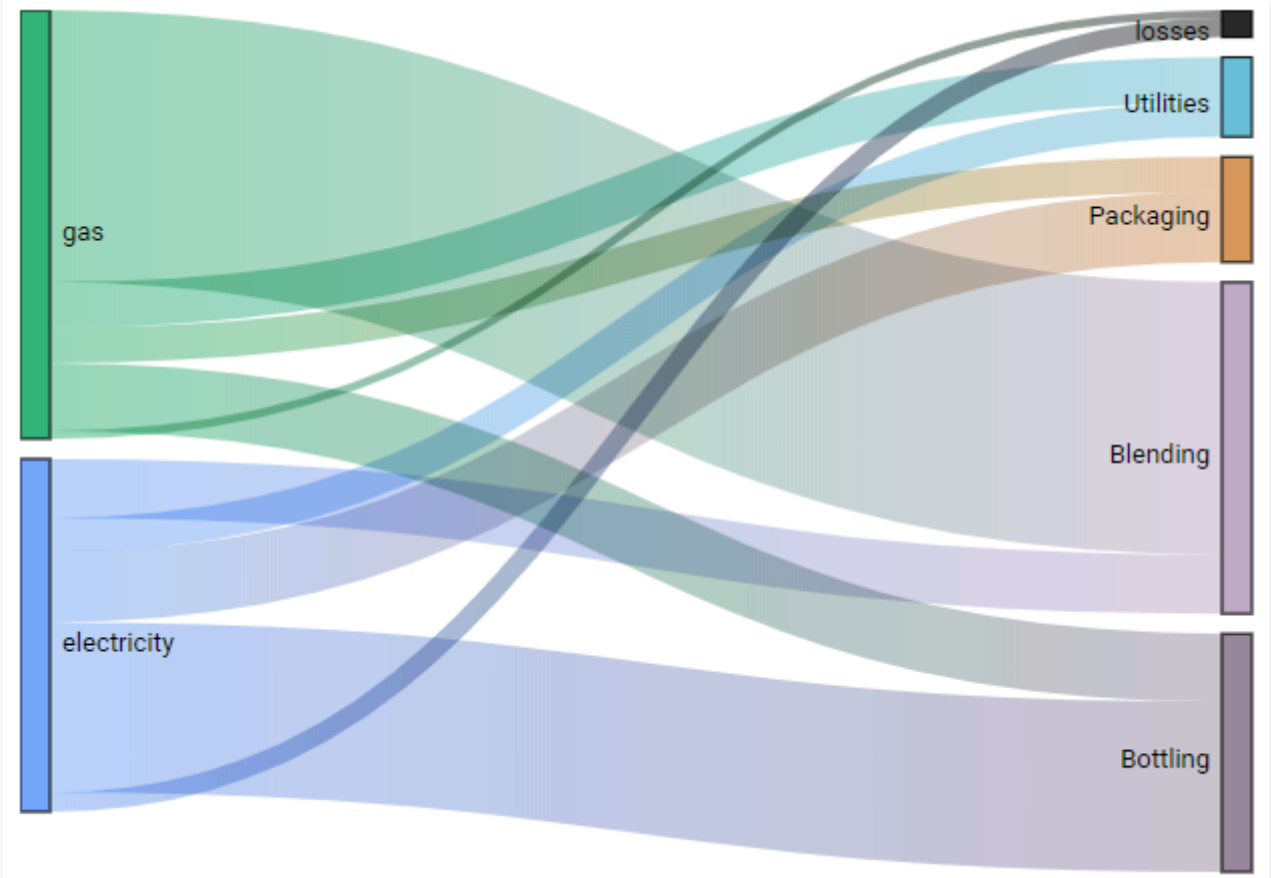
This represents the *category* we want to use to dynamically separate Nodes inside the 1st set of nodes (represented in the first column on the left).

These fields **cannot be aggregated** : they must have several separate values in order to group the

“Value” aggregation.

If the field contains `NULL` values, then the corresponding Nodes are not represented. That means there is no connections with other Node Sets on this data row, and visually the Links just go through the Node Set. (see “NodeSet1 itemB” in screenshots above)





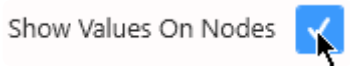
Last modified: Oct 30, 2019

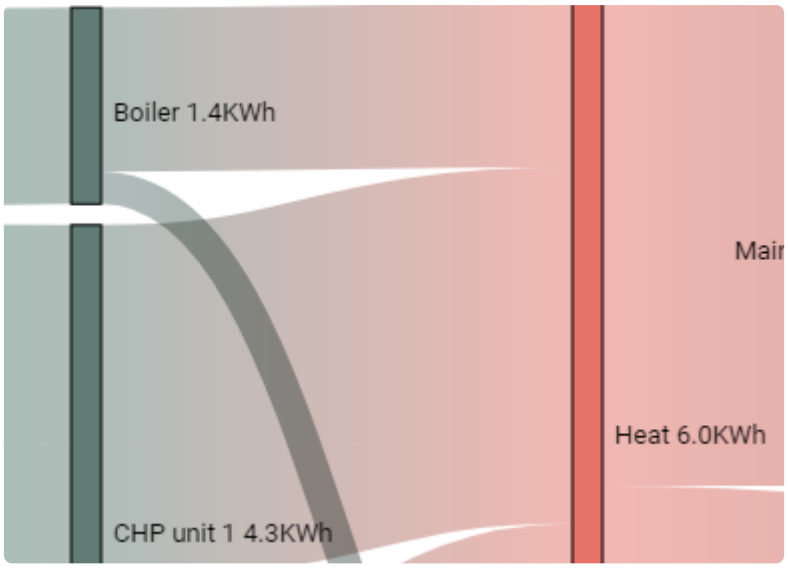
2.8.13.2. Data Point information

Value Labels

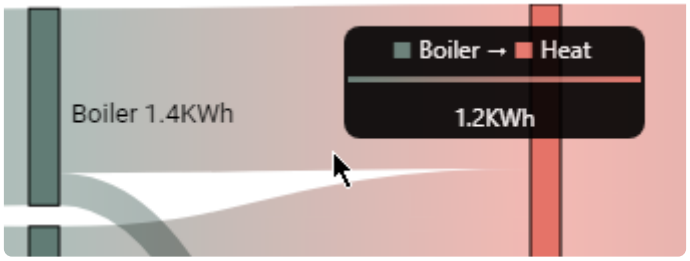
Text labels can be used to display the value next to each Node.

Under *Basic Settings*, check *Show Values On Nodes* :





Regardless of this setting, hovering a Node or a Link shows the corresponding value in the tooltip :



Last modified: Oct 17, 2019

2.8.14. Energy Rating

The Energy Rating widget allows to rate energy consumption and environment footprint on a grading scale.
This visualization is regulatory in European Union, and known in English as EPC.

(new in Alpana 2020 R1)

Last modified: Apr 24, 2020

2.8.14.1. Binding Data

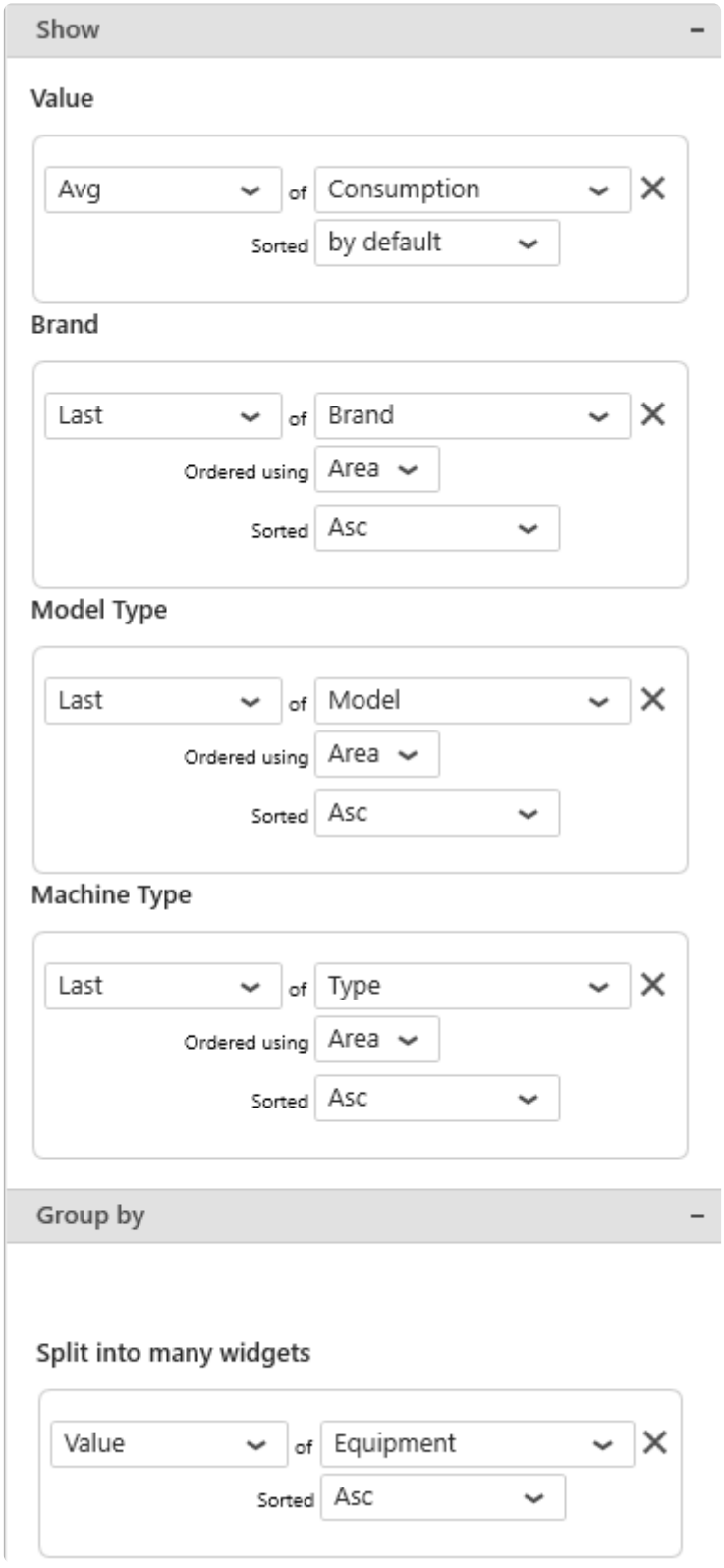
Introduction

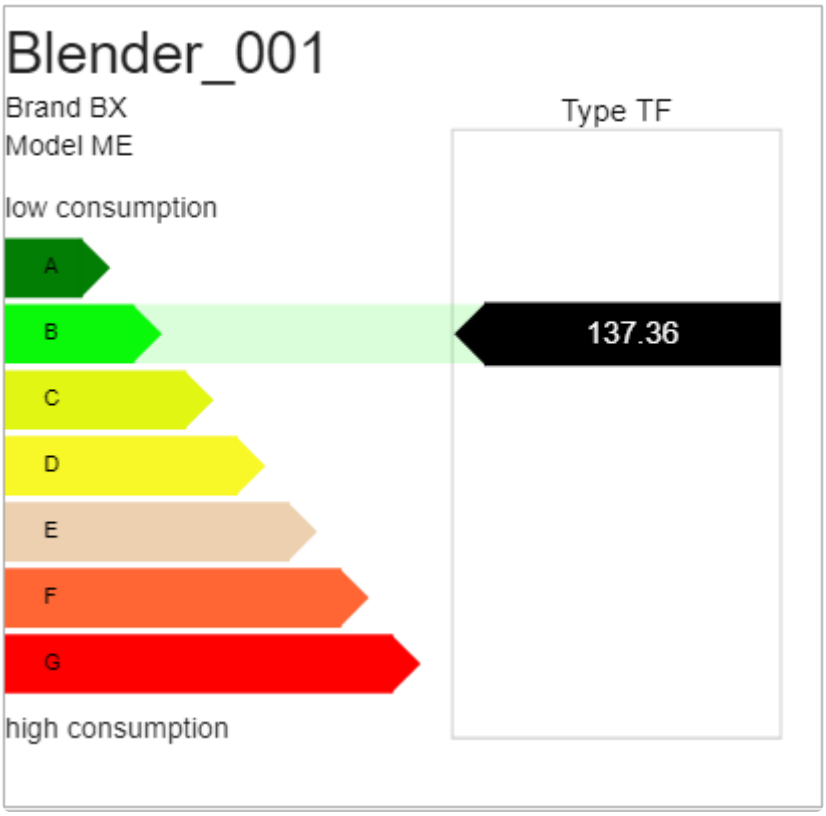
The Energy Rating Widget represents the energy rating of equipments.
The rating is calculated from an analog/real value of energy consumption :

- primary energy consumed, in kWh, or in a normalized unit such as kWh / m²
- mass of CO_2 equivalent, in kg(CO_2eq), or in a normalized unit such as kg(CO_2eq) / m²
  This analog value is converted to a discrete rating as a letter/color combination : A, B, C, D, etc…

The following fields are available for configuring data binding on a Energy Rating Widget :

- *Value* : The analog measurement to be displayed as a grade.
- *Brand*, *Model Type*, *Machine Type* : text properties of the machine
- *Split into many widgets* : allows to create many sub-widgets dynamically, to compare equipments for example


Show
Value
Avg of Consumption
Sorted by default
Brand
Last of Brand
Ordered using Area
Sorted Asc
Model Type
Last of Model
Ordered using Area
Sorted Asc
Machine Type
Last of Type
Ordered using Area
Sorted Asc
Group by
Split into many widgets
Value of Equipment
Sorted Asc



Value

This is the analog input value that will be turned into a grade.

This field **must be aggregated** : it must have a single value in order to be turned into a single grade.
This field is required.

Brand, Model Type, Machine Type

Brand

This is the machine brand name, to comply with regulatory display.

Model Type

This is the machine model type, to comply with regulatory display.

Machine Type

These fields are additional text properties to display in the label.
They can be used to add regulatory information, or other metadata to help explain the consumption behavior of the machine.

Group By : *Split into many widgets*

This represents the series field used to generate a different contained Energy Rating for each different value in this field.
This allows to create many sub-widgets dynamically, for example one for each Equipment, or one per month.

This will turn the widget into a container, and generate a complete different chart for each value in the series field, contained in the Widget.
The layout of the charts inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

This field **cannot be aggregated** : it must have several separate values in order to create several separate series.
This field is optional.

Last modified: Apr 24, 2020

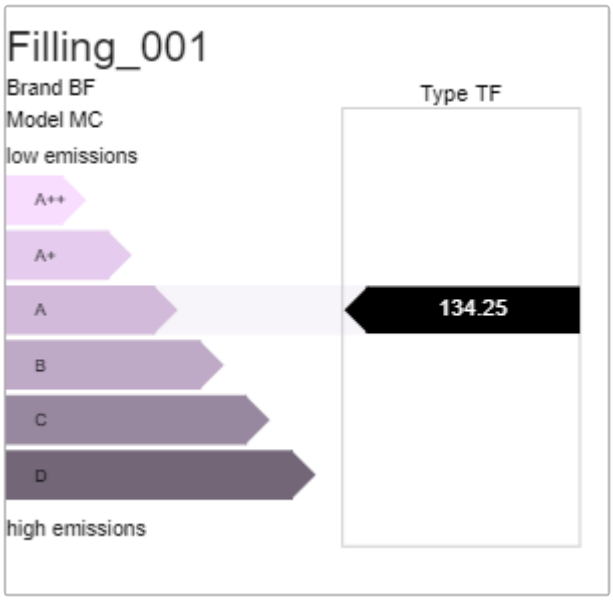
2.8.14.2. Grading Calculation and Display

Grading

In order to turn the analog *Value* into a grade, Alpana will use a list of *Thresholds* :

1. Is the *Value* less than the 1st threshold ? Then apply the 1st grade.
2. Is the *Value* less than the 2nd threshold ? Then apply the 2nd grade.
3. …
4. Else, apply the last threshold.

You can configure those thresholds values and number, and the corresponding labels and colors.
You should configure them at the same time, to ensure a consistent configuration.

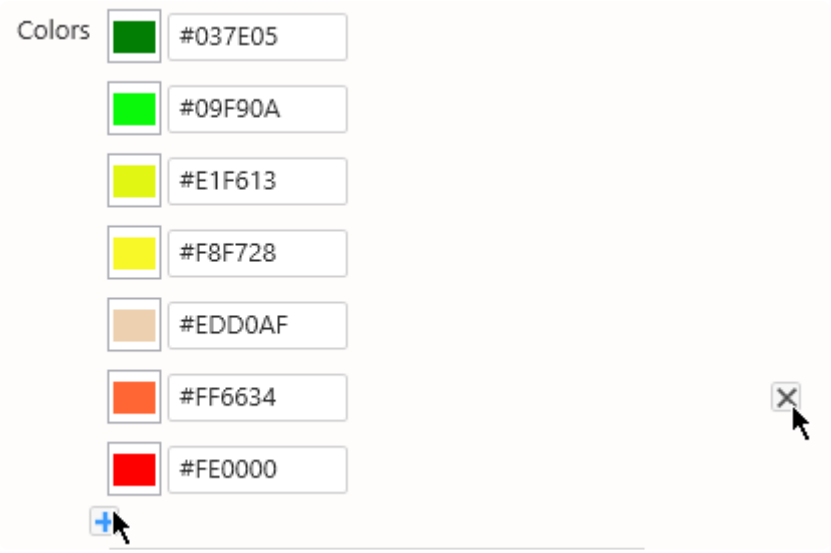


Labels

This is the list of grade labels to apply to each threshold.
By default, they are letter grades : `A`, `B`, `C`, etc…
You can add/remove labels, and change the corresponding text (to use `A++`, `A+` etc for example).

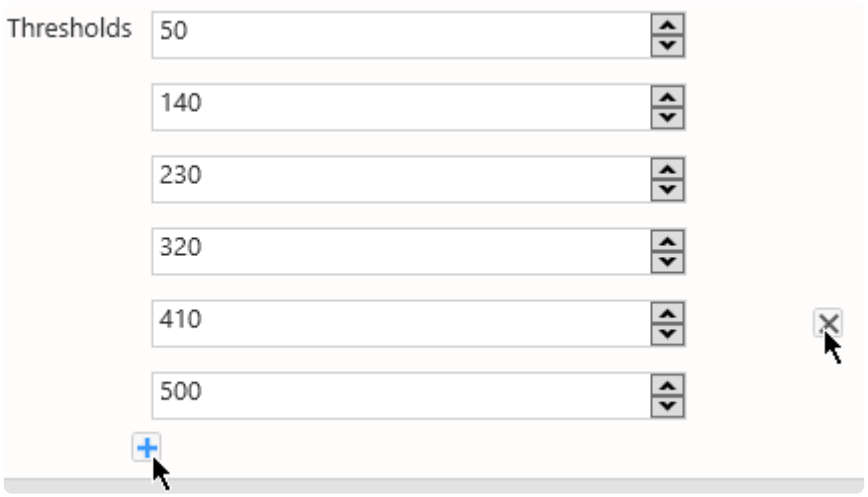
Colors

This is the list of colors to apply to each grade.
By default, this is a gradient from green (good grades) to red (bad grades).
You can add/remove/configure colors.



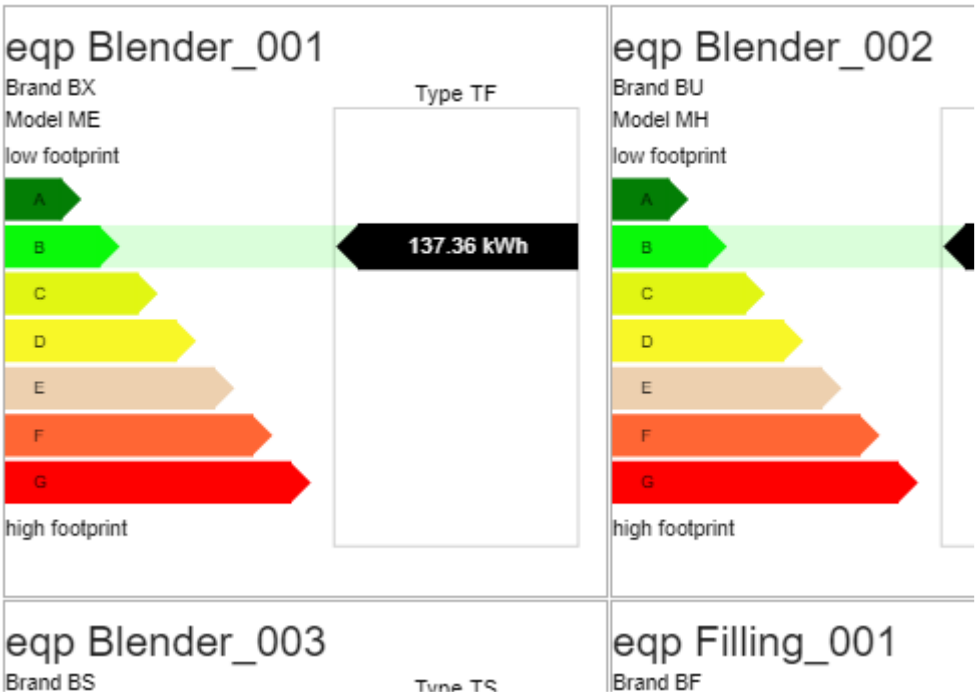
Thresholds

This is the list of thresholds themselves, used to calculate the grade.
You can add/remove/configure thresholds.



Last modified: May 05, 2020

2.8.14.3. Titles and Labels



Title

Optionally a title can be displayed on top of the Widget.
This is useful especially when configuring a *Split into many widgets* field : then configuring the Title as `{{:Row}}` will give you the field value for each sub-widget.

If no title is configured, then it is the same as hiding the title (it doesn't take visual space).

Unit

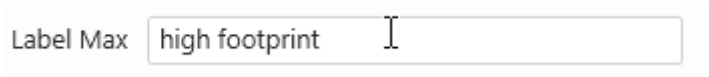
This is the unit of measure of the *Value*.
It will be displayed in the value display of the tooltip.

Label Min

This is the text label to display next to the low values of *Value*, usually corresponding to low energy consumption and good grades ("A").

Label Max

This is the text label to display next to the high values of *Value*, usually corresponding to high energy consumption and bad grades (“G”).



Last modified: May 05, 2020

2.8.15. Filter Widgets

Even though all Widgets can be configured to filter other widgets, some are especially dedicated to filtering :

- CheckBox
- ComboBox
- DatePicker
- RadioButton
- RangeNavigator
- RangeSlider

Last modified: May 03, 2019

2.8.15.1. ComboBox

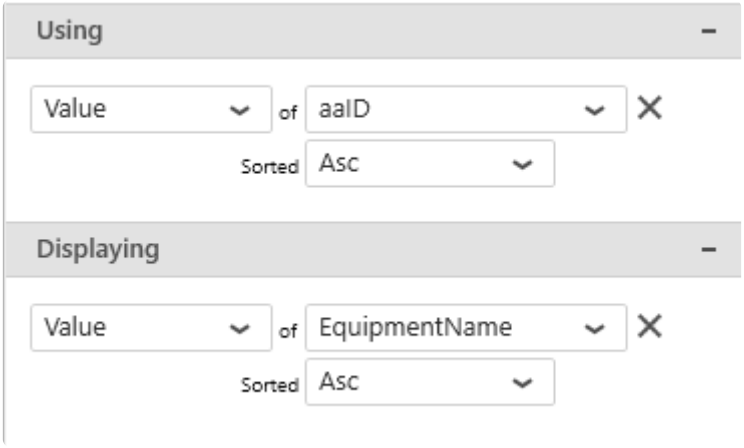
The Combobox presents multiple values in a foldable box and allows to make single or multiple selections.

Last modified: May 02, 2019

2.8.15.1.1. Binding Data

The Combobox has a 2 data fields for binding :

- *Using* : the value used internally for filtering
- *Displaying* : the value displayed for selection



Using : filter value

This represents the category we want to use internally to filter distinct values.

This field cannot be aggregated : it must have several separate values in order to be used as filter. This is usually an internal unique ID.
This is a required field.

Displaying : display value

This represents the text displayed to the user when viewing the Combobox.

This field cannot be aggregated : it must have several separate values in order to be used as filter. Values can be duplicates because the unique values used for filtering are the ones configured for *Using*.
This is an optional field.

If no *Displaying* field is set, the values from the *Using* field are displayed instead.

Last modified: May 02, 2019

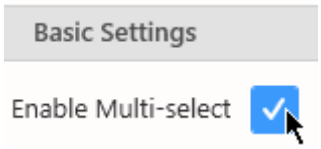
2.8.15.1.2. Configuring behavior

Multiple Selection

Configuring Multiple Selection

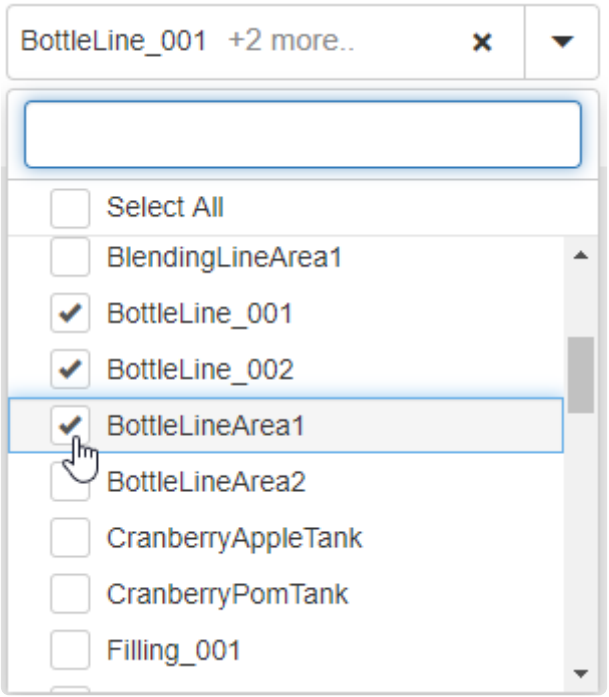
The Combobox can be configured to allow multiple selection.

Check the Enable Multi-Select box under Basic Settings :



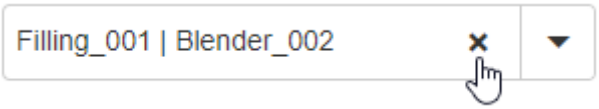
Selecting items at runtime

A checkbox appears in front of every item, allowing multiple selection :



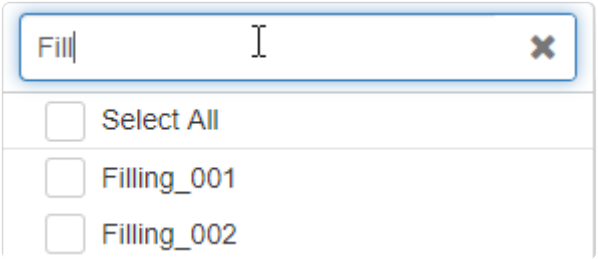
In addition, a new selection item called "*Select All*" (or "*Unselect All*") is added at the top, to allow to select/unselect all items at once.

Clearing item selection at runtime

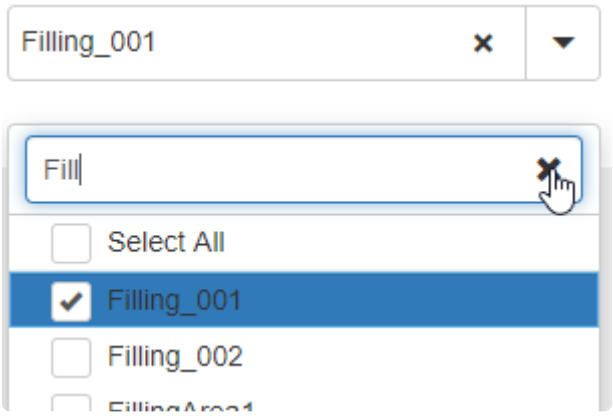


Item search

When Multiple Selection is configured, the Combobox contains a text box that allows the user to search for items



This search filters the content of the Commbobox incrementally and can be cleared using the cross button inside :



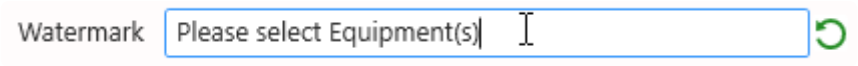
Last modified: May 02, 2019

2.8.15.1.3. Style and Appearance

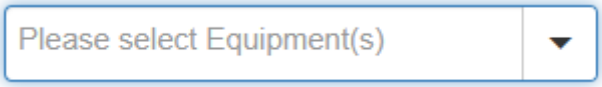
Watermark

Under Basic Settings, the *Watermark* is the text displayed inside the Combobox when no value is selected.
This allows to invite the user to make a selection.

For example :



… at runtime will look like this :



Last modified: May 02, 2019

2.8.15.2. CheckBox

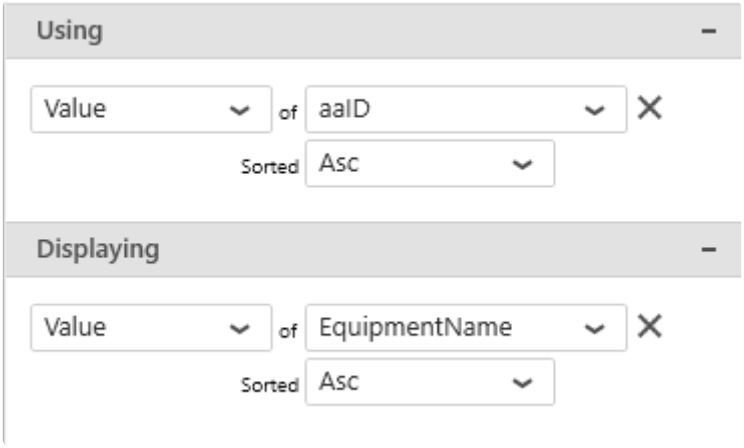
The CheckBox presents multiple values all visible in the widget and allows to make multiple selections.

Last modified: May 02, 2019

2.8.15.2.1. Binding Data

The Checkbox has a 2 data fields for binding :

- *Using* : the value used internally for filtering
- *Displaying* : the value displayed for selection



Using : filter value

This represents the category we want to use internally to filter distinct values.

This field cannot be aggregated : it must have several separate values in order to be used as filter. This is usually an internal unique ID.
This is a required field.

Displaying : display value

This represents the text displayed to the user when viewing the Checkbox.

This field cannot be aggregated : it must have several separate values in order to be used as filter. Values can be duplicates because the unique values used for filtering are the ones configured for *Using*.
This is an optional field.

If no *Displaying* field is set, the values from the *Using* field are displayed instead.

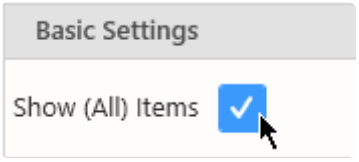
Last modified: May 02, 2019

2.8.15.2.2. Configuring behaviour

Show/Hide “All” item

By default, a item called “(All)” is added at the top, to allow to select/unselect all items at once.

You may show or hide this item from the Show (All) Items check box under Basic Settings :



Last modified: May 02, 2019

2.8.15.2.3. Style and Appearance

Multi-items Viewport

The layout of the checkboxes inside the container is configurable.
Please see the corresponding chapter to configure the view of the Widget container.

Last modified: May 15, 2019

2.8.15.3. RadioButton

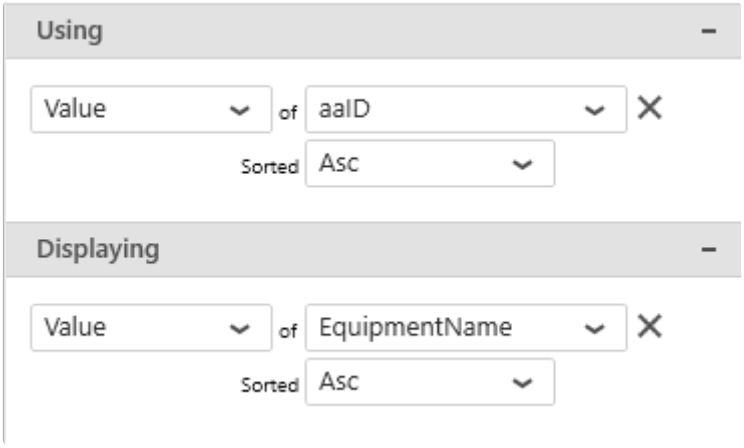
The RadioButton presents multiple values all visible in the widget and allows to make single selections.

Last modified: May 02, 2019

2.8.15.3.1. Binding Data

The Radiobutton has a 2 data fields for binding :

- *Using* : the value used internally for filtering
- *Displaying* : the value displayed for selection



Using : filter value

This represents the category we want to use internally to filter distinct values.

This field cannot be aggregated : it must have several separate values in order to be used as filter. This is usually an internal unique ID.
This is a required field.

Displaying : display value

This represents the text displayed to the user when viewing the Radiobutton.

This field cannot be aggregated : it must have several separate values in order to be used as filter. Values can be duplicates because the unique values used for filtering are the ones configured for *Using*.
This is an optional field.

If no *Displaying* field is set, the values from the *Using* field are displayed instead.

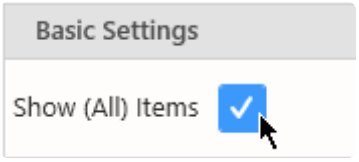
Last modified: May 02, 2019

2.8.15.3.2. Configuring behaviour

Show/Hide “All” item

By default, a item called “(All)” is added at the top, to allow to select/unselect all items at once.

You may show or hide this item from the Show (All) Items check box under Basic Settings :



Last modified: May 02, 2019

2.8.15.3.3. Style and Appearance

Multi-items Viewport

The layout of the radiobuttons inside the container is configurable.
Please see the corresponding chapter to configure the view of theWidget container.

Last modified: May 15, 2019

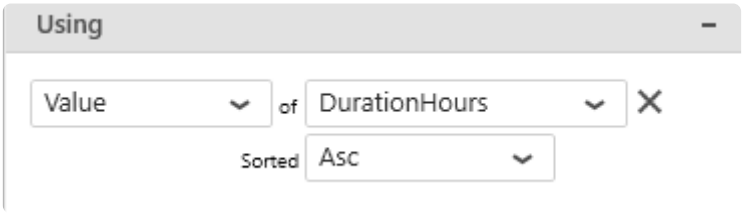
2.8.15.4. RangeSlider

The Range Slider presents a slider and allows to select number ranges or single numbers, as well as dates.

Last modified: May 03, 2019

2.8.15.4.1. Binding Data

The Range Slider has a single data field for binding : *Using*



Using : filter value and display

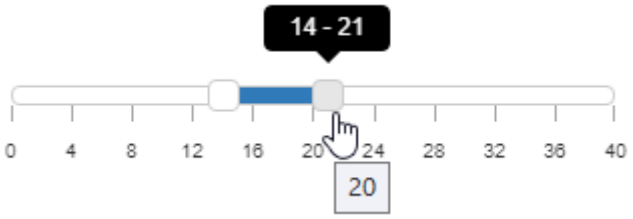
This represents the category we want to use internally to filter distinct values.
This is also the text labels displayed to the user when viewing the RangeSlider

This field cannot be aggregated because it must have several separate values in order to be used as filter.
This is a required field.

Several data types are supported :

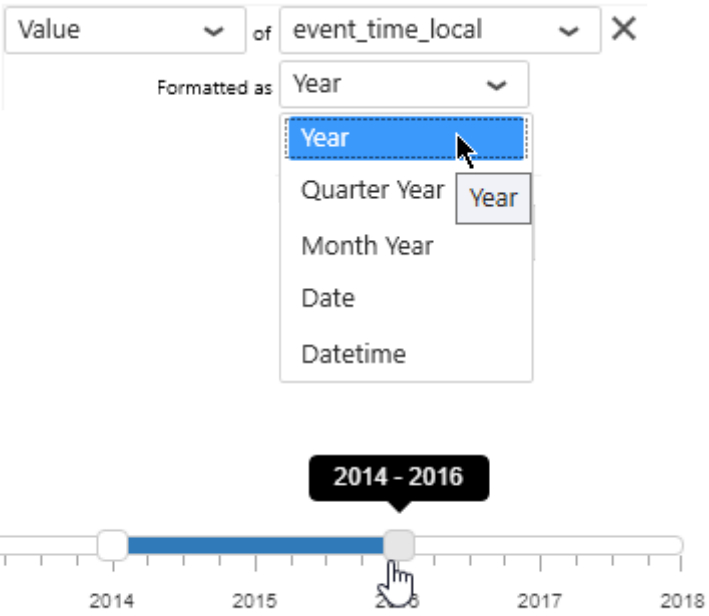
Number field

With a number field, the Range Slider displays numbers in order between the minimum value and the maximum value, and allows to select values :



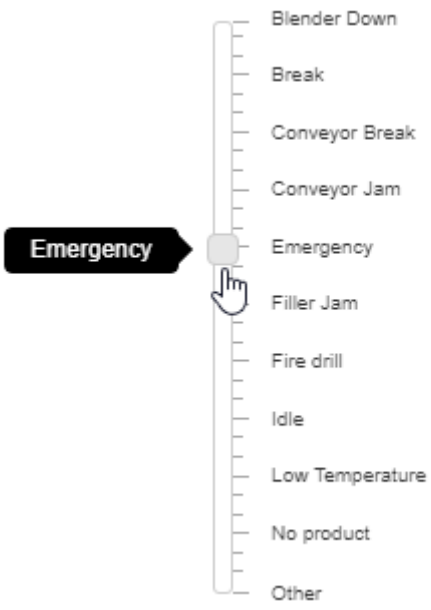
Date/Time field

With a date or time field, the Range Slider allows to select time intervals depending on the Date/Time formatting selected :



String field

The Range Slider could also be used to select strings, especially if using sorting (alphabetically) :



Last modified: May 03, 2019

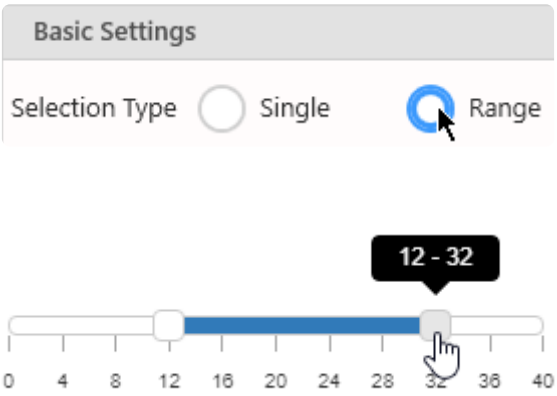
2.8.15.4.2. Configuring behaviour

Range Selection

Range

The Range Slider can be configured to select a range of values (default).

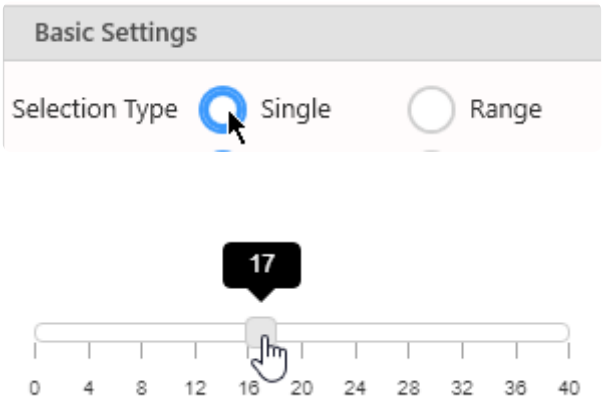
Under Basic Settings > Selection Type, select Range :



Single

The Range Slider can instead be configured to select a single value.

Under Basic Settings > Selection Type, select Single :



Last modified: May 03, 2019

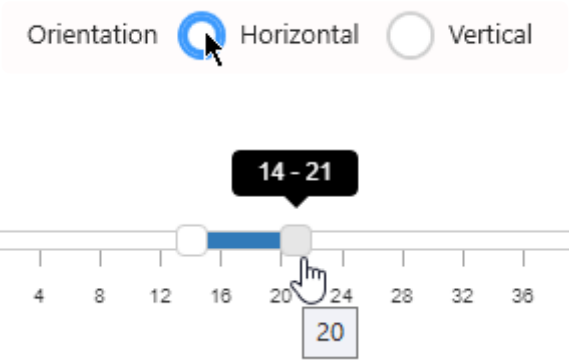
2.8.15.4.3. Configuring style

Orientation

Horizontal

The Range Slider can be oriented horizontally (default).

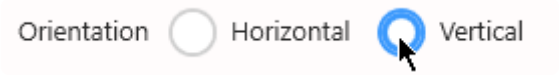
Under Basic Settings > Orientation, select Horizontal :

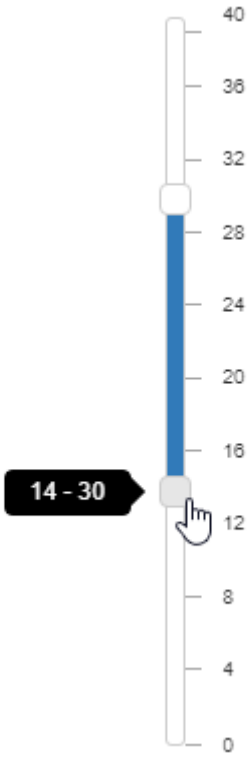


Vertical

The Range Slider can be oriented vertically.

Under Basic Settings > Orientation, select Vertical :





Last modified: May 03, 2019

2.8.15.5. DatePicker

The DatePicker presents a calendar and allows to make single or multiple date selections.

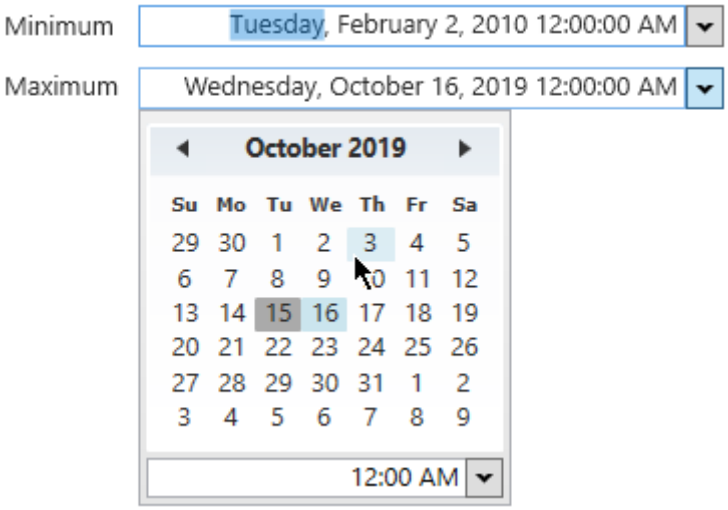
Last modified: May 03, 2019

2.8.15.5.1. Configuring without data

The Date Picker doesn't need to be bound to a Data source.
Instead, you may wish to configure it as a calendar filled with dates.

To choose this, select *Custom Data* :

Then you can just configure a *Minimum* and *Maximum* date to generate the calendar :

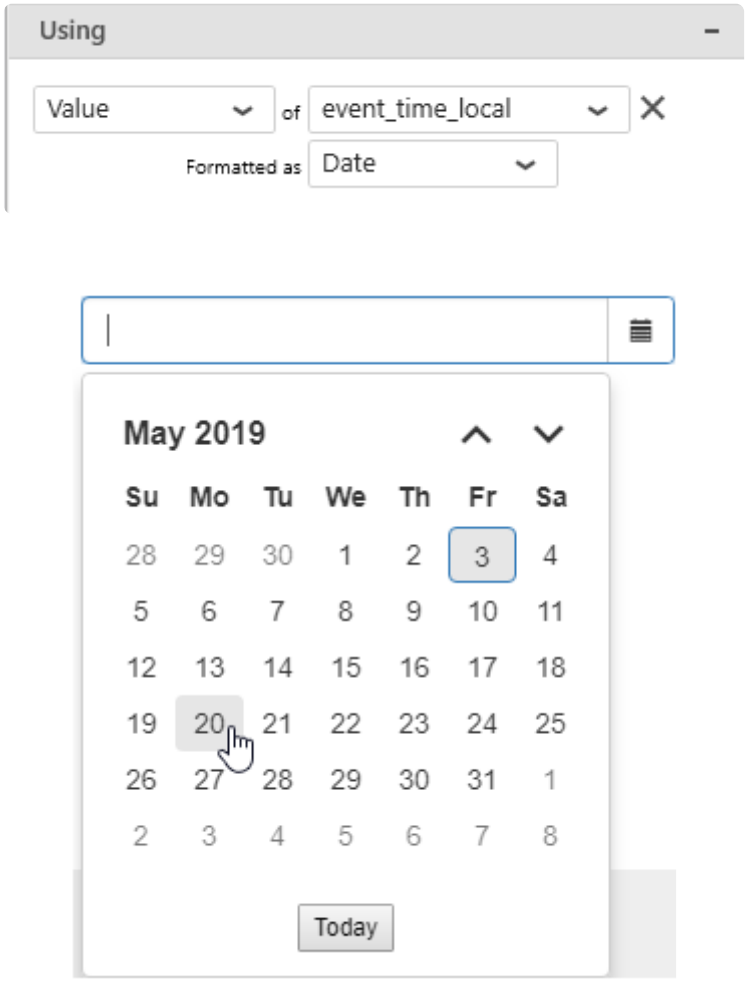


Last modified: Oct 15, 2019

2.8.15.5.2. Binding Data

Instead of filling the calendar with all dates, you may wish to configure it to allow to select only the dates that exist in the data.
For this, select *Datasource Data* :

The DatePicker has a single data field for binding : *Using*



Using : filter value and display

This represents the date field we want to use to present all the available distinct dates.

This field cannot be aggregated because it must have several separate values in order to be used as filter.
This is a required field.

The range of available dates will be the range between the oldest and the latest date in the date field.

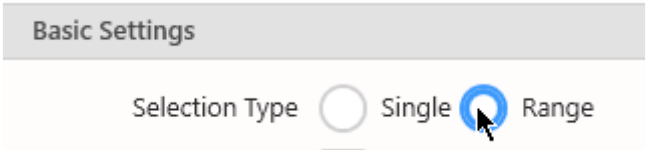
Last modified: Oct 15, 2019

2.8.15.5.3. Configuring behaviour

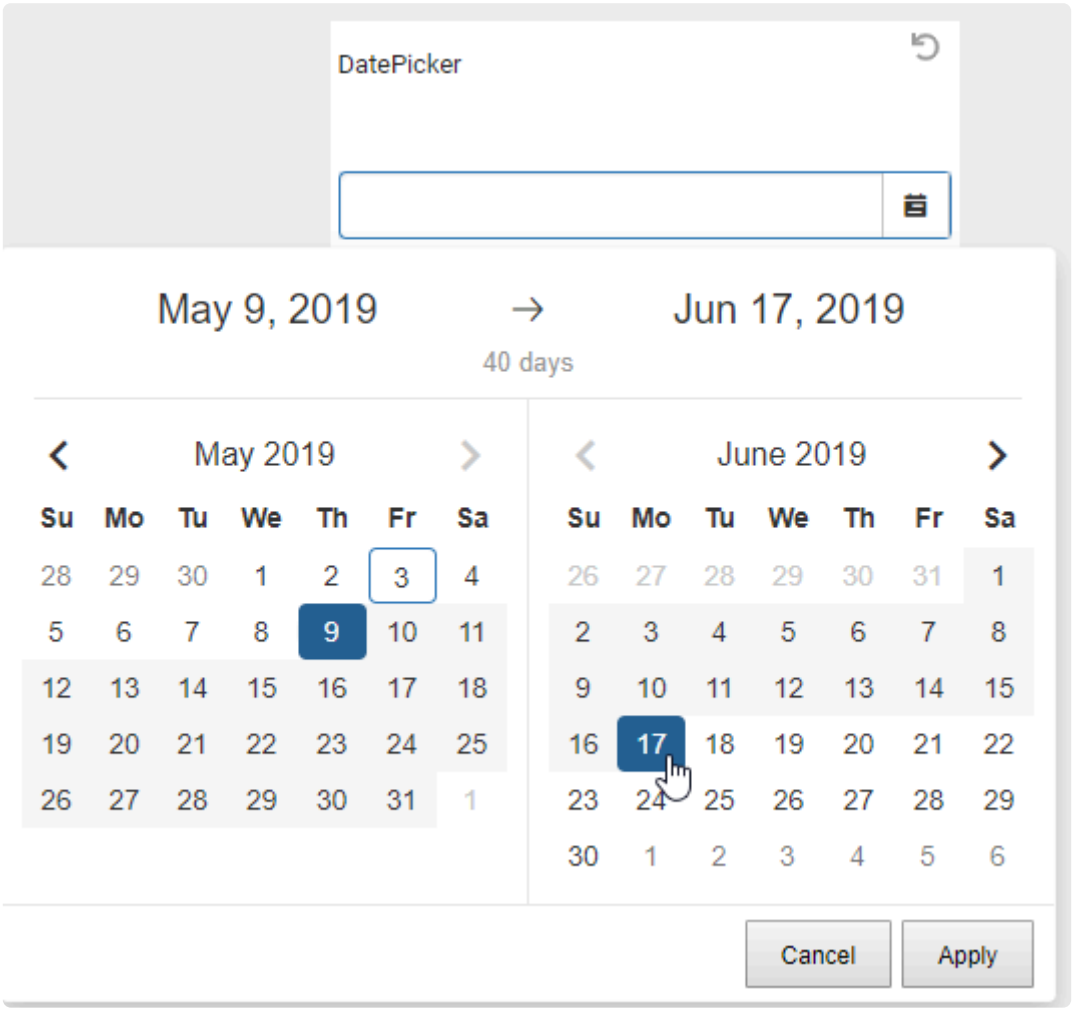
Range Selection

The DatePicker can be configured to allow range selection.

Configure the Selection Type to Range :



The calendar can now show two pages at the same time, to make it easier to select a Start Date and an End Date :



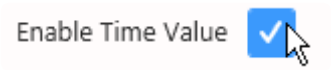
Time Selection

The Date Picker can be configured to allow time selection in addition to date.

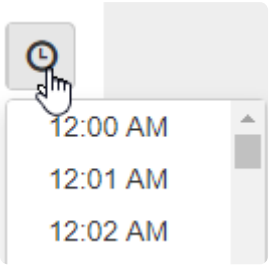
If you configure *Time* selection with *Single* selection, the filter will try to match only the data that has the exact same minute.
This usually gives no result, and you may want instead to use the *Range* selection with *Time*.

Enable Time selection

Select *Enable Time Value* :

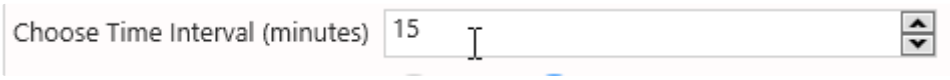


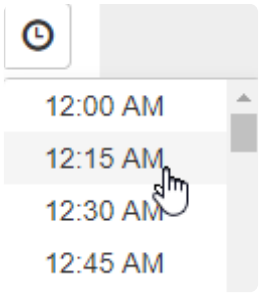
At runtime, this adds a time selector (or two, if Range is configured) :



Configure available times

By default, the time selector shows a list of times from midnight to midnight every 1 minute.
This can be configured with *Choose Time Interval (minutes)* :





Default selection as Relative Date

What is Default selection as Relative Date

Like other Widgets, the DatePicker allows to configure its *default value* by selecting in the preview inside Alpana Designer. This default selection will be a fixed date range.

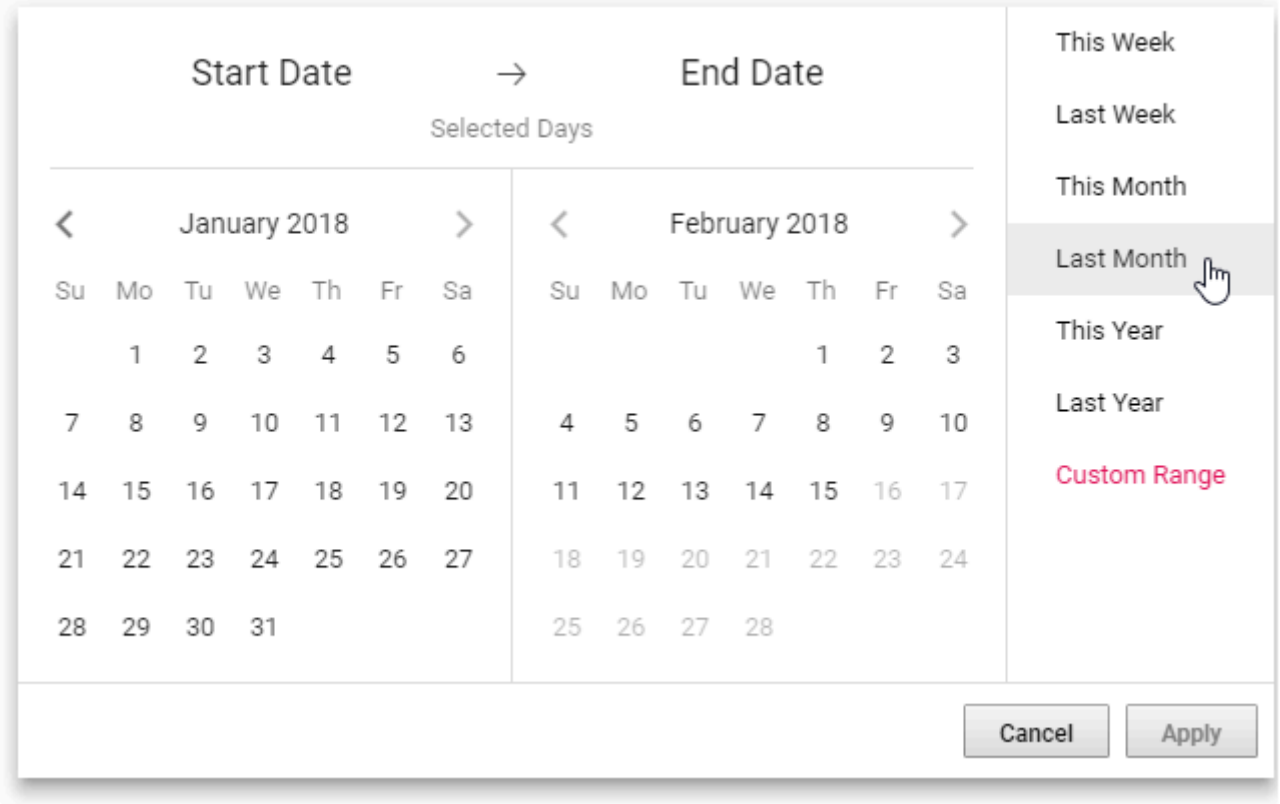
In addition, it is possible to configure a **relative date range as default selection**.
This is described in the corresponding chapter.

Preset Ranges

When the Date Picker is configured with *Range* selection, you can display a list of commonly used presets for relative selection.

Check *Enable Preset Ranges* :

At runtime, in addition to the calendar selection, the user will be able to select from the preset ranges :



- This Week
- Last Week
- This Month
- Last Month
- This Year
- Last Year

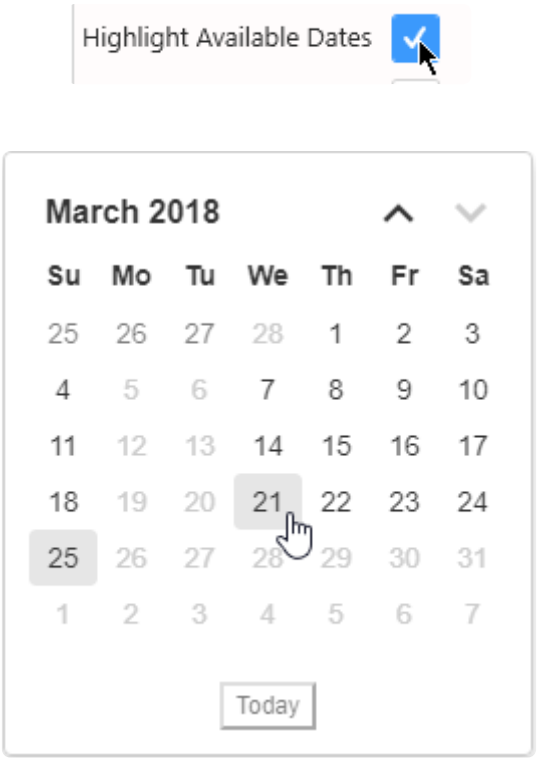
Last modified: Oct 30, 2019

2.8.15.5.4. Configuring style

Highlight Available Dates

By default, all dates in the interval are available for selection. However, sometimes there is no actual data for a given date.

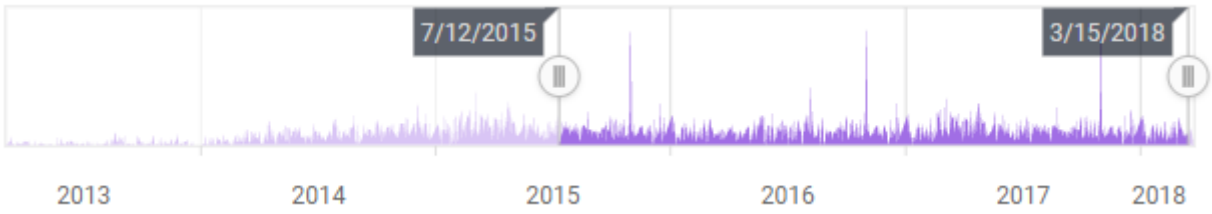
By checking Highlight Available Dates under Basic Settings, you can prevent users from selecting dates that don’t have data.
The dates that don’t appear in the configured date field are greyed-out and can’t be selected :



Last modified: May 03, 2019

2.8.15.6. RangeNavigator

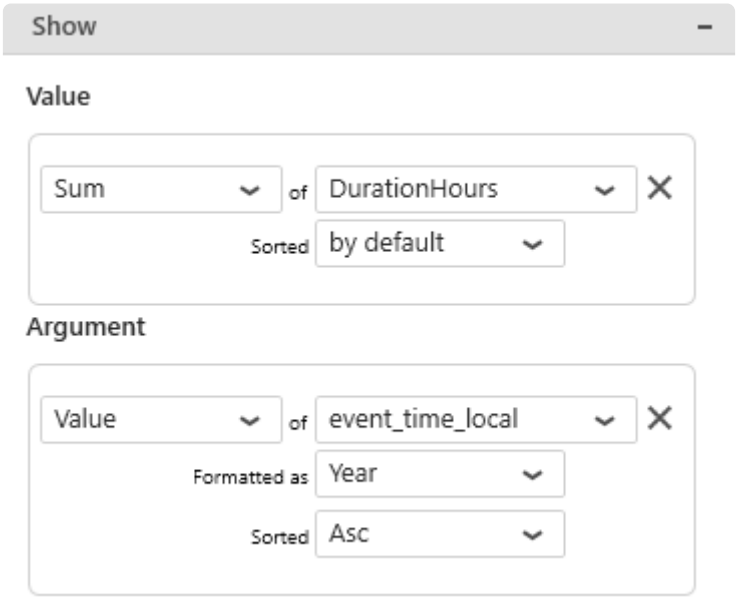
The Range Navigator presents a chart as background and allows to make date range selections on top of it.



Last modified: May 03, 2019

2.8.15.6.1. Binding Data

The following field containers are available for configuring data binding on a Range Navigator :



- Value : the height of the background chart
- Argument : the field used to filter data during selection, and also how we group values together to create points on the background chart

Value

This represents the quantity we want to display on the background chart, and will bind for example to the height of the Area (Area chart).

The Value field must be aggregated : you must choose an aggregation function (SUM, AVG, …)
This is a required field.

Argument

This represents the category we want to use internally to filter distinct values.
This is how we group values together to create points on the background chart.

This field cannot be aggregated because it must have several separate values in order to be used as filter.
This is a required field, and must be a date field.

The date format (Date, MonthYear, …) is used to group the data to display on the chart, but also during filter selection.

Last modified: May 03, 2019

2.8.15.6.2. Configuring behaviour

Default selection as Relative Date

What is Default selection as Relative Date

Like other Widgets, the RangeNavigator allows to configure its *default value* by selecting in the preview inside Alpana Designer. This default selection will be a fixed date range.

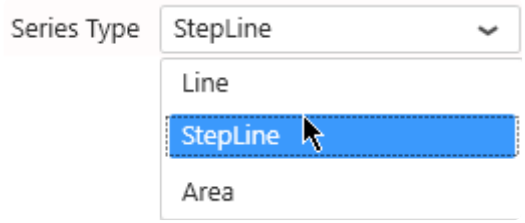
In addition, it is possible to configure a **relative date range as default selection**.
This is described in the corresponding chapter.

Last modified: May 24, 2019

2.8.15.6.3. Configuring style

Background Chart style

The style of the background chart can be changed by selecting from the Basic Settings > Series Type list :



Axis Interval Type

(new in 2020 R1)

This configures the labels that will be displayed on the time (X) axis :

- *Years* : only the year is displayed
- *Quarters* : both the year and the quarter is displayed
- *Months* : both the year and the month is displayed
- *Days* : both the year and the “day + month” is displayed
- *Auto* : the interval is selected automatically depending on the time span of the data

Clicking an axis label (like the year number) will select that period.

Note : this does not define the date format of the Parameter when filtering, this is defined by the formatting of the Argument (see Binding Data)

Last modified: Jun 19, 2020

2.8.15.7. TreeView

(new in Alpana 2020 R1)

The Tree View allows to present hierarchical context in a foldable tree, and select values from any node or leaf to filter the Dashboard.

Last modified: Apr 27, 2020

2.8.15.7.1. Binding Data

Introduction

The Tree View is composed of Nodes representing the hierarchy.
This allows the user to select context values from the top level, or drill-down into the lower levels to filter more specifically.
Each Node Group corresponds to a hierarchy level.

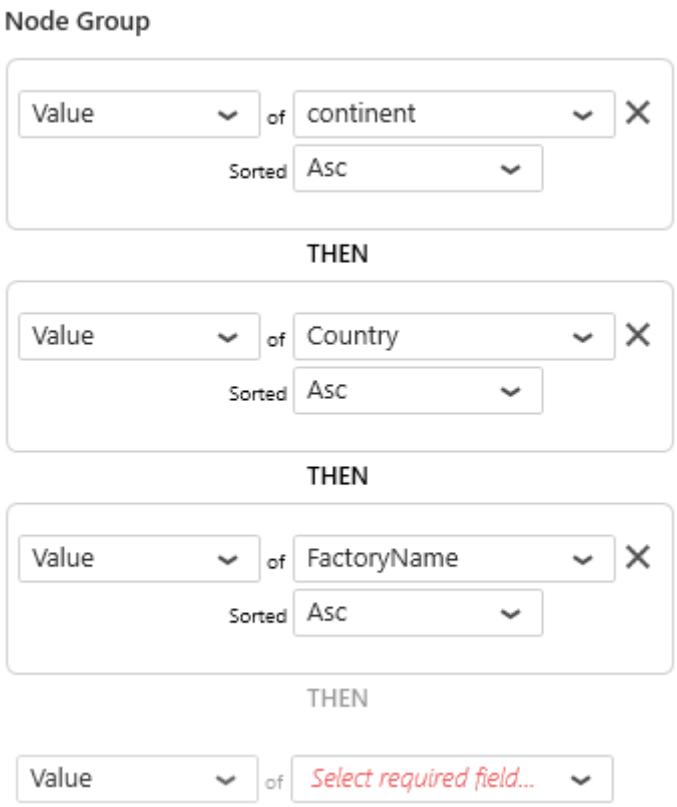
The following fields are available for configuring data binding on a Tree View :

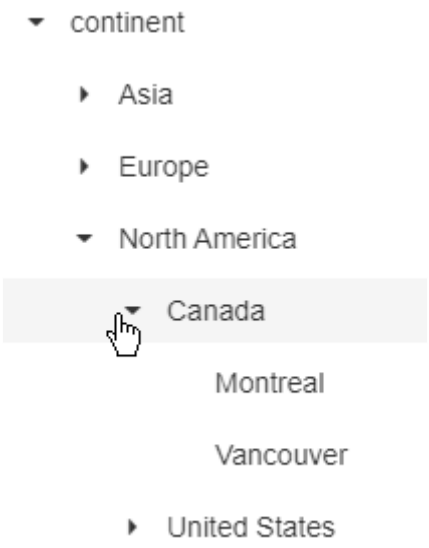
- *Node Group 1, 2, 3, …* : each field corresponds to one hierarchy level, its values are the corresponding user choices at that level.

Node Group 1

This represents the *category* we want to use to dynamically separate Nodes inside the 1st hierarchy level.

At least 1 field is necessary, to have 1 level of values presented.
These fields **cannot be aggregated** : they must have several separate values in order to be used as filter.





Note :
If the last field contains `NULL` values, then the corresponding Nodes are not represented. If another field has `NULL` values, then the corresponding Node is represented as `No <FieldName> (<parentValue>)`. This allows to list children of nodes which are `NULL`.

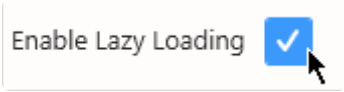
Last modified: Apr 27, 2020

2.8.15.7.2. Configuring Behaviour

Lazy Loading

Usually, a large list of values means a lot of nodes to calculate before displaying the Widget, which could mean a performance degradation.
Instead, you can optionally let the Tree View only load the lists of values on demand : until the user unfolds a node, the children values are not loaded.
If the node unfolded has many values, this may mean that it will take some time to unfold.
On the other hand, the initial loading of the Widget is much faster, and most nodes will open fast if they each have few values inside their immediate children.

To enable this behavior, check *Enable Lazy Loading* :



Last modified: Apr 27, 2020

2.8.16. Static Widgets

Alpana allows to create some Widget types that are not bound to data.
The following chapters describe those *Static Widgets*.

Last modified: May 02, 2019

2.8.16.1. Image

The Image Widget allows to display static images on a dashboard, and handles transparency.

This can be useful for displaying a logo or a reference image for example.

Last modified: Jun 20, 2019

2.8.16.1.1. bind image

To configure, simply browse for a supported image file : `.png`, `.jpg`, `.bmp`, `.jpeg`, `.gif`, `.emf`, `.tif`, `.tiff`, `.ico`

Last modified: May 02, 2019

2.8.16.1.2. Image Modes

Images can be configured with an Image Mode which determines how the image is cropped/stretched to fit the Widget container

Default

In Default mode, the image is simply centered and cropped, but not resized.

> If you use a transparent image on the image, the margins will not be visible.

Fill

In Fill mode, the image is stretched to fit the container.

This can be useful to apply a background image and add Widgets on top of it : the Widget's position will be constant relative to the image, independently of the web page resizing.

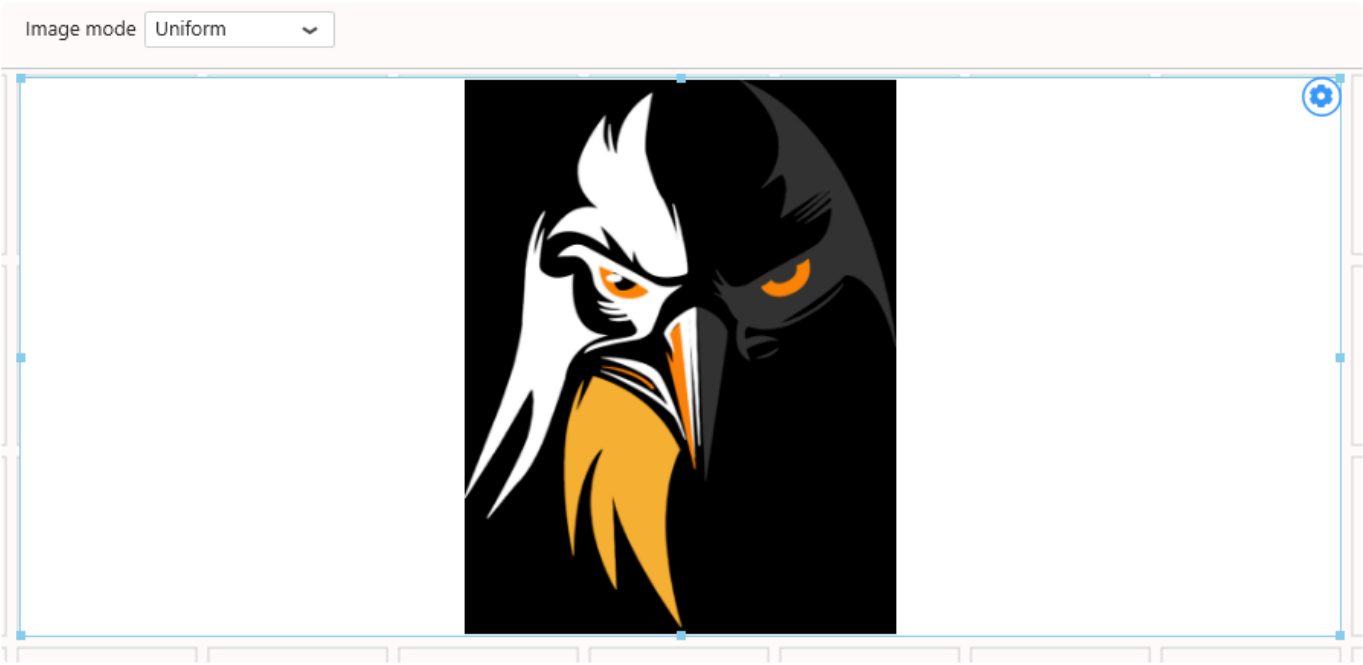
Uniform

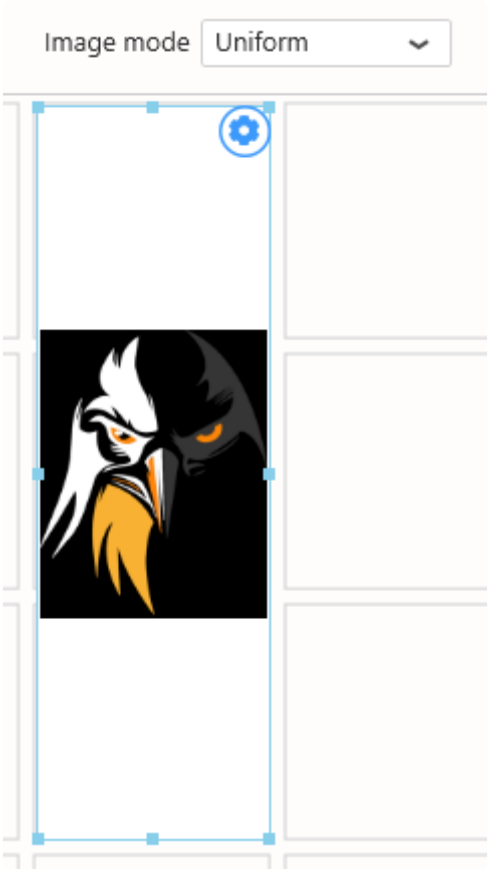
In Uniform mode, the image is resized to fit the container without being stretched or cropped. The container is not filled completely.

This is useful to make sure the full image is always visible without being stretched.

This is usually desired for a logo.

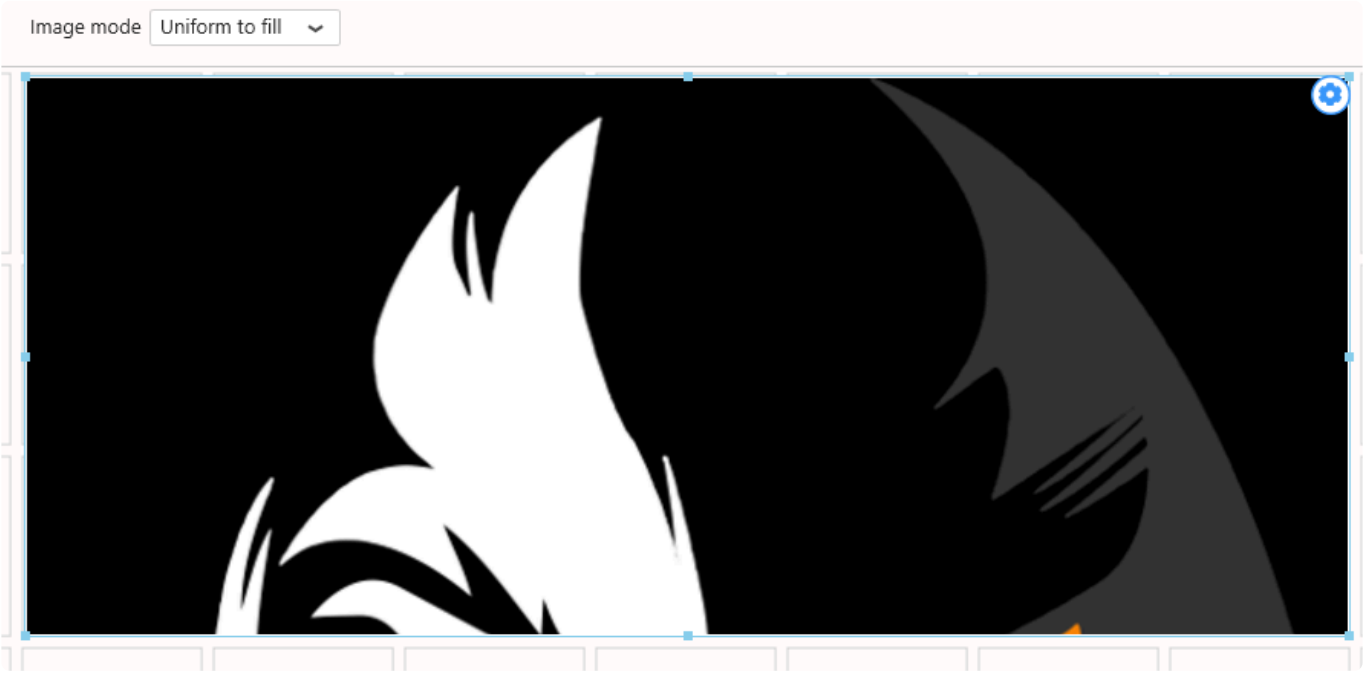
If you use a transparent image on the image, the margins will not be visible.

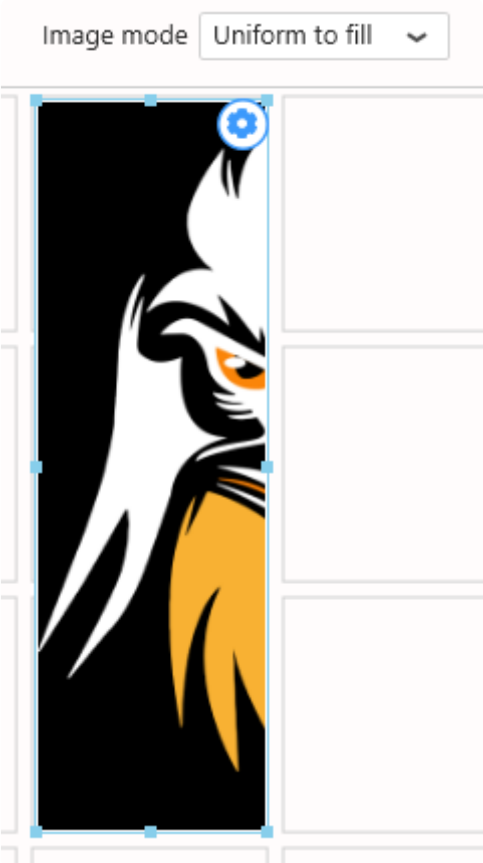




Uniform to fill

In Uniform to fill mode, the image is resized and cropped to fill the container without being stretched.





Last modified: May 02, 2019

2.8.16.2. Label

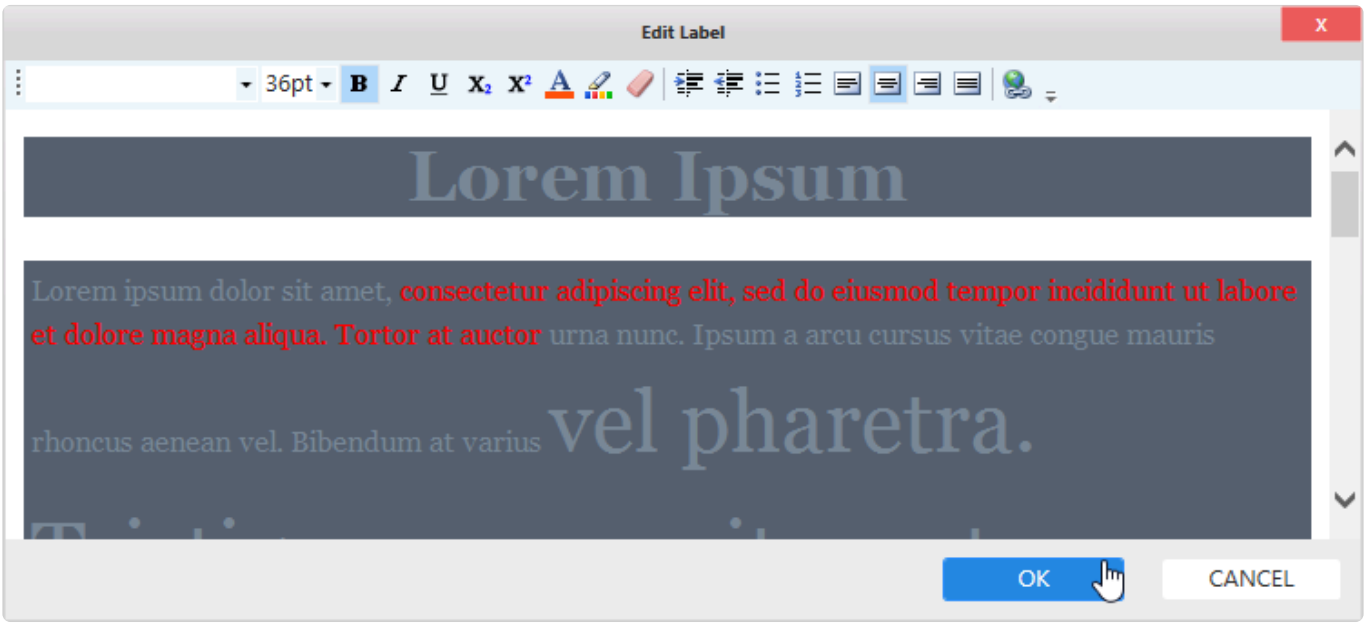
The Label Widget allows to insert rich text into a Dashboard.

Last modified: May 02, 2019

2.8.16.2.1. Configuration

Text editor

The Label Widget can be configured like any rich text editor, with font type, size, style, color, alignment, bullet lists, hyperlinks, …



Label Parameters

The text content can include placeholders that will be replaced by Parameter values, see Label Parameters

Last modified: Aug 26, 2020

2.8.16.3. Embed

The *Embed* Widget is a widget dedicated to embedding other content into Alpana :

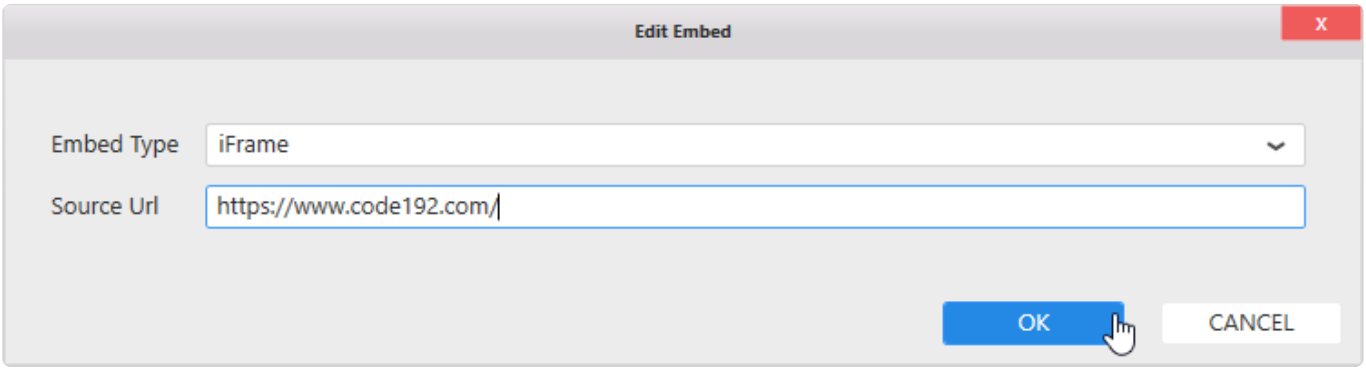
- other websites
- components from other applications (like Widgets from Aveva Insight)
- video or image streams from security cameras
- online documents published on other websites
- other web widgets, like streaming videos, calendars, etc…

Last modified: May 02, 2019

2.8.16.3.1. Configuration

Source Url

This is the Url that points to the content you wish to embed. Simply paste the Url in this text box.



Dynamic URL

You can update the URL displayed in the widget when a Parameter value is updated.
For this, use the Parameter placeholder :

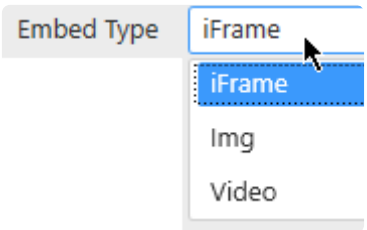
```
${MyParameterName}
```

Example :
When the Parameter `param_Country` is updated, the Embed widget will display the corresponding Wikipedia page :

```
https://en.wikipedia.org/wiki/${param_Country}
```

Embed Type

The *Embed Type* determines how the content will be embedded :



iFrame

The content will be embedded “as-is”.
This is usually the default.

Video

This is for video streams that require to be embedded inside `<video>` HTML markup.
This can be the case for some security cameras.

Img

This is for image streams that require to be embedded inside `<img>` HTML markup.
This can be the case for some security cameras / webcams.

Last modified: Oct 15, 2019

2.8.16.3.2. Note : Cross-origin resource sharing

Embedding a content from one website (your content) into another website (Alpana) often requires to be allowed from the embedded website (your content).
This is called Cross-origin resource sharing.

For example, the https://www.google.com/ URL currently refuses to be embedded inside an iframe of another website. This is a normal thing, for security purposes.
In some cases, you can create an exception, please see with the documentation of your content website.

Last modified: May 02, 2019

2.8.16.4. PDF viewer

The PDF Widget allows to insert PDF documents into a Dashboard.

Last modified: Oct 16, 2019

2.8.16.4.1. Configuration

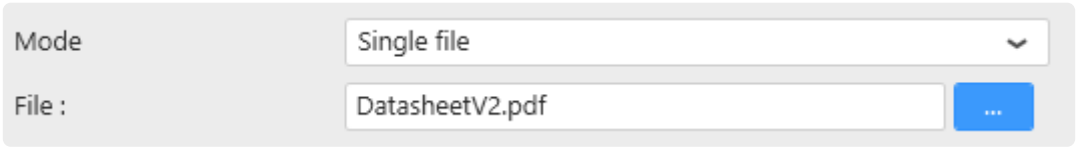
Selecting PDF file

There are two *Modes* for selecting a PDF file to display :

- *Single file* : a single file is displayed, always the same, and cannot be changed at runtime.
- *Based on parameter value* : a single file is displayed at a time, selected from a list of possible files. At runtime, which file is displayed depends on the value of a Parameter.

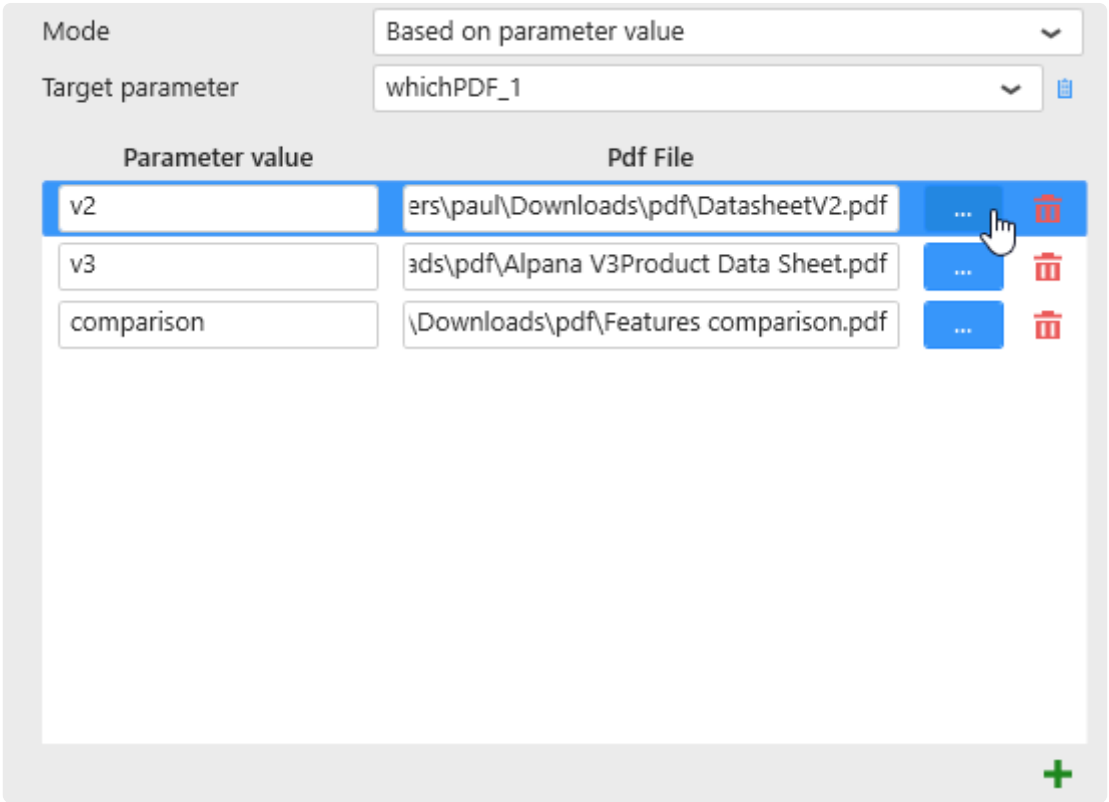
The files become embedded inside the Dashboard as dependencies, and will stay the same even if the original file is modified or deleted.

Single file



Click the ... button to browse for a file to select.

Based on parameter value



In this mode, you must bind the Widget to a Parameter, then configure which value of the Parameter will open which PDF file.

Selecting a Parameter

Select the *Target Parameter* that will hold the desired value.
At runtime, when this Parameter changes, the widget will display the corresponding PDF file.

Selecting PDF files and binding to Parameter values

Click the + button to add a new binding :

- Enter the *Parameter value* in the text box
- Click the ... button to browse for a file to select

When the selected *Target Parameter* will take the value in *Parameter value*, then the corresponding PDF file will be displayed.

To remove a binding, click the trash icon on the right of it.

Configuring the display

Several options allow to configure the way the PDF file is displayed :

Page Displayed

Select the page number of the PDF that should be displayed when the Dashboard opens.

Zoom (Percent)

Select the zoom factor to display the PDF when the Dashboard opens.

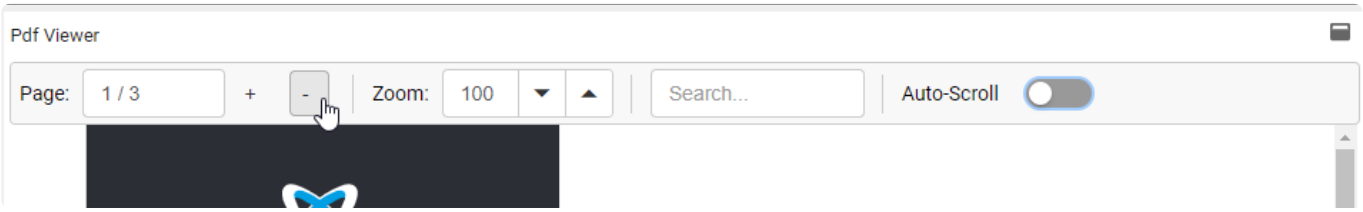
Auto-Scroll

If you check *Enable Page Auto-Scroll*, the PDF will be played like a slides presentation : each PDF page will be displayed for a certain duration, then the next page will be displayed automatically. When the end has been reached, the first page is next.

The duration to display each page is defined by *Auto-Scroll Timer (Seconds)* : the number of seconds to display each PDF page.

Runtime Toolbar

When you check *Enable Runtime Toolbar*, a toolbar is displayed at runtime with some controls to let you interact with the PDF file :



- *Page* : displays the current page number / and total page number. You can input the desired page number in this box.
- + − : Go to the next page by clicking + or the previous page by clicking −
- *Zoom* : enter the desired zoom factor (%) in the text box, or use the up/down arrows to increase/ decrease the zoom factor
- *Search…* : enter strings of text to search the PDF
- *Auto-Scroll* : enable the Auto-Scroll feature (see above)

Last modified: Oct 16, 2019

2.8.17. Input Grid

The *Alpana Runtime Connection Input Grid* widget allows to configure an input form for an existing Runtime Connection and include it inside a Dashboard.

This widget is a bit special :

- it allows to input data.
- it is not bound to a Data Source
- it is not displayed on a Public Dashboard because there would be no control over who inputs data

Pre-requisite : login

In order to configure this Widget from the Designer, you need to be logged-in to the Alpana Server, see Login to Alpana Server

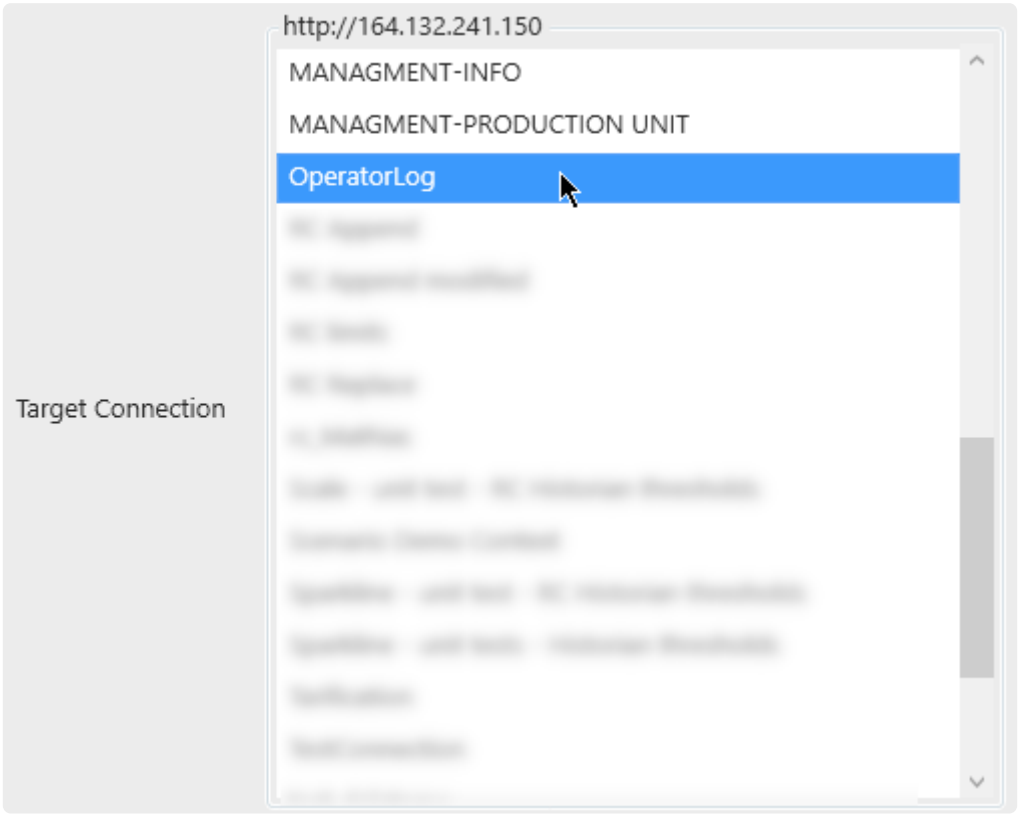
Last modified: Jun 16, 2020

2.8.17.1. Configuration

When clicking the widget's settings, the *Edit Input Grid* dialog opens and lets you configure the options of that widget :

Selecting a *Target Connection*

Once you are connected to Alpana Server, the dialog presents the list of all Runtime Connections published on the Server.

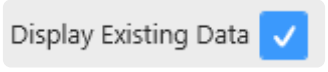


Select the desired Runtime Connection from the list.

Display and Permission options

Display Existing Data

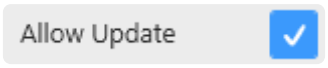
You can optionally *Display Existing Data* :



This allows the user to see all data rows that were entered previously.
If this is disabled, then the user will only see the data row that was input during this session.

Allow Update

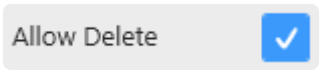
If you *Display Existing Data*, you can also optionally *Allow Update* :



This allows users to modify the data on any existing entry in the table.

Allow Delete

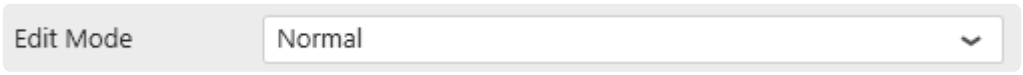
If you *Display Existing Data*, you can also optionally *Allow Delete* :



This allows users to delete any existing entry in the table.

Input form options

The *Edit Mode* defines how a new data row is created or how an existing data row is edited :

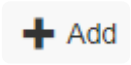


Normal

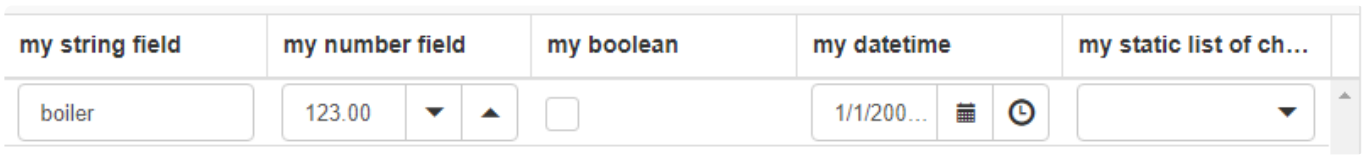
In *Normal* mode, the data is edited inline inside the grid :

Adding data

The user clicks the *Add* button :

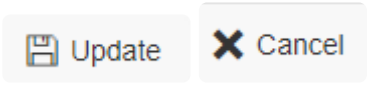


This creates a new empty row (or pre-filled with default values) :



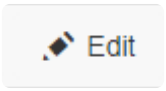
Note : the auto-fields cannot be edited since they are created automatically by Alpana.

The user then updates all the fields as desired, and either clicks *Update* to save the change, or *Cancel* to cancel creating the entry :



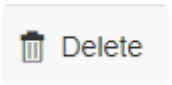
Editing data

On an existing data row, the user can edit by clicking on the row and clicking the *Edit* button :



Deleting data

On an existing data row, the user can delete the row by clicking on the row and clicking the *Delete* button :

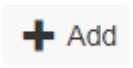


Dialog

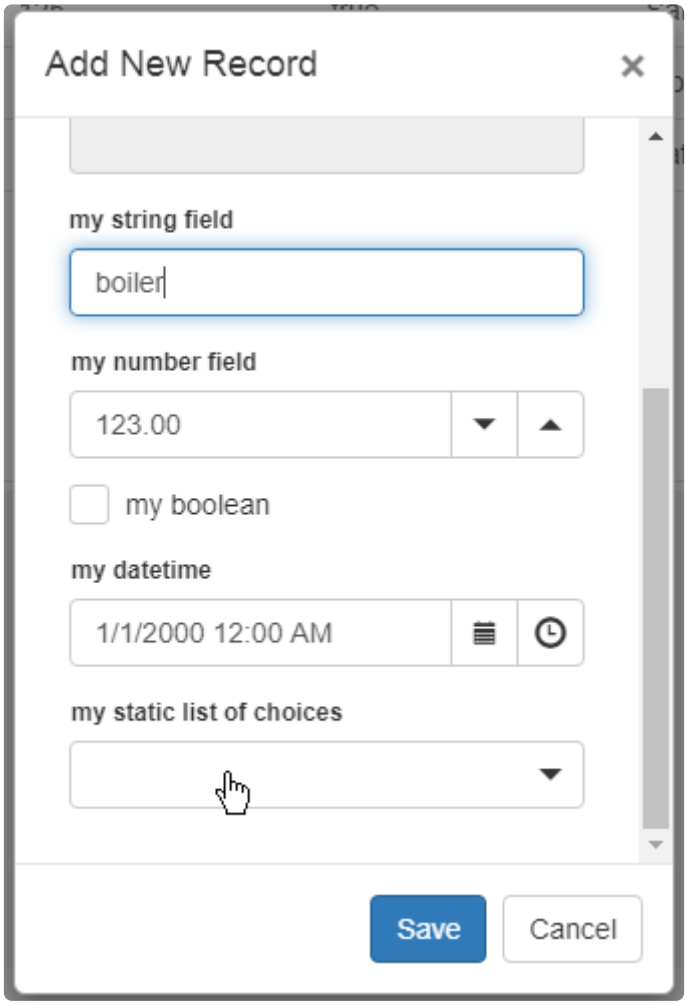
In *Dialog* mode, a pop-up dialog allows to edit/input data :

Adding data

The user clicks the *Add* button :



This opens the input dialog (empty or pre-filled with default values) :

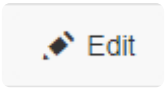


Note : the auto-fields cannot be edited since they are created automatically by Alpana.

The user then updates all the fields as desired, and can either *Save* or_Cancel_ the dialog :

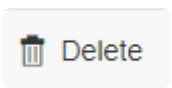
Editing data

On an existing data row, the user can edit by clicking on the row and clicking the *Edit* button :



Deleting data

On an existing data row, the user can delete the row by clicking on the row and clicking the *Delete* button :



Last modified: Jun 16, 2020

2.9. Using Dashboard Parameters

Last modified: May 24, 2019

2.9.1. What is a Parameter ?

A **Parameter** is a proxy between actions and results in a *Dashboard*.

What is this for ?

This is the mechanism behind Dashboard Filters, dynamic Stored Procedure calls, URL Parameters, …

How does it work ?

A Parameter acts as a **variable** : its value can be assigned to and read from.

Listeners can subscribe to update automatically when the Parameter value changes.

A Parameter can take 0, 1 or more values.
A Parameter can listen to one or more Master Widgets to update its value.

Examples

Dynamic filtering

Dashboard Widgets can listen to Parameters :
when the Parameter changes, the Widget is **filtered** accordingly.

Dynamic data calls

Stored Procedures can also be set by Parameters :
when the Parameter changes, the Stored Procedure is updated accordingly.

Getting values from the URL

Parameters values can also be given through URLs to change their default value.

Automatically generated Parameters

In some cases, Parameters are created automatically : see Chapter Widget as Master Filter.

Last modified: May 24, 2019

2.9.2. Managing Parameters

Parameters are configured and managed in a central menu in the Parameters Editor.
Read the next chapters to know more.

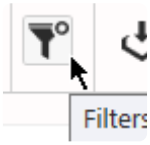
Last modified: May 24, 2019

2.9.2.1. Opening the Parameters Editor

How to open

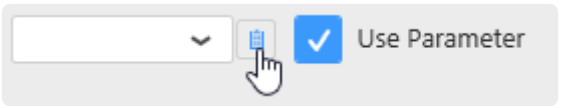
From the Dashboard activity tab

In the top bar, use the Edit Filters icon :



From a “Use Parameter” configuration

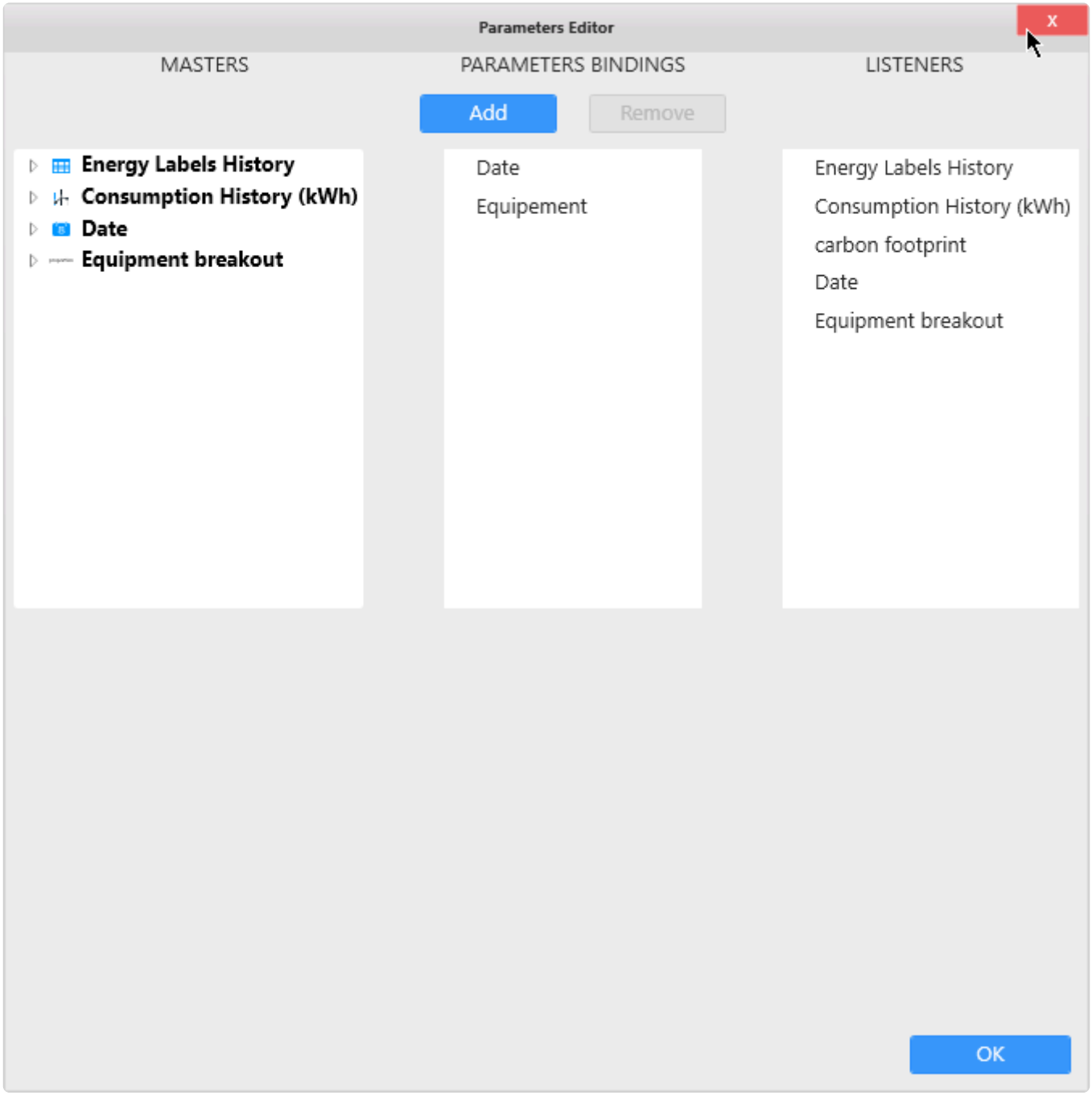
In many places where a Parameter can be used, a list icon acts as shortcut to open the Parameters Editor :



This appears for example in Data Source filters, in Stored Procedures parameters, etc…

Result : Parameters Editor

This opens the “Parameters Editor” dialog :



From there, you can manage Parameters :

- create Parameters
- rename Parameters
- remove Parameters
- change their default value
- change bindings between Widgets and Parameters

Masters

On the left is a list of Master Widgets and their selection events : this is a source of data for Parameters.

Parameters Binding

In the middle is a list of Parameters : this is the variable that contains the value.

Listeners

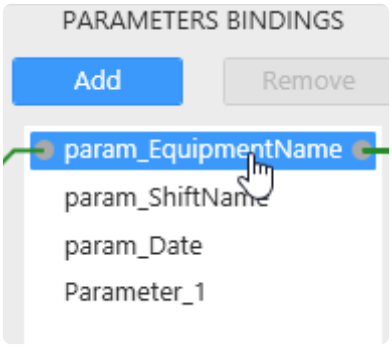
On the right is a list of Listener Widgets : this is a destination that will update according to the value.

Last modified: May 24, 2019

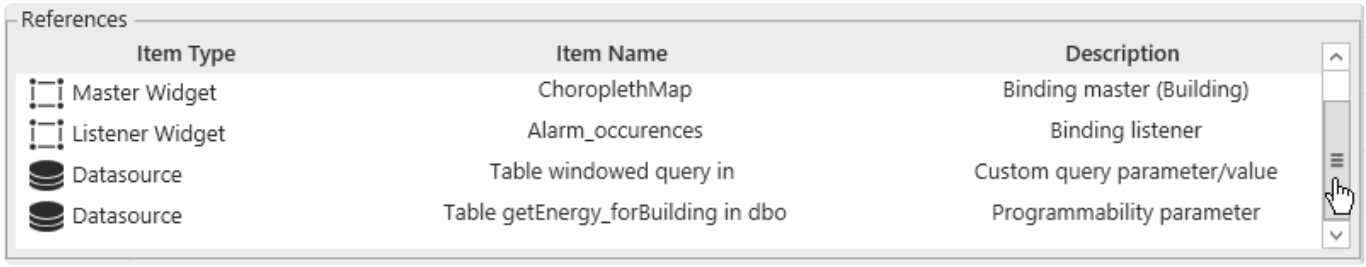
2.9.2.2. Visualizing Parameter binding

See the references to a Parameter

There are many ways to use a Parameter (see the following chapters).
To see a summary of all p laces where a Parameter is used, click on the Parameter name under the “Parameters Bindings” list :



The list at the bottom of the dialog shows all the references to this Parameter :



Each reference is described as :

- *Item Type* : type of item referencing the Parameter
- *Item Name* : name of the item referencing the Parameter
- *Description* : description of the reference

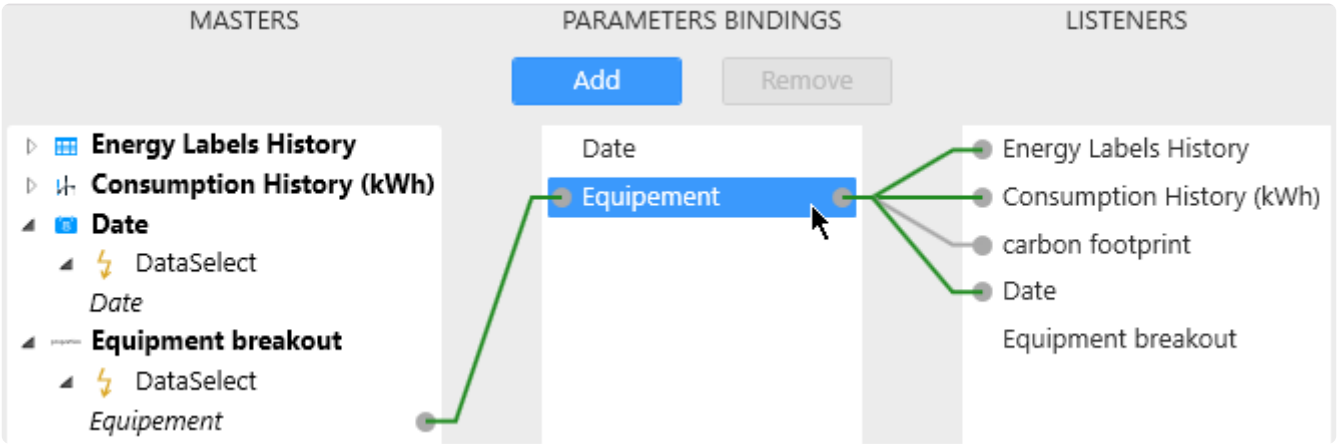
The types of references can be :

- *Binding master (FieldName)* : the Parameter is receiving values from the field “FieldName” of a Master Widget
- *Binding listener* : the Parameter is sending filter values to a Listener Widget
- *Custom query parameter/value* : the Parameter is being used inside a Custom Query
- *Programmability parameter* : the Parameter is being sent as argument to a Stored Procedure
- *Expression argument/value* : the Parameter is being used as a value inside an Expression
- *Data main filter* : the Parameter is being used as a filter value for a Data Source
- *Widget main filter* : the Parameter is being used as a filter value for a Widget
- *Query time parameter* : the Parameter is being used for the time settings of a Historian Query

- *Column conditional formatting* : the Parameter is being used as a comparison value in a Grid Widget's conditional formatting
- *Label content* : the Parameter is being used as a text value to replace text inside a Label Widget, or to replace part of the URL of an Embed Widget
- *Widget header* : the Parameter is being used as a text value to replace part of a Widget's header
- *PDF document selection* the Parameter is being used to select one of multiple PDF files inside the PDF Widget

See the binding to a Parameter

To see which Widgets are connected to a Parameter, click on the Parameter name under the "Parameters Bindings" list :



List of Parameters

To the middle is a list of dashboard Parameters.

Master binding

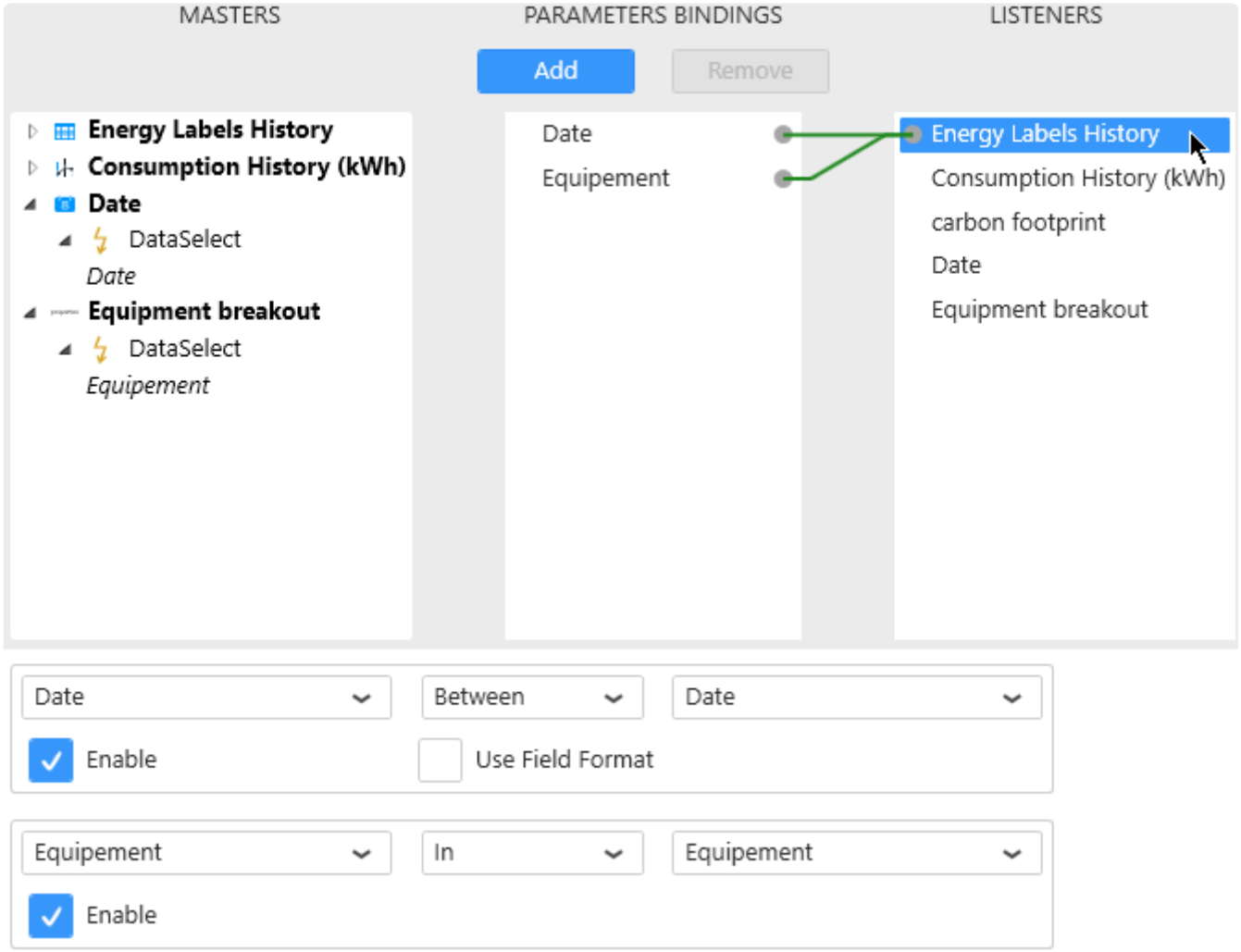
To the left are the links to Master Widgets : this is the Widgets that can **send a value** into the Parameter.

Listener binding

To the right are the links to Listener widgets : this is the Widgets that will **receive a value** from the Parameter.

See Parameter-Listener relationship

To see how a Listener Widget will become filtered, click on the Listener Widget :



The link to Parameters is displayed in the middle.

At the bottom, is displayed a list of all filter actions configured for each Parameter :

- on the left : the field name from the Data Source
- on the middle : a boolean comparison operator
- on the right : the Parameter name
- at the bottom left : a checkbox to disable this filtering
- at the bottom right : an optional formatting applied to the Data Source field

Last modified: Aug 26, 2020

2.9.2.3. Editing Parameter bindings

To edit Parameters bindings, see the following chapters.

Last modified: May 24, 2019

2.9.2.3.1. Necessary connections

No connection is absolutely necessary :

- a Parameter that is not linked to a Master Widget can still receive values from a URL Parameter.
- a Parameter that is not linked to a Listener Widget can still send values into a Stored Procedure, a Custom Query Table, an Expression Value, a filter condition, …

Multiple connections are possible :

- a Parameter can receive values from several Master Widgets
- a Parameter can send values into several Listener Widgets

Last modified: May 24, 2019

2.9.2.3.2. Removing a connection

Why remove a connection ?

By default, a Filter Widget becomes connected to all Listener Widgets using the same Data Source. Sometimes, you only want to filter some Widgets and not others.

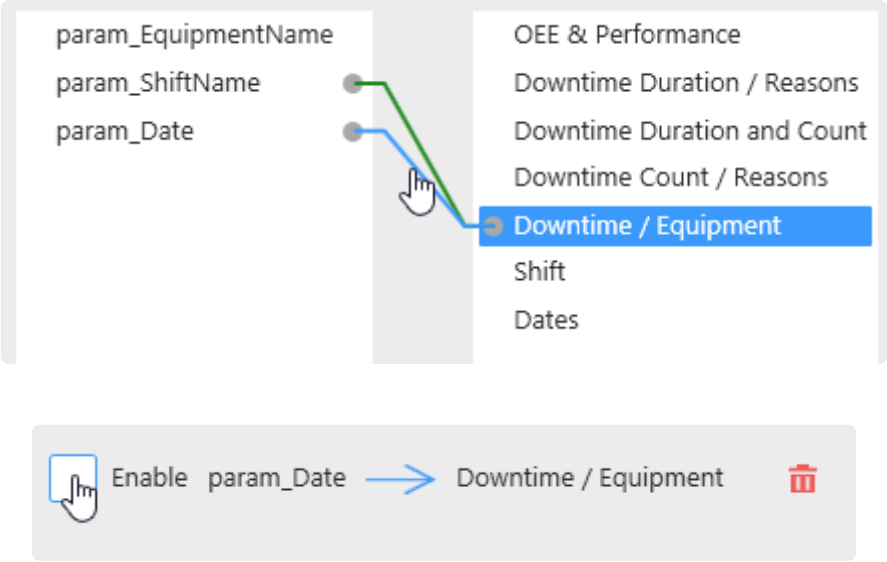
This can also be useful for sending values into a Parameter without Listener Widgets :
The Master Widget sends a value into the Parameter, and the Parameter is used as Stored Procedure parameter for example.

How to disable a connection

You can disable a connection without removing it.

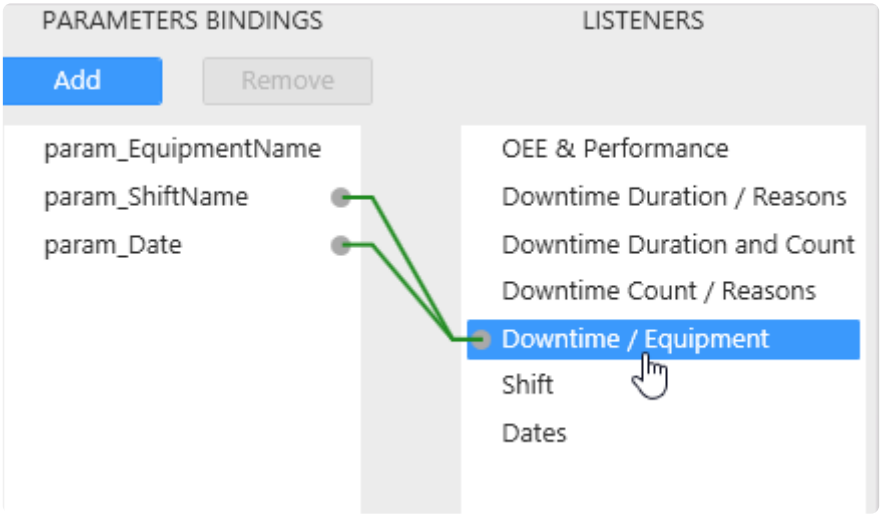
From the connection

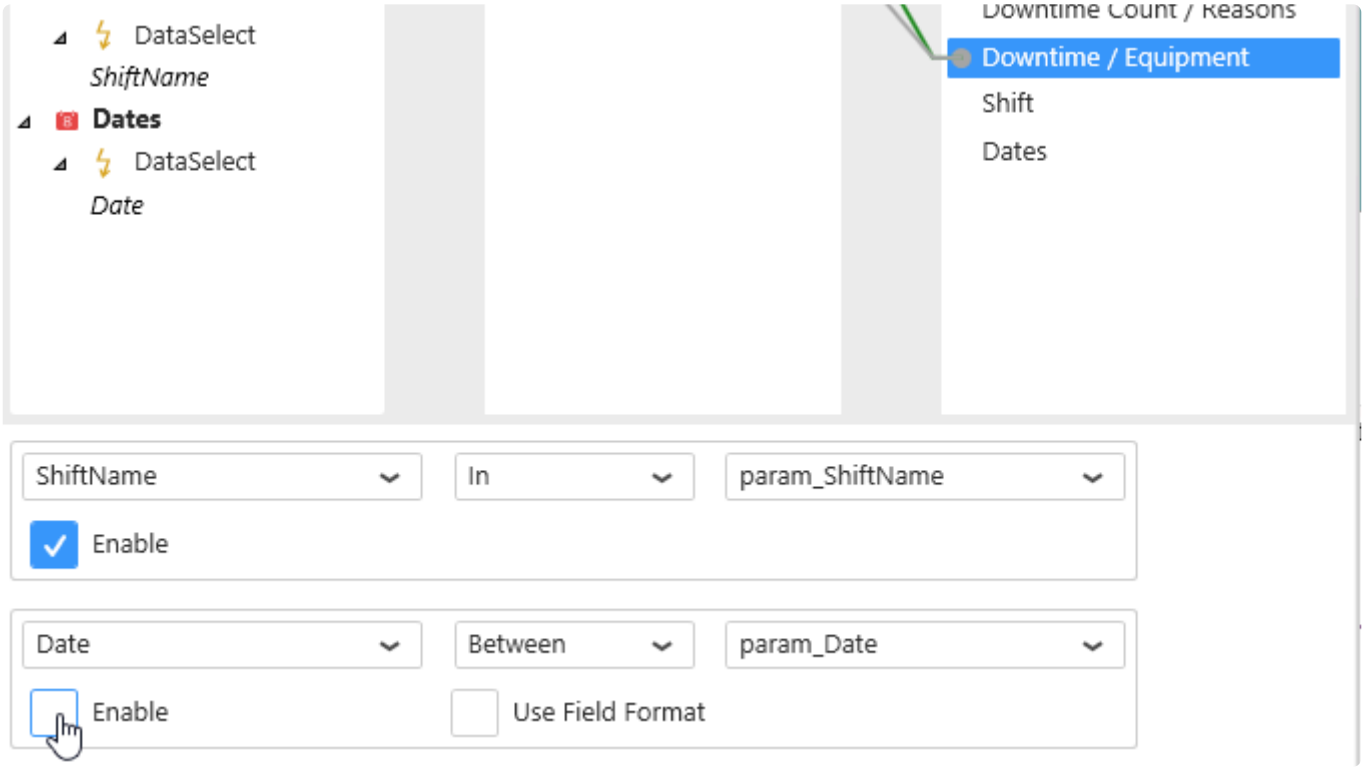
Click on the connection, and un-check “Enable” :



From the Listener Widget

Alternately, you can click on a Listener Widget, and un-check “Enable” on the corresponding relationship :



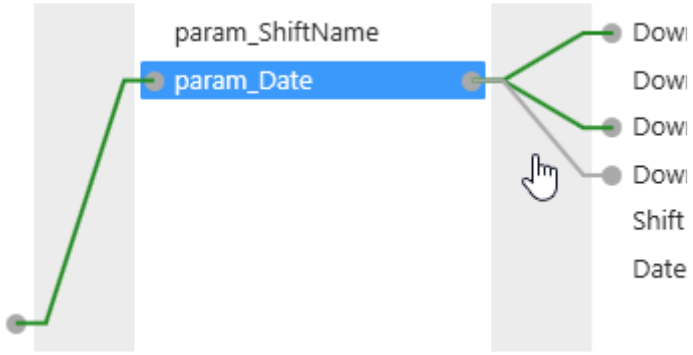


Re-enable connection

Similarly, you can re-enable the connection by checking again “Enable”.

See disabled connections

A disabled connection is displayed as a greyed-out link :



How to remove a connection

To remove a connection, click on it, then click on the red bin button :

Last modified: May 24, 2019

2.9.2.3.3. Adding a connection

Why add a connection ?

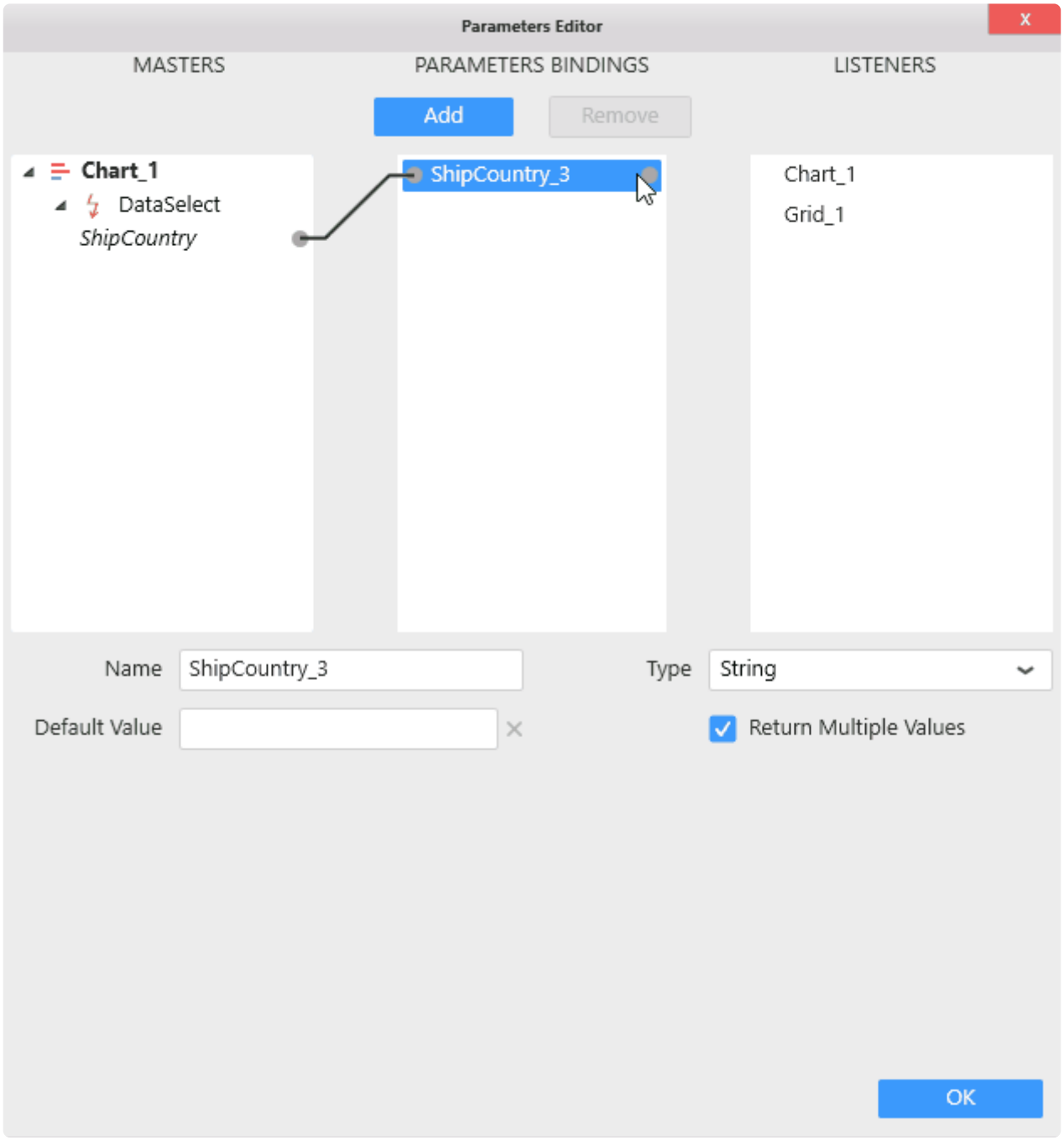
This can be useful for **filtering across Data Sources** :

Only Widgets from the same Data Source are linked by default.
If you wish to use a Master Widget from DataSource_A to filter a Listener Widget from DataSource_B, you have to manually define the relationship : *what field in DataSource_B will become filtered by the Parameter and how ?*

This can also be useful for sending values into a Parameter without Listener Widgets. The Parameter can be used as Stored Procedure parameter for example.

How to add a connection

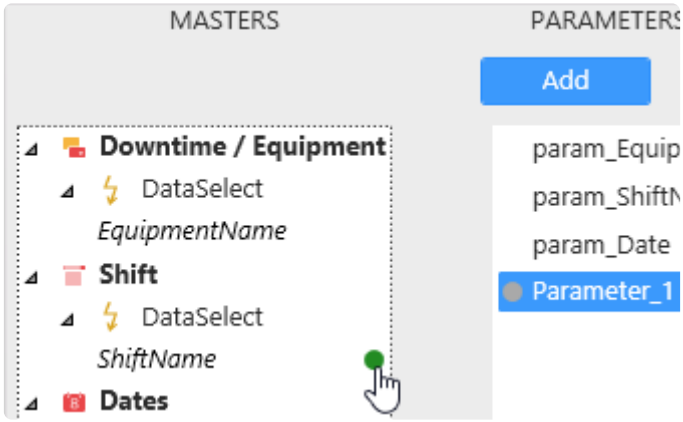
To add a connection, click on the right dot of the desired field to link from, and drag to the left dot on the field to be linked to :

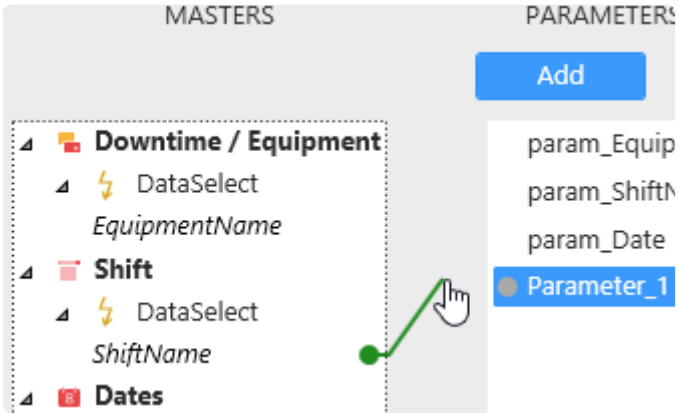


A connection can only be created from left to right.

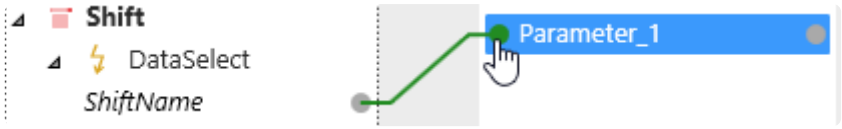
Connecting a Master to a Parameter

Browse the Master Widget on the left. Unfold the tree to find the desired DataSelect event (the name of the event is the name of the Data Source field selected).
Hover the DataSelect event. A grey dot appears on the right of the event name, this is where you will click and drag to draw a link :





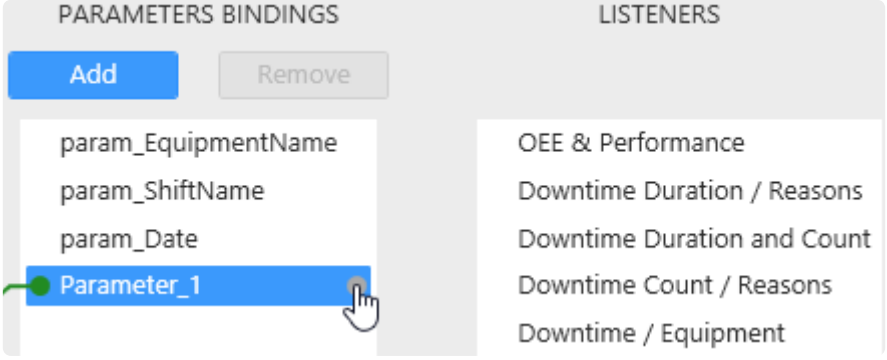
Complete the link into the grey dot on the left of the destination Parameter :

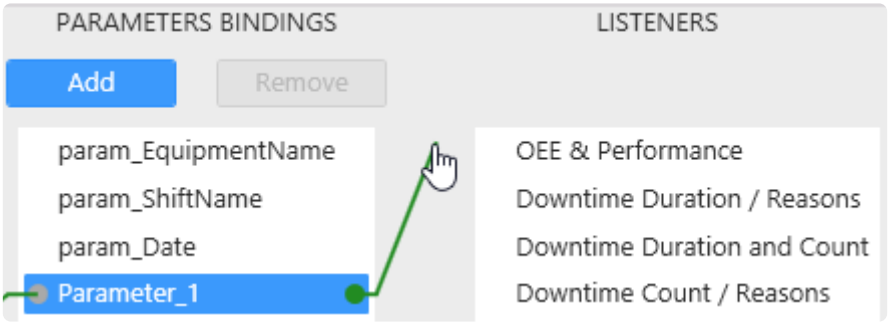


The Master Widget is now linked to this Parameter :
When a user will generate a DataSelect event for this field, the corresponding data value(s) will be sent into the Parameter.

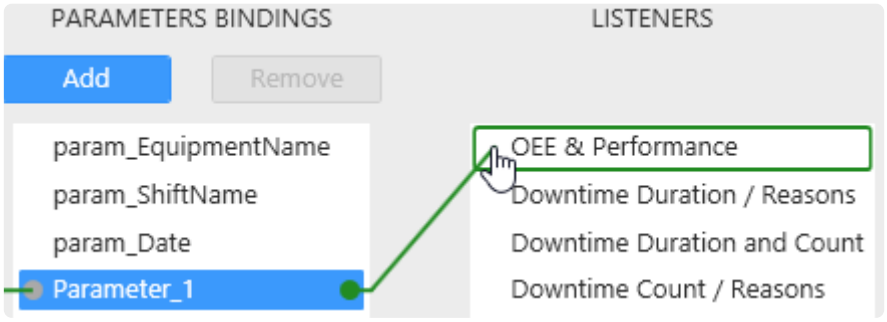
Connecting a Parameter to a Listener

Select a Parameter. A grey dot appears on the right of the Parameter name, this is where you will click and drag to draw a link :

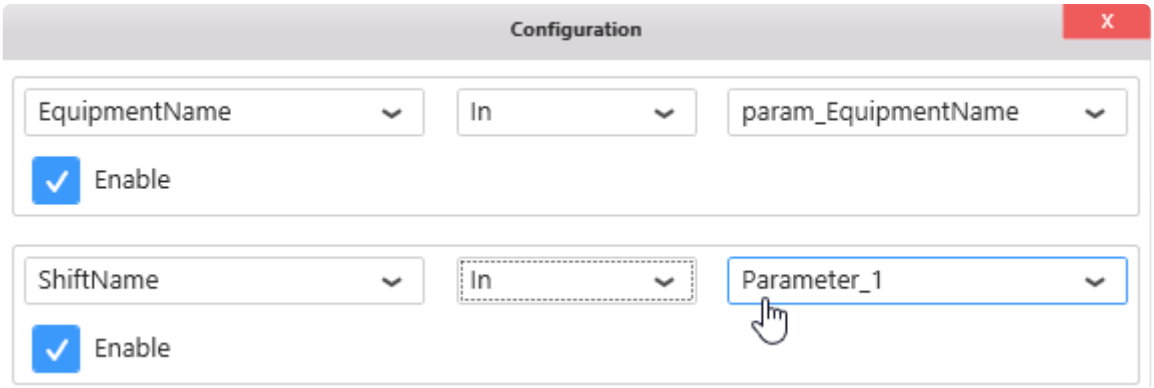




Complete the link into the grey dot on the left of the destination Listener Widget :



Upon completing the link, a relation ship editor appears to let you set the relationship between the Parameter value and the Data Source fields of the Listener Widget :



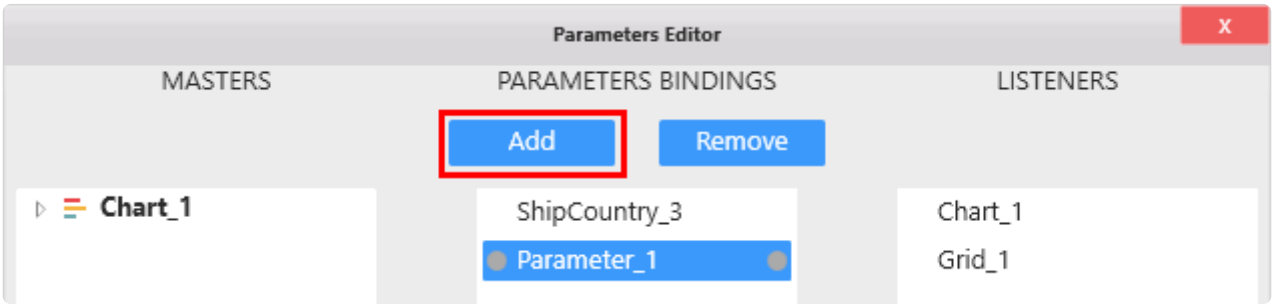
Set up the desired relationship, and close the dialog with the red cross.
The relationship should read like a SQL statement. For example here : `WHERE ShiftName IN( <values of Parameter_1>)`

Last modified: May 24, 2019

2.9.2.4. Adding/Removing Parameter

Manually Adding a Parameter

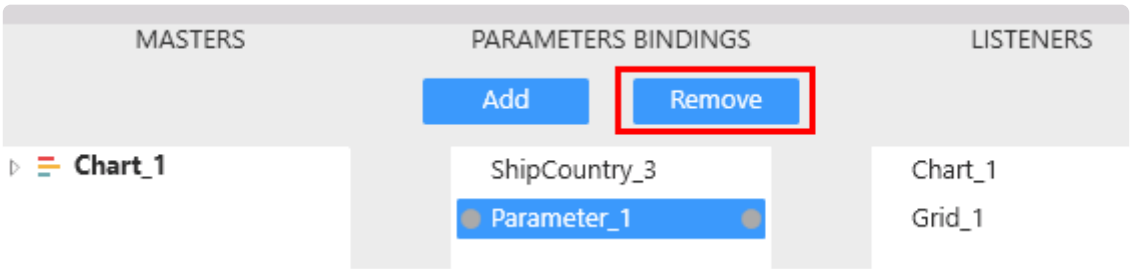
To add a Parameter simply click on the Add button under “Parameter Bindings” :



The new Parameter is added, with no links and default settings.

Manually Removing a Parameter

To remove a Parameter, select the Parameter and click the *Remove* button :



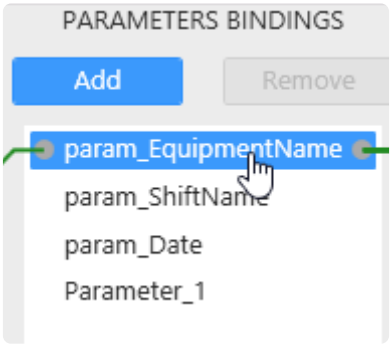
If the Parameter was used in the Dashboard, then the following happens :

- for Listener Widgets, the Parameter its filter condition are removed
- if it was used in a Stored Procedure or a Custom Query, then it will be replaced by its default value, in order to maintain correct SQL syntax in those objects.

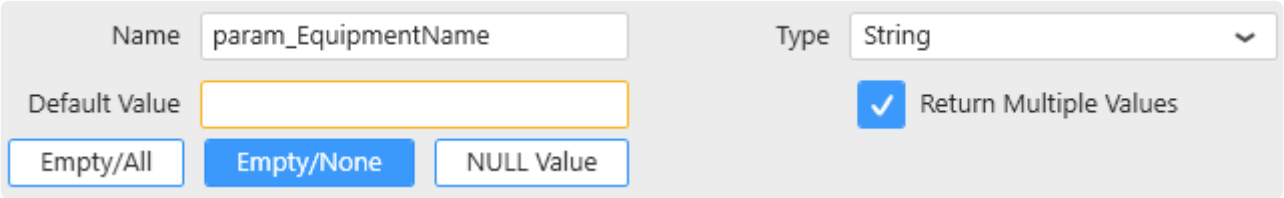
Last modified: Jun 19, 2020

2.9.2.5. Edit Parameter properties

To edit a Parameter, click on it in the middle list :



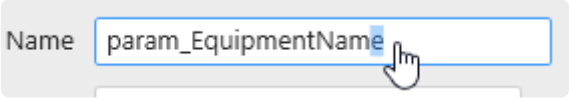
The Parameter properties are editable at the bottom :



Last modified: Jan 22, 2020

2.9.2.5.1. Parameter Name

You can give a custom name to the Parameter by using this text box :



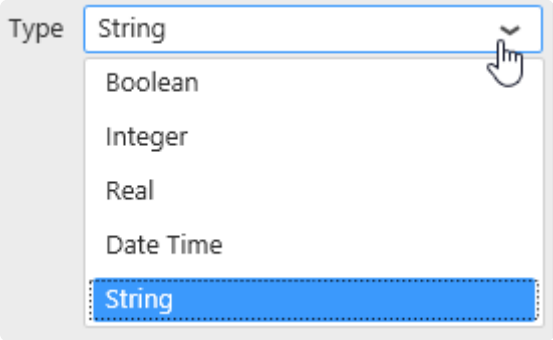
This is especially useful when using the Parameter as URL Parameter : the Parameter name will be used in the syntax to pass values to the Dashboard from the URL.

Parameter names are also listed in the Filter Overview at runtime.

Last modified: May 24, 2019

2.9.2.5.2. Value Type

To set the Type of the Parameter, select it from the “Type” listbox :



Values sent into this Parameter will be converted to this Type.

Last modified: May 24, 2019

2.9.2.5.3. Parameter Default Value

The value that the Parameter has by default is called the *Default Value*.
This is the value that the Parameter has before any manual filtering at runtime for example.

Setting Default Value from URL Parameters

URL Parameters actually replace the Default Value when the Dashboard loads.

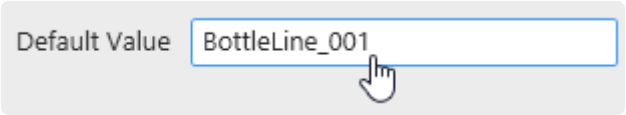
Setting Default Value from Widgets

It is possible to set a Parameter Default Value by making a selection in the Widget Preview of one of its Master Widgets.

This is explained in the corresponding chapter.

Setting Default Value manually

Default Values can be set manually from the “Default Value” text box :

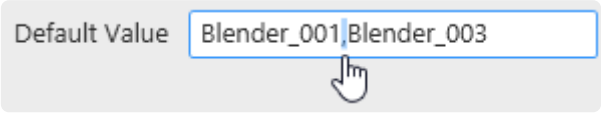


Text or number Default Values

Text or number default value can be set manually by writing the value in the text box.

Multiple Default Values

Multiple Default values can be set using a comma , as separator :

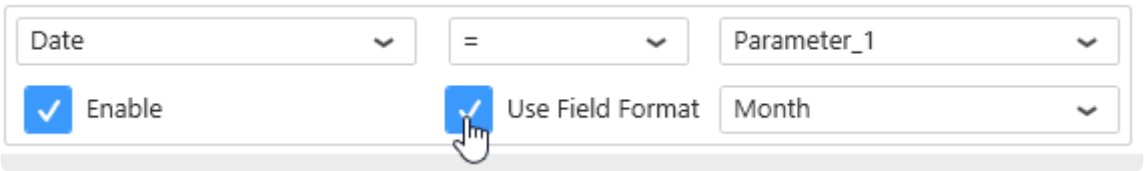


Date Default Values with Format

For Date type values, it is possible to use a date *Format*.
This will behave similarly as Data Source Filters on formatted date fields.

Example :

In the Listener Widget relationship, use format “Month” :



Set the Parameter as Integer and in the *Default Value* text box, you enter `2`.

⇒ By default, the month of February (month number `2`) is filtered.

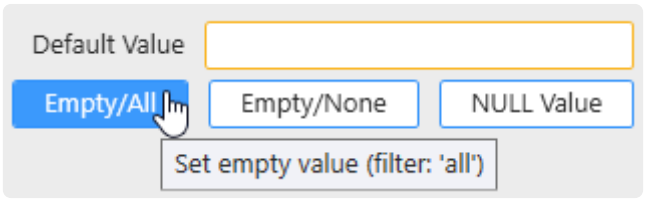
Special Default Values

A Default Value with “no content” could actually have different meanings :

- don’t filter (select all items in the Listener Widgets)
- don’t select anything (select nothing in the Listener Widgets)
- select explicitly the `NULL` value
- select explicitly the empty string `""`

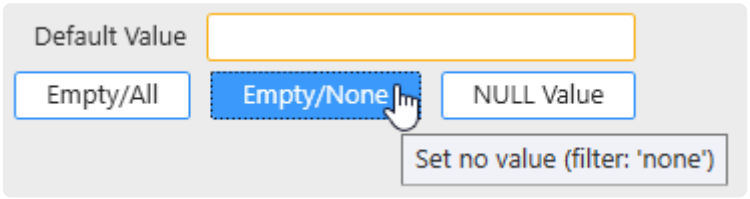
Select All

If you want the Parameter by default to not filter anything (select all values in the Listener Widgets), click the button *Empty/All* :



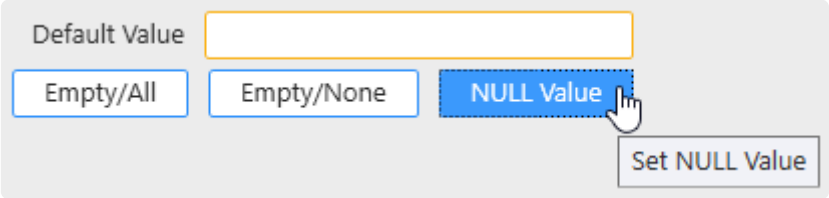
Select None

If you want the Parameter by default to remove every value (select no corresponding value in the Listener Widgets), click the button *Empty/None* :



Select NULL

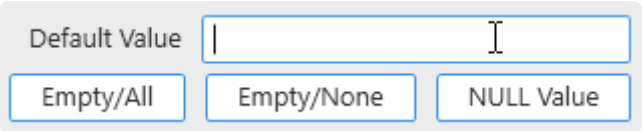
If you want the Parameter by default to select the `NULL` value (`NULL` in SQL-like Connections, empty cells in Excel, etc), click the button *NULL Value* :



Empty string

> ! Erasing the content of the *Default Value* text box does not remove the Default Value ! Actually it sets the Default Value to be the empty string "".

To let the Parameter have the empty string "" as Default Value, erase all text content :



> ✱ Note : the buttons for special values are all white (unselected)

Last modified: Jan 22, 2020

2.9.2.6. example : Multi-Data Source filtering

Multi-Data Source filtering

You can bind Parameters to any Listener Widget independently of its Data Source and independently of the corresponding field name.

> This can be useful if you want to use for example a single date filter to apply to several Listener Widgets from different Data Sources.

Thus, you will avoid duplication and have a consistent filtering.

To achieve this, the Master Widget must be able to send the required values to the Parameter.

Example scenario 1

For example if DataSource1 has fewer values than DataSource2 (maybe a narrower date range, maybe not all equipments appear in this equipment list), then using the field from DataSource1 in the Master Widget will render you unable to send those missing values to the Parameter.

To solve this, you can use the field from DataSource2 which contains the most values (the widest date range, or the complete equipment list).

Note however that in this case you will try to filter Widgets from DataSource1 which may not contain the values and will return empty results.

Example scenario 2

If DataSource1 has some values that are not in DataSource2, **and** the other way around, then you cannot list all possible values in a Filter Widget by binding a field from one of your existing Data Source directly.

To solve this, you can create an additional Data Source dedicated to contain all the filter values you need to use. It could be a Custom Query with an UNION of both data sets, or it could come from one of your reference tables listing all equipments for example.

Note, once again, that when you will filter for values that exist only in one of the Data Sources, the Widgets bound to the other Data Source will return empty results.

Last modified: May 24, 2019

2.9.3. Widget as Master Filter

Alpana is dedicated to data exploration and interactive display of information.
Using a filter at runtime allows the user to focus the visualizations on a sub-set of the data.
Dashboard Parameters are the mechanism behind this dynamic filtering.

Dynamic filtering : Master Widget / Listener Widget

A ***Master Widget*** can be used at runtime to make selections that **apply a filter** to other Widgets.
Widgets that are configured to listen to this filter are called *Listener Widget*.

Internals

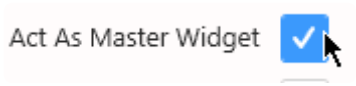
When the user clicks on a shape in a Master Widget, a DataSelect event is generated and sends the selected value into a Dashboard Parameter.
Listener Widgets receive the Dashboard Parameter value and add it as a filter to their data.

Last modified: May 24, 2019

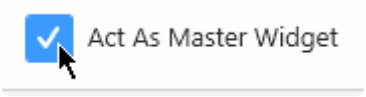
2.9.3.1. Configuring a Widget as Master

Configuration

To turn a Widget into a Master Widget and allow it to filter other Widgets, check *Act As Master Widget* under *Filter* :



There is also a shortcut in the top bar :



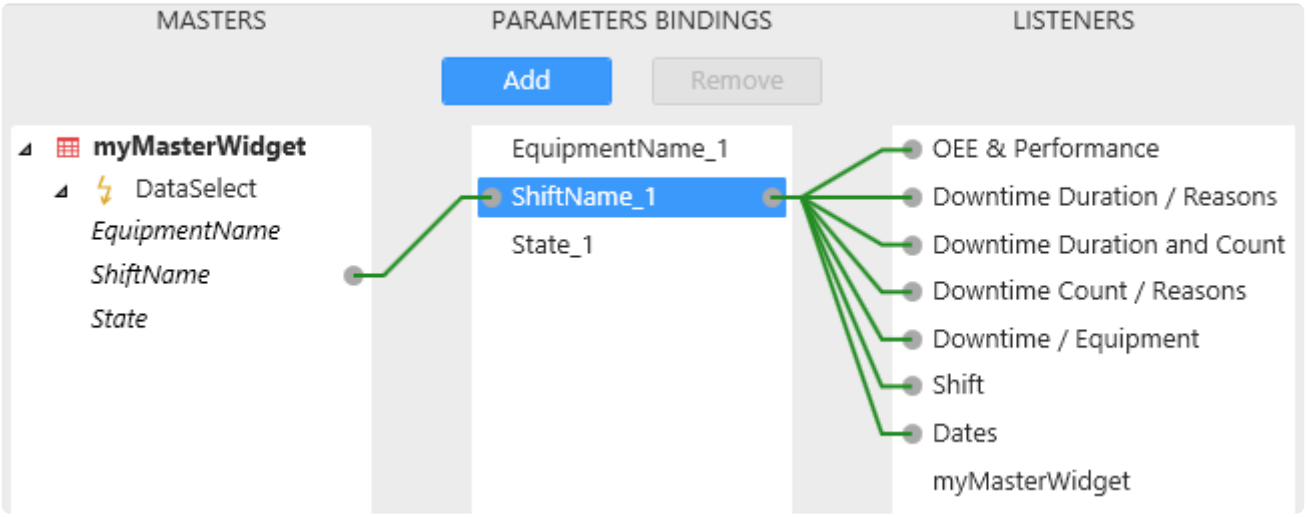
Master Widget by default

Some widgets do not require to perform this setting manually : all Widgets in category “Filter” are configured as Master Widget by default.

Internals

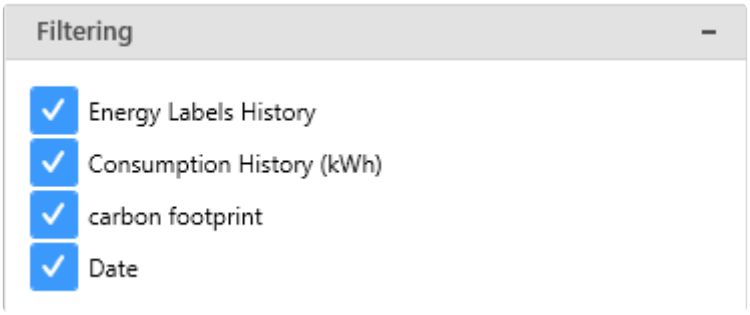
This will automatically :

- make the Widget and its DataSelect events available as “Master” to the left column in the “Parameters Editor” dialog
- generate as many Dashboard Parameters as there are DataSelect events, and name them like `FieldName_index` (example : `Equipment_1`)
- bind the events as Master to the corresponding Parameters
- bind to these Parameters all the Listener Widgets that use the same Data Source as the Master Widget

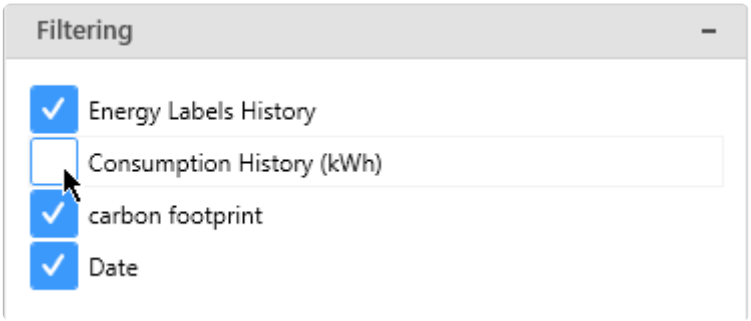


Viewing filtered widgets

When the Master Widget configuration is active, a new section appears at the bottom of the data panel on the left, called *Filtering* :



This is a list of Listener Widgets that are bound to this Master Widget.
The checkboxes are a shortcut to disable the Parameter linking to that listener.



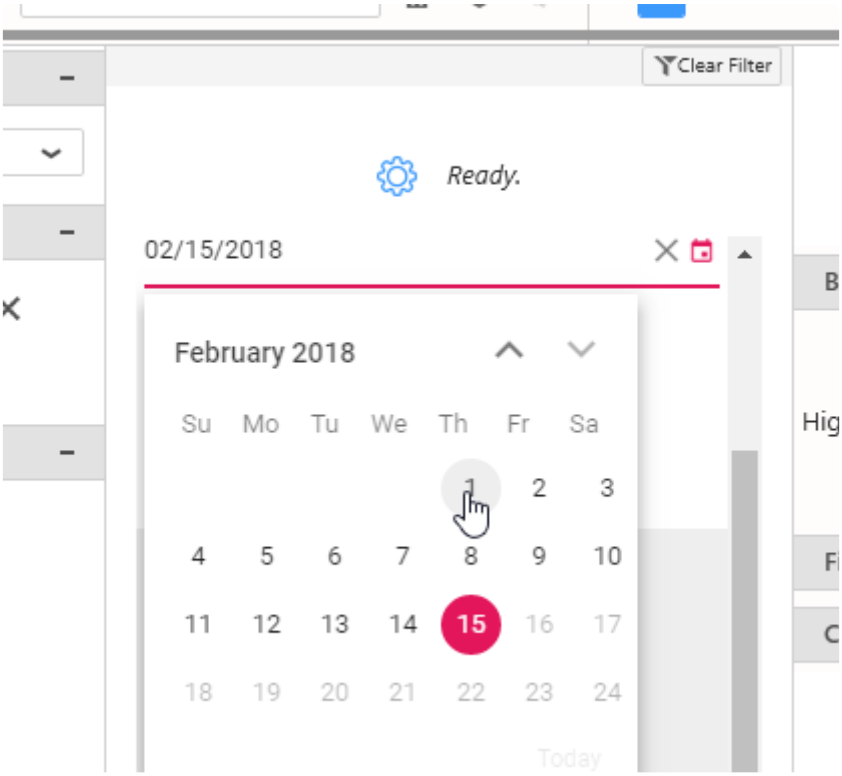
Last modified: May 24, 2019

2.9.3.2. Setting a Widget Default Selection

The Master Widget can be configured to generate a selection by default.
This means that every time the Dashboard will load, the Widget will send this default selection into the Parameter Default Value.

Setting a static default selection

In Alpana Designer, use the Widget preview in the middle part of the view to select the desired default value :



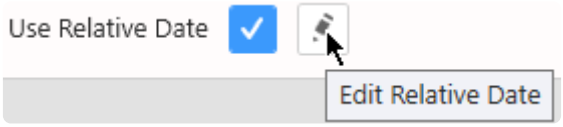
Default selection as Relative Date

In addition, some widgets (date/time filter widgets) allow to configure a **relative date range as default selection**.

> Note :
> This is different from the notion of Widget data filtered based on a Relative Date.
> If you have a Widget that needs to have its data filtered based on a relative date, see instead the chapter on Widget-level Filtering.

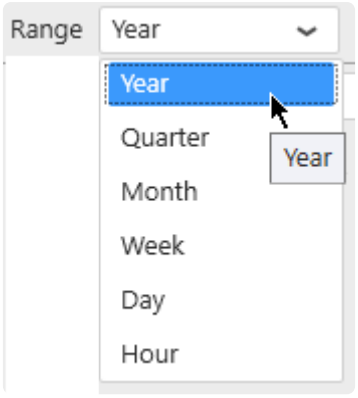
Enabling Relative Date for the Default selection

For this, check *Use Relative Date* under Basic Settings and click the pen icon to configure :

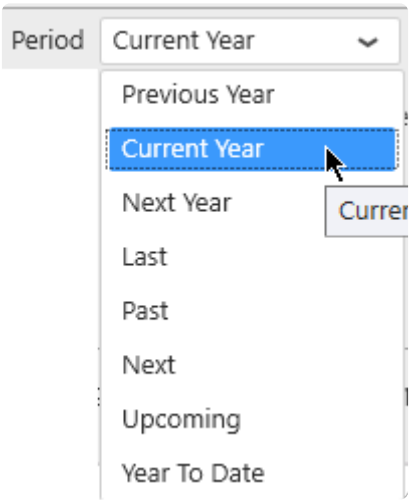


Configuring a Relative Date Period

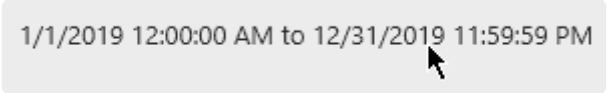
Then for each time range :
Year, Quarter, Month, Week, Day, Hour



… a Period can be selected :
Previous 1, Current, Next 1, Last N, Past N, Next N, Upcoming N, To Date



A preview of the selected date range for the current time appears at the bottom of the window to help you configure :



Last modified: May 24, 2019

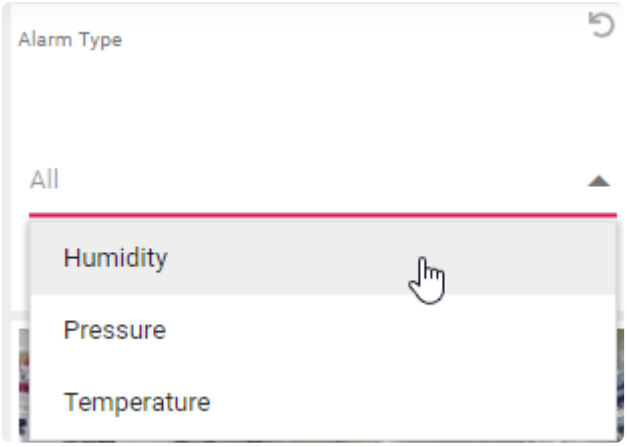
2.9.3.3. Using filters at runtime

Default values

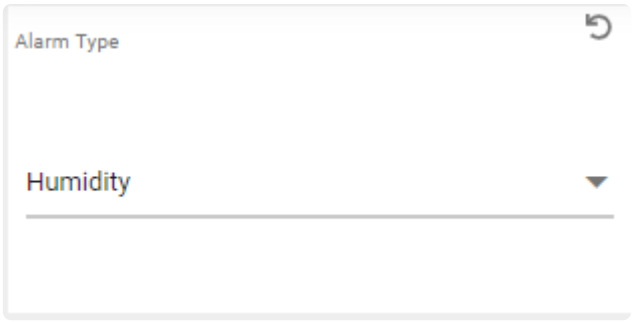
When a Dashboard loads, all Parameters have their Default Value.
It may be empty (by default) or configured with values (see above in the Managing Parameters chapter).

Applying a filter

At runtime, clicking a shape in the Widget applies a filter and changes the corresponding Dashboard Parameter values :

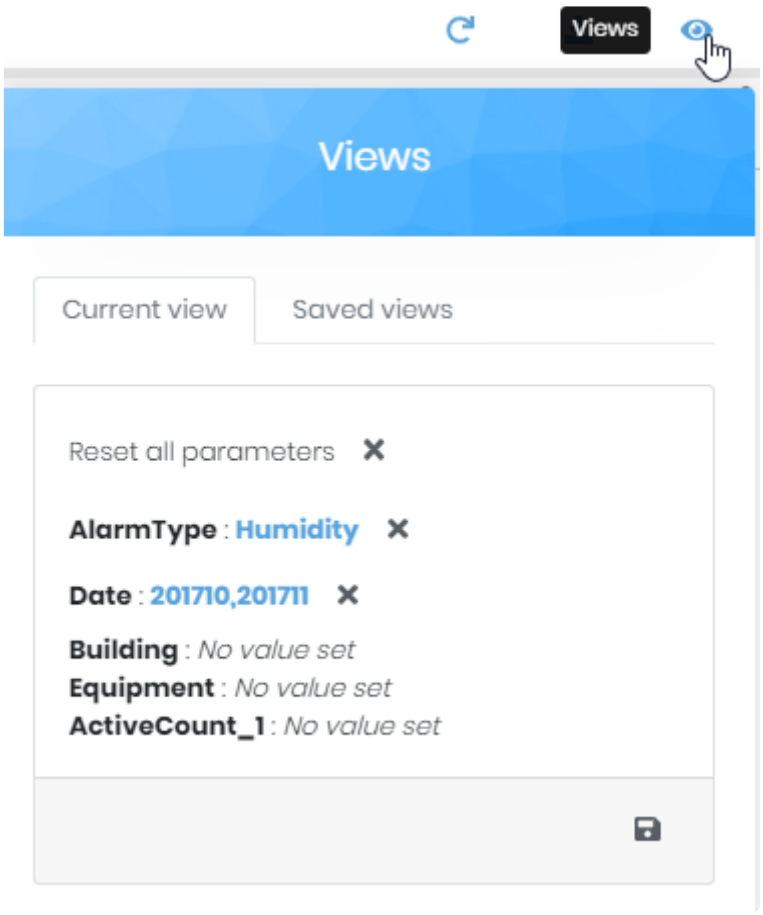


A reset icon appears in the title bar of the Widget to indicate that the Widget is currently applying a filter :



Filters Overview

When filters are applied, the state of Dashboard Parameters can be listed by clicking *Views* on the top right of the Dashboard :



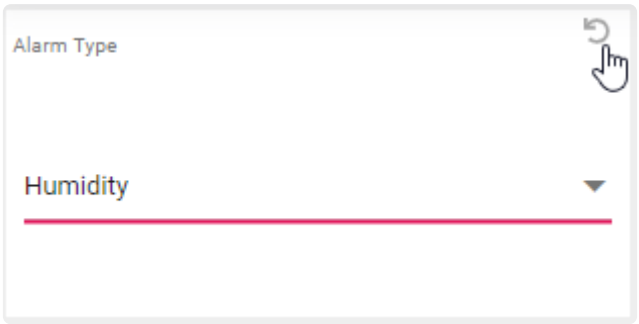
This shows a list of Parameter names and corresponding values.

Clearing filters

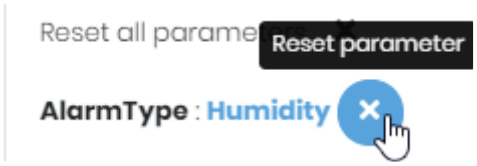
Parameters can be reset to their default values to remove the filter effect.

Clearing single filter

To clear a single filter, click on the reset icon in the title bar of the Widget :

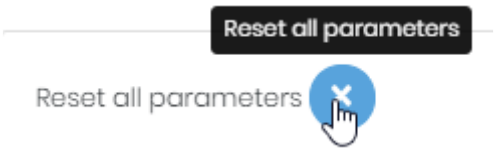


You can also click on the cross on the corresponding Parameter in the *Views* :



Clearing all applied filters

To clear all applied filters at once and reset all Parameters to their default value, open the *Views* and click *Reset All Parameters* :



Last modified: May 24, 2019

2.9.4. URL Parameters to receive values

The default value of any Parameter can be set from a **URL Parameter**.

This means that a particular set of filters can be used when opening the dashboard if the correct URL Parameters are configured.

> This can be used to receive context from other tools that will simply link to a Alpana Dashboard and concatenate filter values as context.

See also Wikipedia for more information : https://en.wikipedia.org/wiki/Query_string

Last modified: May 24, 2019

2.9.4.1. Example

Your dashboard contains a Parameter called `Parameter1` and another called `Parameter2` which accepts multiple values.

The dashboard is published at `http://yoursite.com/viewer/dashboard/YourDashboardID`

By adding this :

```
http://yoursite.com/viewer/dashboard/YourDashboardID?Parameter1=ValueA&Parameter2=ValueB&Parameter2=ValueC&Parameter2=ValueD
```

It will give a default value for your Parameters :

- `Parameter1` takes the single value `ValueA`
- `Parameter2` takes the multiple values `ValueB,ValueC,ValueD`

Last modified: May 24, 2019

2.9.4.2. Reserved words

> **Important**
> The following names are reserved and cannot be used as URL Parameters. Please rename the corresponding Parameters.

- Id
- Category
- Comment
- View
- Dashboard
- Tab
- Viewid

Last modified: May 24, 2019

2.9.5. Parameter usage

Parameters are a fundamental concept used for many advanced configurations.
See some examples in the following chapters.

Last modified: May 24, 2019

2.9.5.1. Label Parameters

Parameters can be used as a placeholder inside a text string : the placeholder is replaced dynamically by the Parameter's value.

Syntax

`${parametername}` will be replaced in-place by the values of the Parameter called "*parametername*".

Example :

```
The high limit is ${parametername} minutes.
```

will get replaced at runtime by :

```
The high limit is 123 minutes.
```

Multiple values

Parameters that return multiple values will return them separated with commas.

So that you can write :

```
The date interval is ${multiDateParameter} for this Dashboard.
```

and it may get replaced at runtime by :

```
The date interval is 2020-08-26T14:50:35,2020-08-26T15:50:35 for this Dashboar
d.
```

Note : In Alpana Designer, the text will be displayed with the default parameter values.

Usage

Label Parameters can be used to replace static text in :

- the Label widget
- any widget Header

Last modified: Aug 26, 2020

2.9.5.2. Dynamic calls to Stored Procedures or Table-Valued Functions

Parameters can be used as arguments to **Stored Procedures** or **Table-Valued Functions** : when the Parameter changes, the Stored Procedure or Table-Valued Function is called again with the new values as arguments.

For more information, see in chapter Stored Procedures.

Last modified: May 24, 2019

2.9.5.3. Dynamic calls to Custom Queries

Parameters can be used inside **Custom Queries** : when the Parameter changes, the Custom Query is called again with the new values.

For more information, see in chapter Custom Query Tables.

Last modified: May 24, 2019

2.9.5.4. Dynamic Expression Values

Expressions can use Parameter values as “Value” : this allows to change the calculation depending on the value of the Parameter.

For more information, see in chapter Calculating Expressions.

Last modified: May 24, 2019

2.9.5.5. Dynamic Data Source filters conditions

The value of a **Data Source filter** condition can be a Parameter value : this allows to change the Data Source-level filter depending on the value of the Parameter.

For more information, see in chapter Filtering Data.

Last modified: May 24, 2019

2.9.5.6. Dynamic Widget filters conditions

The value of a **Widget-level filter** condition can be a Parameter value : this allows to change the Widget-level filter depending on the value of the Parameter.

For more information, see in chapter Widget-Level Filtering

Last modified: May 24, 2019

2.9.5.7. Dynamic Historian Queries

Parameters can be used inside Historian queries to let users change the query at runtime : when the Parameter changes, the Historian query gets updated.

For more information, see in chapter Parameter-Based Settings

Last modified: Jun 18, 2019

2.9.5.8. Include Parameter value in a Link to URL

Links To URL can use Parameter values as part of the URL : this allows to include the value of the Parameter inside the URL.

For more information, see in chapter Link to URL.

Last modified: Oct 15, 2019

2.9.6. Widget-specific binding

Some specific Widget types use Parameters in their configuration.
See the following chapters for a reference.

Last modified: May 24, 2019

2.9.6.1. Grid conditional formatting

Grid conditional formatting conditions values can be a Parameter value : this allows to change the conditional formatting depending on the value of the Parameter.

(see in the chapter about Grid conditional formatting)

Last modified: May 24, 2019

2.9.6.2. Dynamically updating embedded content

Embed widget

The URL displayed in the Embed widget can be composed dynamically using a Parameter : this allows to change the displayed content depending on the value of the Parameter.

PDF widget

The PDF file displayed in the PDF widget can be selected dynamically using a Parameter : this allows to change the displayed content depending on the value of the Parameter.

Last modified: Oct 17, 2019

2.10. Dashboard Layout

(new in version 2020 R1)

Widgets are laid out on a bi-dimensional (2D) canvas called a *Layout*.
A single Dashboard can contain **multiple Layouts** that can present the same widgets with a different look-and-feel corresponding to different visualization situations : embedding, viewing on a big screen, etc
By default, there is only one layout called “Default Layout”.

Each Layout automatically contains 3 views :

- *Desktop* : this presentation is displayed when viewing from a desktop web browser
- *Mobile* : this presentation is displayed when viewing from the Alpana Mobile app
- *Pdf export* : this presentation is used when exporting the Dashboard to a PDF report

For each layout, you can configure different Layout-level options : theme, background options, …

For each view of each layout, you can configure the layout of Widgets differently :

- any widget may be displayed in a different position and size
- any widget to be visible or hidden

This is only a configuration of the presentation, and is all configured within the same Dashboard, with the same Widgets already configured, and still using the same Data Sources.

Last modified: Jun 17, 2020

2.10.1. Managing Layouts

By default, only 1 layout configuration exists : `Default Layout`
Its properties can be configured as described below, and it contains 3 different views for *Desktop*, *Mobile* and *Pdf exports*.

Adding a Layout

Click on the *New Layout* button in the top bar to add a new layout to the Dashboard :

By default, it is a copy of the current Layout.

Alternately, you can also click the *Duplicate* button from the desired Layout's properties to duplicate it :

Deleting a Layout

To delete a Layout, click the *Remove* button from its properties :

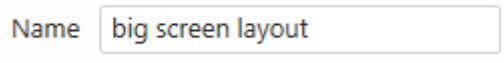
> Note : the Layout configured with the *Is Default Layout* property cannot be deleted

General Layout properties

The Layout's properties can be opened by clicking the *Edit Current Layout* button in the top bar :

Name

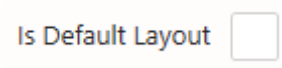
The Layout's *Name* can be used to identify it when you have multiple Layouts :



Default Layout

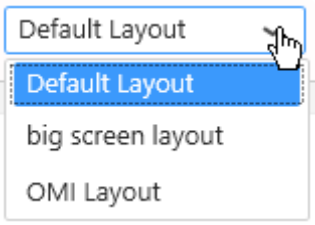
The *Is Default Layout* option defines which Layout will be displayed by default :

- when viewing the Dashboard from a desktop web browser
- when viewing the Dashboard from the Mobile app
- when exporting to PDF
- when opening the Dashboard in the Designer



Selecting a Layout in the Designer

To select which Layout you want to edit in the Designer, use the box in the top bar :



Viewing a layout at runtime

By default at runtime, the Default Layout is displayed.
If you want to display other layouts, you can use the URL Parameter `layout`.
For example, `(your dashboard URL)?layout=my_big_screen_layout` will display the layout called “my_big_screen_layout”.

Last modified: Jun 17, 2020

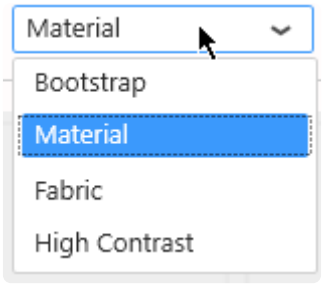
2.10.2. Dashboard Theme and Background

The default visual style for runtime widgets and controls is defined by the currently applied **Theme**.

> This Theme is Layout property and will only apply to the current Layout of the Dashboard (and all its contained Widgets).

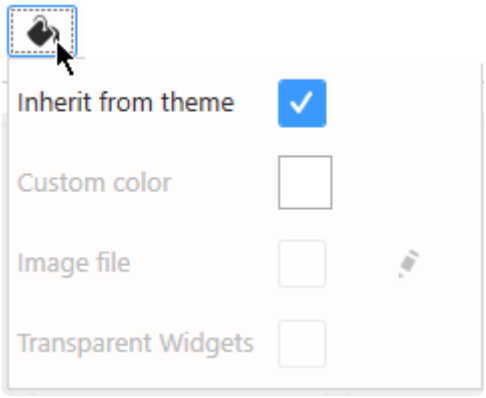
Selecting a Theme

Use the theme Selector in the Layout properties :



Background Settings

The background settings are inherited from the currently selected Theme.
This can be overridden by using the *Background* button in the Layout properties :



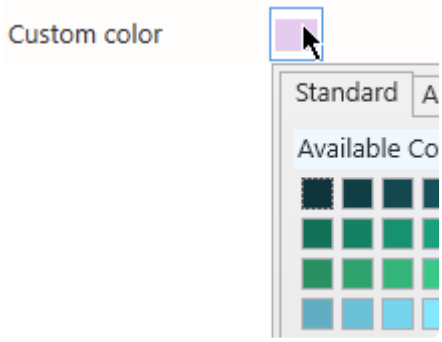
Inherit from theme (default)

The background style is defined by the currently selected Theme.

Un-select this option to allow to customize the following options.

Custom Color

The Background Color can be configured manually using a color picker :



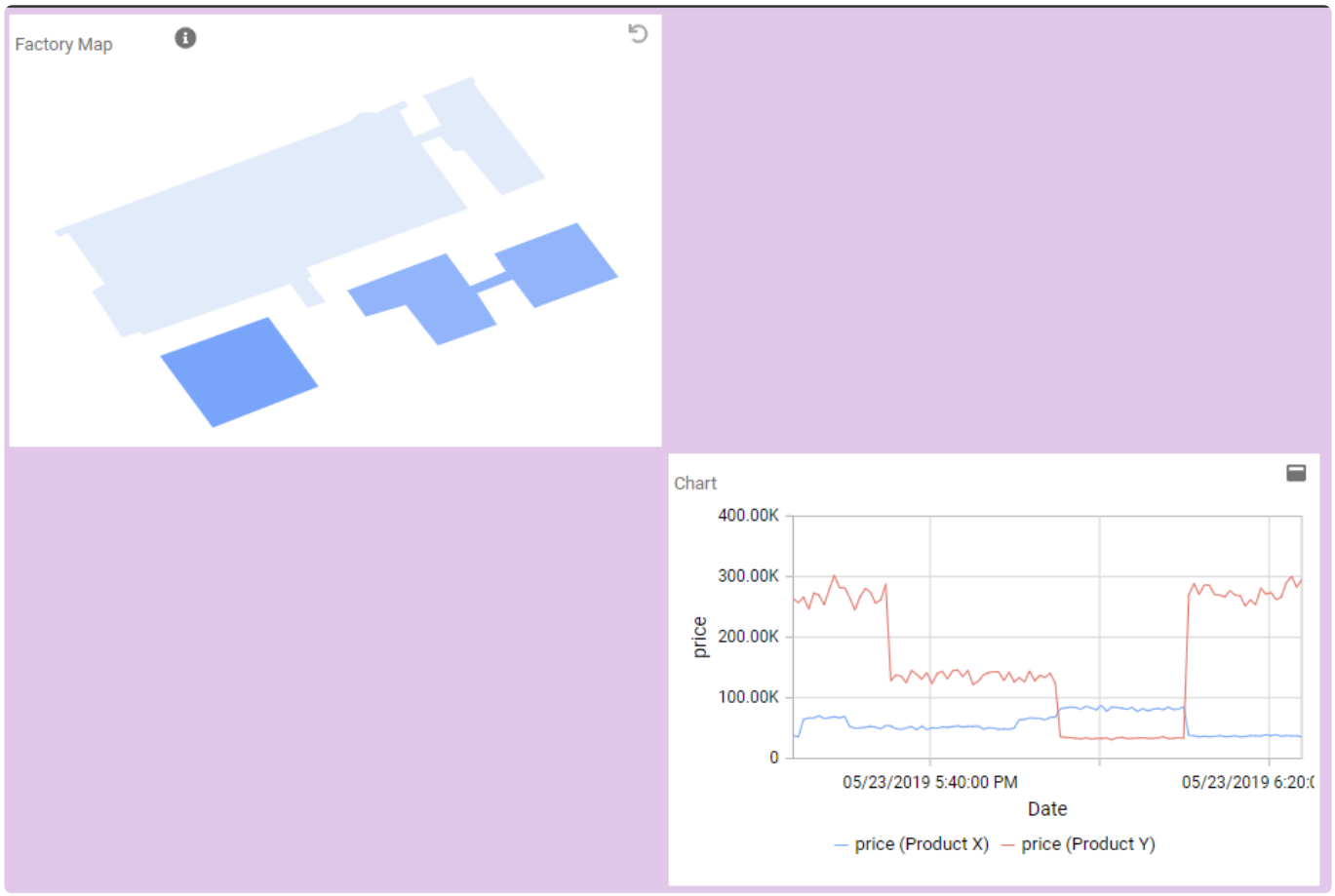


Image file

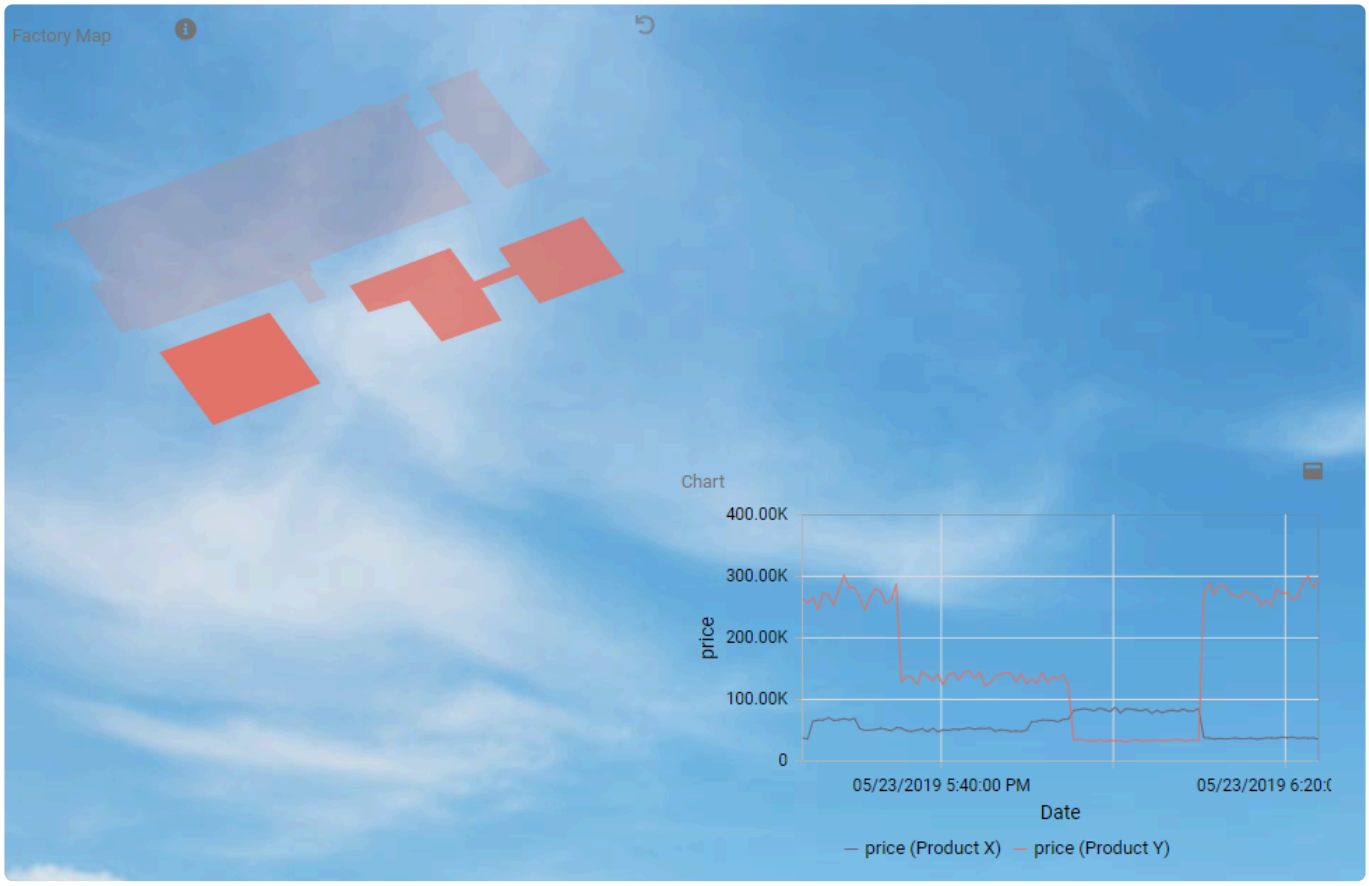
Instead of a plain background color, you can use an image as background.
Click on the pen icon to select a file.

> The maximum size accepted is 1.00Mb
> Supported formats are : `*.png`, `*.jpg`, `*.bmp`, `*.jpeg`, `*.gif`, `*.emf`, `*.tiff`, `*.tif`, `*.ico`

Transparent Widgets

Even when configuring a different background color for the dashboard canvas, the Widgets' background is still inherited from the Theme.
This is in order to maximize readability.

However, you may select *Transparent Widgets* to use the custom background (color or image) as widget background too :



Last modified: Jun 17, 2020

2.10.3. Layout size and Responsiveness

Layout Grid

To help for easy alignment and page composition, a layout grid is displayed in Alpana Designer. This grid is drawn in Alpana Designer only to guide the user, but doesn't appear at runtime.

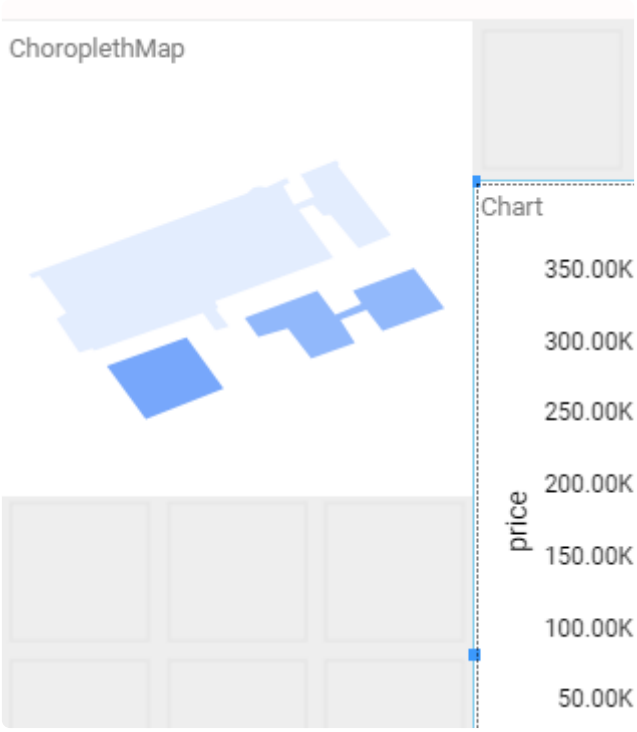
When moving and resizing Widgets, they will snap to this grid.

Grid Resolution

By default, the layout grid shows 12 grid tiles horizontally. (this corresponds to 6 vertically on most screens).

This *Grid Resolution* can be configured in the Layout properties :

For example, setting a higher resolution allows for finer Widget alignments and sizes :



Vertical size

Since Alpana v3.0.0, the layout size is not limited vertically : as long as you add Widgets at the bottom, the web page will be scrollable for any length as desired.

Responsiveness

This bi-dimensional layout allows to dispose Widgets horizontally and vertically.

On many systems this is a comfortable way to view Dashboards.
On systems with larger or smaller screens, the layout gets stretched, and maintains a consistent appearance, with some widget elements resizing and gaining/losing details depending on the widget's own responsiveness.

However, when the screen is very narrow (and this is especially the case on mobile devices), the displayed layout will switch from the *Desktop* Layout view to the *Mobile* Layout view.

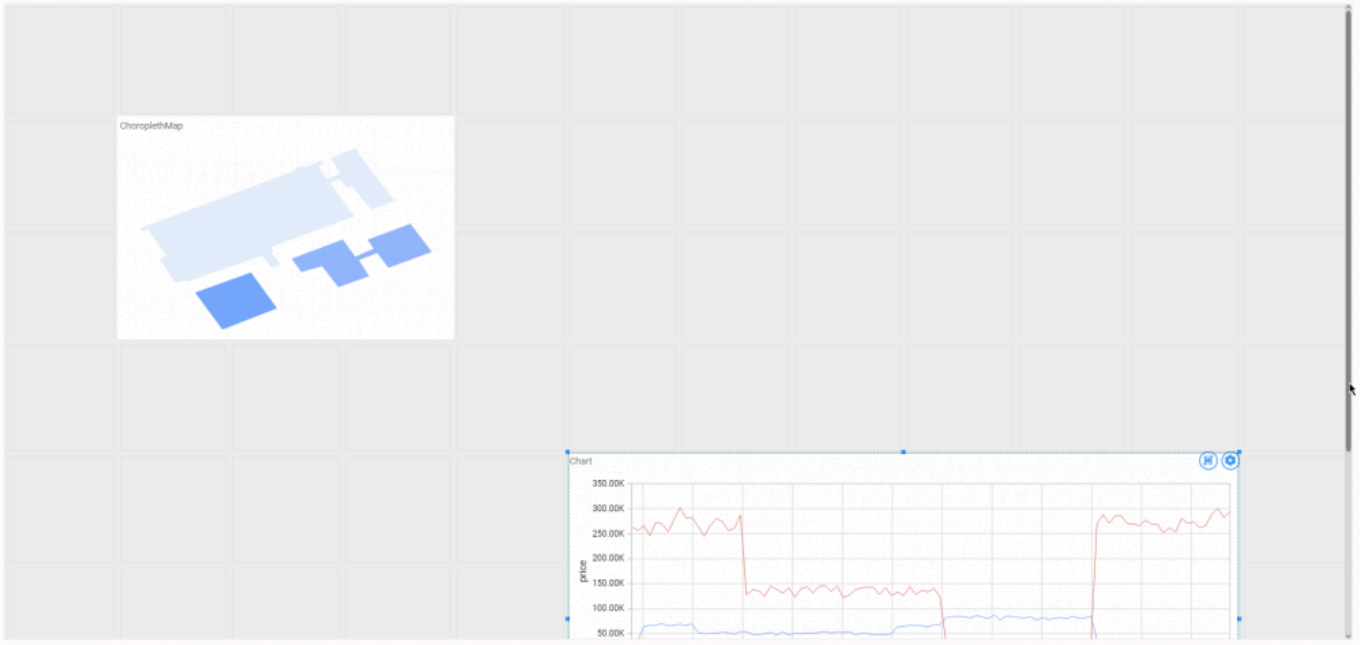
Note : the threshold size actually depends on system zoom and browser zoom too.

Last modified: Jun 17, 2020

2.10.4. Configuring Widgets on the Layout

Layout canvas

Widgets are laid out on a bi-dimensional (2D) canvas.
In Alpana Designer, the layout is configurable in the center part of the *Dashboard* activity tab :



This is where you will configure the Widget's position and size.

Navigating the canvas

The canvas (or design area) can be scrolled and zoomed :

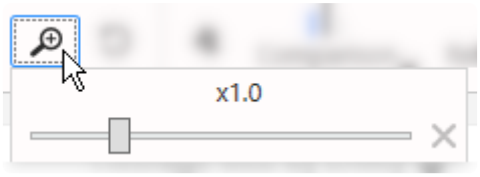
Scrolling

To navigate the canvas vertically, you can use the vertical scroll bar on the right, or your mouse wheel. This is necessary when you have many widgets vertically.

The canvas width always fits the Designer window's width, so there is no need to scroll horizontally.

Zooming

To zoom the canvas in and out, use the *Design Area Zoom* control :



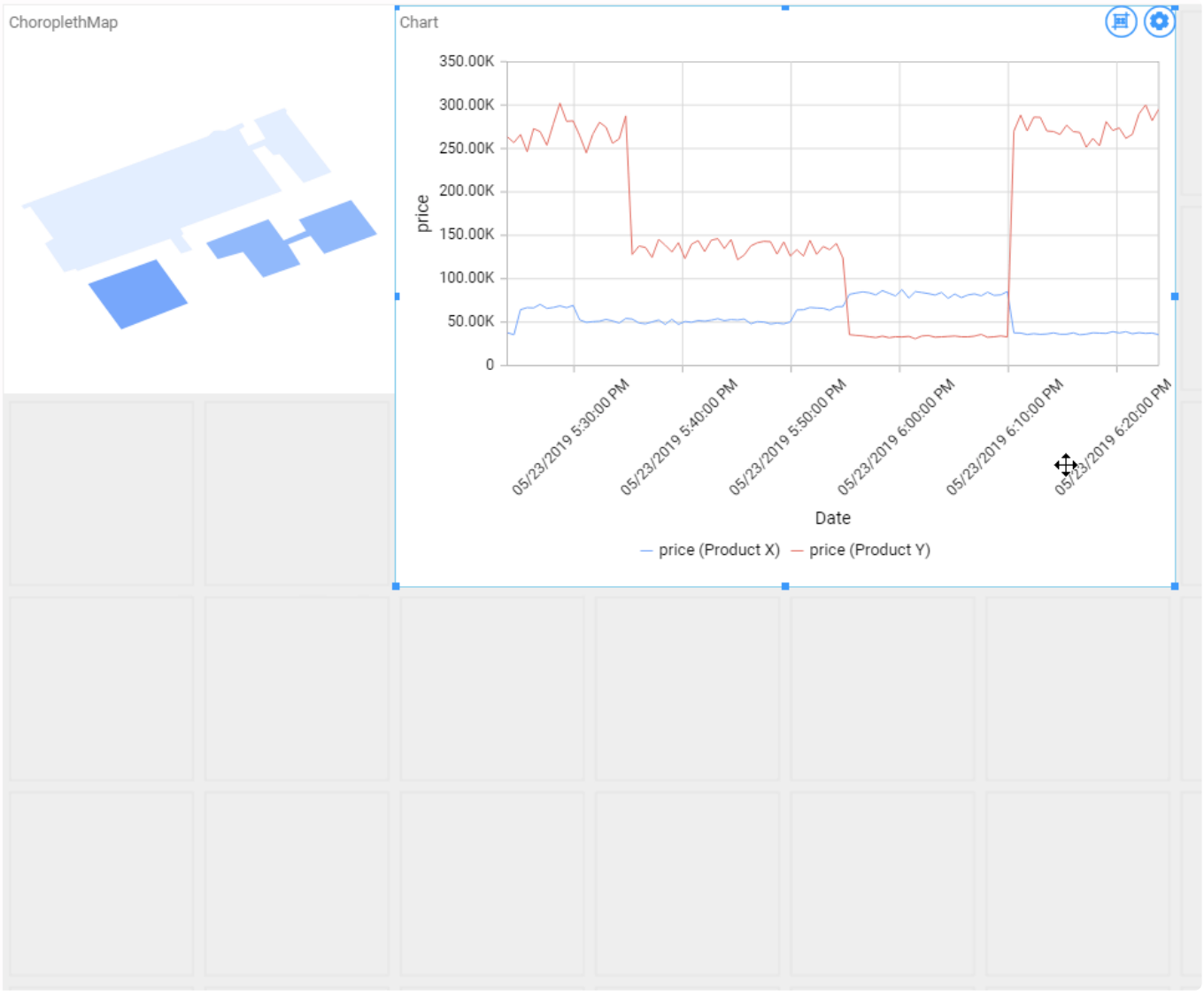
Zooming out allows to have a better view of a multi-page layout (on a PDF for example), and allows to move Widgets quicker.
Zooming in allows to have finer control on a widget position.

The canvas can be zoomed from `x0.1` to `x4.0`.
Use the cross button to reset the zoom level to normal (`x1.0`).

Widgets position and size

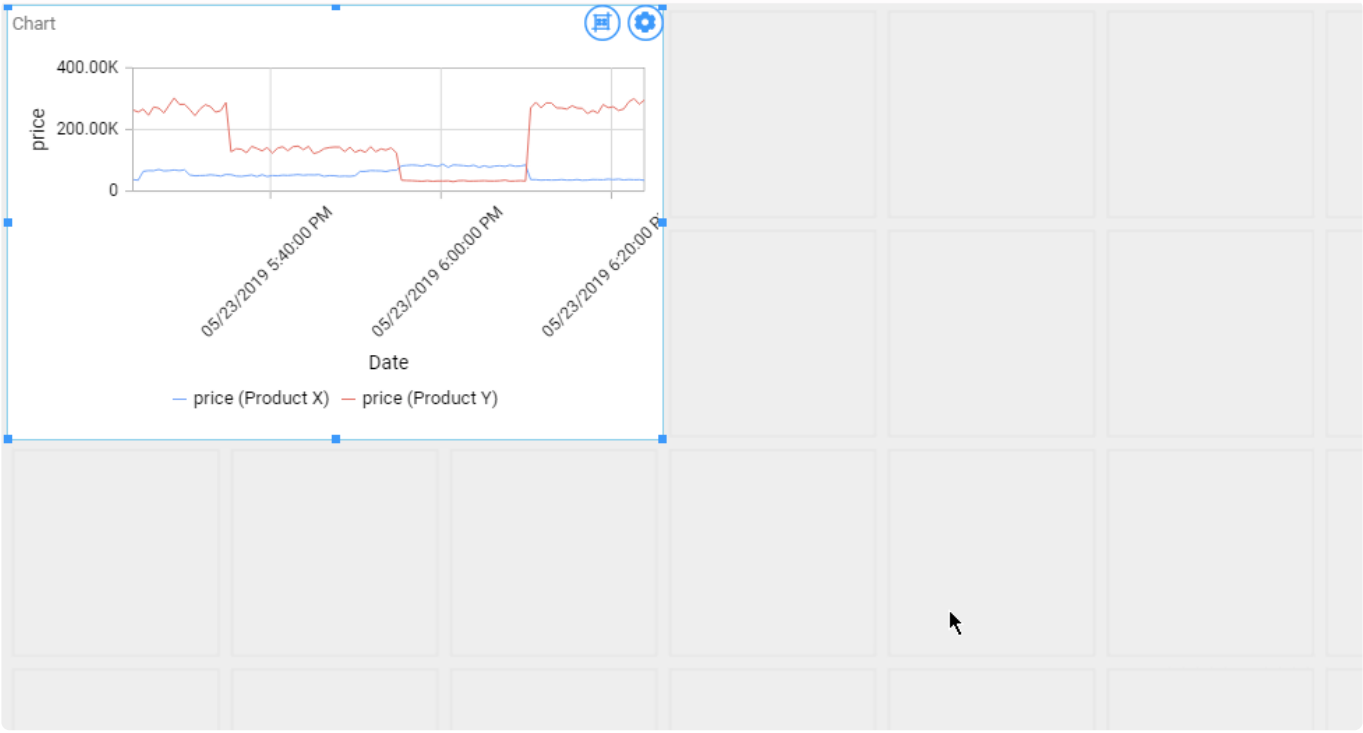
Moving Widgets

Widgets can be moved on the Dashboard layout by selecting and dragging them :



Re-sizing Widgets

To resize a widget, select it to let the handles appear.
Then drag the corner or side handles to resize :



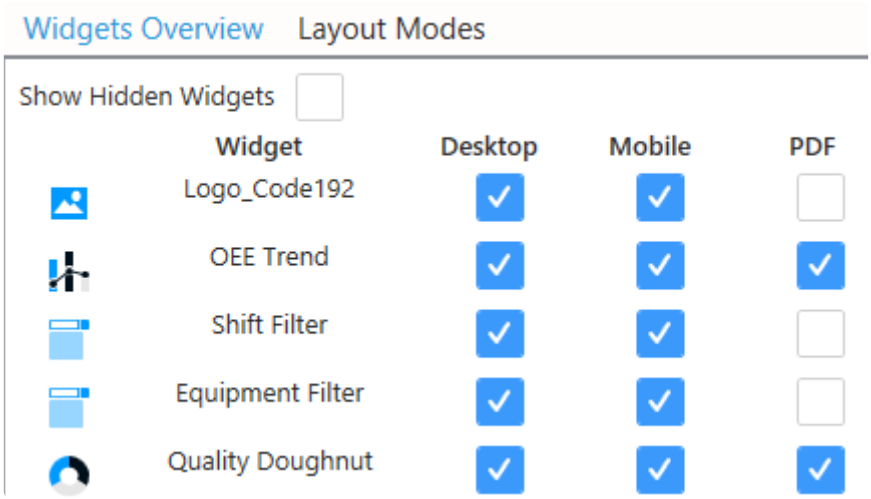
Hiding Widgets

Hiding Widgets allows to create layouts which are simpler or adapted to smaller screens, or maybe to hide images on the PDF exports, etc.
When a widget is hidden on a specific view of a specific Layout, that means it will not be loaded or displayed at runtime.

On each Widget, the *Toggle Visibility* button allows to hide/show the widget :

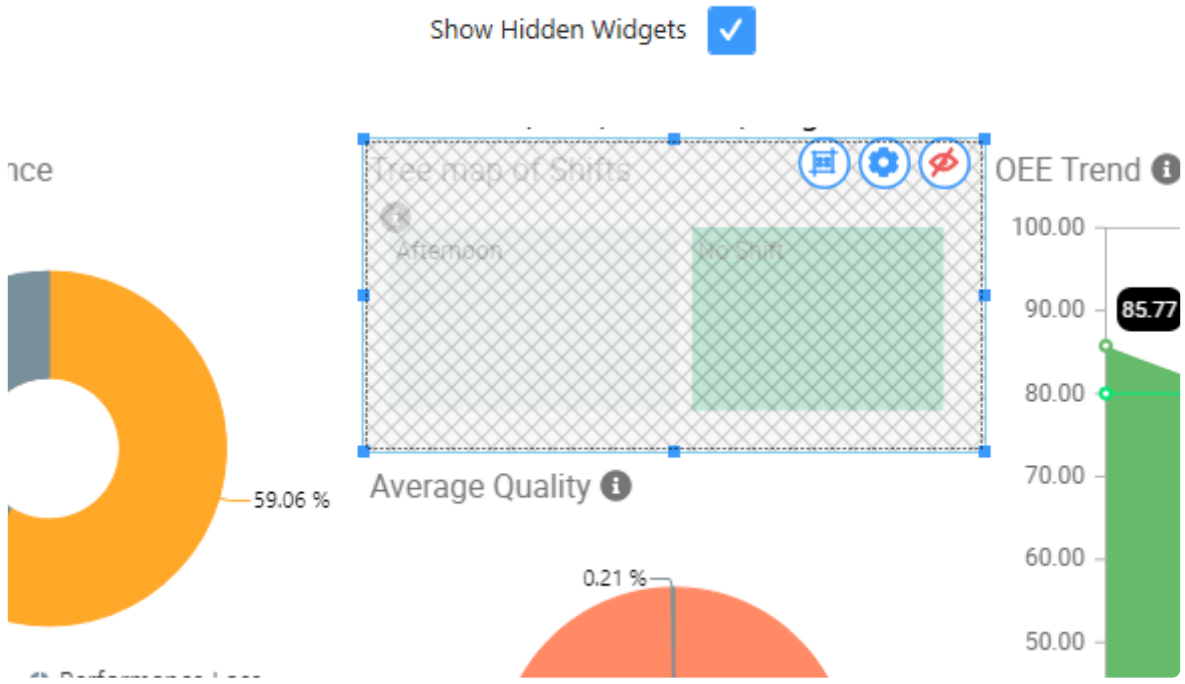
This setting is specific to each view of each layout.

Alternately, you can configure the visibility of each Widget on each view by using the *Widgets Overview* from the Layout properties :



When a Widget is checked on a particular view of this Layout, then it is visible.
When it is unchecked, then it is hidden on that view

When the *Show Hidden Widgets* option is checked, then the hidden Widgets are made visible on the Designer layout :



This can be used to remember that those configured widgets exist, but they will not be displayed at runtime.

Last modified: Jun 17, 2020

2.10.5. Layouts for PDF Exports

Introduction

The *Pdf export* view of each Layout allows to configure the layout that will be rendered when exporting to PDF

How to edit the PDF view

Selecting the PDF view

Select the desired Layout, then use the view selector to select *Pdf Export* :

Using page guides

The canvas displays page guides as an overlay :

Note : This is displayed in the Designer only, to guide the configuration.

Note : There is a slight difference between the position of this overlay and the actual end of page in the PDF exports.
This is for a simple geometrical reason : the overlay is displayed between two canvas square guides, but the page export formats (A3, A4, etc) have a size ratio that cannot be reduced to a simple integer ratio.

Default behaviors

Automatic layout

When creating Widgets from another view (the *Desktop* view for example), the *Pdf Export* layout is generated automatically depending on available space and widget size.
Often, you may need to later tweak this automatic PDF layout manually in order to create a good-looking PDF export.

Overlapping pages

When a Widget is configured to overlap multiple pages, it will be resized automatically during the PDF export, to span a single page only.

Important : Widgets with hidden content

Some widgets have behaviors that allows them to hide some of their content :

- a Grid widget with pagination
- a Doughnut legend with pagination
- etc

! When the PDF generation engine detects that such a situation occurs, then the Widget is removed from its current position in the layout, and moved to its own full page in the PDF.

* Note : This is often depending on the responsive behavior of the widget, so it may be behaving differently with the *Desktop* viewing conditions and the *Pdf export* size and DPI.

Important : Incompatible Widget types

Some Widget types are incompatible with the PDF export functionality.
Those Widgets will simply be ignored during the PDF export, and are automatically made invisible on the *Pdf export* view.

The incompatible Widget types are :

- PDF Viewer
- Alpana Runtime Connection Input Grid
- Embed

Preview

When the selected view is the *Pdf export*, then the *Preview* button shows the PDF export.

Last modified: Jun 17, 2020

2.10.6. Layouts for Mobile

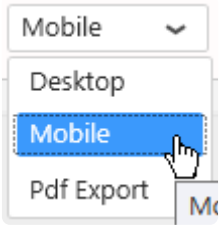
Introduction

The *Mobile* view of each Layout defines how it will be displayed when viewed on a mobile device.

How to edit the Mobile view

Selecting the Mobile view

Select the desired Layout, then use the view selector to select *Mobile* :



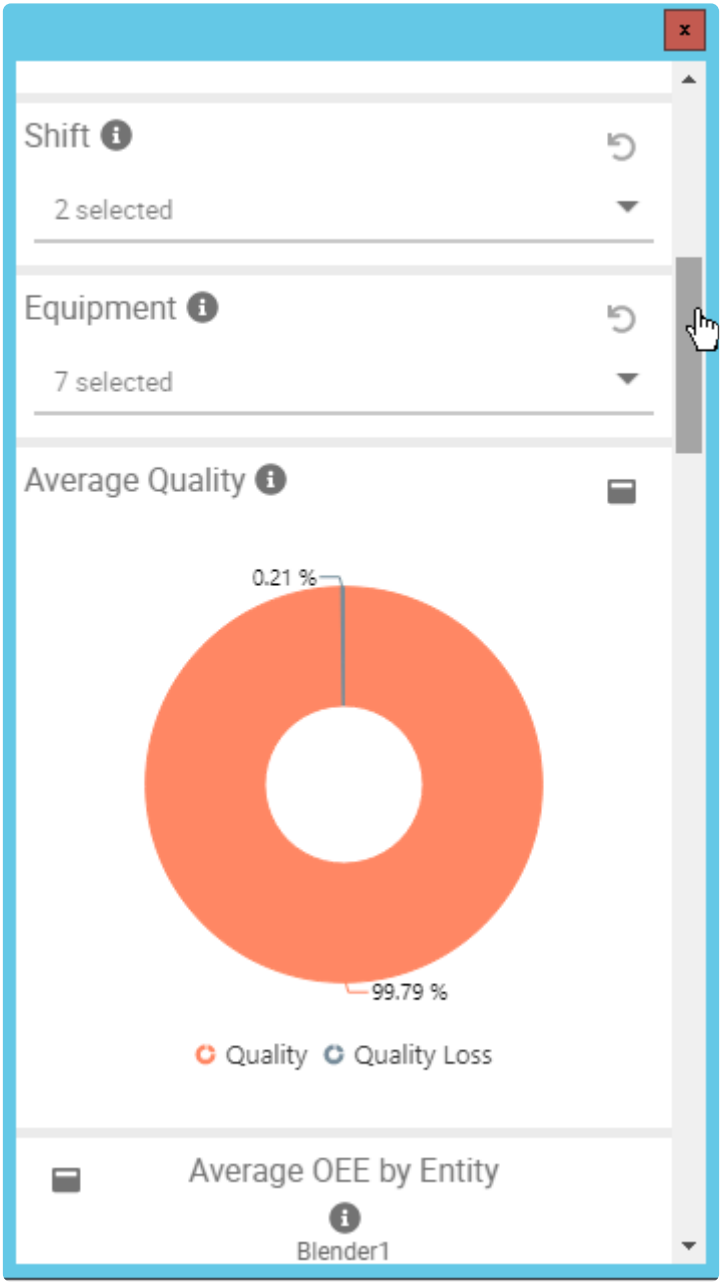
Default behaviors

Automatic layout

When creating Widgets from another view (the *Desktop* view for example), the *Mobile* layout is generated automatically depending on available space and widget size.
Often, you may need to later tweak this automatic Mobile layout manually in order to create a good-looking Mobile experience.

Preview

When the selected view is the *Mobile*, then the *Preview* button shows a small pop-up window shaped like a mobile phone.

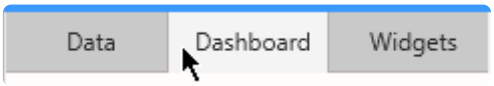


Last modified: Jun 17, 2020

2.11. Dashboard Settings and Publication

Once you prepared data, you can start to compose a Dashboard with Widgets.
There are several ways to proceed, as we will see in this chapter.

Since Alpana Designer v3.0, all the layout and dashboard-level properties are configured inside the *Dashboard* activity tab.



Other Dashboard-level settings are possible, such as :

- theme
- autorefresh
- title
- description
- disabling comments

Last modified: Jun 17, 2019

2.11.1. Dashboard Preview

When working in Alpana Designer, you can easily preview your work in the *Dashboard Preview*.

> This is useful for testing interactions and making sure the dashboard layout is correct before Publishing.
> This *Dashboard Preview* is local, so you don't need access to *Alpana Server* or Internet.

Starting the Dashboard Preview

To launch the current Dashboard in preview mode, click the *Preview* button in the top toolbar :

The result depends on which layout view you are currently selecting :

Desktop

For the *Desktop* view, the Preview starts your default web browser and displays the fully interactive dashboard as it would look when published.

Mobile

For the *Mobile* view, the Preview shows a small pop-up window shaped like a mobile phone.

Pdf Export

For the *Pdf Export* view, the Preview shows the PDF export.

Refreshing the Dashboard Preview

When the Dashboard is modified, the opened preview is not updated.
To update it, click again the *Preview* button to launch another Preview page.

Last modified: Jun 17, 2020

2.11.2. Create and Export / Import Dashboards

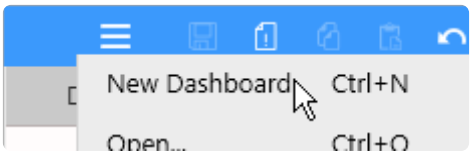
In Alpana Designer, a project is a Dashboard.
Continue reading to learn how to create/export/import Dashboards.

Last modified: May 27, 2019

2.11.2.1. Creating a New Dashboard

When Alpana Designer opens, an empty Dashboard is available for you to start working.

When a Dashboard is already open, you can replace it by a New Dashboard by clicking on the burger menu at the top and selecting *New Dashboard* (or keyboard shortcut Ctrl+N) :



A confirmation dialog appears, allowing you to save your current dashboard, and a fresh dashboard is created in Alpana Designer.

Last modified: May 27, 2019

2.11.2.2. What is a Dashboard file

A complete Dashboard can be saved to a file with format `*.alpd`"

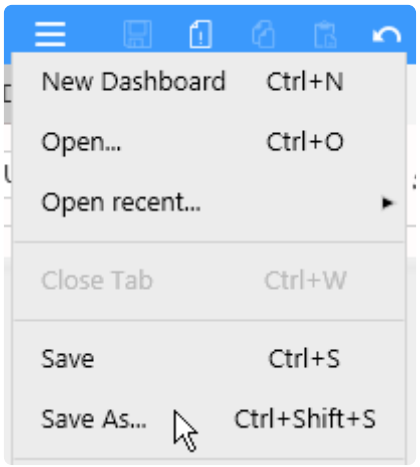
This Dashboard.alpd file contains :

- all Dashboard-level settings (theme, layout, filters, …)
- all Widgets and their configuration
- all Data Sources
- all Connections, including their credentials
- a copy of the source data files for all file-based Connections
- the map GeoJSON content for all Map widgets
- the image files for all Image widgets

Last modified: May 27, 2019

2.11.2.3. Save a Dashboard as file

Click on the burger menu at the top and use *Save* (keyboard shortcut Ctrl+S) to save the Dashboard with the same file name as currently, or use *Save As…* (keyboard shortcut Ctrl+Shift+S) to specify a file name and location :



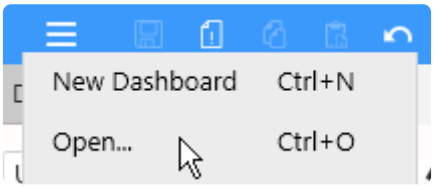
Last modified: May 27, 2019

2.11.2.4. Open a Dashboard file

A Dashboard.alpd file can be opened in a few ways :

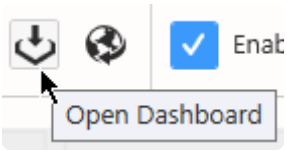
File > Open

To open a Dashboard.alpd file in the currently open Alpana Designer, click on the burger menu at the top and use *Open…* :



Open / Import

You can also use the top toolbar *Open Dashboard* button :

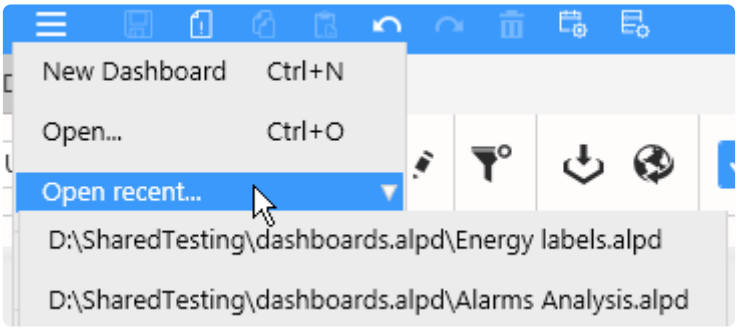


In the *Import Dashboard* dialog, under *Computer*, click … to browse for a Dashboard file and click *Import* :

Open recent

Alpana Designer also maintains a list of recently opened Dashboards.

Click on the burger menu at the top, hover the *Open recent…* item to pop-up a list of recent dashboards, and click an item to open it :



Double-click file in Windows

By double-clicking the file in Windows Explorer, a new instance of Alpana Designer opens this file.
If Alpana Designer is already open, the window becomes focused and displays a confirmation message to close the current dashboard and open the new one.

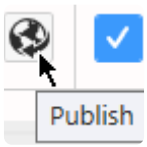
Last modified: May 27, 2019

2.11.2.5. Publishing a Dashboard to Alpana Server

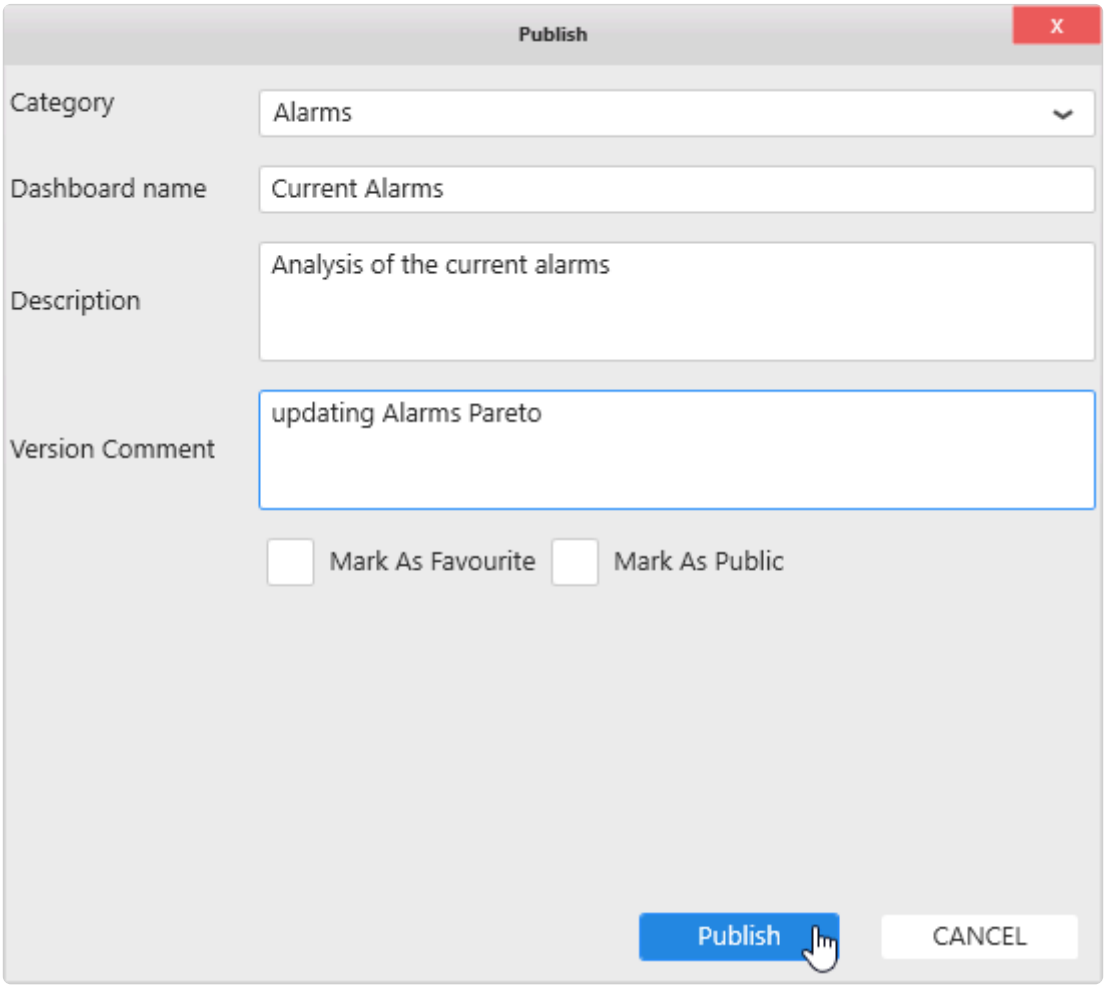
“Publishing” means pushing the Dashboard to a Tenant on Alpana Server to make it available to other users.

> In order to Publish a Dashboard to a Tenant on Alpana Server, you need to be logged in from Alpana Designer to that Alpana Server tenant, and have the necessary **permissions** to create Dashboards in a Category.

In the top toolbar, click the *Publish* button :



In the *Publish* dialog, fill in details and click *Publish* :



The Dashboard is now published to Alpana Server and available to other users depending on

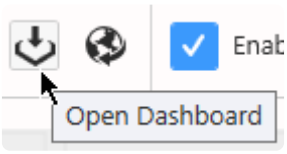
permissions.

Last modified: May 27, 2019

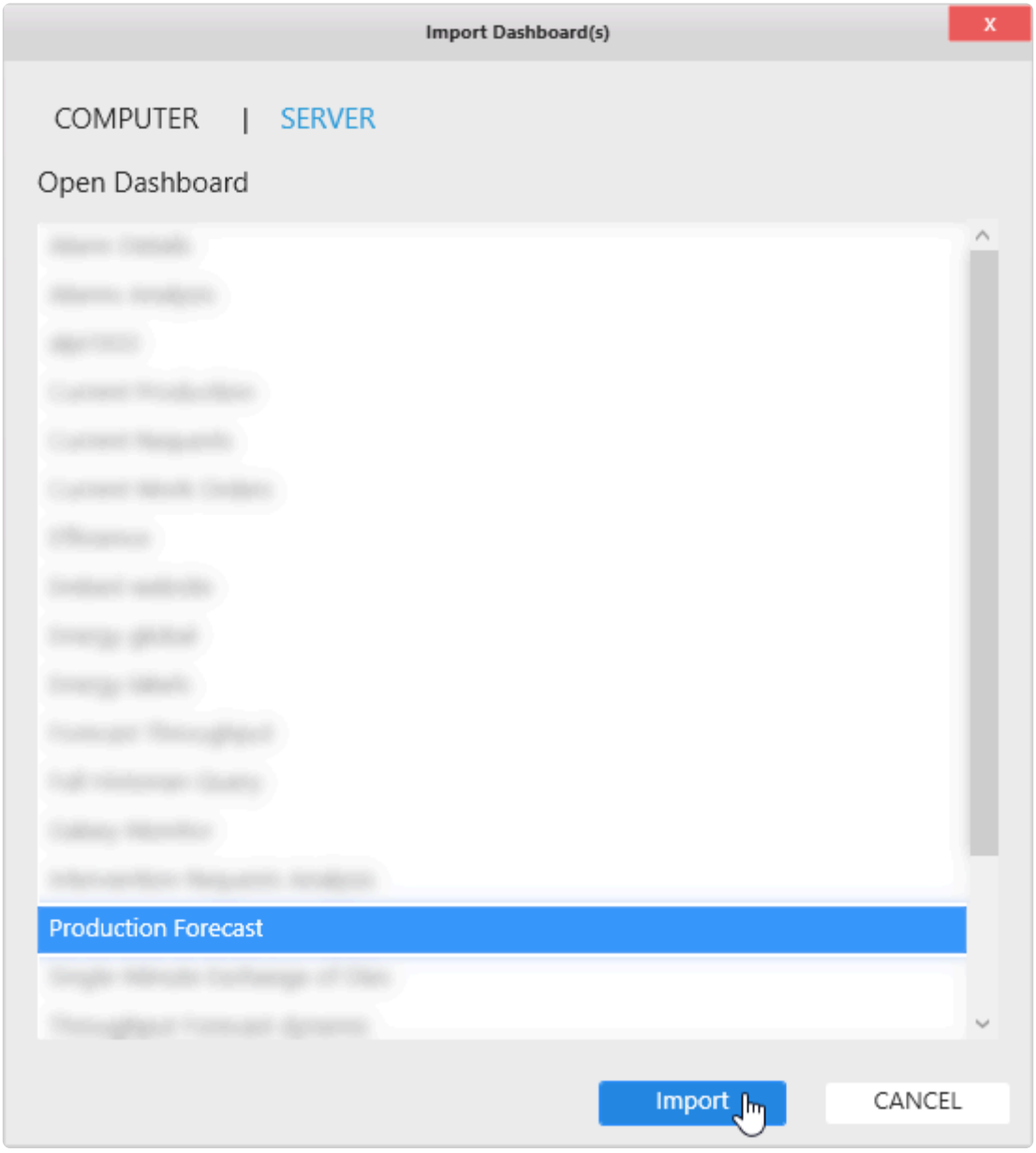
2.11.2.6. Importing a Dashboard from Alpana Server

In order to Publish a Dashboard to a Tenant on Alpana Server, you need to be logged in from Alpana Designer to that Alpana Server tenant, and have the necessary **permissions** to access the Dashboard.

Click the top toolbar *Open Dashboard* button :



In the *Import Dashboard* dialog, click *Server*, select a published Dashboard and click *Import* :



The Dashboard is now open in Alpana Designer ready to be edited.

Last modified: May 27, 2019

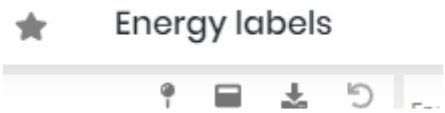
2.11.3. Dashboard-level Settings

Below are documented some settings that are configured at the Dashboard-level.

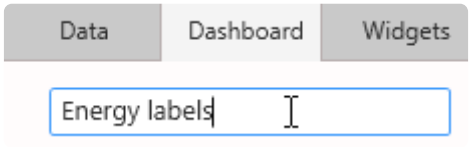
Last modified: May 28, 2019

2.11.3.1. Dashboard Title

Dashboards can have a Title displayed at runtime at the top :



To configure a Dashboard Title, edit the Title textbox on the top-left :



Last modified: May 28, 2019

2.11.3.2. Widget Auto-refresh

What is Auto-Refresh

At runtime, when the Dashboard loads in the browser, data is fetched from the data sources and Widgets are drawn.
After this, if the user doesn't interact with the Dashboard, no new data is fetched and the data displayed can become outdated.

Instead, when Auto-refresh is configured, new data is fetched at regular intervals without needing user intervention.

> This is useful for example when Dashboards are displayed on a TV screen without mouse and keyboard.

Auto-refresh ? Buffer refresh ?

Auto-refresh will perform the same action on data as a manual refresh of the web page.
This may be insufficient to get new data if your Connections are bufferized.
For more information on the Buffer and how to refresh it, see the relevant Chapter in the Connecting to Data chapter.

Important note on Performance

> Setting a small Auto-refresh interval will make more queries to the data.
> Your system (databases, network, I/O, …) needs to be able to support as many queries as (number of open widgets with Auto-refresh) X (frequency of Auto-refresh).
> Make sure to do stress and performance tests for your application to scale appropriately.

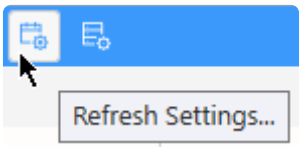
> The Auto-refresh system will perform decently with simultaneously 20 open Dashboards containing each 20 data measures (data points), and 10 Users, when using the recommended hardware pre-requisites.

The minimum allowed Time Interval is 30s.
This is for performance reasons on the web server as well as the data server when many users will leave open an auto-refreshed dashboard.

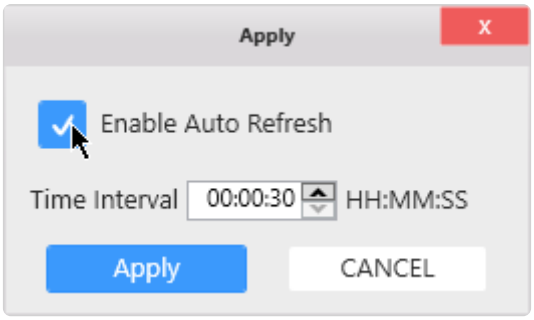
Use this setting sensibly, considering the number of queries that can result from many users leaving the dashboard open on different computers.

Configuring Auto-refresh

To configure Auto-refresh, use the title bar button *Refresh Settings…* :



In the configuration dialog, check Enable Auto Refresh and set a time Interval :

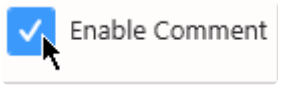


Last modified: Jul 23, 2019

2.11.3.3. Disabling Comments

Dashboard-level comments can be Disabled.

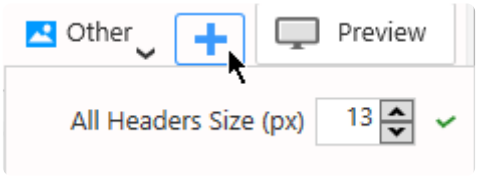
To Enable or Disable Dashboard-level comments, use the checkbox Enable Comment :



Last modified: May 28, 2019

2.11.3.4. Widget Headers size

Widget Headers have a responsive size by default.
If you wish to override the Header size for all Widgets, open the *Additional Options* menu on the top right of the *Dashboard* activity :



Set a Header size in pixels and click the *Apply* button.

This will apply the size at the Dashboard level.
You can override this setting for each Widget separately.

Last modified: Oct 15, 2019

2.11.4. Designer-level Settings

Alpana Designer maintains some application settings.
See the below chapters for more details.

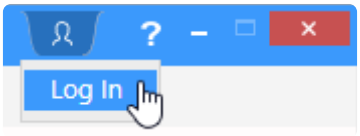
Last modified: May 27, 2019

2.11.4.1. Login to Alpana Server

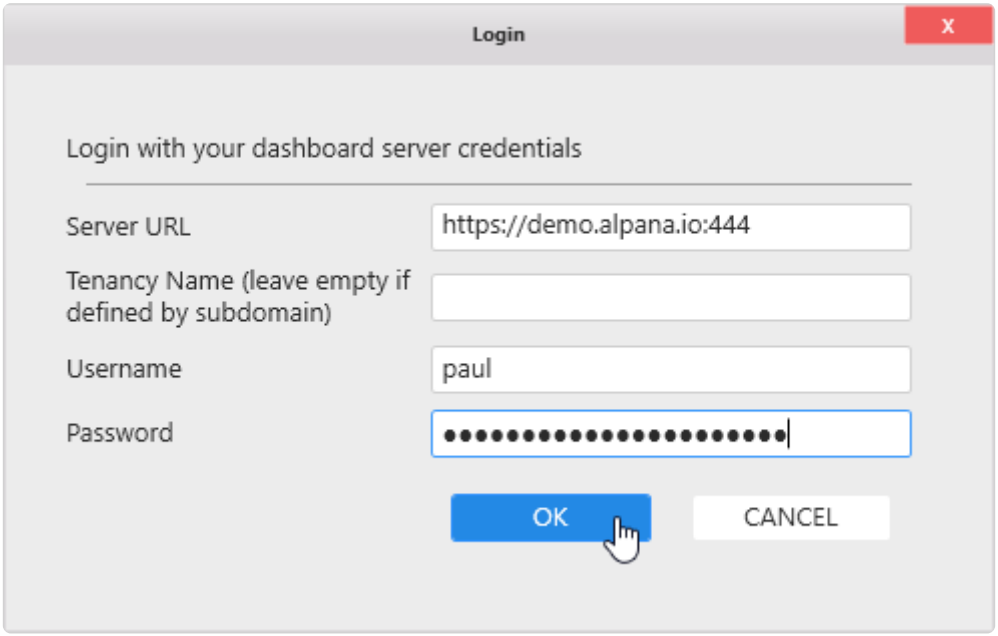
Alpana Designer can log in to Alpana Server.
This is useful for importing/exporting objects to Alpana Server from the Alpana Designer interface.

Logging in

To login, click the profile icon on the top right of the Alpana Designer Window and click *Log In* :



The *Login* window pops up and lets you enter your credentials :



Once you login, you will stay logged in until you explicitly logout (see below). This includes closing Alpana Designer.

Server URL

The full URL to connect to the Alpana Server installation.
This must include :

- the right protocol (HTTP / HTTPS)
- the full name or IP address. If Alpana Server is configured to have Tenants as sub-domains, you can include the sub-domain here too.
- the port number of the Alpana backend website

> ! Remember : the Alpana Server backend website is usually running on a different port number than the frontend (which you use for viewing dashboards in your web browser)

Tenancy Name

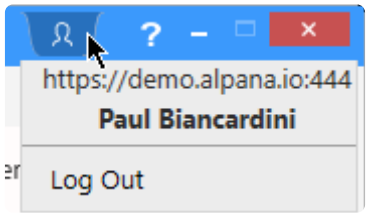
The name of the Tenant you want to connect to.
If Alpana Server is configured to have Tenants as sub-domains, you can leave this field empty.

Username / Password

Use the same credentials as for logging in to Alpana Server from the web interface (frontend).

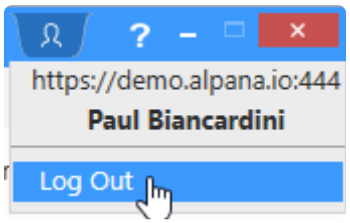
Checking login

To check if you are logged in, click the profile icon on the top right of the Alpana Designer Window and the details window pops up :



Logging out

To logout, click the profile icon on the top right of the Alpana Designer Window and click *Log Out* :



Last modified: May 27, 2019

3. Alpana Server

Alpana Server is a website application that allows to make use of the content created with Alpana Designer : display and interact with visualizations, export, notify, socialize.
Securely share your Dashboards and Widgets to any modern web browser. Dashboards can be embedded in your Scada Application, your intranet or presented through our secured Server.

Read below the documentation of Alpana Server to learn more.

Last modified: Dec 26, 2019

3.1. About Alpana server

Alpana Server is a web server that allows to share and manage your Alpana content (Dashboards, Widgets, Data Sources).
After creating content in Alpana Designer, you simply push it to Alpana Server. This is called Publishing.

Securely share your Dashboards and Widgets to any modern web browser.
Dashboards can be embedded in your Scada Application, your intranet or presented through our secured Server.

The Alpana Server lets you efficiently organize and share Dashboards through a web interface.
Installed on premise or in the cloud, Alpana Server can host multiple tenants in complete isolation.
Use the web API talk to Dashboards live.
Embed Dashboards within your application either as web pages, or directly within Wonderware Intouch with our native container.

Last modified: Nov 04, 2019

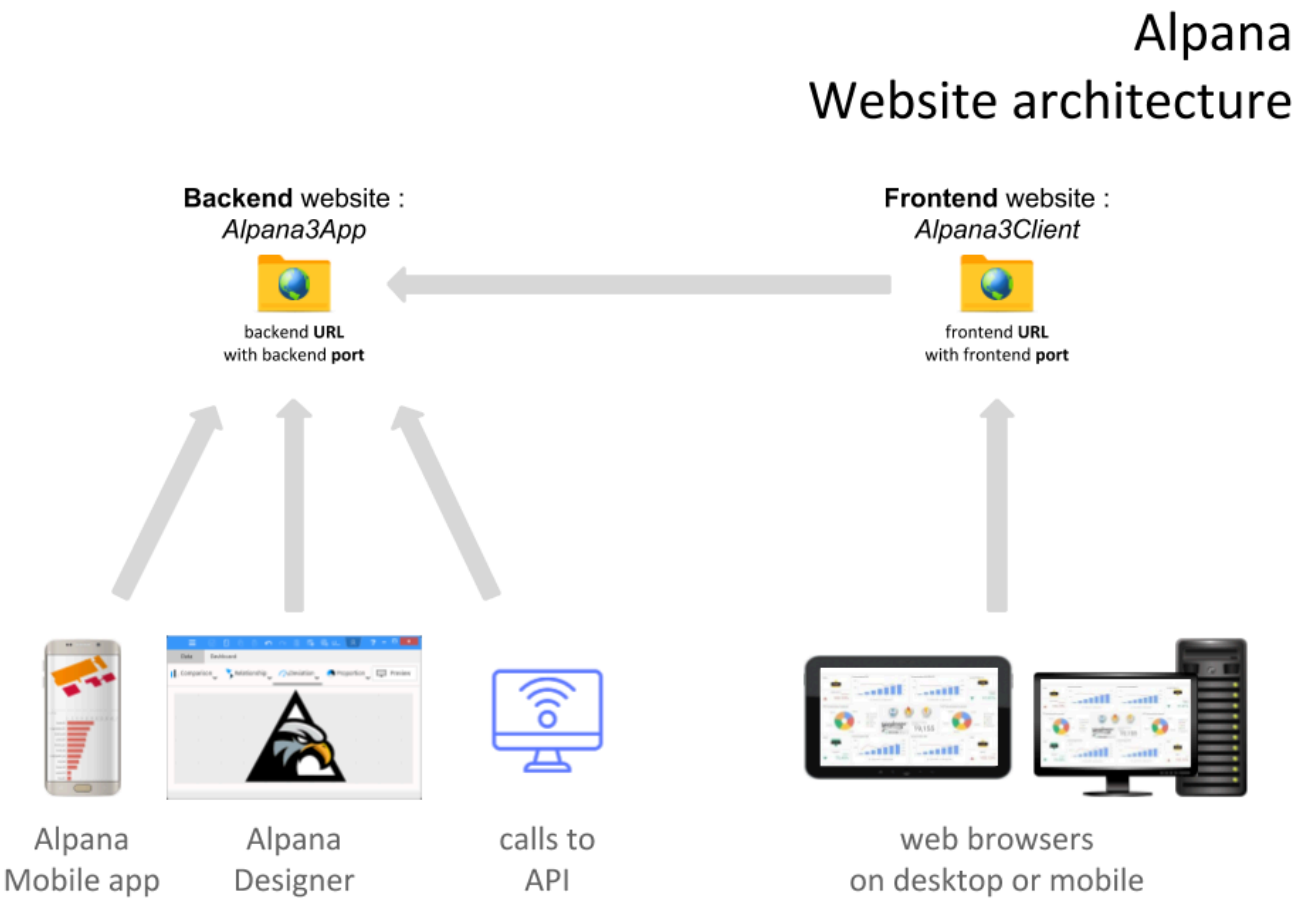
3.1.1. Architecture

This chapters explains the components of Alpana Server.
Alpana Server is a website that deploys on Microsoft IIS and stores data to databases in Microsoft SQL Server.

Websites

Alpana Server runs two websites on Microsoft IIS :

- the *Back-End Site*
- the *Front-End Site*



The application is run under Microsoft IIS by the Windows User called `IIS APPPOOL\Alpana3App`

The *Back-End Site*

The *Back-End Site* is named `Alpana3App` in IIS.
This is where the application runs (this is where everything is actually managed, permissions,

dashboards, authentication, etc…).

Any client application needs to connect to this Back-End :

- Users will connect to this site for publishing from Alpana Designer
- Users will log in to this site when using Alpana Mobile
- User-created applications will connect to this site when using the API
- actually, the *Front-End Site* also connects to the Back-End in the background

The *Front-End Site*

The *Front-End Site* is named `Alpana3Client` in IIS.
This is where the web portal is made available.

Users will connect to this site to view dashboards from a web browser.

The Front-End site actually connects to the Back-End site for every action.

Application Databases

Alpana Server also requires two types of databases :

- the actual Alpana Server database
- the Buffer database(s)

These databases are accessed by the Back-End application using SQL Windows Authentication under the Windows User called `IIS APPPOOL\Alpana3App`.

Alpana Server database

The Alpana Server database is named `AlpanaDb` by default.
It stores all information about the application (users, dashboards metadata, logs, …)

Buffer database(s)

Each Tenant of Alpana Server has its own Buffer database called `AlpanaBuffer_X`, where `X` is the internal ID of the Tenant. (e.g. : on a single-tenant installation, the database is called `AlpanaBuffer_1`).

For more explanations about the Buffer, see the dedicated chapter.

Access to user data

Dashboards point to data through Connections.

When a Dashboard makes a new query to a Connection, new data is fetched by the Back-End application.

This means in particular :

- When using Windows Authentication in connections, they will be run by the corresponding IIS Windows user `IIS APPPOOL\Alpana3App`. So this user must have appropriate permissions to the data.
- This data access must be allowed by firewalls, port redirection, anti-virus, etc…

Folders

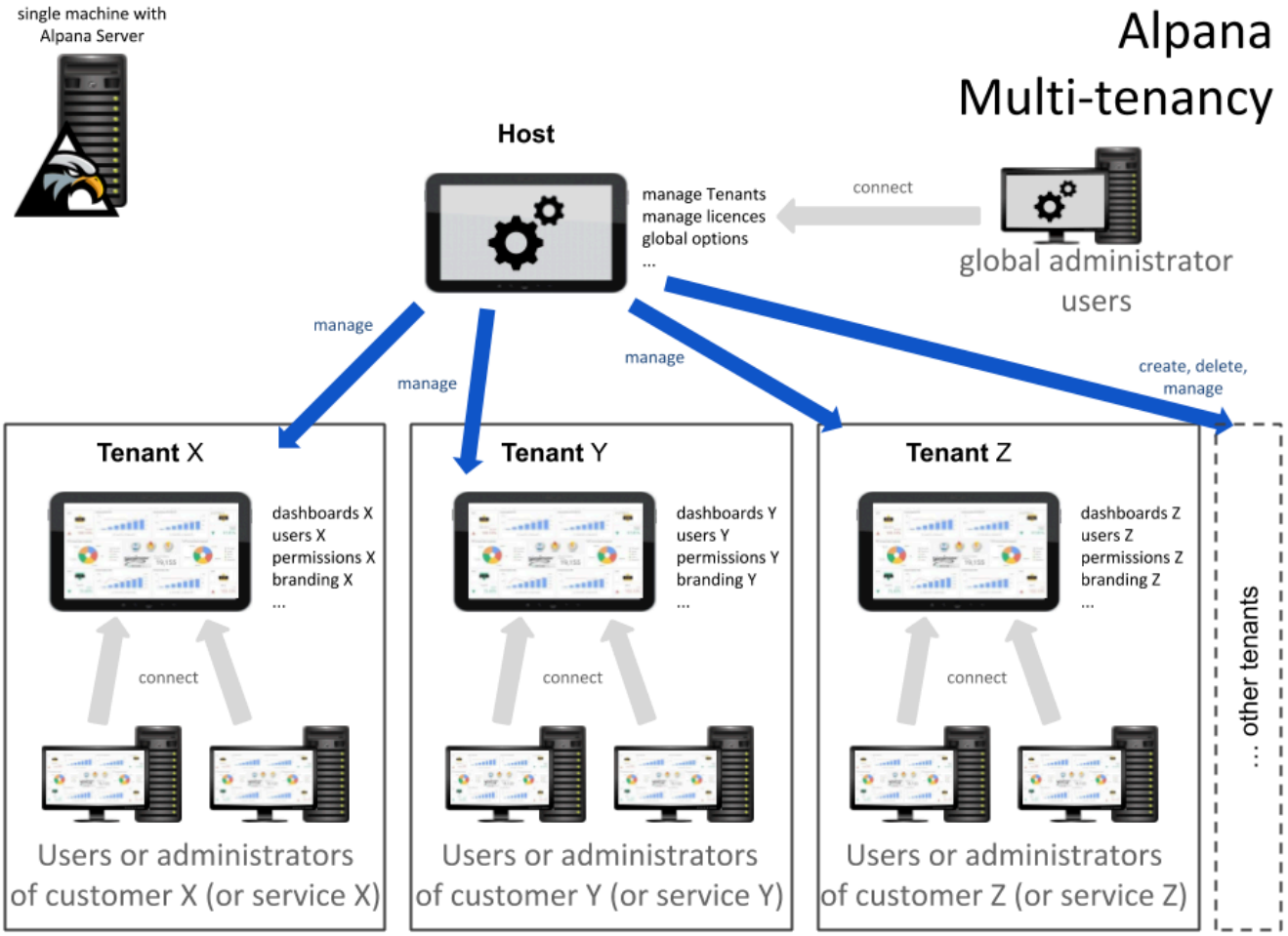
Website folders are by default located inside `C:\inetpub\alpana3\`.
Under this folder, user data relevant for backup is located under `\app\App_Data\`.
In particular, the actual published Dashboards are located inside `\app\App_Data\Published\`.

Last modified: Dec 26, 2019

3.1.2. Multitenancy

Alpana Server is a multi-Tenant application.
This means that a single instance of Alpana Server can manage any number of customer-facing isolated instances.

What Is Multi-Tenancy?



> The term “software multitenancy” refers to a software architecture in which a single instance of software runs on a server and serves multiple tenants. Systems designed in such manner are often called shared (in contrast to dedicated or isolated).
> A tenant is a group of users who share a common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance – including its data, configuration, user management, tenant individual functionality and non-functional properties.
> Multitenancy contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants.

(Wikipedia)

In short, multi-tenancy is a technique that is used to create **SaaS** (Software as-a Service) applications.

Host > Tenant

We define two notions for a multi-tenant system :

- *Tenant*: A customer which has its own dashboards, users, permissions, settings… and uses the application completely isolated from the other tenants.
  A “*tenant user*” is a user owned by a tenant.
- *Host*: There is always a **single** Host. The Host is used for creating and managing tenants.
  A “*host user*” is at a higher level : it is independent from all tenants and can control them.

Last modified: Oct 21, 2019

3.2. Installation

To install Alpana Server, please follow the below chapters.
Pre-Requisites are **very important**, in particular “IIS Configuration”.

Last modified: Dec 26, 2019

3.2.1. Pre-Requisites

This section explains the system requirements to run Alpana Server.

Alpana Server Pre-Requisites : Software

Operating System

Windows OS (x64) :

- Windows Server : 2012 R2, 2016, 2019
- (see note below) Windows 10 (Version 1607+)

> ! Important : A **server** OS is strongly advised.
> Alpana Server runs a web server and should be allowed to handle many incoming connections, but desktop versions of Windows apply hard limitations on those.
> In practice, running a web server on Windows 10 will only let you open a few dashboards at the same time.

Database

Microsoft SQL Server is required, with version 2012 or later.
Microsoft SQL Server Express is possible, but see limitations related to this edition of SQL (concurrency, database size, resource usage, etc).

During installation, you will need a login to SQL Server which has the permissions to create users and databases.

> ! We strongly recommend to install the Alpana Server Database on the same machine that the web application: the web app being constantly communicating with its repository, if the repository and the application are on different machines, it will affect the performance and the user experience.

Other software

The following software needs to be installed and configured as described below.

> ! The pre-requisites **order** is very important.
> If you modify any IIS setting after installing AspNet Core, then you will have to Repair AspNet Core : see Installation Troubleshooting.

You must restart IIS after installing AspNet Core.
You must restart the computer after installing a new version of the .NET Framework.

Pre-requisites must be installed **in the following order** :

1. Microsoft IIS

This must be installed and configured **first**.
The IIS server must be in the “Started” state for installing properly.

How to configure

On server editions of Windows (Server 2012 R2, Server 2016, 2019), IIS can be installed and configured by using the Server Manager, with the menu “Manage > Add roles and features”.
On desktop editions of Windows (Windows 10), Microsoft IIS can be installed and configured by opening “Turn Windows features on or off”

Required IIS configuration

In order to install Alpana Server, Microsoft IIS must be installed and configured with the following options :

- Under *Common HTTP Features* :
  - *WebDAV Publishing*
- Under *Performance Features* :
  - *Dynamic Content Compression*
- Under *Security* :
  - *Windows Authentication*
- Under *Application Development Features* :
  - *.NET Extensibility 3.x*
  - *.NET Extensibility 4.x*
  - *ASP.NET 4.x*
  - *ISAPI Extensions*
  - *ISAPI Filters*
  - *WebSocket Protocol*
- Under *Web Management Tools* :
  - *IIS Management Console*

2. PowerShell

PowerShell must be active.

You can just check that you can run Powershell : `Windows+R` to run the command : `powershell`

3. Microsoft .NET Framework

Microsoft .NET Framework 4.8 must be installed.

An offline installer exists at the following direct link :
https://dotnet.microsoft.com/download/dotnet-framework/thank-you/net48-offline-installer

(Alternately, go to the download page then download the “Runtime”.)

> Installing this component requires to restart the machine.

4. AspNet Core

“ASP.NET Core 3.1 Runtime – Windows Hosting Bundle” must be installed.

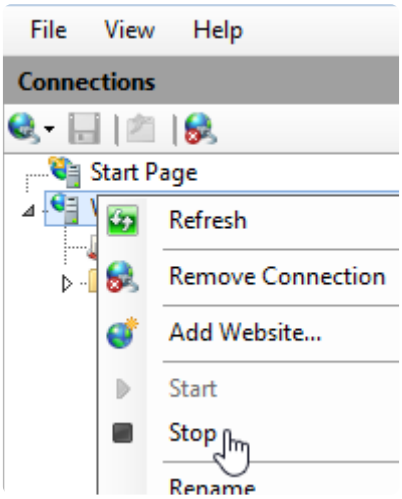
Installers for latest v3.1.3 can be found at the direct link here :
https://download.visualstudio.microsoft.com/download/pr/ff658e5a-c017-4a63-9ffe-e53865963848/15875eef1f0b8e25974846e4a4518135/dotnet-hosting-3.1.3-win.exe

(Alternately, go to the .NET Core 3.1 download page and select version ASP.NET Core Runtime “Hosting Bundle” for Windows.)

> Installing this component requires to restart IIS.

5. Restart IIS

After installing AspNet Core, restart IIS (or the machine altogether) :
Run `inetmgr`, and right-click the machine name to press *Stop* and then *Start* :



6. IIS URL Rewrite

“IIS URL Rewrite” must be installed.

A direct link to the offline installer is available here :
https://download.microsoft.com/download/1/2/8/128E2E22-C1B9-44A4-BE2A-5859ED1D4592/rewrite_amd64_en-US.msi

(Alternately, go to the URL Rewrite downloads and select your platform and language)

This should be installed after IIS

Alpana Server Pre-Requisites : Hardware

Hardware requirements highly depend on usage.
Typically, SQL and IIS will take disk space, disk I/O, RAM and CPU depending on several factors like database usage, dashboards complexity and website traffic.
Please see with your website and database administrators to size your application appropriately.

Hardware Minimum

Those are the bare minimum that will get you a running installation.
As soon as you will add content, add data, and increase traffic, **you will need more**.

- HDD : 1GB
- RAM : 3GB including OS
- CPU : any dual core CPU, x86 or x64 architecture

Hardware Recommended

To go beyond the minimal installation, you will need evaluate your requirements based on traffic, number of dashboards, number of tenants, other software and databases hosted on the same machine, etc…

Typically, it’s recommended to have at least :

- SSD for fast I/O, 50GB+ with free space always available (usual system administrator advice apply)
- RAM : 8GB for a small/medium site with few databases an no other big software installed
- good server CPU, like a recent Xeon with several cores

This can be much more if your application serves many dashboards to many users on many tenants.

Last modified: Aug 26, 2020

3.2.2. Installing

Important

> ! Before installing, make sure you meet the Pre-Requisites.

> ✻ Alpana Server will create folders and SQL databases.

Download installer

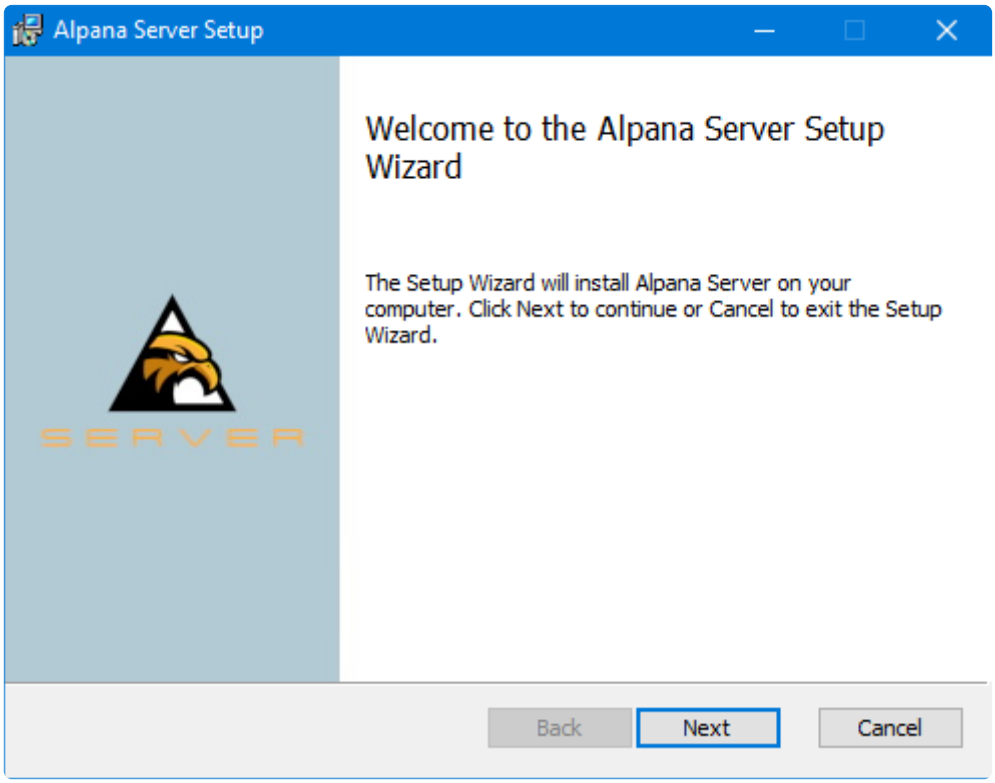
See here to download the installer.

Running the Alpana Server installer

Login with administrator rights to the machine where you want to install Alpana Server.

Start the Alpana Server installer `.msi` file to start the installation.

At the installer welcome message, click *Next* :

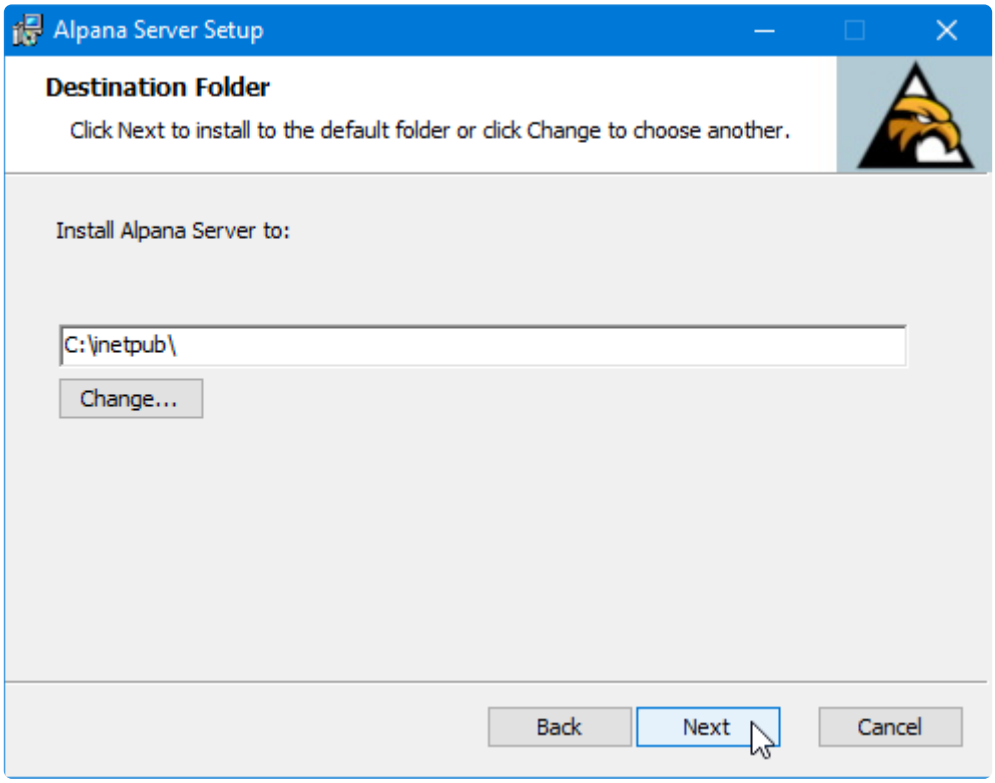


Accepting EULA

Accept the End User License Agreement and click *Next* :

Configuring website folder

Alpana Server will be installed as IIS websites in this machine.
Use this page to configure the folder on the disk where the website files will be hosted :



Configuring SSL or DNS

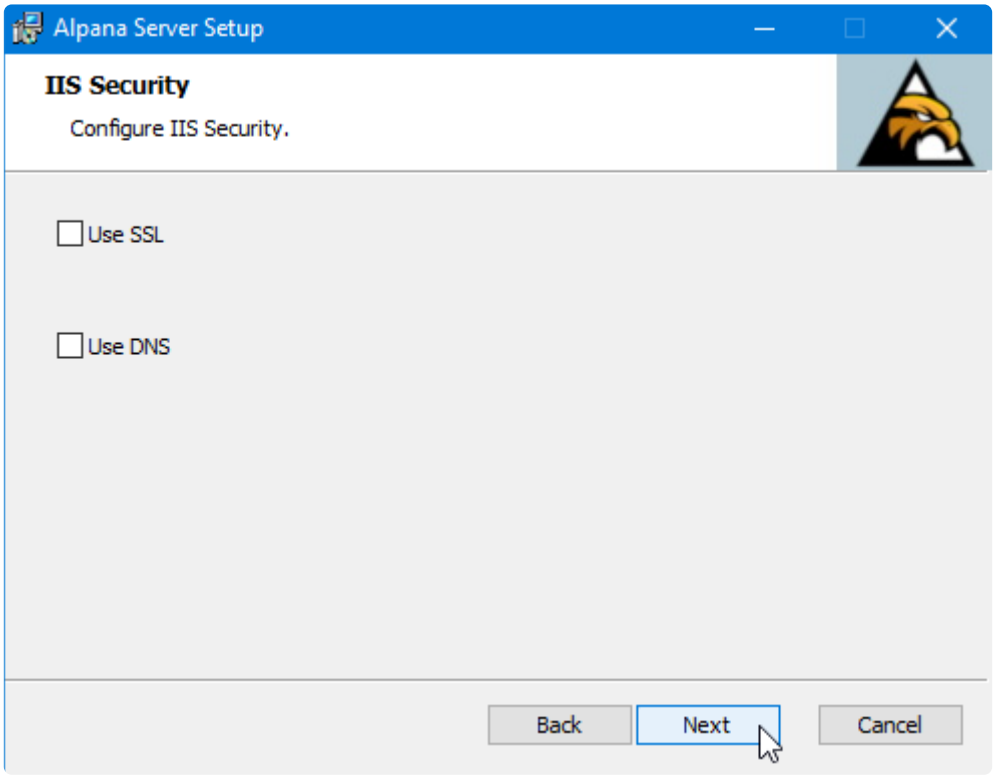
Use this page to optionally configure :

- SSL : insert a certificate to let users connect to your Alpana Server website using HTTPS
- DNS : bind a domain name to let users connect to your Alpana Server using a friendly name like www.alpana.io, or write a static IP address that is not currently available

Tenant names as subdomains

If you use DNS with a domain name, you will be able to create subdomains.
The installer can be configured to automatically generate a subdomain for each created Tenant.
For example, if your domain is “mydomain.com”, and you have 3 Tenants, they each will have the following URL :

- tenant “Default” : URL `http://default.mydomain.com`
- tenant “CustomerX” : URL `http://customerx.mydomain.com`
- tenant “CustomerY” : URL `http://customery.mydomain.com`



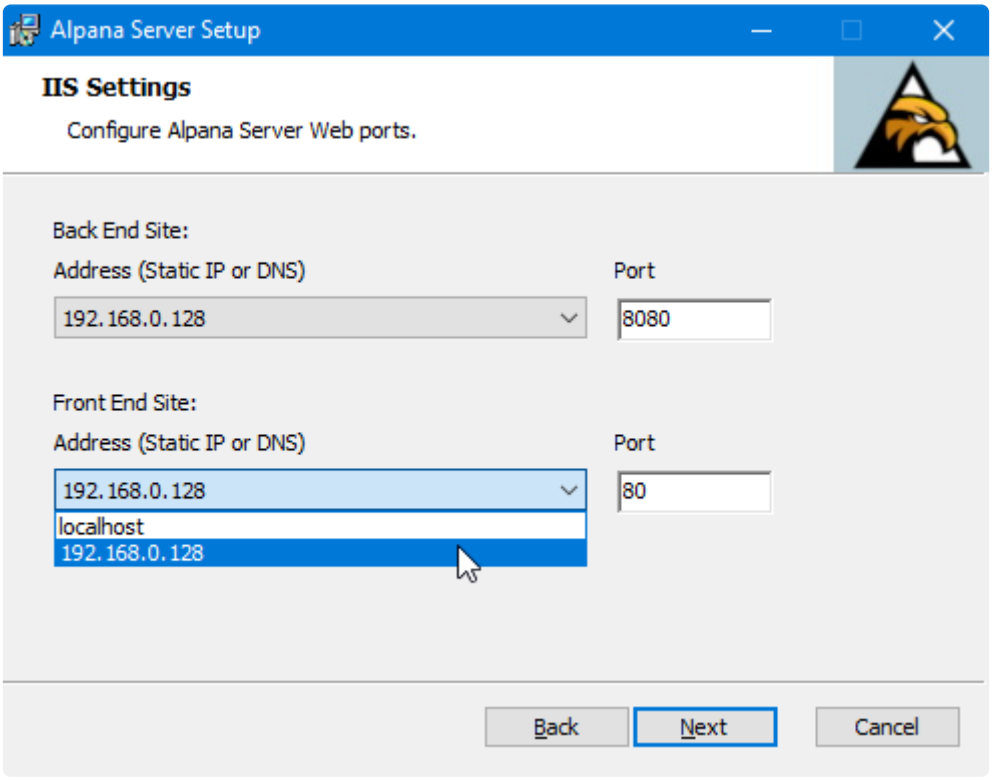
Configuring address and ports of websites

Alpana Server actually consists of two IIS websites :

- the *Back-End Site* is where the application runs. Users will connect to this site for publishing from Alpana Designer, for logging in using Alpana Mobile, or when using the API
- the *Front-End Site* is where the web portal is made available. Users will connect to this site to view dashboards from a web browser.

If a port is invalid or unavailable, select another port.

If you have several IP addresses (Network Interface Cards), select the desired IP address from the list, or “localhost”.



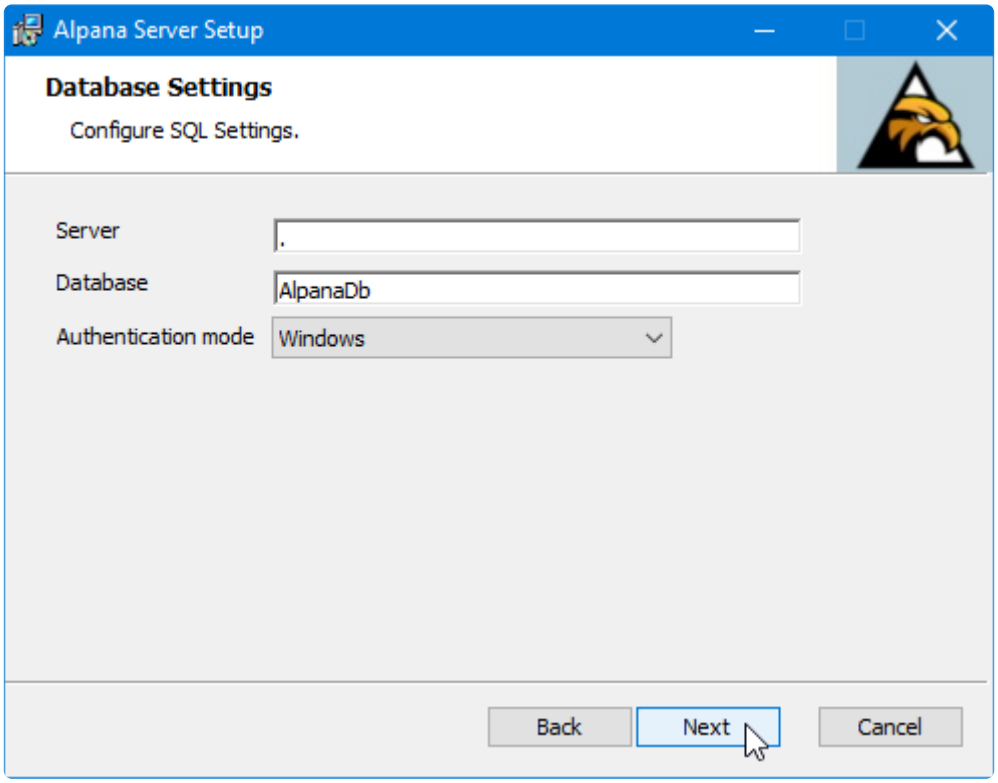
! When selecting “localhost”, Alpana Server will not be reachable from another computer.

! Please make sure that those ports are open in the firewall and accessible to the client machines.

* When using a domain name, Alpana will not respond with the IP address. See Troubleshooting.

Configuring the SQL connection

Alpana Server stores its state inside a SQL database.
Use this page to configure the connection string to the database.
This connection string must allow the installer to create databases and give permissions to users.



✻ Two databases are created : the Alpana Server database for storing the state of Alpana Server, and the Buffer database for temporarily storing some dashboard data.

! The Alpana Server database is created by default with recovery mode *Simple*.

Server

This is the address of the SQL Server instance you need to access. Possible values are :

- `.` or `localhost` : local server
- : server named with a NetBIOS name
- : server accessible through a IP v4 address
- : server accessible through a IP v6 address
- `\` : connect to a specific instance. For SQL Express, it may be necessary to use `\SQLEXPRESS`
- `,1433` : 1433 is the default port for SQL Server, but can be changed like this

Database

The desired name for the Alpana database.

Authentication Mode

“SQL Server” : will use SQL Server authentication with “User name” and “Password”.

“Windows” : will use Windows Authentication from the Windows user that generates the query.

Note on Windows Authentication

Please note that **two** Windows users will be using this connection string :

- The Windows user who runs the installer. The installers authenticates to SQL as this Windows user to create databases and transfers rights to the second user below :
- The Windows user who runs the Application Pool of the IIS Website for Alpana Server. Its name is `IIS APPPOOL\Alpana3App`. The Alpana Server app will use this user to connect to its owned databases. The dashboard Connections will use this user to connect to dashboard data if the Connections are configured with Windows Authentication.

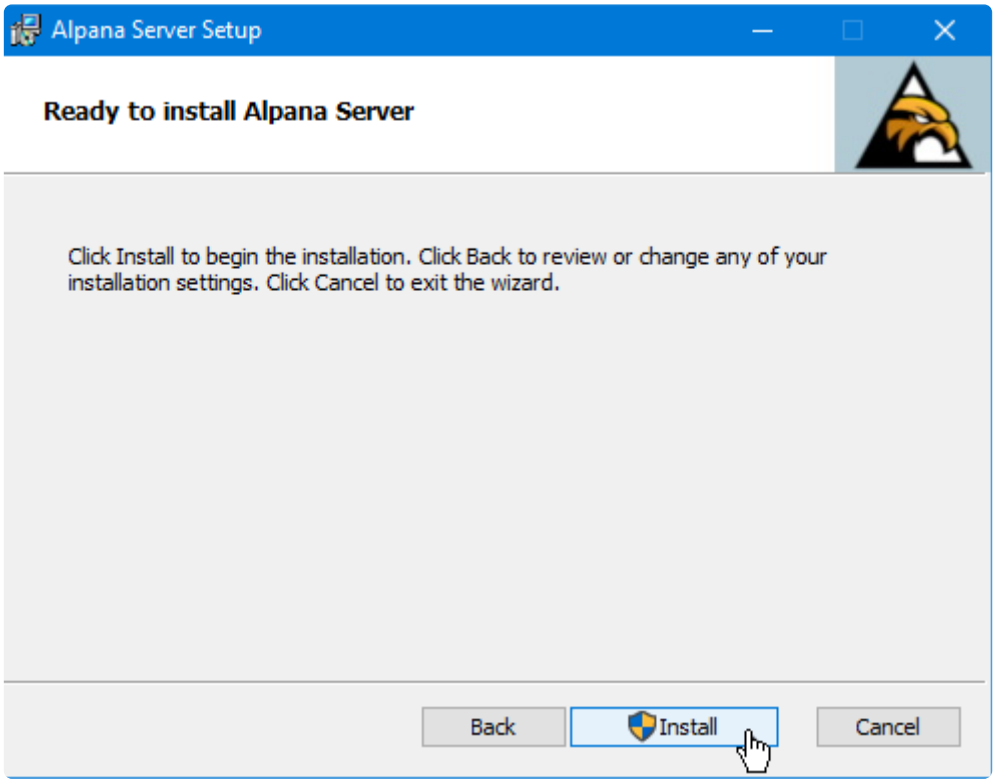
> ! Both Windows users must have the necessary permissions to SQL.
> In particular, the Windows user you are using to run the Alpana Server installer must have admin privileges in SQL.

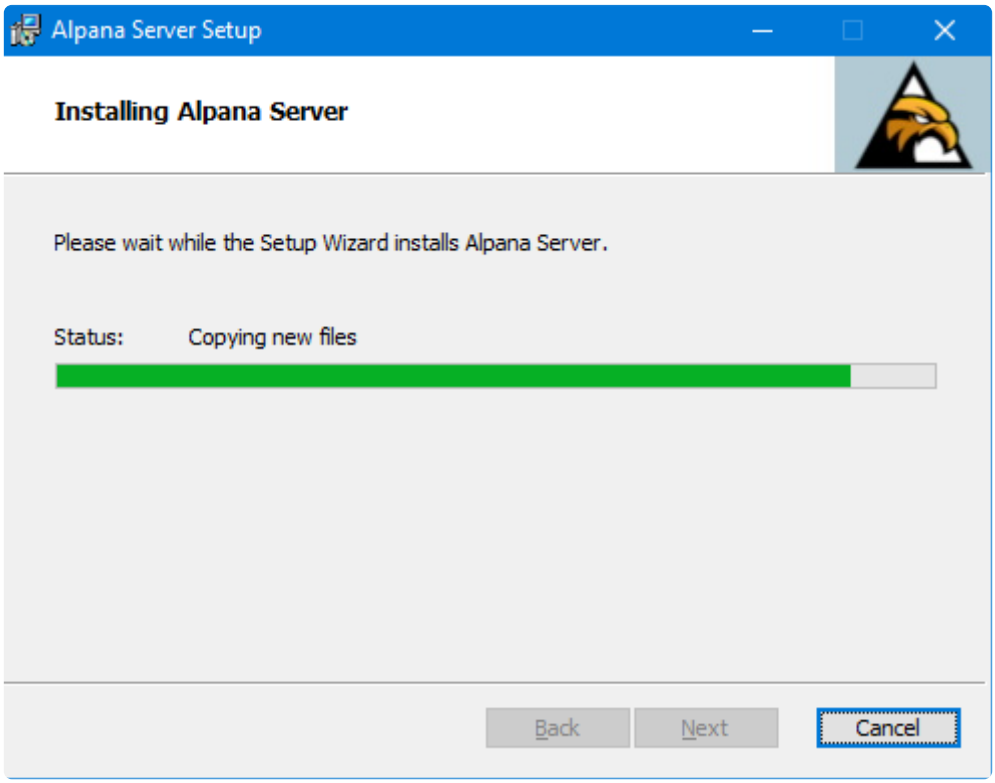
Note on SQL Authentication

Please note that when selecting SQL Authentication, the SQL user for authenticating as the IIS Application Pool of the Alpana Server (`IIS APPPOOL\Alpana3App`) will not be created in the SQL Server.
If you need some dashboards to be able to use Windows Authentication in their database Connections, then you will need to manually create the SQL user for `IIS APPPOOL\Alpana3App` and give it the necessary permission to the databases of those Dashboards.

Completing the installation

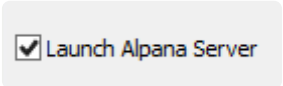
Confirm the configuration by clicking *Install* :





Launch Alpana Server

Upon completion, you can opt to *Launch Alpana Server* :



This will open your default web browser to navigate on the Front-End.

If you un-check this option, you may still start the websites at a later time.

Installation troubleshooting

If you encounter some issues during or after installation, see Installation Troubleshooting and contact Support.

Next Steps

After successfully installing, you will have perform the Post-Installation steps.

> ! Application startup takes some time, this is normal.
> When starting the IIS website, your browser should display a “loading” animation for some time (usually about 1min).

Last modified: Aug 26, 2020

3.2.3. Anti-virus and Firewall set up

Ports

Reaching Alpana Server

Alpana Server is composed of 2 websites that must both be reachable by the web browser on the client machines : the_Back-End Site_ and the *Front-End Site*
The ports number are defined during installation.

Allowing Alpana Server to get data

Alpana Server will make queries to data stores, as defined within the Connections and Data Sources inside the Dashboards.
Please take the necessary steps to allow this data access (open ports to reach SQL Server or Historian on a distant machine for example).

Anti-virus

Alpana Server may make queries to external IP addresses (for fetching data from data sources, for example on port 1433 for SQL).
This may be detected as suspicious behavior by some security software.
For normal operation, you should let your security software allow this access to Alpana Server.

Last modified: Aug 26, 2020

3.2.4. Post-Installation

The steps required to activate and configure the application after installation are listed below.

Startup

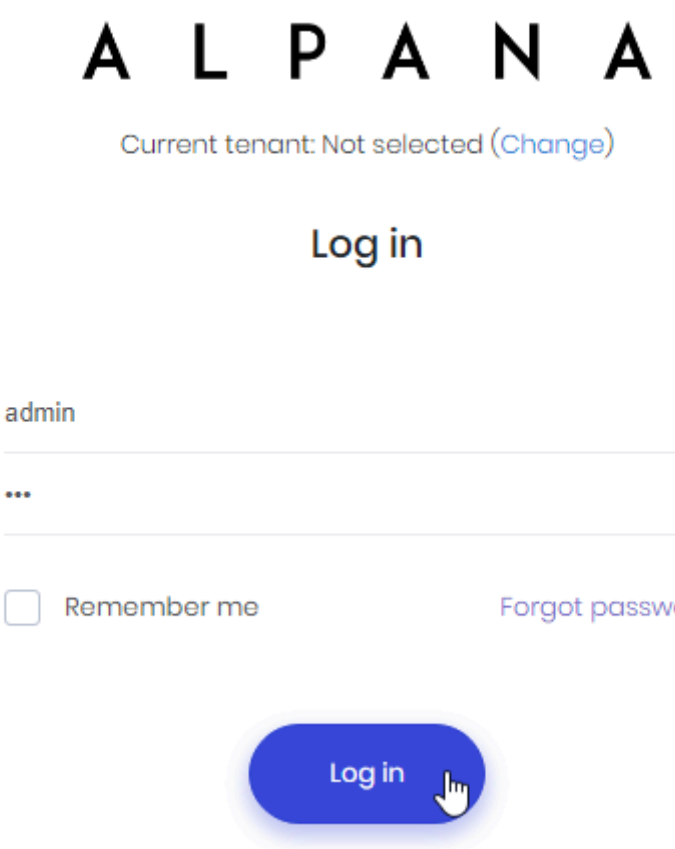
If you didn't start the website during the installation procedure, manually start the `Alpana3App` and `Alpana3Client` websites from Microsoft IIS Management Console.
Open your web browser to point to the `Alpana3Client` website.

> ! Application startup takes some time, this is normal.
> When starting the IIS website, your browser should display a "loading" animation for some time (usually about 1min).

First login to the Host

When Alpana Server is started for the first time, you can login to the Host using the following credentials :
User name : `admin`
Password : `123qwe`

> ! Important : As with any default password, please change this password as soon as possible !

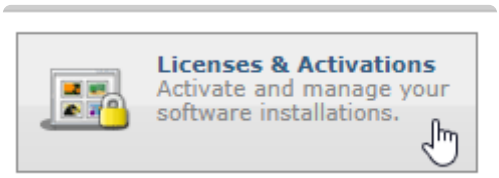


License Activation

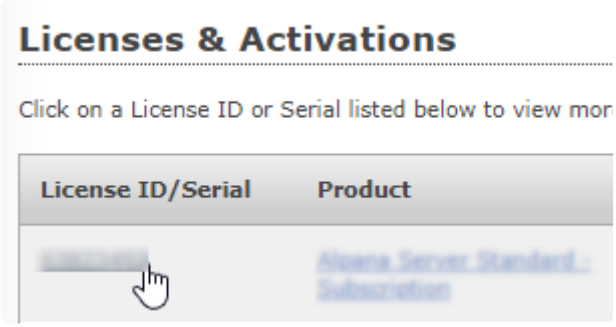
You may now activate your license.
Without this step, you won't be able to use your installation.

Browsing your license portal

Login to the web License Portal.
Navigate to *Licenses & Activations* :

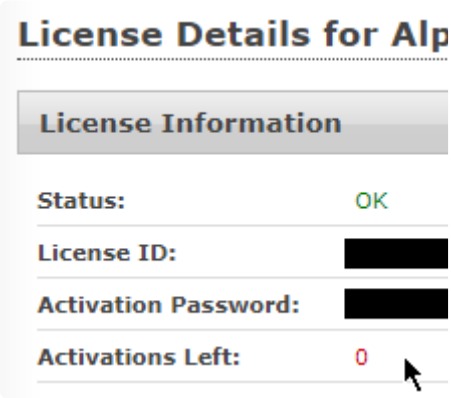


Under *Licenses & Activations*, click the *License ID* you want to activate :



Important note

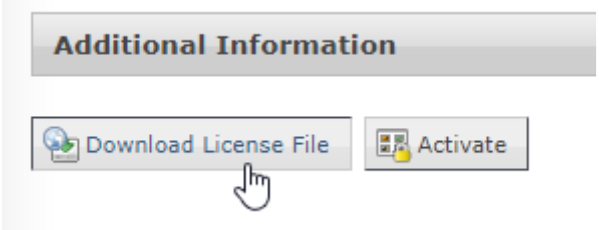
> ! Before starting the license activation, please make sure that you have enough Activations left !
> If your *Activations Left* is 0, then your license is probably used elsewhere. Please contact your Software Distributor for support.



> ! To continue, the *Activate* button should be clickable.

Getting the License File

At the bottom, click the button *Download License File* :

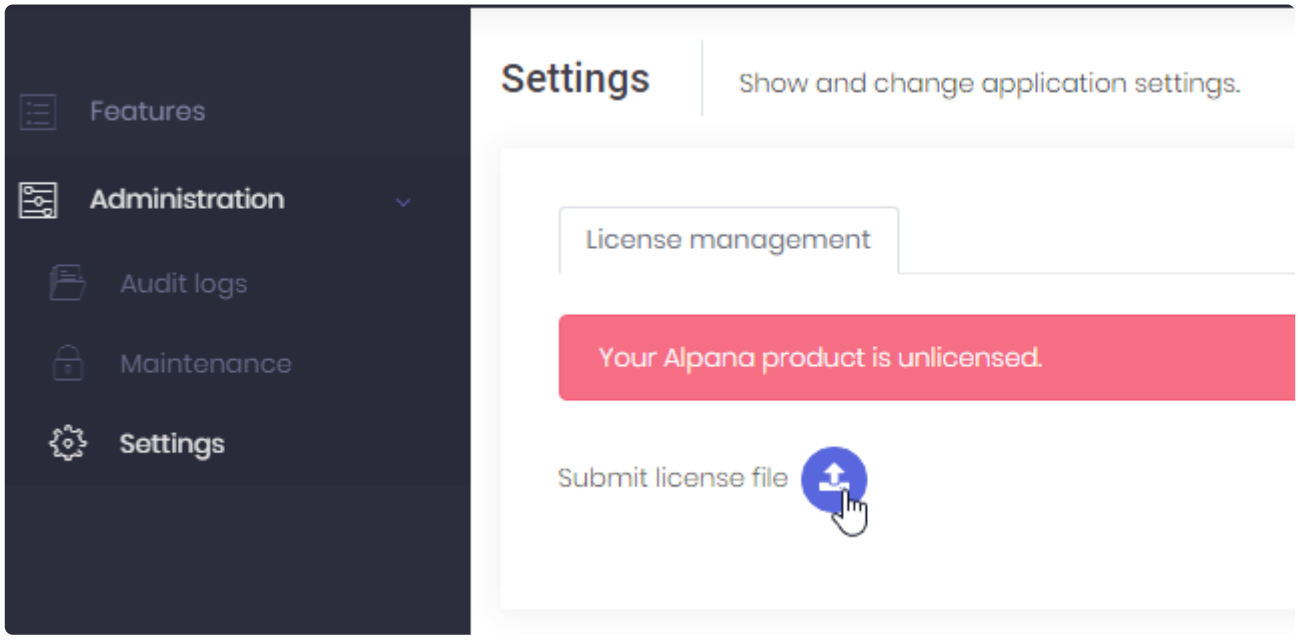


⇒ A file `AlpanaServer.lic` is downloaded.
Move this file to your Alpana Server installation.

> You can leave this web page open, it will be used below to complete the Activation.

Activating the License File

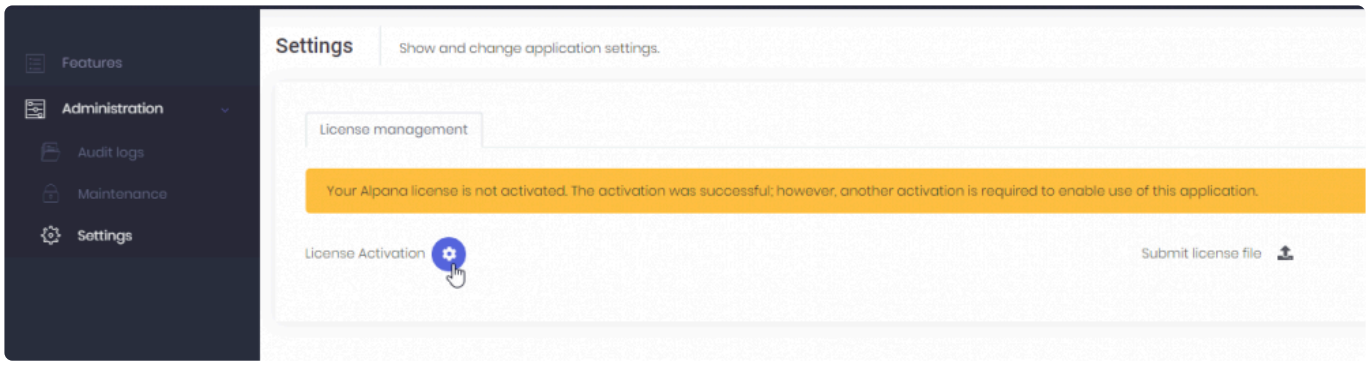
On the Host of your Alpana Server, navigate to *Administration* > *Settings*, tab *License Management*.
Click *Submit License File* and upload the `AlpanaServer.lic` file :



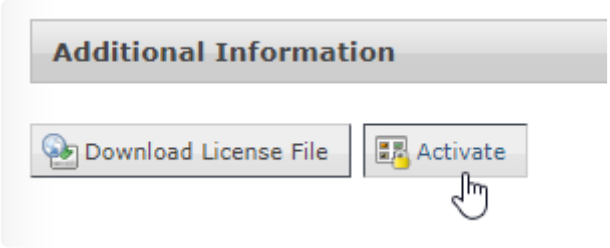
⇒ The first step of license activation is complete, but your license is not yet activated.

Activating the License

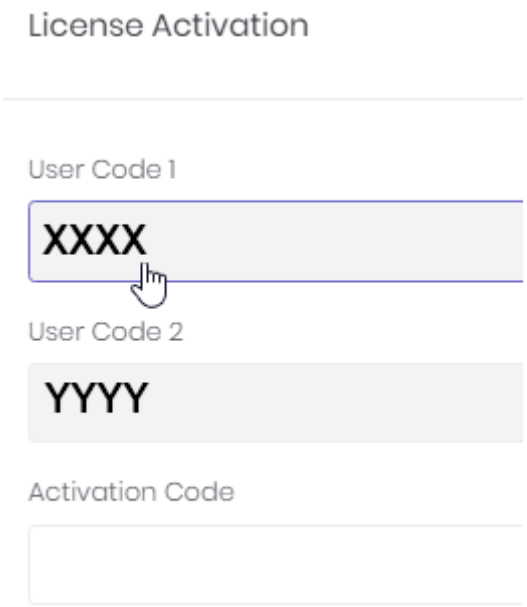
On the Host of your Alpana Server, click *License Activation* and leave the pop-up window open :



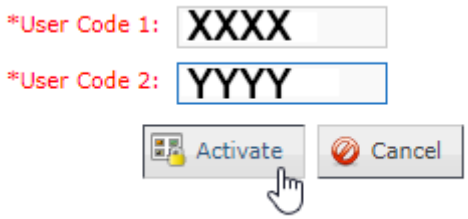
On the web License Portal, click *Activate* and leave the pop-up window open :



On the Host of your Alpana Server, copy the *User Code 1* and *User Code2* :

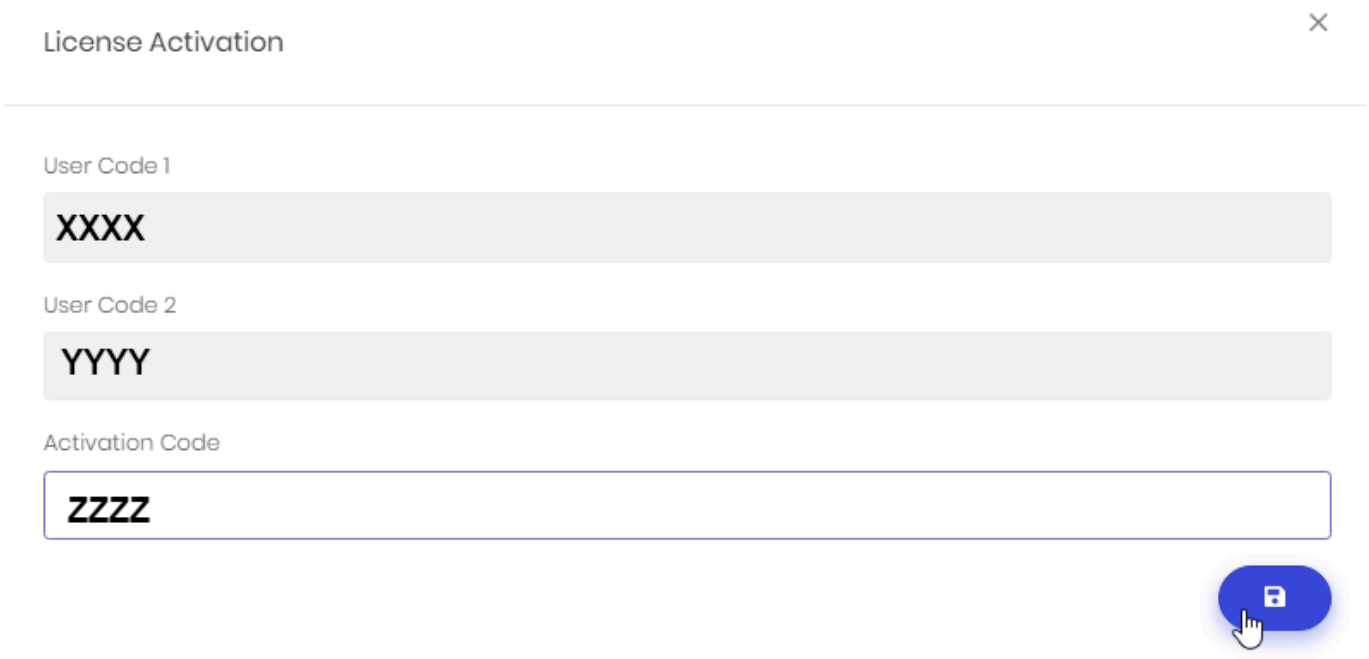


…and paste them in the correct fields in the web License Portal, and click *Activate* :



⇒ As a result, the web License Portal displays an *Activation Code* :

Copy the *Activation Code* and in the Host of Alpana Server, paste it and click to save :



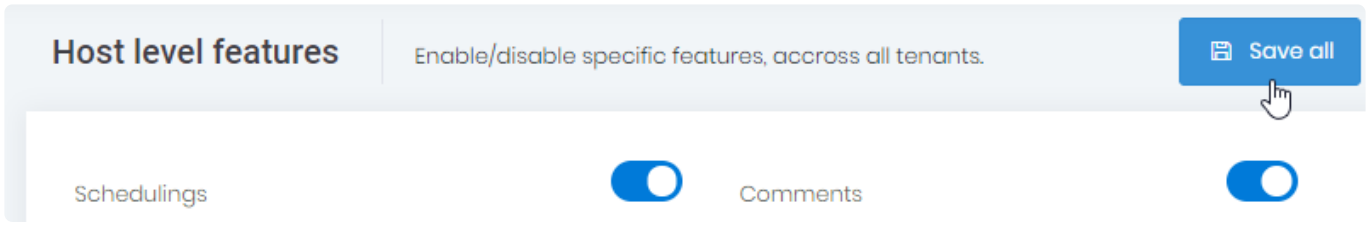
⇒ The License should be activated.

> If the License Activation was successful, you may delete the `AlpanaServer.lic` file. It cannot be used later, and keeping it would only create confusion during a later activation.

Enabling the features

Depending on your license type, some features may be activated at the Host level.
Login to the Host and browse to *Features* :

Check the required Features and click *Save* :



> Important : After changing the allocated features, you must restart the application (from IIS Manager).

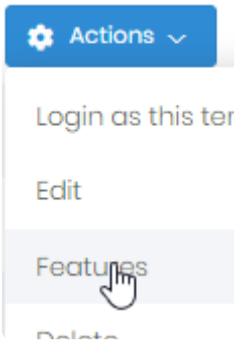
Then, in order for Tenants to benefit from the features, you must allocate them.

Assigning features to the Tenants

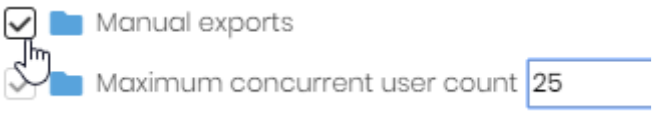
You will have to assign license features to the 'Default' Tenant.

By default, 1 Tenant exists, called "Default". If your license allows only 1 Tenant, this is it !
Otherwise you may need to create your other Tenants.

On the Host, Navigate to *Tenants* management.
Then on the desired Tenant, select *Action* > *Features* :



In the *Features* window, check which Features to allocate, including resources, and click *Save* :



don't forget to allocate at least 1 Concurrent User, 1 account, and 1 Dashboard !

What's next ?

Login to a Tenant

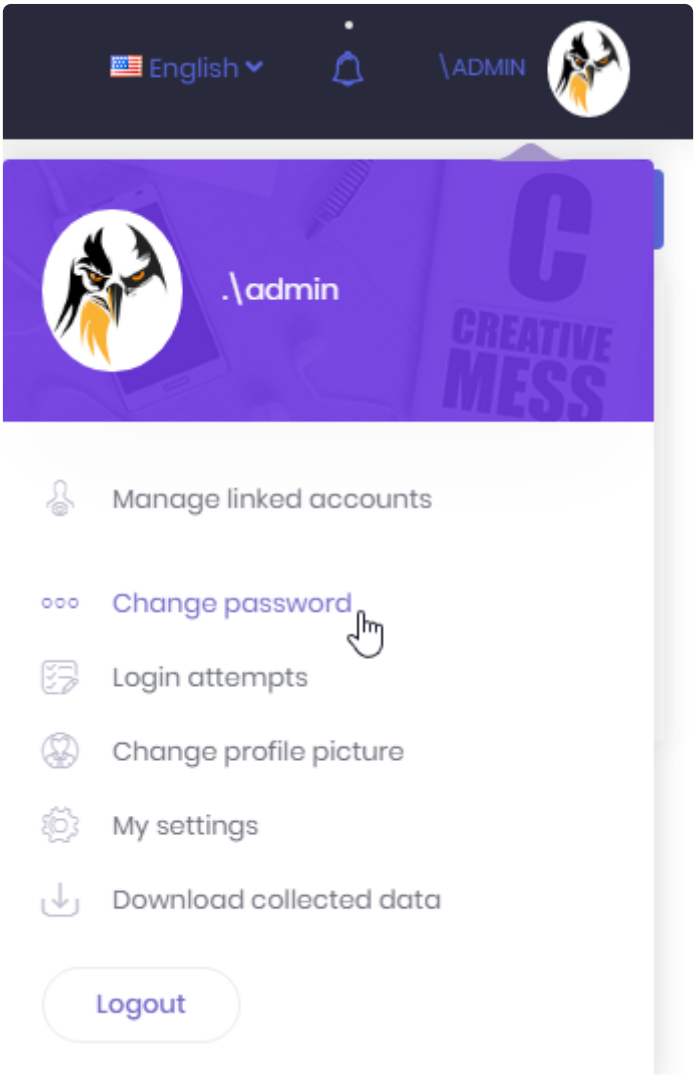
By default, 1 Tenant exists, called "Default".
You can login to this Default Tenant using the following credentials :
User name : `admin`
Password : `123qwe`

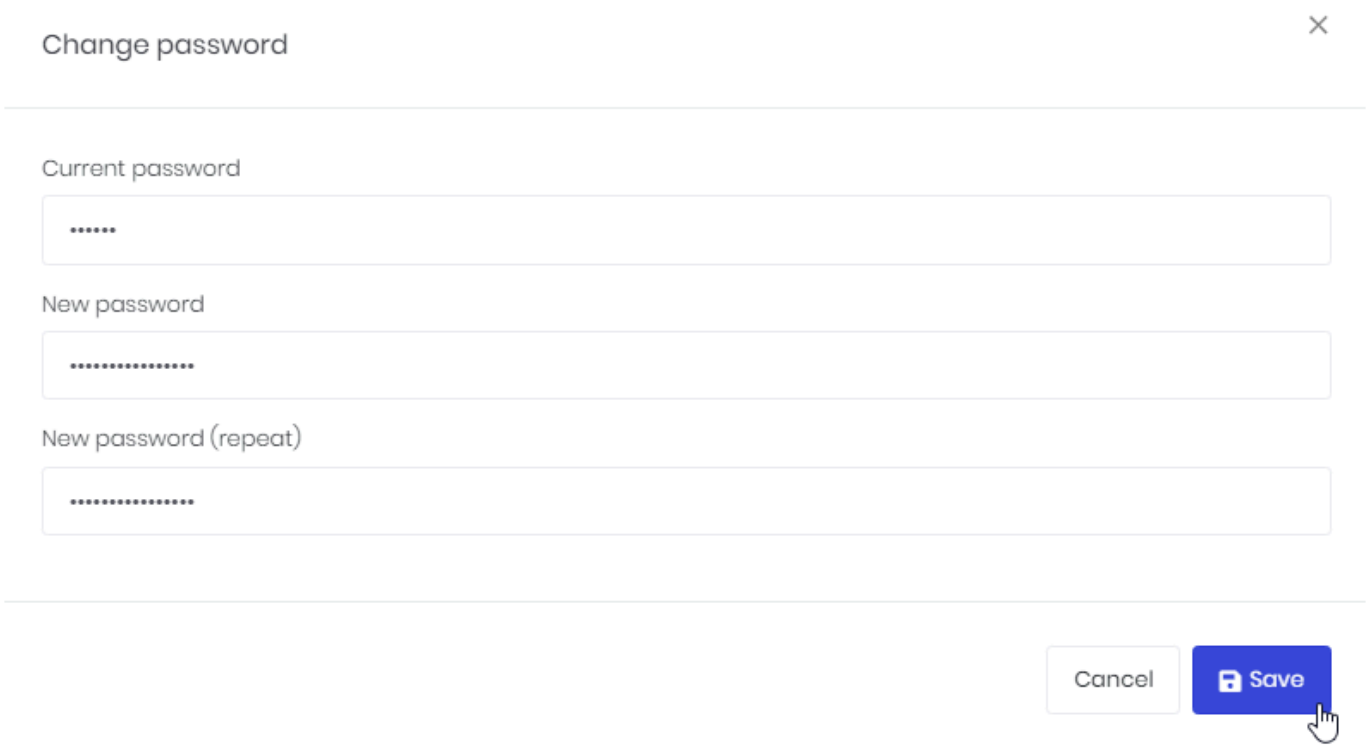
Note : this user is **not the same** as the "admin" user of the Host : it is 2 different accounts.

> ! Important : As with any default password, please change this password as soon as possible !

Changing default passwords

To change the password, click on your “admin” username on the top right, and in the menu select *Change Password* :





> Note :
> There are 2 accounts called `admin` that have the same default password `123qwe` :
> - the admin account of the Host
> - the admin account of the “Default” tenant
> They are different accounts.
> Don’t forget to **change both passwords**.

Create Categories

Create Categories on the Tenant.
This will allow users to publish Dashboards.

Configure the application

There are several configuration options as an Administrator, please see the Administration pages.
For example, you may wish to Configure e-mail settings.
This will allow to invite users by e-mail, let them recover their passwords, and more.

Create users

Create Users and Roles to let users access Alpana Server.
Configure the relevant security options.

Last modified: Aug 10, 2020

3.2.5. Un-Installing

Tips

Don’t lose your license !

Before un-installing, perform a license de-activation in order to be able to re-use your license later.

Avoid a reboot

If you don’t stop the application, Windows will probably require you to reboot.
If you don’t want to reboot, you must stop the application before uninstalling.
To completely stop the application, you may need to stop the IIS server itself.

⇒ If you do this, don’t forget to start IIS again after uninstallation.

Uninstall

You can un-install Alpana Server using Windows Control Panel.

Last modified: Jan 31, 2020

3.3. Login and User Profile

Alpana Server allows users to login to the application to share content.
This section explains how to login and how to use the user-specific features.

Last modified: Oct 22, 2019

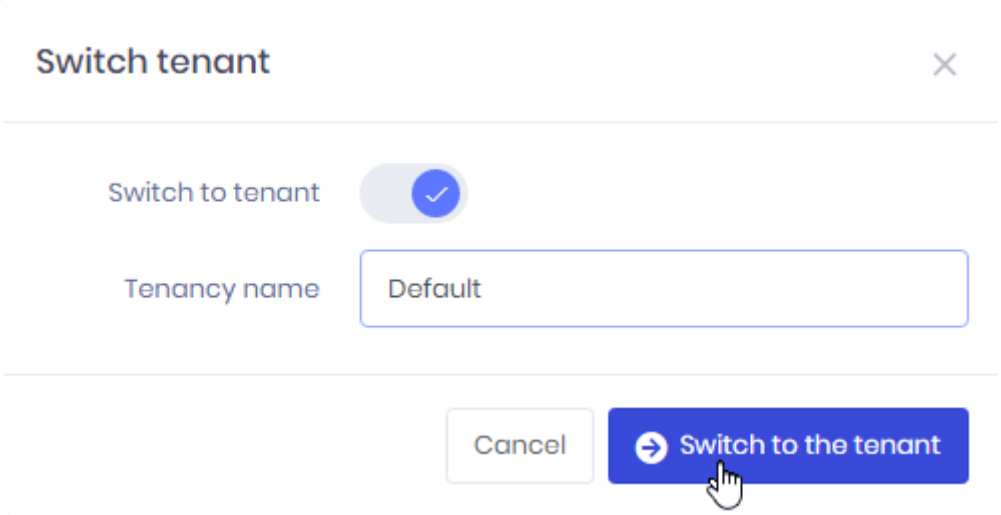
3.3.1. Selecting a Tenant or the Host

The login screen allows to select which Tenant to connect to, or allows to connect to the Host.

Selecting a Tenant

To login to a Tenant, you must first enter that Tenant's name on the login screen.
Click on "Current Tenant" : *Change* :

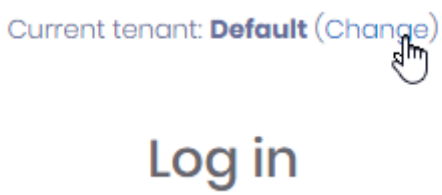
In the popup, write the name of the Tenant you wish to connect to :



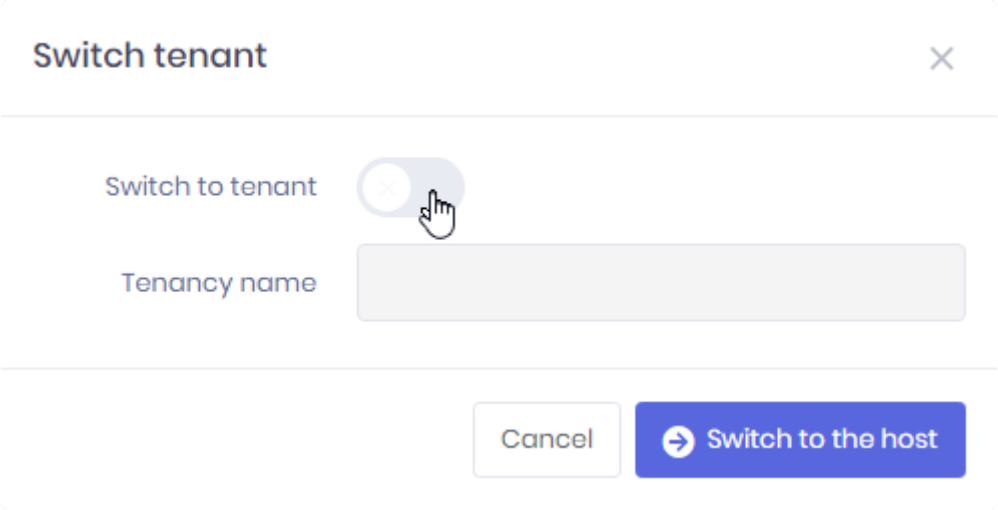
Selecting the Host

To login to the Host, you must make sure that "Current tenant" is "Not selected".
Not selecting a Tenant means you will login to the Host.

To switch from a Tenant to the Host, click on "Current Tenant" : *Change* :



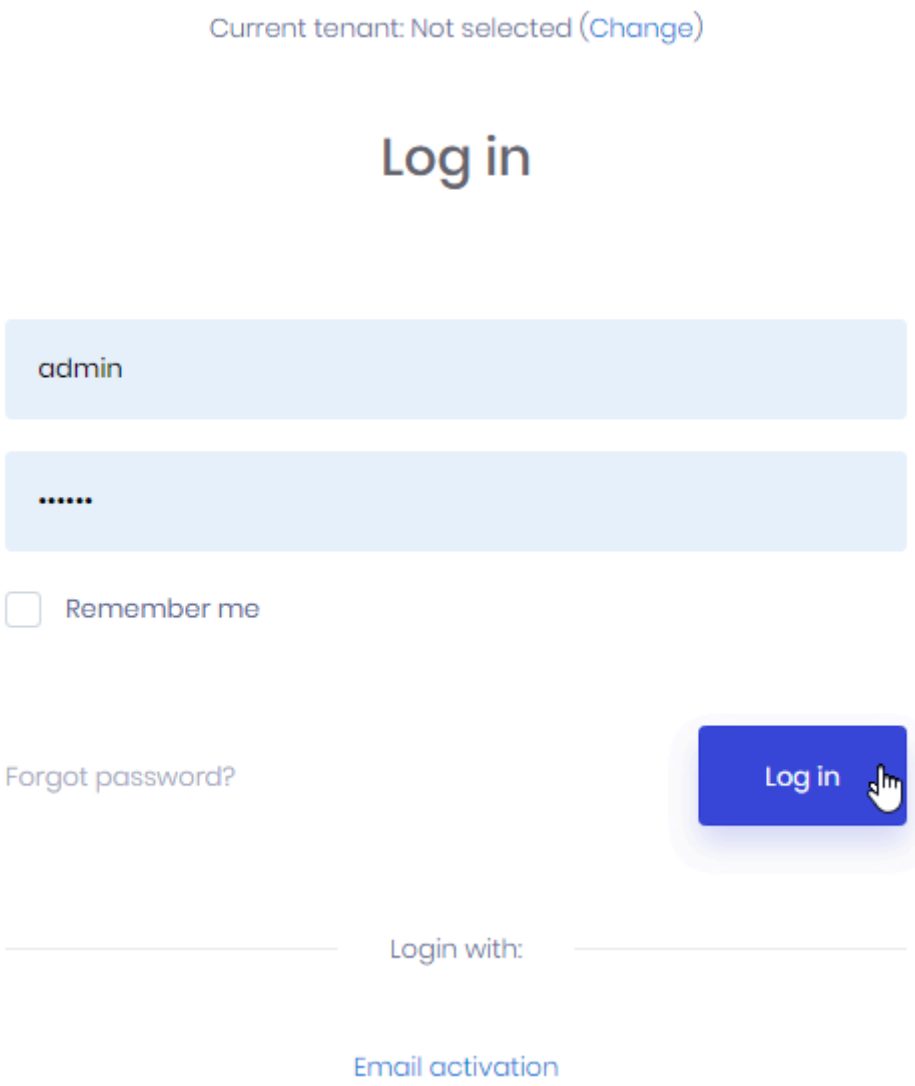
Then un-check *Switch to tenant* and click *Switch to the host* :



Last modified: Nov 04, 2019

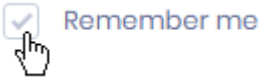
3.3.2. Login options

The login screen allows several options :



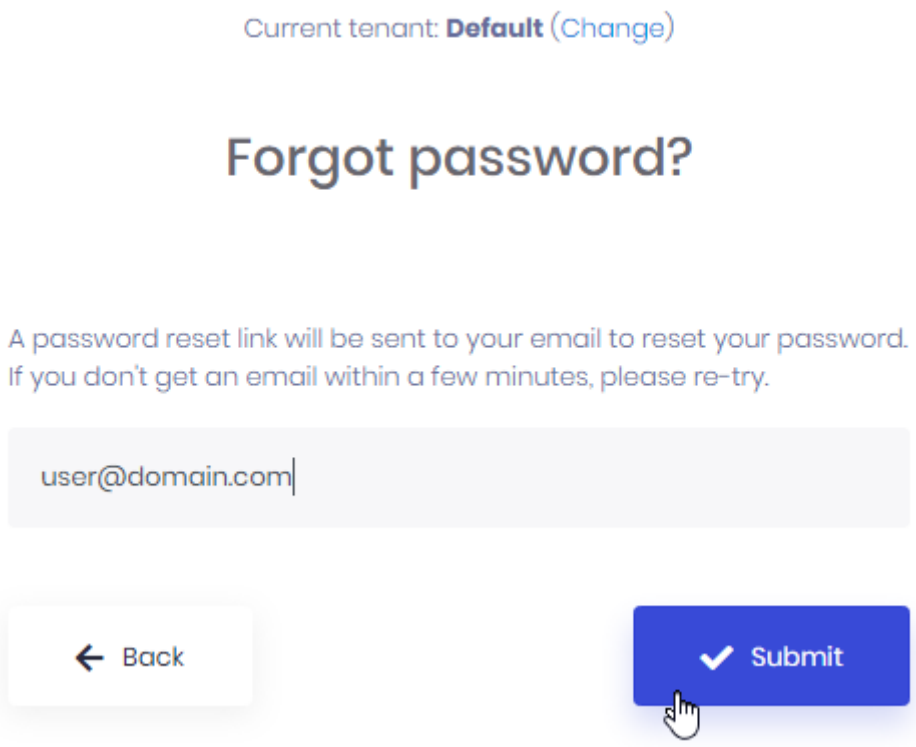
Remember Me

Check this box if you want to stay logged-in even when the web browser is closed :



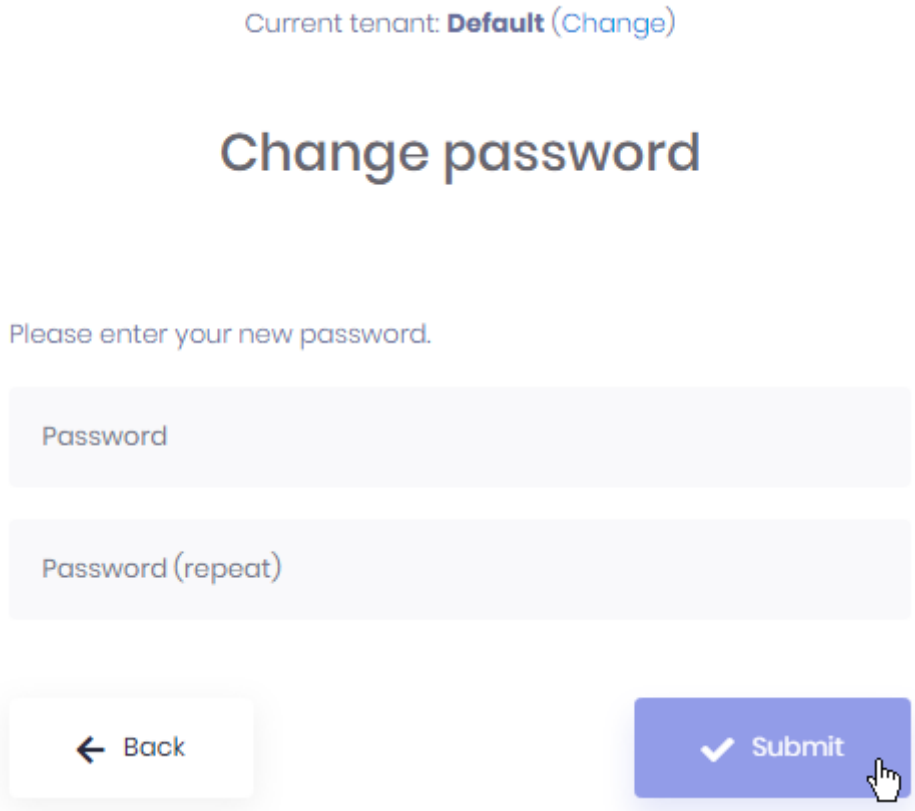
Forgot Password

If you forgot your password, you can click “*Forgot password ?*” and enter your e-mail address to receive a password reset link :



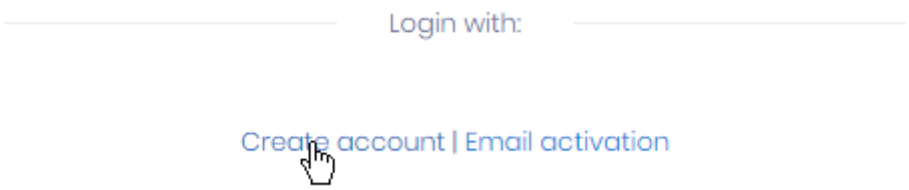
Note : your user must already exist with the corresponding e-mail address.

When you click on the *Reset* link in the e-mail, you will be able to provide a new password :

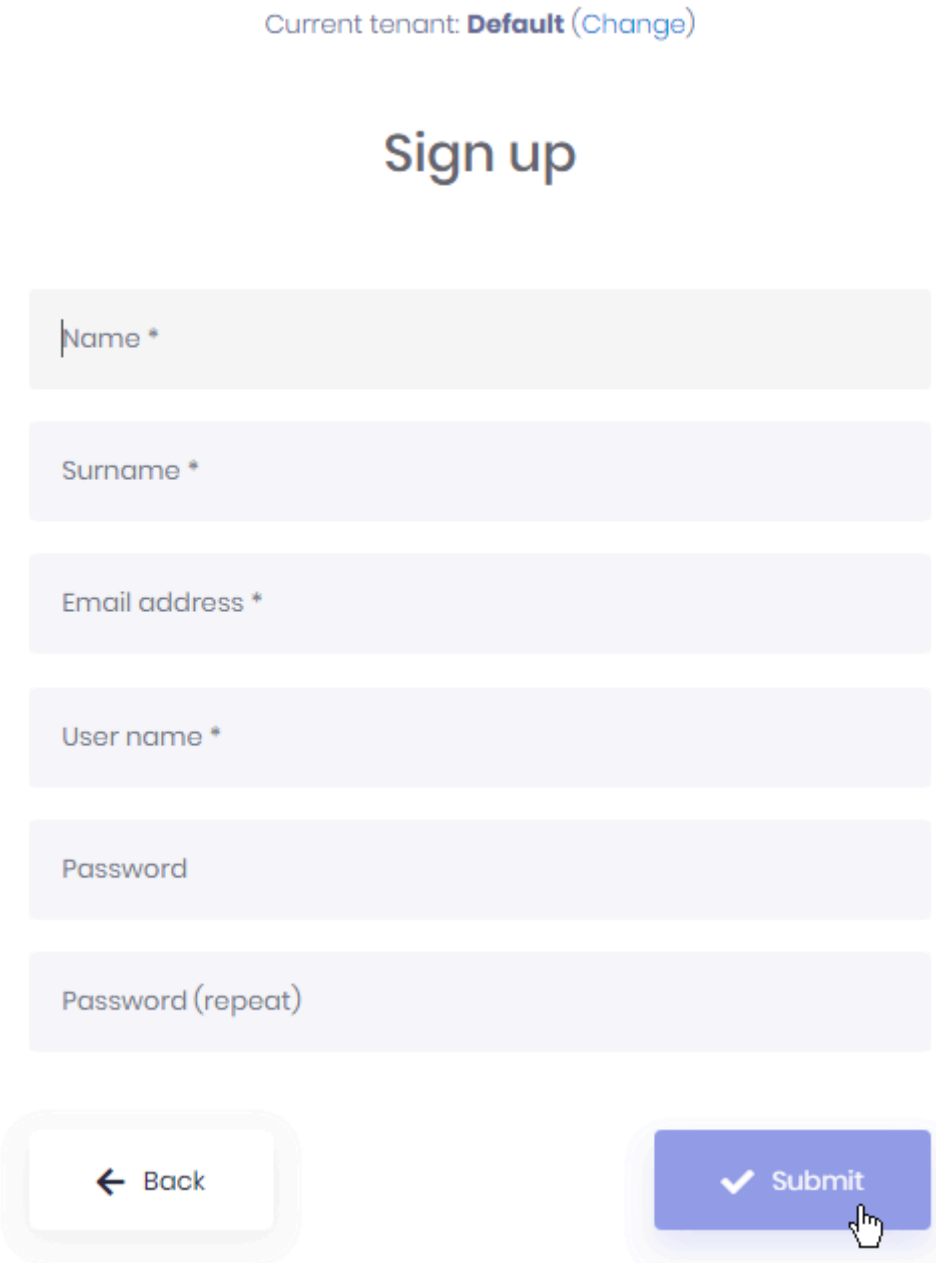


Creating an account from the login page

Your administrator may have allowed new users to sign up by themselves to create their own account. To do this, click on *create account* :



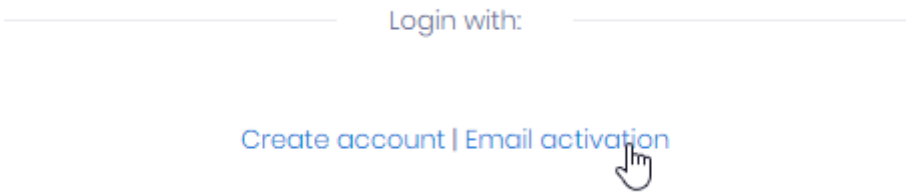
Then fill-out the form :



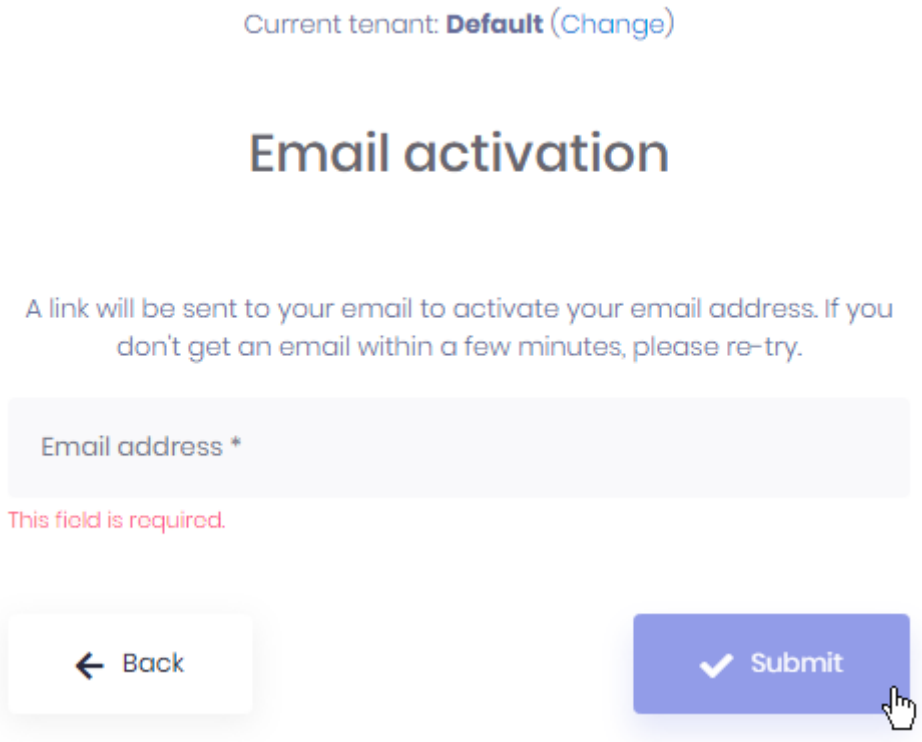
Sending email Activation from the login page

Your administrator may require you to confirm your e-mail address before logging-in.
If you didn't receive an activation e-mail, you can re-send it from the login page.

Click *Email activation* :



Then fill-in your e-mail address to receive the activation e-mail :



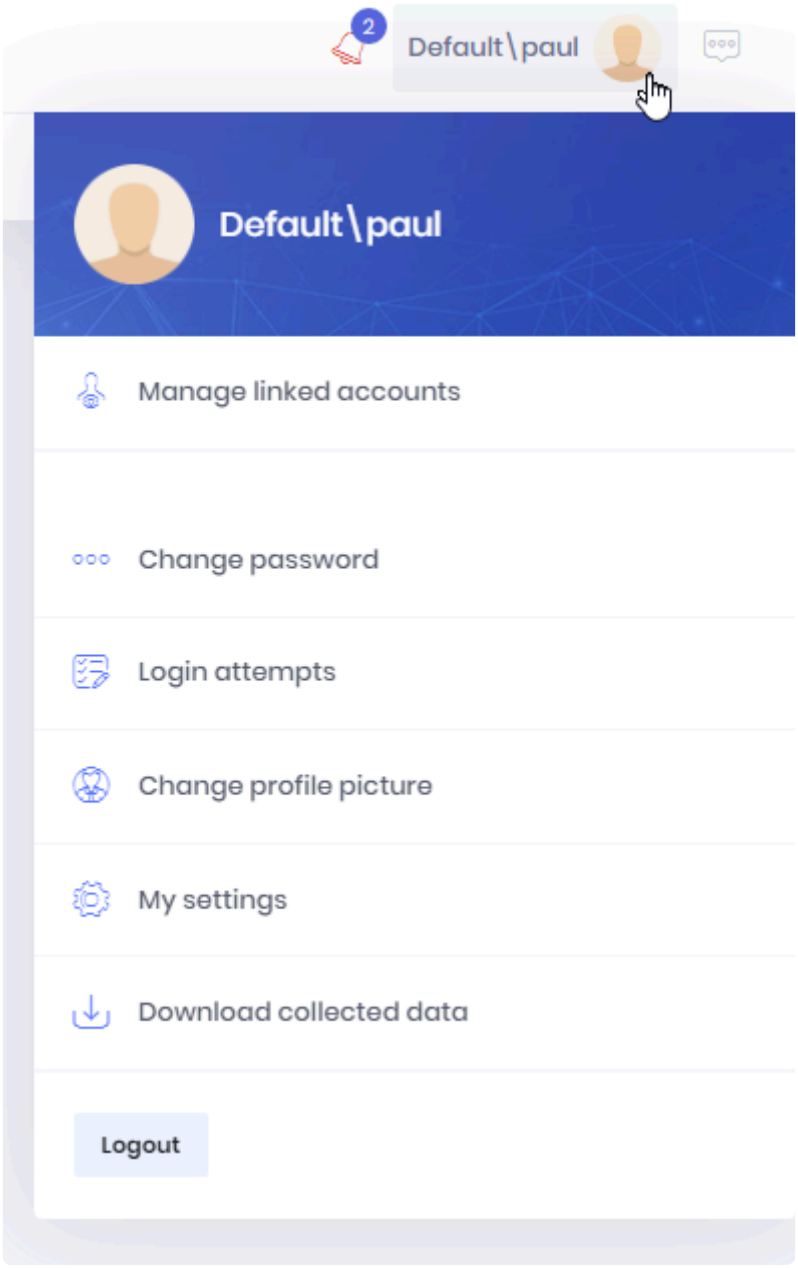
Note : your user must already exist with the corresponding e-mail address.

Last modified: Oct 22, 2019

3.3.3. User Settings and Logout

When logged-in, you can see your user name and current tenant like this :
`tenant\username`
If you are currently logged-in to the Host instead of a Tenant, there is nothing left of the `\` :
`\username`

Click on your user name to open the user profile popup and manage your settings :



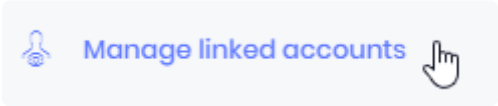
Click on *Logout* to be disconnected.

Last modified: Oct 22, 2019

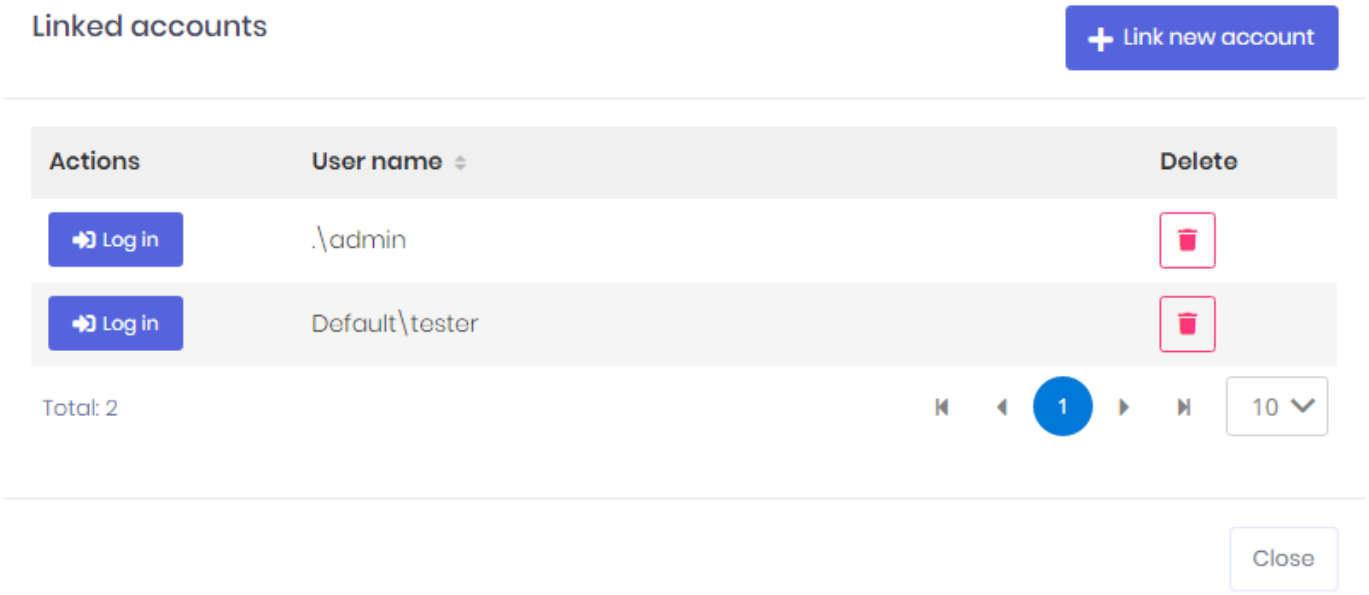
3.3.4. Linked Accounts

If you have accounts on several Tenants or the Host, you can Link them together.
This allows you to switch between accounts and Tenants without having to logout, enter credentials, and login.

To manage Linked Accounts, open your user settings and click *Manage Linked Accounts* :

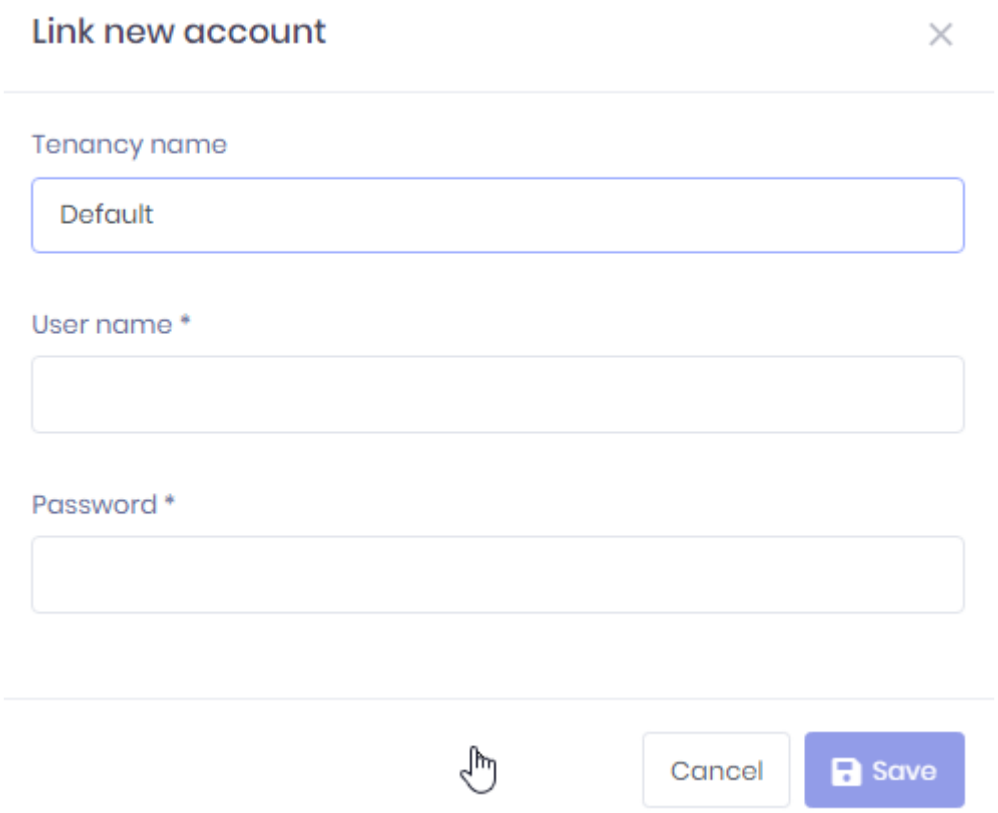


This opens the *Linked accounts* pop-up :



Link new account

To add a linked account, click *Link new account*.
Fill-in the form for the account you want to link to :



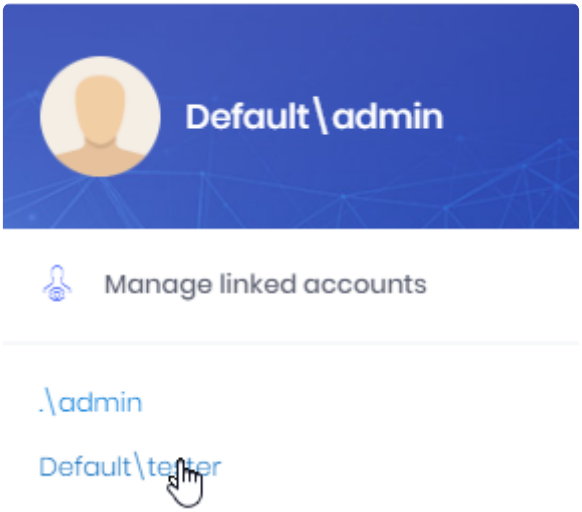
Removing a linked account

If you want to un-link an account, click the trash icon in front of it.

Login as linked account

To login to a linked account, click the *Login* button in front of it in the *Linked accounts* pop-up.

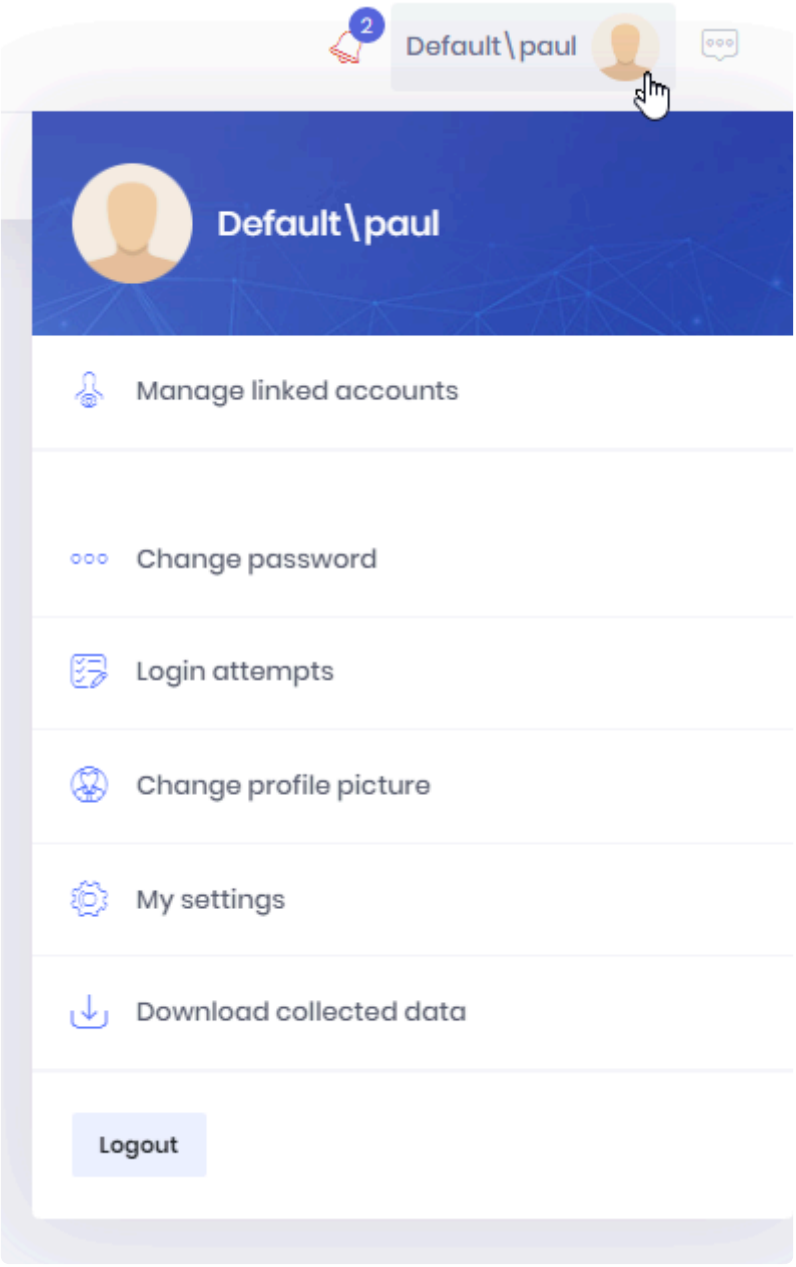
There is also a shortcut in your user settings :



Last modified: Oct 22, 2019

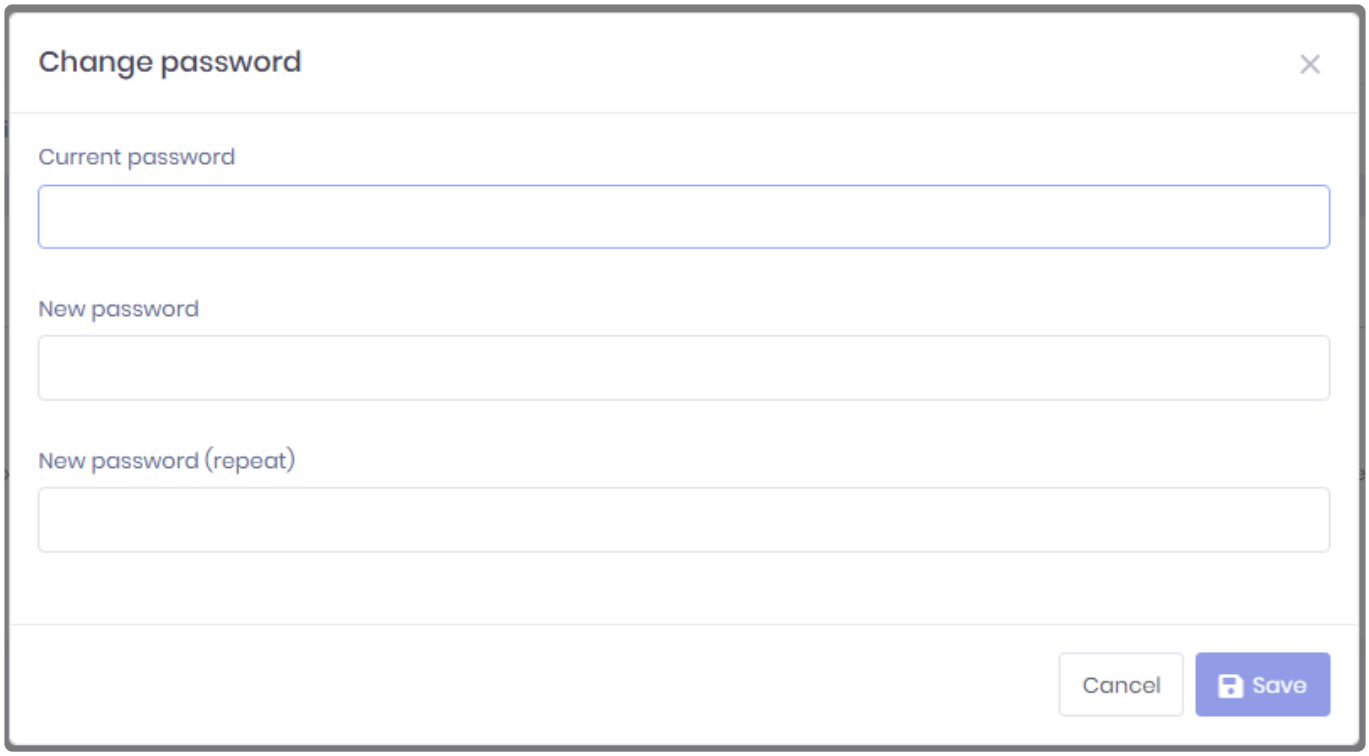
3.3.5. Account details and password

Your user profile popup allows you to manage your settings :



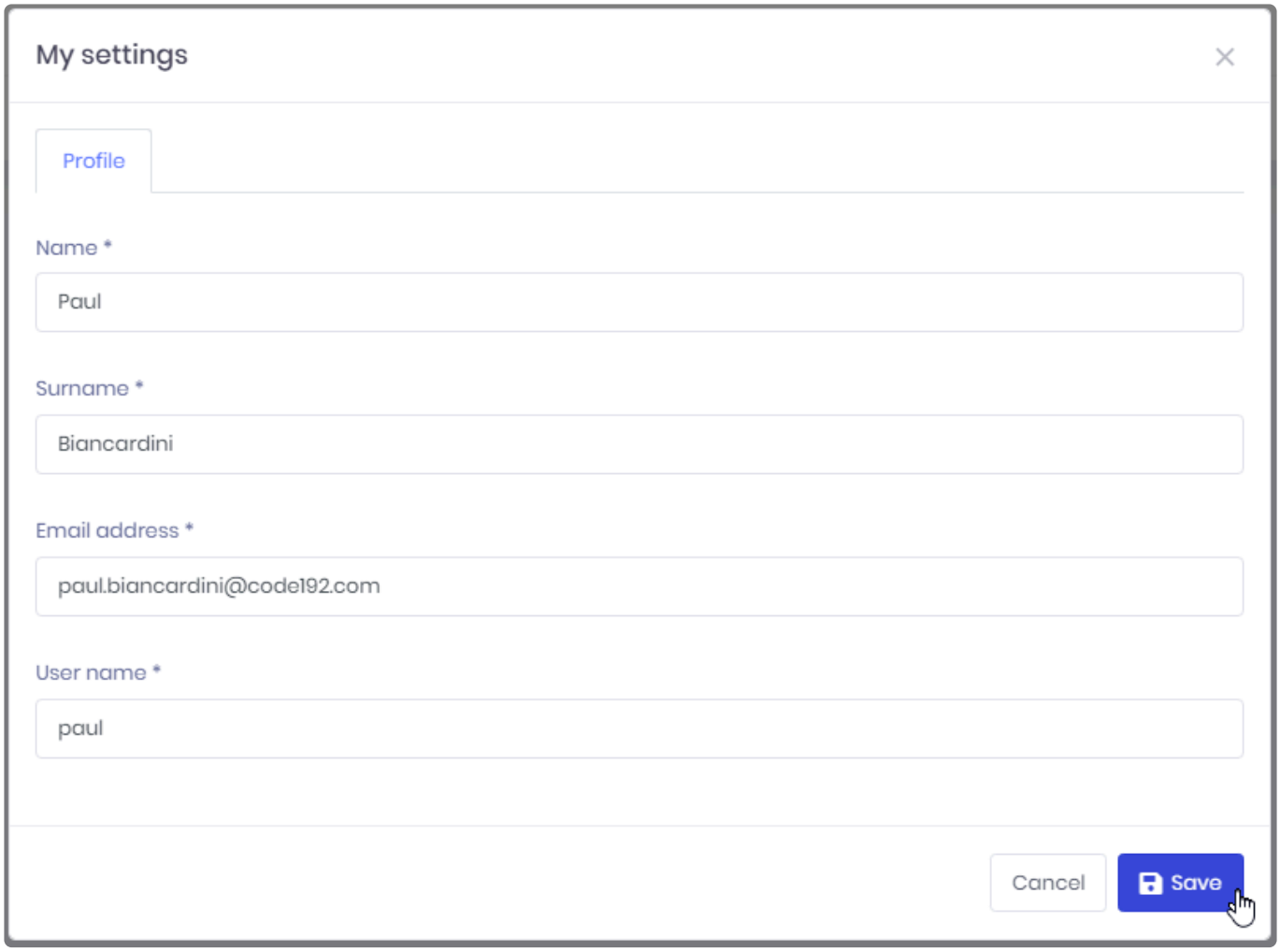
Change Password

Click *Change password* to update your password :



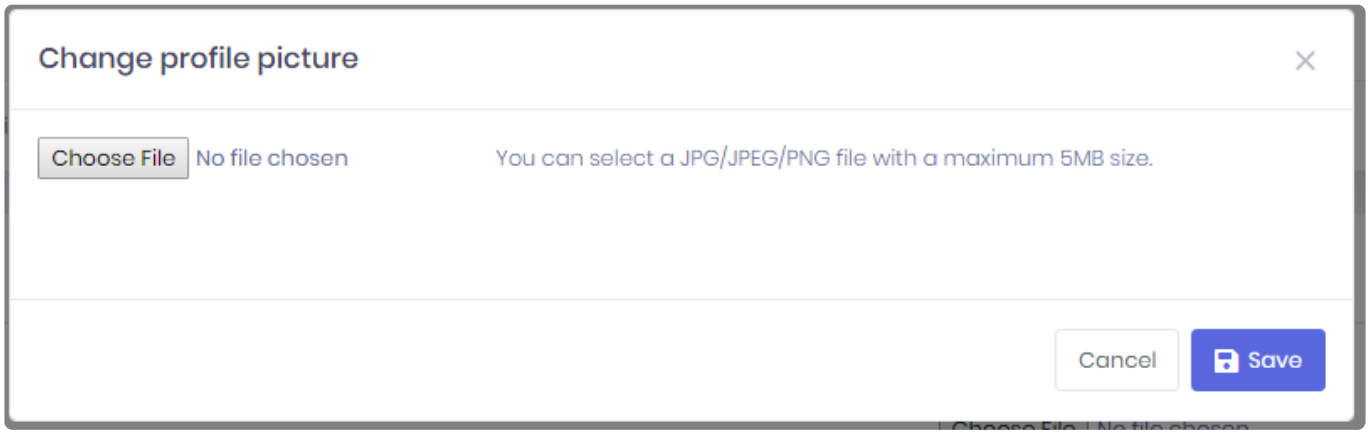
Update Profile

Click *My Settings* to update your profile (name, e-mail, user name) :



Change profile picture

Click *Change profile picture* to update your profile picture :



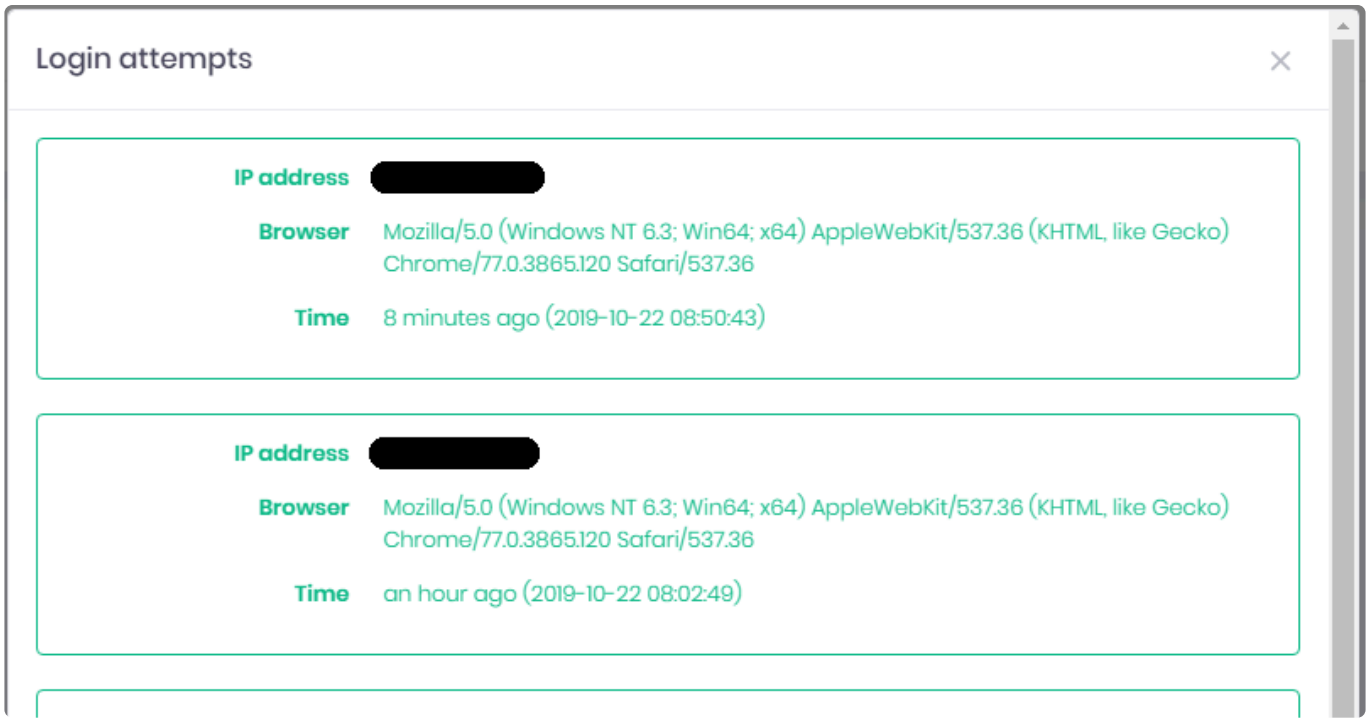
Last modified: Dec 26, 2019

3.3.6. Account Security and Privacy

Your user profile popup allows you to manage your Security and Privacy settings :

View login attempts

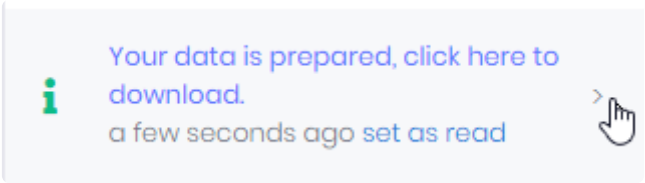
For security purposes, you can review login attempts to your account.
Click *Login attempts* under your profile to view the attempts, with the corresponding IP addresses and User Agent :



Download collected data

For privacy purposes, you can download a comprehensive list of the data that was collected by Alpana Server during your usage (profile information, chat history, …)

Click *Download collected data* under your profile.
Alpana Server prepares your data, and notifies you when the download is available :

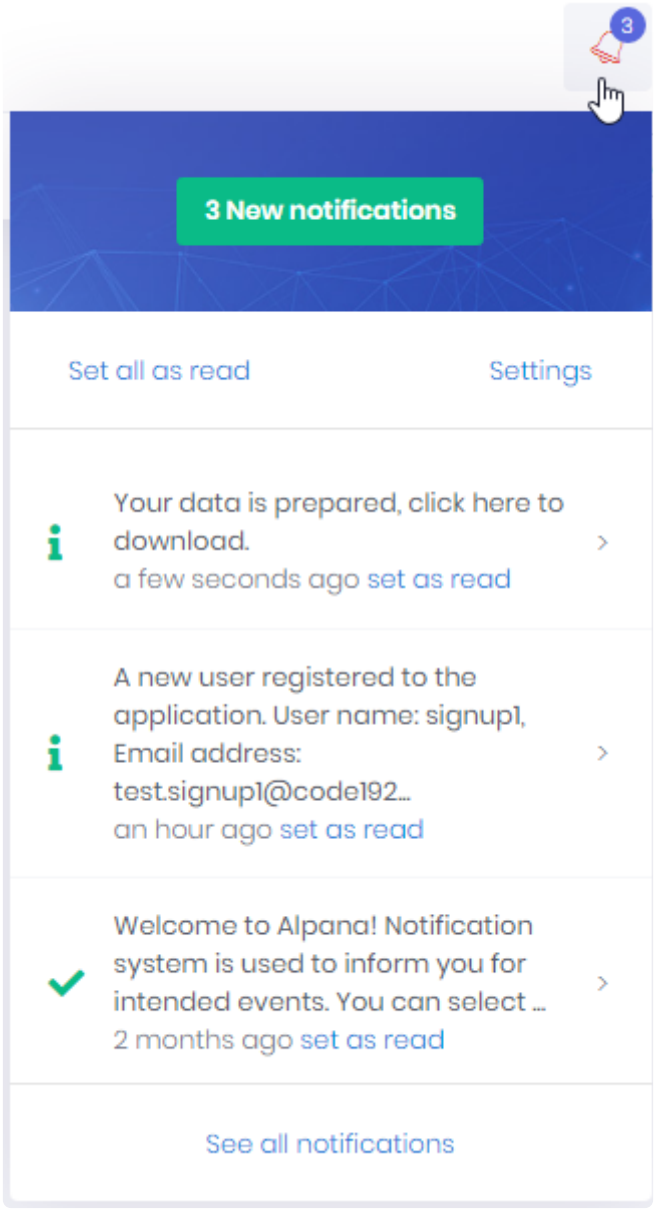


Last modified: Oct 22, 2019

3.3.7. Notifications

Alpana Server includes a notification system to let you be aware of events that are relevant to you.

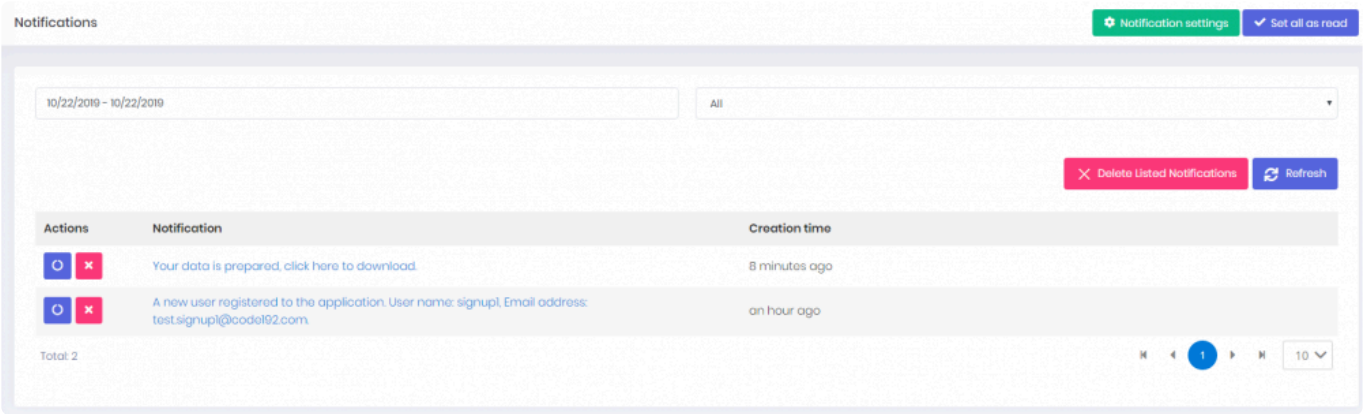
Click the bell icon to open the Notifications pop-up :



Viewing notifications

New notifications appear with a number on the bell icon, and a message in the Notification pop-up.

Click *See all notifications* to see a full list of your notifications in a dedicated page :



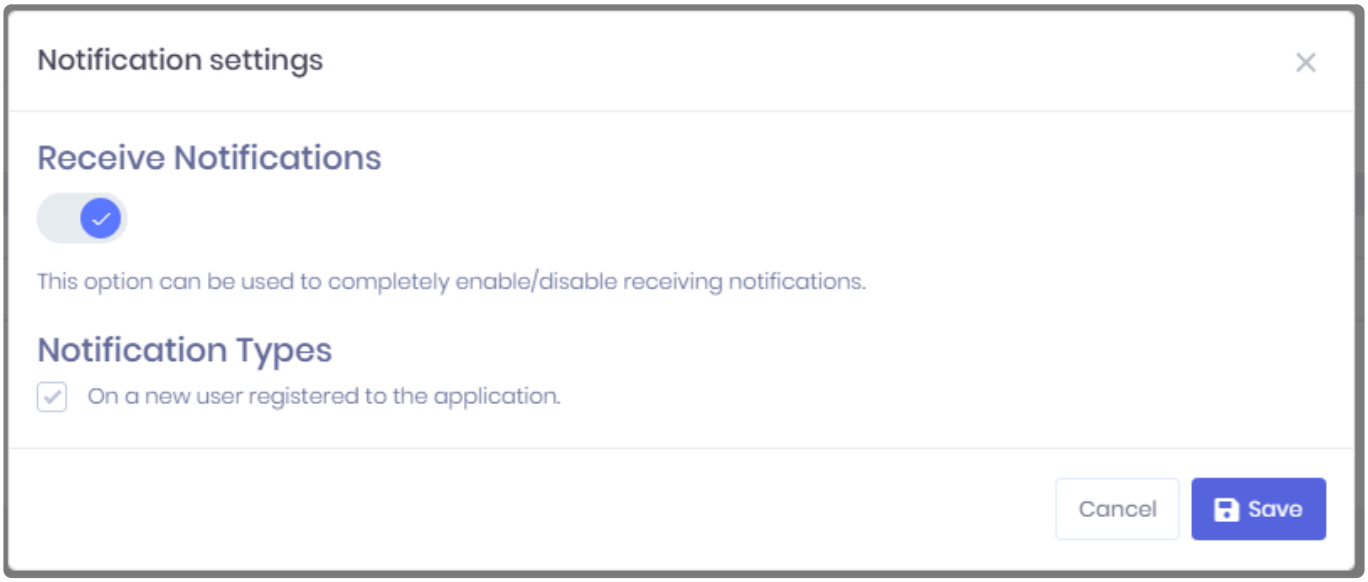
In this page, you can browse notifications and filter by date and “read/unread” status.

Delete / Mark as read

From the Notifications popup or from the Notifications page, you can *Set all as read*, or individually *set as read* a notification.
From the Notifications page, you can *Delete Listed Notifications*, or individually *Delete* a notification.

Notifications settings

From the Notifications popup or from the Notifications page, you can open the *Notification settings* :

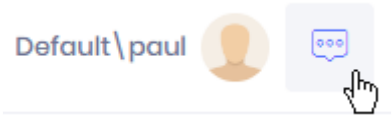


This allows to disable notifications, and select which notification types you wish to receive.

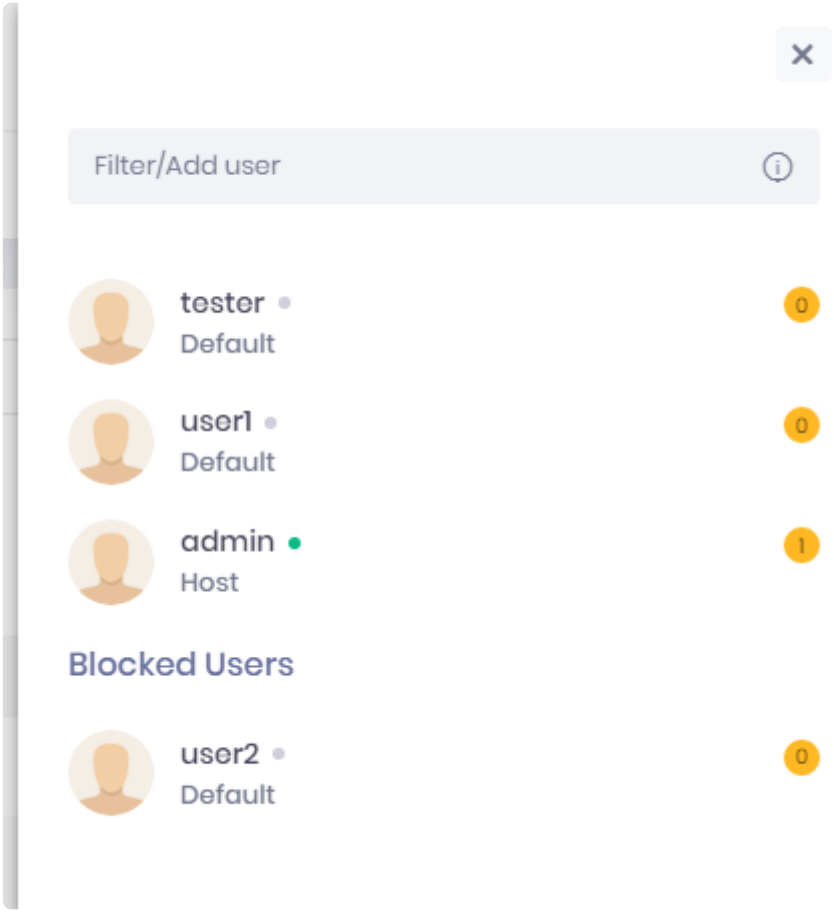
Last modified: Oct 22, 2019

3.3.8. Chat

Next to your user profile, you can click the *Chat* button to open the Chat panel :



The panel opens on the side with your list of contacts :



Click on the cross to close the panel.

Managing Contacts

Contacts list

The contacts list shows :

- your current contacts
- your blocked contacts
- their status (connected or not)
- the number of unread messages from that contact

You can filter your contact list by entering part of their User name or Tenant name in the *Filter/Add user* box.

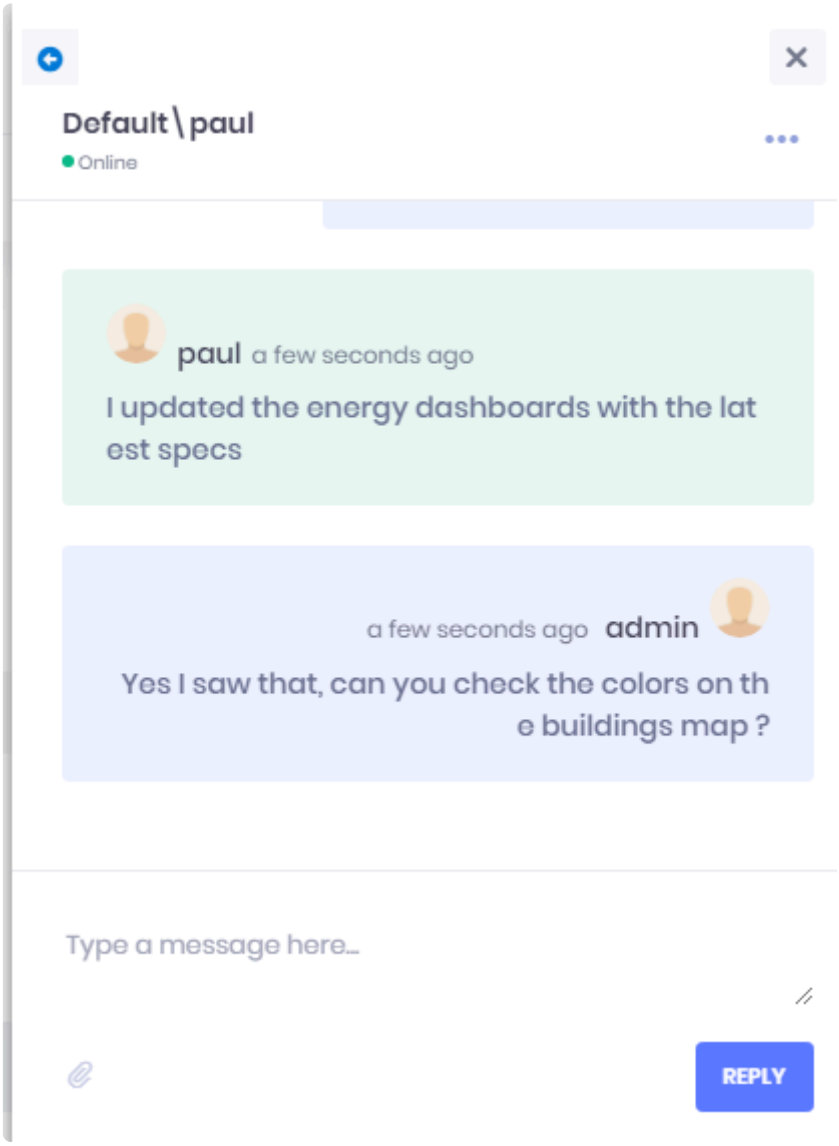
Adding contacts

To add a contact, enter the full Tenant name and User name in the *Filter/Add user* box with the following format :

- `username` : a user on the same Tenant
- `tenant\username` : a user on a different Tenant
- `.\username` : a user on the Host

Conversation

Click on a contact to open the Conversation view for that contact :

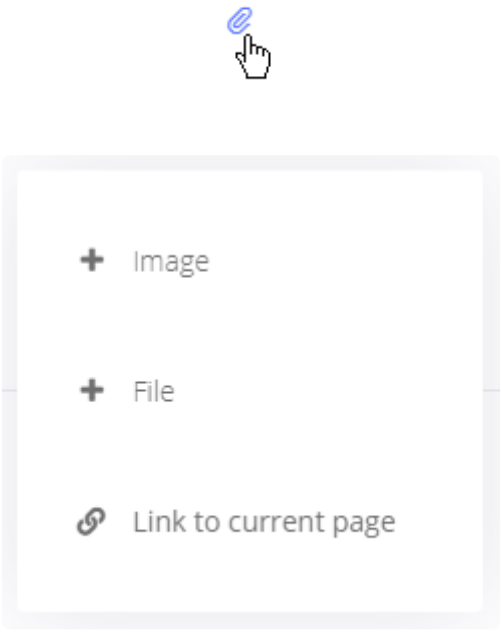


Return to the contacts list by clicking on the left arrow at the top.

Sending messages

You can send a text message by typing in the “Type a message here…” box and hitting *Reply* or the *Enter* key on your keyboard.

Click on the Attachment button to attach content to your message :

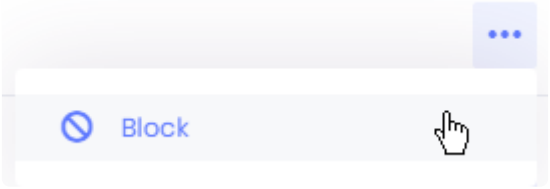


This allows to send :

- an image displayed inline
- a file attached to your message
- a link to the current page

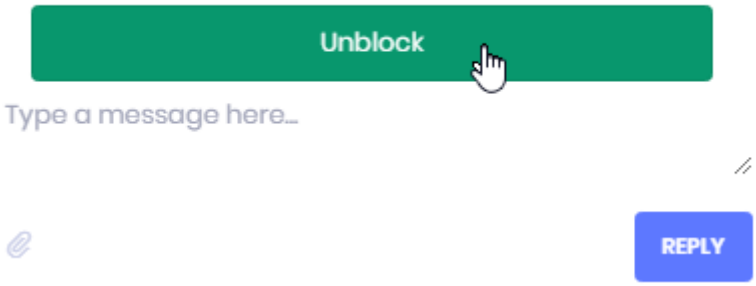
Blocking/Unblocking contacts

Clicking on the ellipsis “…” button allows to *Block* the contact :



You won’t receive any more message from this contact.

To unblock a contact, click *Unblock* :



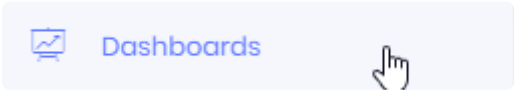
Last modified: Oct 22, 2019

3.4. Dashboards

Dashboards are the core concept of Alpana.
They are created using Alpana Designer, and are composed of Widgets that are laid out on a single scrollable page.
“Publishing a Dashboard” means pushing it to Alpana Server to make it available to other users on the platform.

This section describes publishing, viewing and managing Dashboards.

To access Dashboard management, use the *Dashboards* menu :



Last modified: Oct 22, 2019

3.4.1. Categories

Published Dashboards are organized in Categories.
Categories can contain other Categories, like a tree of folders.

This Chapter documents the management of Categories.

Creating Categories

> When you install Alpana Server or create a new Tenant, no Categories exist by default. You will need to create at least one Category for publishing.

Categories are managed like a tree of folders.
There are two types of Categories :

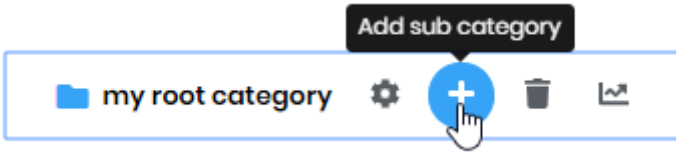
1. *root Categories* are at the root of the tree structure.
2. *sub Categories* are nested under root Categories or other sub Categories

Creating a root Category

To create a root Category, click *Add new root Category* :

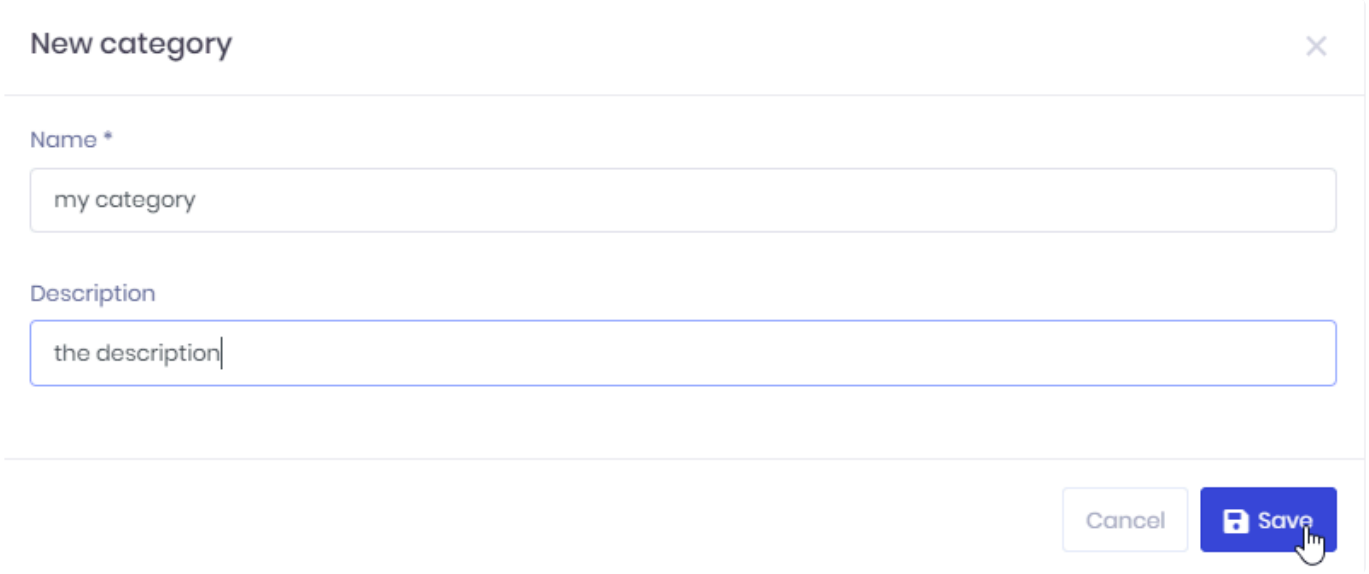
Creating a sub Category

To create a sub Category, click on the desired parent Category, then click the *Add sub Category* icon in front of it :



New Category menu

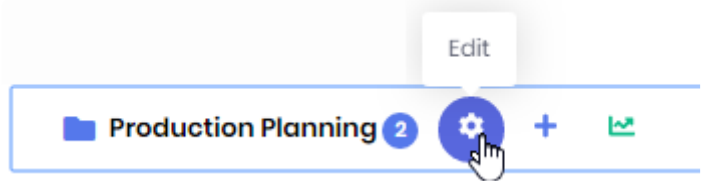
When creating a Category, fill in the desired name and description, and click *Save* :



Managing Categories

Editing a Category

To Edit a Category, click on it, then click the *Edit* icon in front of it :

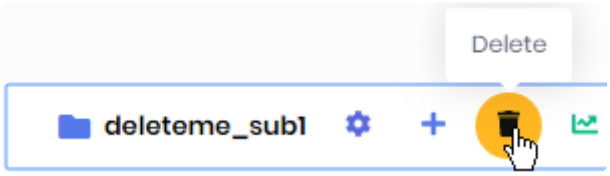


This allows you to change the Category name and description :

Deleting an empty Category

> A Category cannot be deleted while it still contains Dashboards, and while any of its Sub Category contains Dashboards.

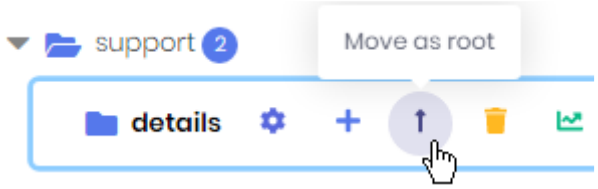
To Delete a Category, click on it, then click the *Delete* icon in front of it :



Moving a Category

To move a Category, drag it and drop it onto another Category.

To move a Sub-Category to become a Root Category, use the dedicated button option on the Category :

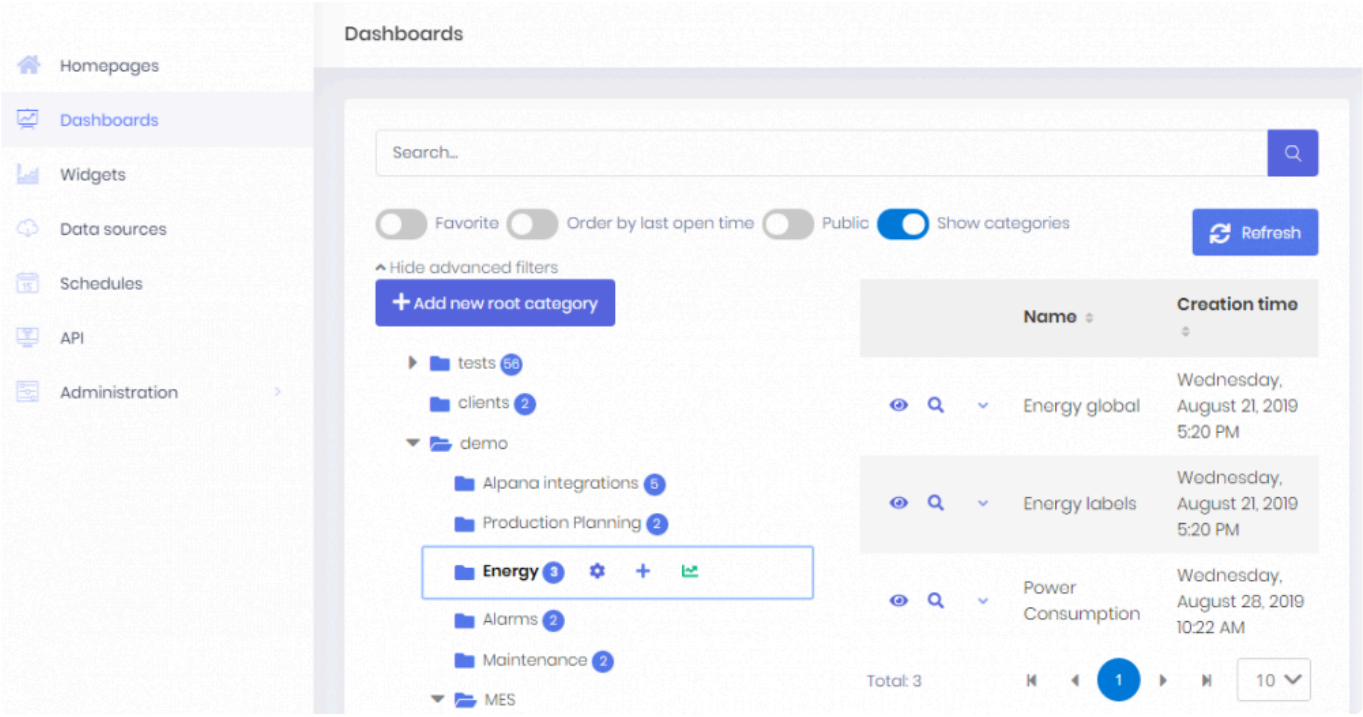


Last modified: Aug 26, 2020

3.4.2. Browsing the Dashboards

The Dashboards list is composed of :

- a Search bar at the top
- the list of Categories on the left
- the list of Dashboards from the selected Category on the right



Browsing Categories and listing Dashboards

Click on a Category to select it.
The Dashboards from that Category will be listed on the right.

The list is paginated and the page size can be configured at the bottom.

The list can be sorted by Dashboard Name or by Creation Time.

Filtering the list of Dashboards

The Search bar allows to filter Dashboards by name.
To clear the filter, just use an empty search.

You can also use options to filter only Public Dashboards, or your Favorites.

Last modified: Oct 22, 2019

3.4.3. Viewing Dashboards

Viewing Dashboards is the core functionality of Alpana.

This section describes how to view Dashboards and what features are available when viewing a Dashboard.

Last modified: Oct 22, 2019

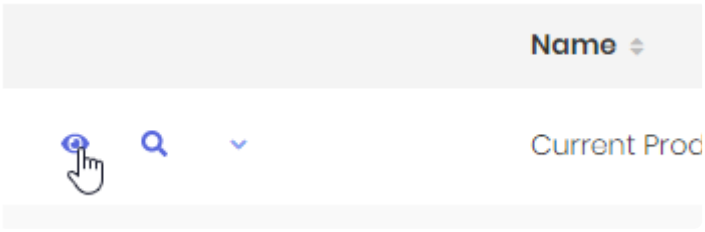
3.4.3.1. Opening Dashboards for Viewing

From the list of Dashboards, you can open any dashboard for viewing using any of the two following options :

- *Open* in a new page
- *Preview* in the current page

Opening to view in new page

Click the eye button in front of the Dashboard to open the full-page viewer in a new tab of your web browser :

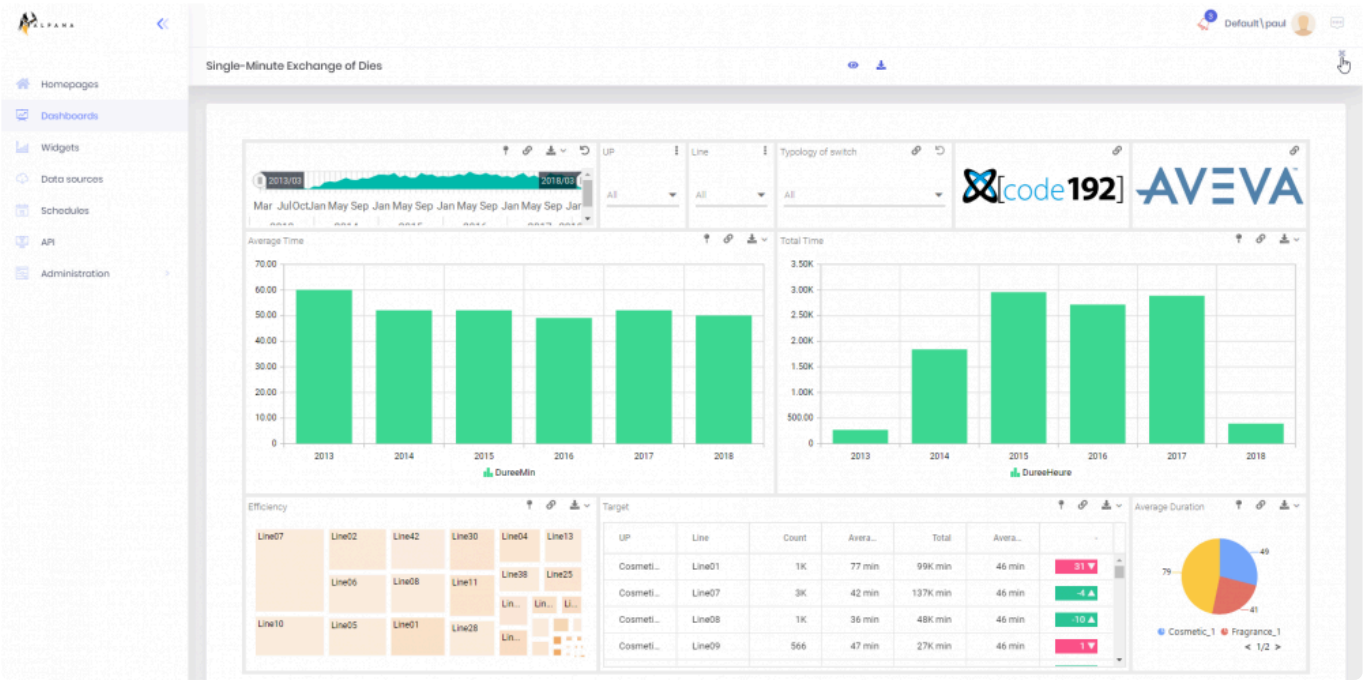


This is the preferred method, as it will include all viewing features.
This is what we will use in the next chapters.

Previewing in the current page

Click the magnifying glass button in front of the Dashboard to open the preview in the same page :

You can close the preview by clicking the cross at the top right :



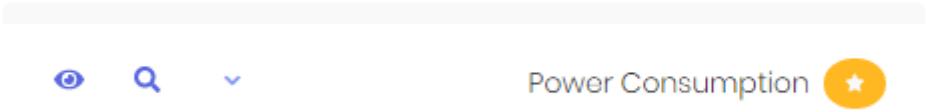
Last modified: Oct 22, 2019

3.4.3.2. Favorite Dashboards

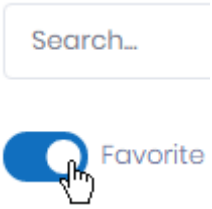
You can mark Dashboards as Favorite to better find them later.

When viewing a Dashboard, click the star next to the title :

Favorite Dashboards are marked with a yellow star :



You can filter the Dashboards list by Favorites, just check the corresponding option in the top bar :



Last modified: Oct 23, 2019

3.4.3.3. Refreshing Dashboards

When a Dashboard page is loaded, each Widget makes a query to its Data Source as described in Alpana data model

Then, each time an interaction changes the query, a new query is make for the Widgets that need it. For example : using filters or drill-down.

If you want to get new data in the Dashboard without having to interact, you can use the *Reload* button in the top bar of the Dashboard viewer :

This refreshes all widgets as if refreshing the browser page, except you keep the filter state.

If instead you want the Dashboard to refresh automatically, you have several options that developers can implement :

- Auto-refresh
- API actions

Last modified: Oct 23, 2019

3.4.3.4. Manual Exports

From the Dashboard viewer, Dashboards can be manually exported to several formats.
After completion of the export, a Notification informs you that your file is ready.
Click on the notification, and your web browser proposes a “download” or “save as” dialog.

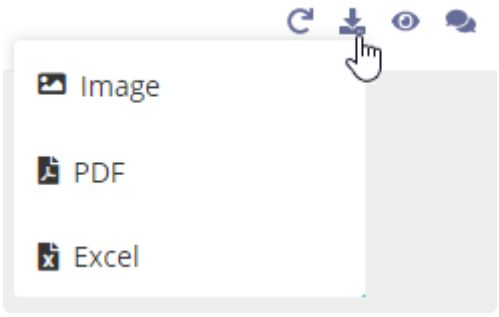
> Note : If instead you wish to schedule automated exports, please the chapter on Schedules.

Two levels of exports are possible :

- exporting the full Dashboard
- exporting a single Widget from the Dashboard

Exporting the Dashboard

To export the full Dashboard, click the *Export* button in the top bar :



Several formats are available for exporting :

Image

The Dashboard is exported as an image file.
Choose a *File name*, a *Layout Configuration*, and an *Image Type* :

Layout Configuration

This allows to select which Layout of the dashboard you wish to export.

Image Type

Supported image types are :

- JPEG (default)
- PNG

- BMP

PDF

The Dashboard is exported as a PDF file.

Choose a *File name*, and a *Layout Configuration* :

Layout Configuration

This allows to select which Layout of the dashboard you wish to export.

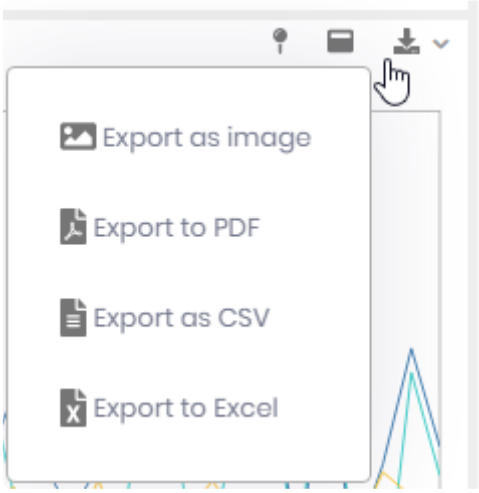
Excel

The Dashboard is exported as an Excel XLSX spreadsheet.
The data for each Widget is in a separate sheet of the same file.

Just choose a *File name*.

Exporting a Widget from the Dashboard

To export a single Widget, click the *Export* button on the corresponding Widget :



Note : this may be enabled or disabled for each Widget, see the corresponding chapter.

Several formats are available for exporting :

Image

The Widget is exported as an image file.
Choose a *File name* and an *Image Type* :

Image Type

Supported image types are :

- JPEG (default)
- PNG
- BMP

PDF

The Widget is exported as a PDF file.

Just choose a *File name*.

Excel

The Widget is exported as an Excel XLSX spreadsheet.

Just choose a *File name*.

CSV

The Widget is exported as a CSV text file with comma , separators.

Just choose a *File name*.

Last modified: Jun 17, 2020

3.4.3.5. Dashboard Views

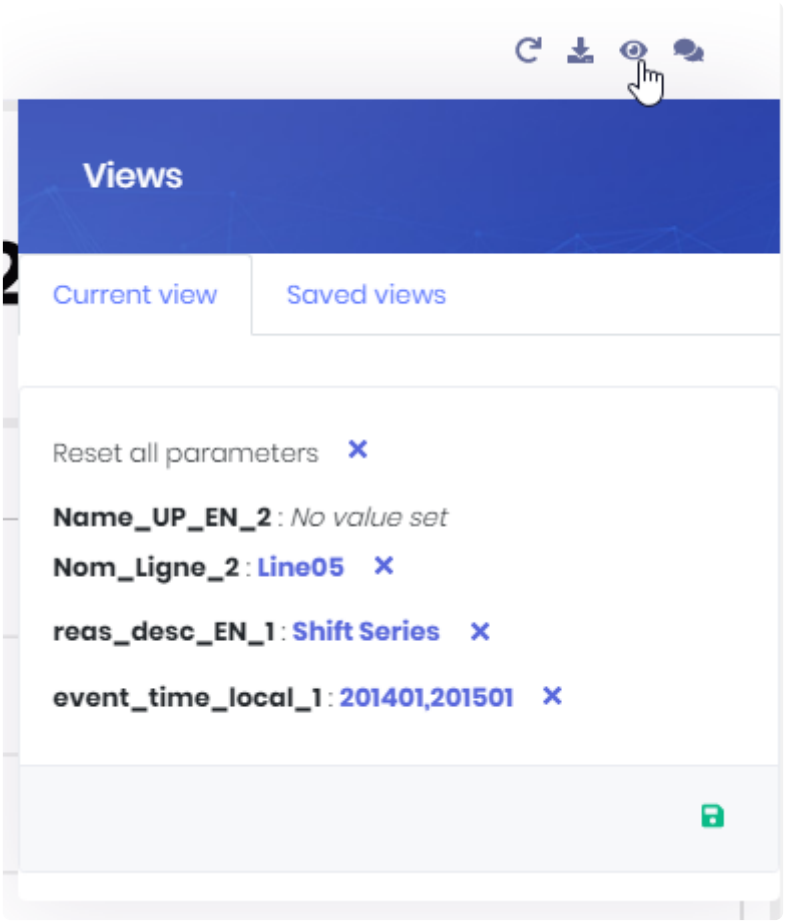
The Dashboard *Views* popup allows you to manage the current state or Parameters :

- you can view the current values of Parameters
- you can reset Parameters
- you can save sets of Parameters to reuse them later

This is very useful for example when you have many filters and want to monitor their state, or bookmark them.

Viewing Parameters

The *Views* pop-up shows the name and value of each Parameter of the Dashboard :

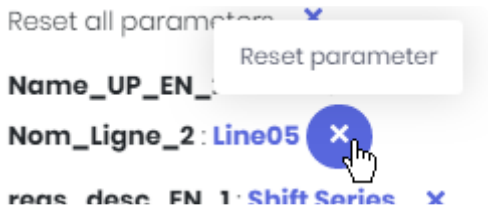


If a Parameter has no value at the time, "*No value set*" is displayed.

Resetting Parameters

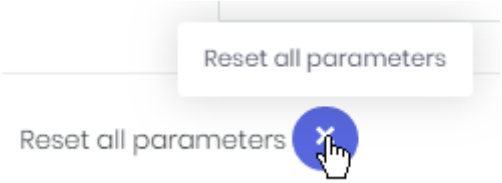
Single Parameter

To reset a Parameter to its default value, click the cross button "*Reset parameter*" in front of it :



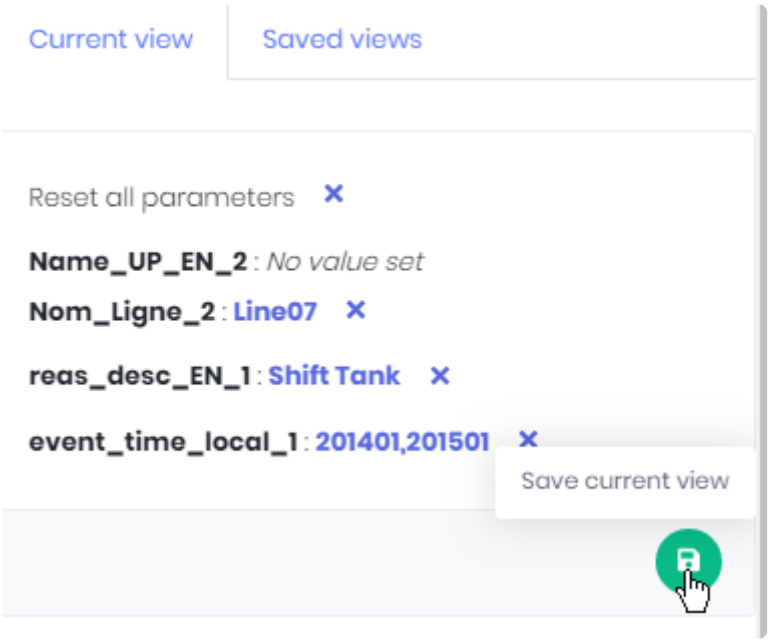
Single Parameter

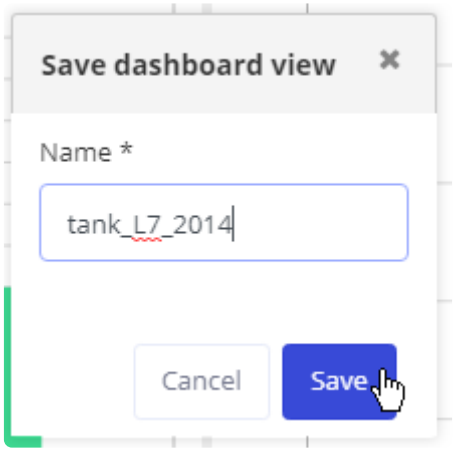
To reset all Parameters to their default value, click the cross button in front of *Reset all parameters* :



Saving Views

You can save the current state of Parameters to restore it later.
Click on the *Save current view* button :



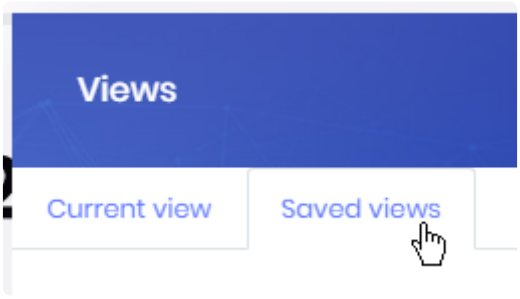


Re-opening saved Views

When a Dashboard contains previously saved Views, you can restore them.
This return Parameters to their previous state, like a bookmark.

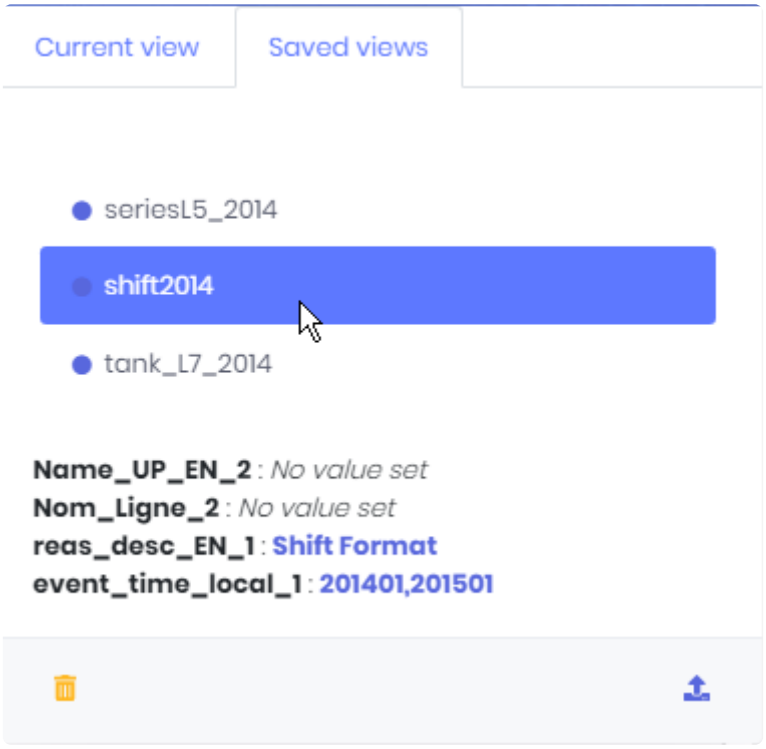
From the Views pop-up

In the Views dialog, switch to the *Saved views* tab :

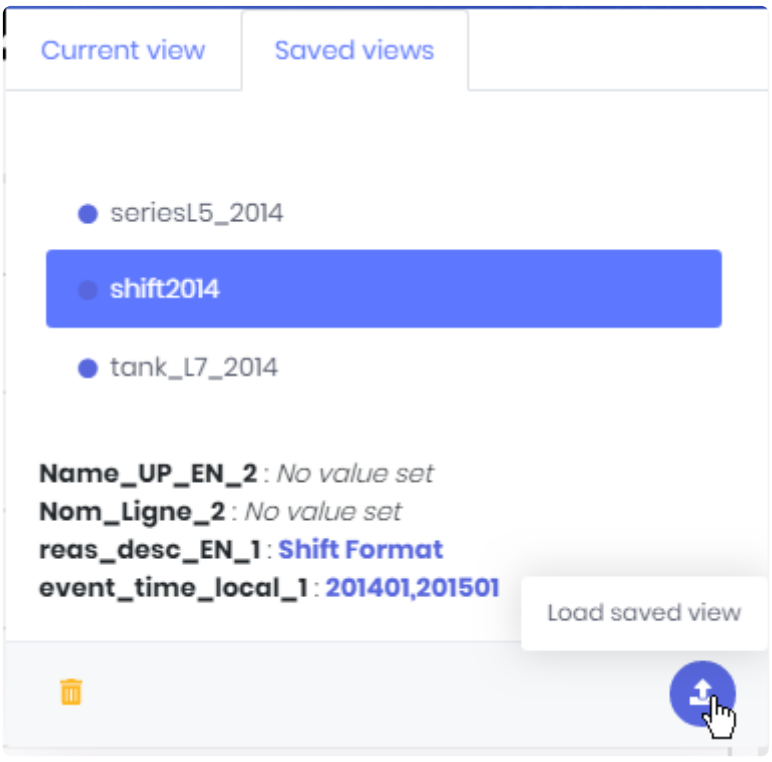


This tab lists all saved Views.

Click a View name to see the corresponding Parameters values at the bottom of the dialog :

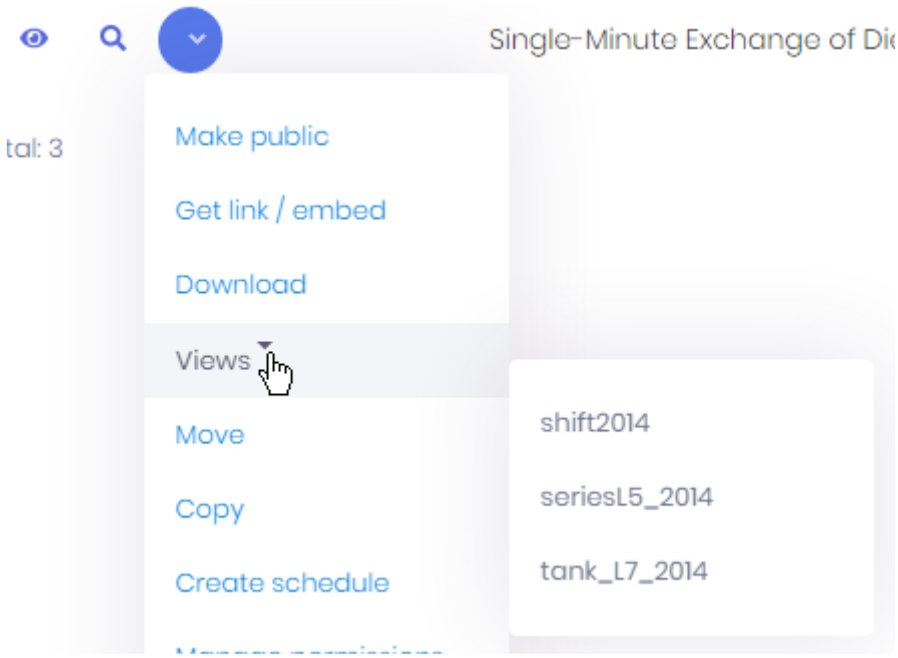


Click the *Load saved view* button to restore the Parameters state in the Dashboard :



From the Dashboards list

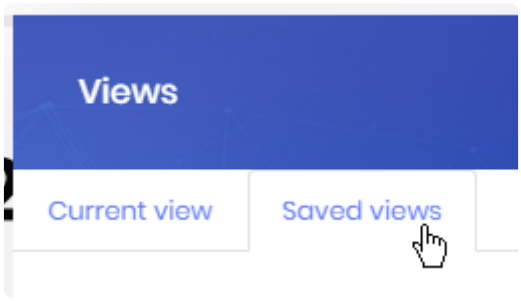
In the Dashboards list, in the Dashboard's actions menu, hover the *Views* item.



This unfolds the list of Views of this Dashboard.
Click an item in the list to open the Dashboard and restore the corresponding View.

Deleting Views

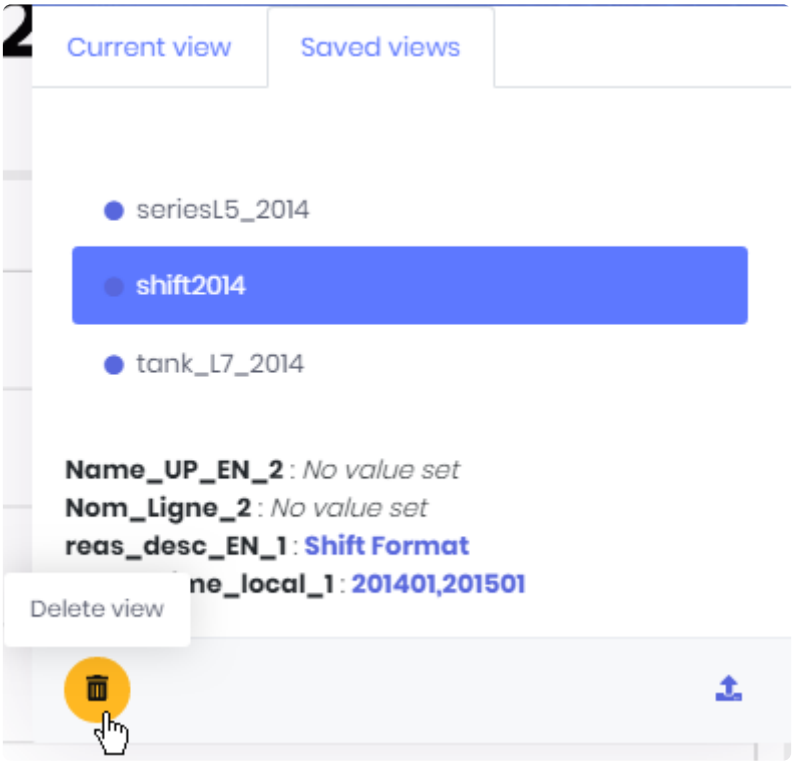
In the Views dialog, switch to the *Saved views* tab :



This tab lists all saved Views.

Click a View name to see the corresponding Parameters values at the bottom of the dialog.

Click the *Delete view* button to delete that View :



Last modified: Oct 24, 2019

3.4.3.6. Comments

When viewing Dashboards, you can leave comments and have discussions with other users.
You can create comment threads and choose to be notified.
This is a good way to have your analysis and discussion in the same place.

Opening the Comments

To open the Comments panel and start commenting or reading comments, click the *Comments* button in the top bar or in the menu of widgets :

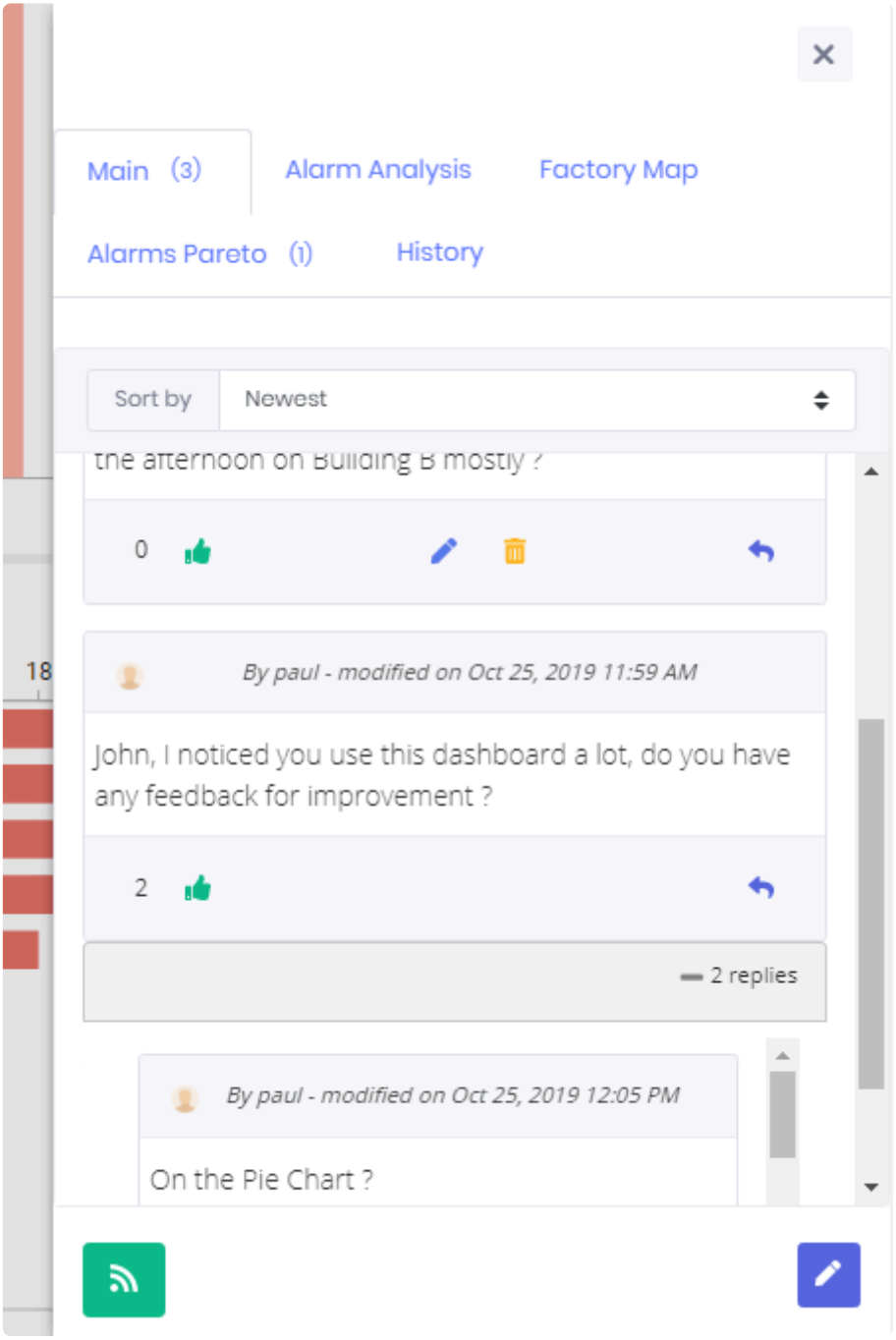
To close the panel, click the cross at the top right.

Viewing Comments

The Comment panel is divided in Threads, with :

- one Thread for each Widget in the Dashboard that has Comments enabled.
- one Thread called *Main* for the Dashboard itself.

Each Thread is inside a tab.



Each Comment shows :

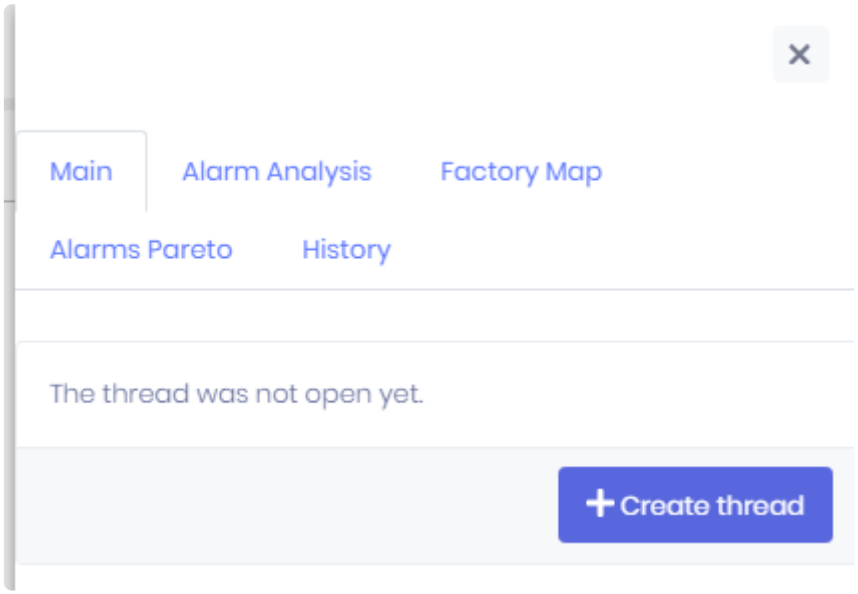
- the user's name and profile image
- the last modified date
- the number of Likes
- the Comment-level controls (see below for a description)

Comments can be sorted by date or by top Likes.

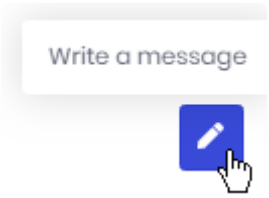
Commenting

New Comment

To open a Thread, click the *Create Thread* button.

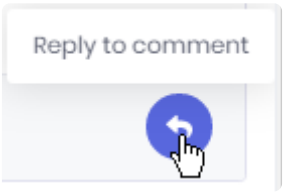


To write a new Message, click *Write a message* at the bottom of the Comments panel :



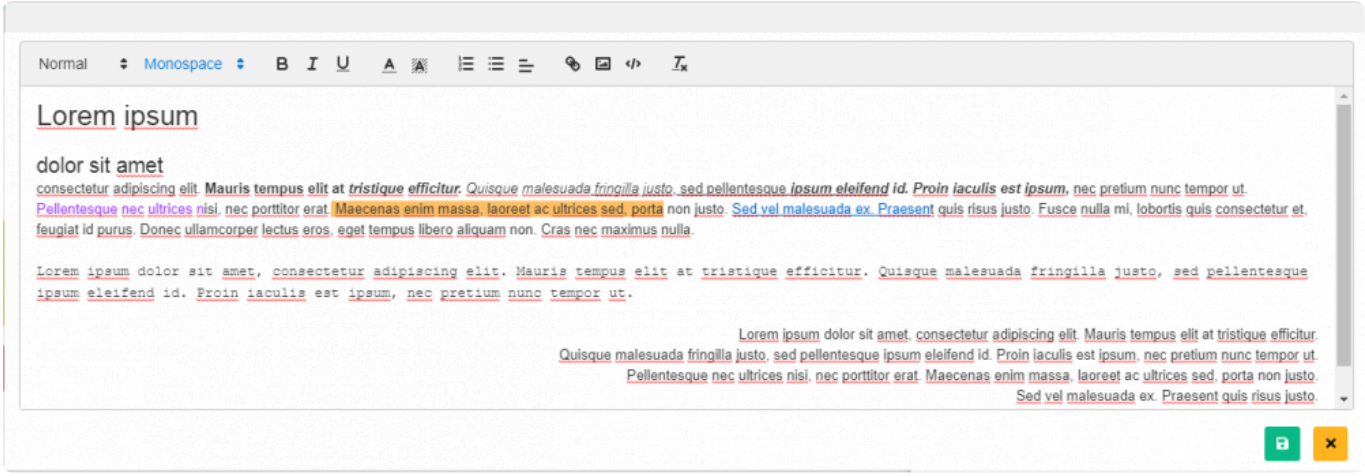
Replies

To attach your Comment to an existing comment, you can Reply to it.
Locate the desired Comment and click *Reply to comment* :



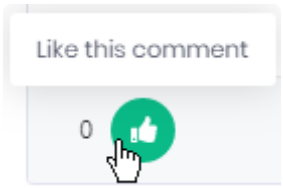
Comment editor

When creating or editing a Comment, the text editor allows you to write rich text spanning multiple lines (style, links, alignment, etc) :

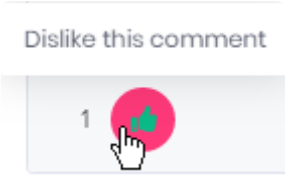


Liking Comments

You can Like Comments by clicking the *Like* button :



You can remove your Like by clicking again the same button :

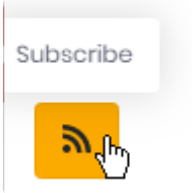


Getting notified

For each Comment thread, you can choose to be notified of new Comments.
This will send a new Notification to you when a new Comment is added to the Thread.
A link in the Notification will take you to the Dashboard and Comment itself.

Subscribing

Click the *Subscribe* button to be notified :



Unsubscribing

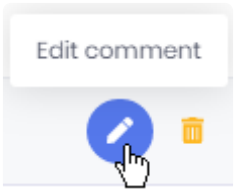
Click the same button to *Unsubscribe* :

Modifying Comments

You can modify your own Comments :

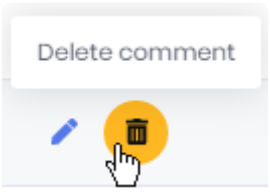
Editing

Click *Edit comment* to edit your Comment :



Deleting

Click *Delete comment* to delete your Comment :



Last modified: Oct 25, 2019

3.4.3.7. Pinning Widgets

Widgets can be configured to show a “*Pin*” control.
This control allows to Pin the Widget to a Homepage.

To learn how to use Homepages and Widget Pinning, please see the dedicated chapter.

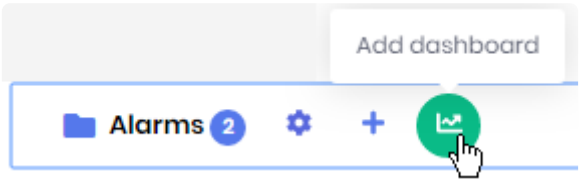
Last modified: Oct 28, 2019

3.4.4. Managing / Publishing Dashboards

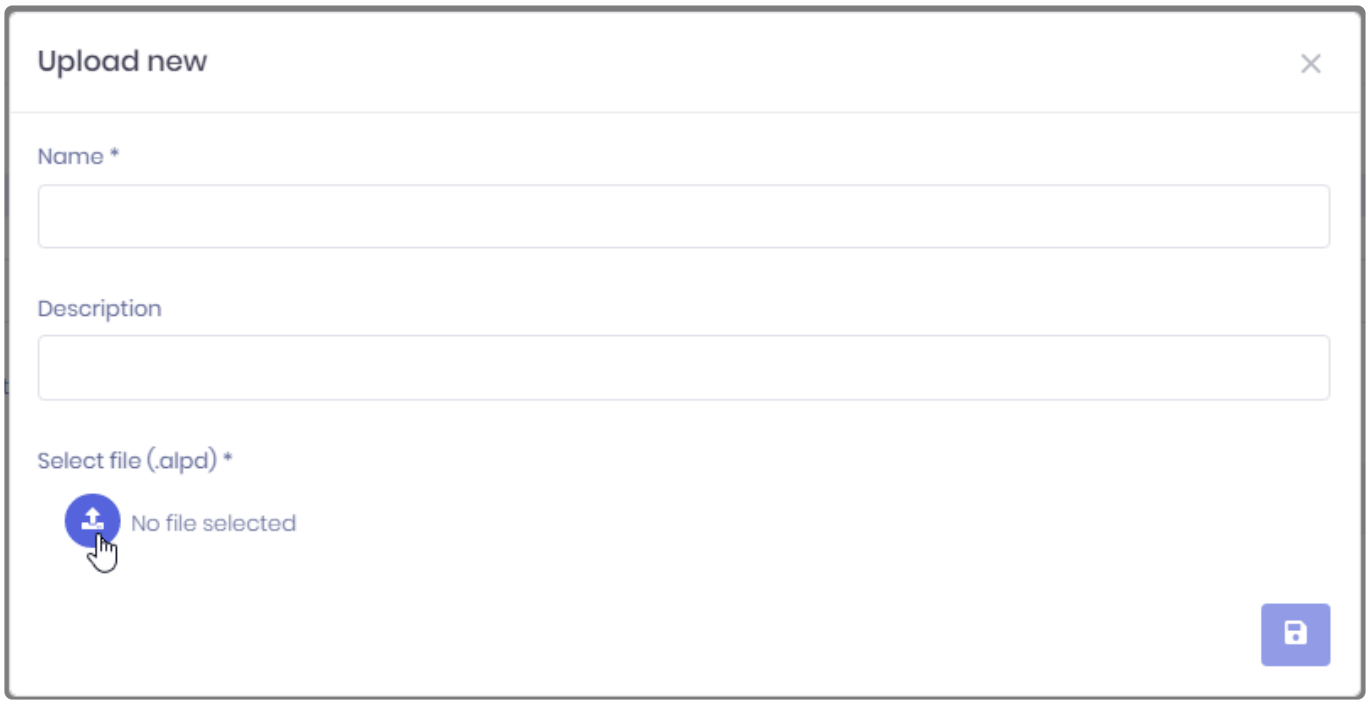
Publishing from Alpana Server

Dashboards can be published from Alpana Designer.
Alternately, you can publish a dashboard `.alpd` file directly to the Server.

Click on the destination Category and click the *Add Dashboard* button :



This opens the *Upload New* dialog :

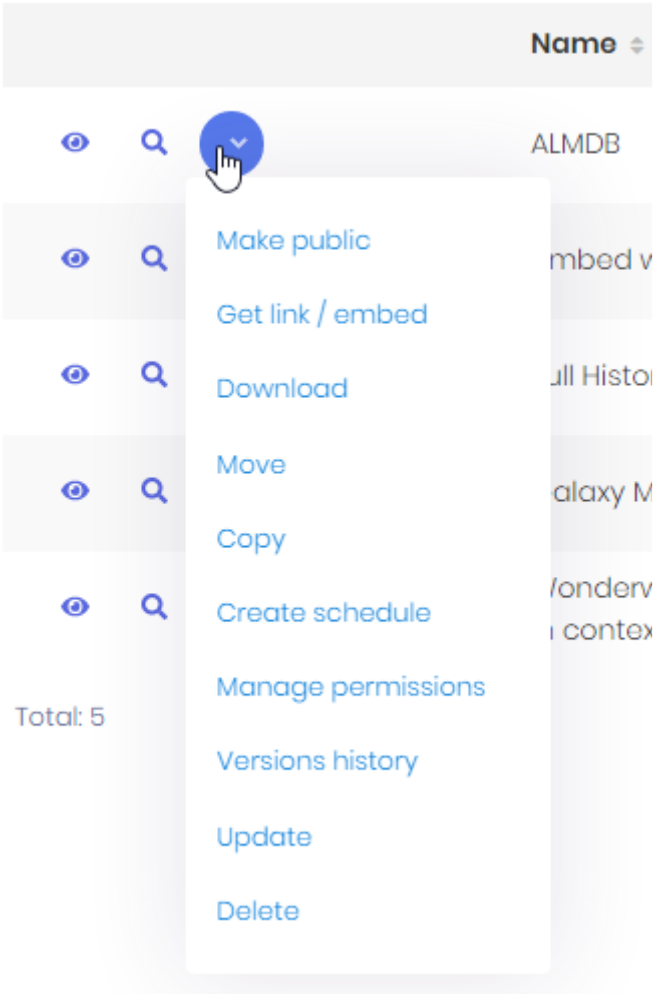


Click the *Select File* button to browse your filesystem for a dashboard `.alpd` file.
Fill-in a *Name* and optionally a *Description*.

When you click the *Save* button, the Dashboard is published to the Category.

Actions menu

Each published Dashboard has a set of actions available.
To list those actions, open the dashboard menu next to the Dashboard's name :



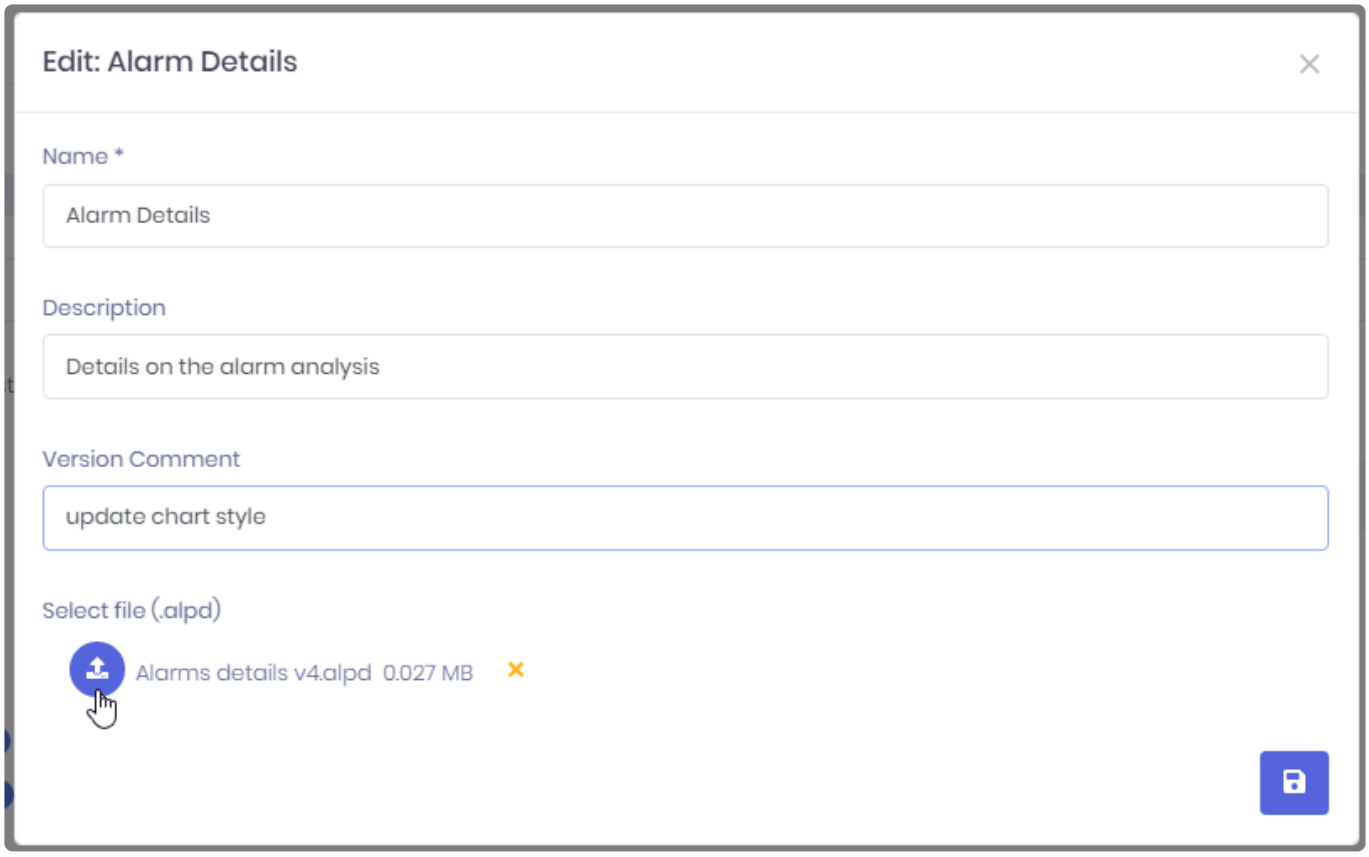
Some of those actions are detailed in the corresponding chapter :

- *Make public* / *Make private*
- *Get link / embed*
- *Download*
- *Move*
- *Copy*
- *Create schedule*
- *Manage permissions*
- *Versions history*
- *Update*
- *Delete*

Update

This opens the *Edit* dialog to update the Dashboard's name and description.

You can also upload a new dashboard `.alpd` file to replace the current one.
If you do so, this will create a new version, so you can input a *Version Comment*.

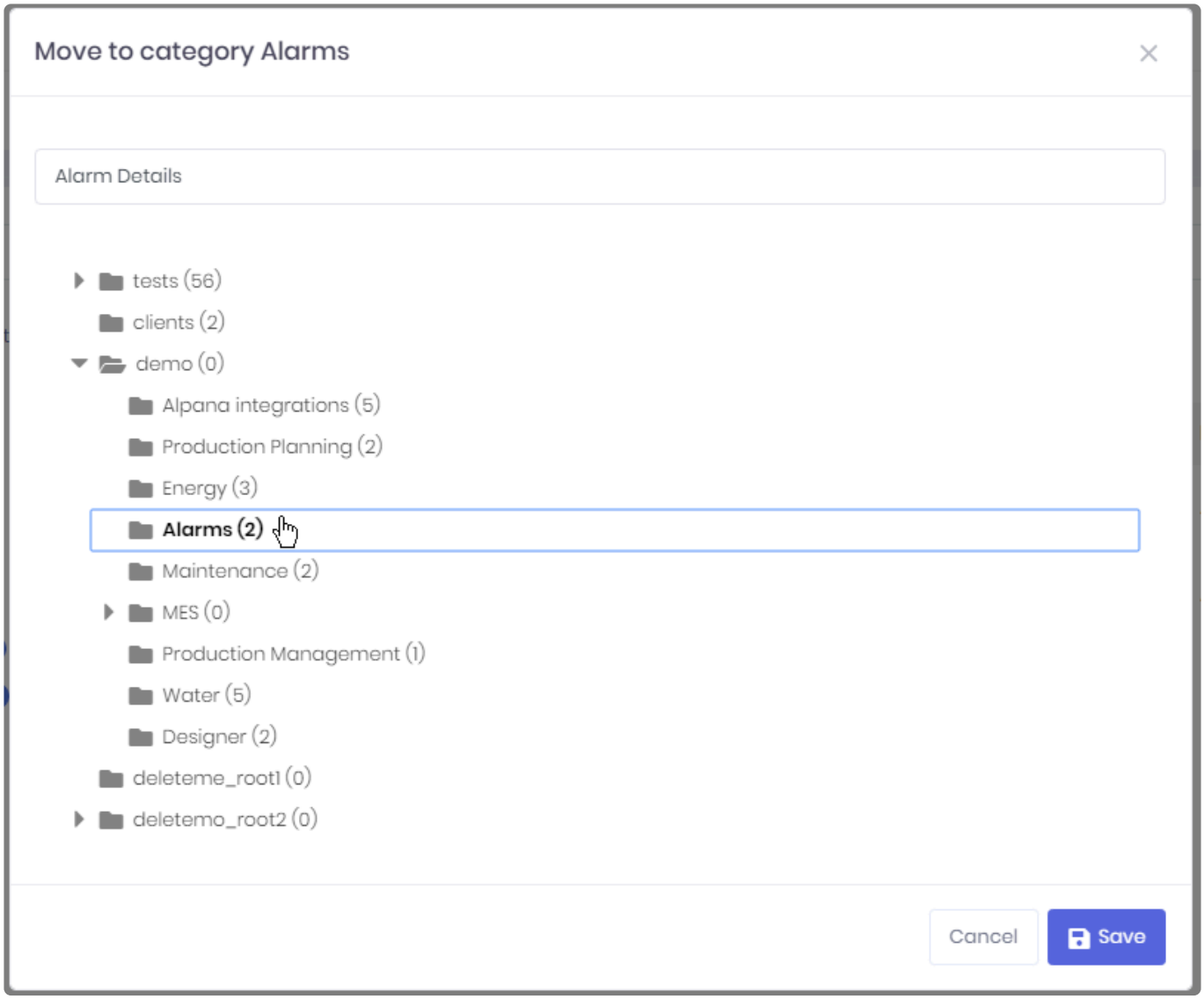


Download

This downloads the dashboard `.alpd` file.
This allows for example to open it in Alpana Designer.

Move

This opens the *Move* dialog which allows to select a destination Category for moving this Dashboard to :



Copy

This opens the *Copy* dialog which allows to select a destination Category for copying this Dashboard to.

Delete

This deletes the selected Dashboard.

Make public / Make private

(see corresponding chapter)

Manage permissions

(see corresponding chapter)

Get link / embed

(see corresponding chapter)

Create schedule

(see corresponding chapter)

Versions history

(see corresponding chapter)

Last modified: Oct 28, 2019

3.4.5. Dashboard Version History

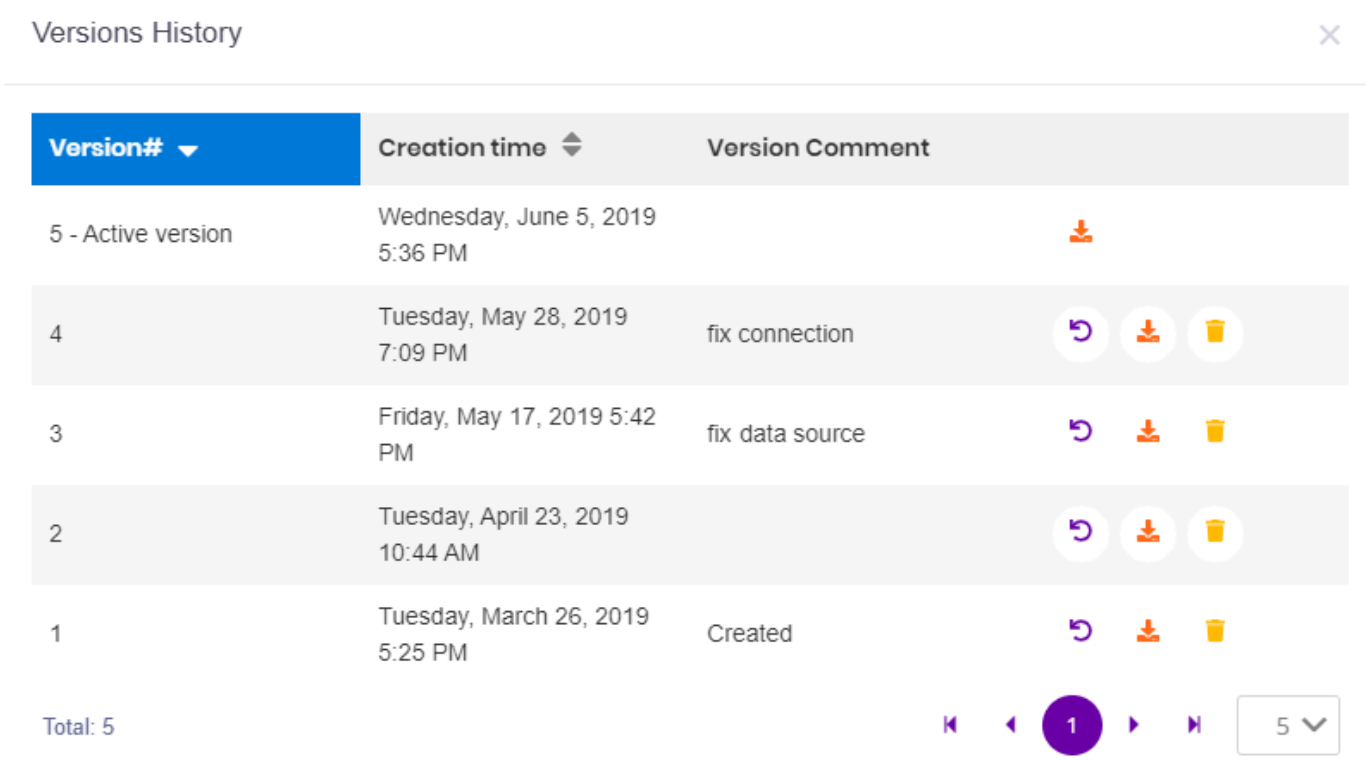
Every time you over-write an existing Dashboard, the previous version in saved also.
The Version History allows you to see the changes applied to a Dashboard, and restore a previous state.

Opening the Versions History

From each Dashboard action menu, select *Versions history* to open the dialog.

Viewing Version History

The Versions History dialog shows a paginated list of Versions :



Each Version displays :

- a Version Number
- its Creation Time
- an optional Version Comment

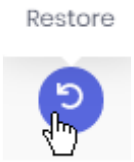
You can sort the list by version number and creation time bu clicking on the corresponding column headers.

You can browse versions pages and choose a page size at the bottom of the dialog.

Restoring an older Version

By default, the Active Version of a Dashboard is the latest published.
However, you can manually restore any previous Version.

Click the _ Restore_ button in front of the selected Version :

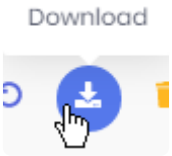


The *Active version* indicator will move back to that older version.

So if you want to restore the latest Version again (or any other Version), just do the same process again.

Downloading an older Version

Click the *Download* button to download the corresponding Version :



This will download the corresponding dashboard `.alpd` file.

This is useful for example for studying a previous version, and maybe turning it into a separate Dashboard.

Deleting an older version

You can delete any previous version.
Click the *Delete* button in front of the corresponding Version :

Last modified: Oct 25, 2019

3.4.6. Public / Private Dashboards and Dashboard-level Permissions

Dashboard access can be managed through the Permissions system, or by making it public.
Both items are listed in the actions menu of the Dashboard.

Make public / Make private

What is a Public dashboard

By default, content on Alpana Server is secured behind the authentication an permissions system. However, in some cases you may wish to share content to users without requiring a login.

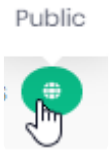
When you turn a Dashboard *Public*, it becomes accessible for viewing to everyone who can access the URL.

> ! This means that Alpana can no longer protect unwanted access to this Dashboard.
> If the Dashboard contains sensitive data, you will have to find other means to prevent unwanted access.

This is useful for example if you want to share this dashboard to many users on a public website.

Making it public/private

From the Dashboard's action menu, select *Make public* to make the Dashboard Public.
The Dashboard becomes marked with the *Public* icon :



To make the Dashboard private again, select *Make private* from the Dashboard's action menu.

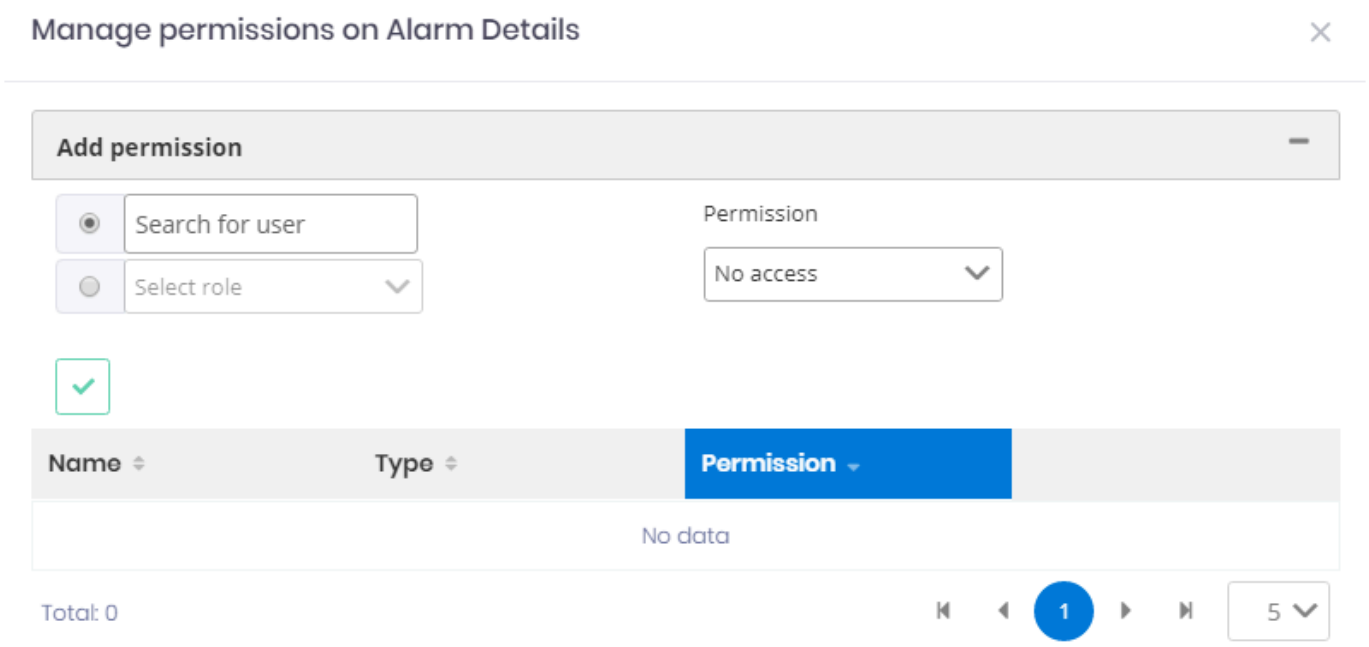
Managing Public Dashboards

You can filter the Dashboards list to see only Public Dashboards, just check the corresponding option in the top bar :

Manage permissions

> ! When managing permissions, be careful not to remove access to your own user.
> If this happens, you can work around it by : creating a new Role with all permissions, assigning this as the only Role to a User, and letting that User remove the Permissions restriction.

From the Dashboard's action menu, you can manage its Permissions.
Select *Manage permissions* to open the corresponding dialog :

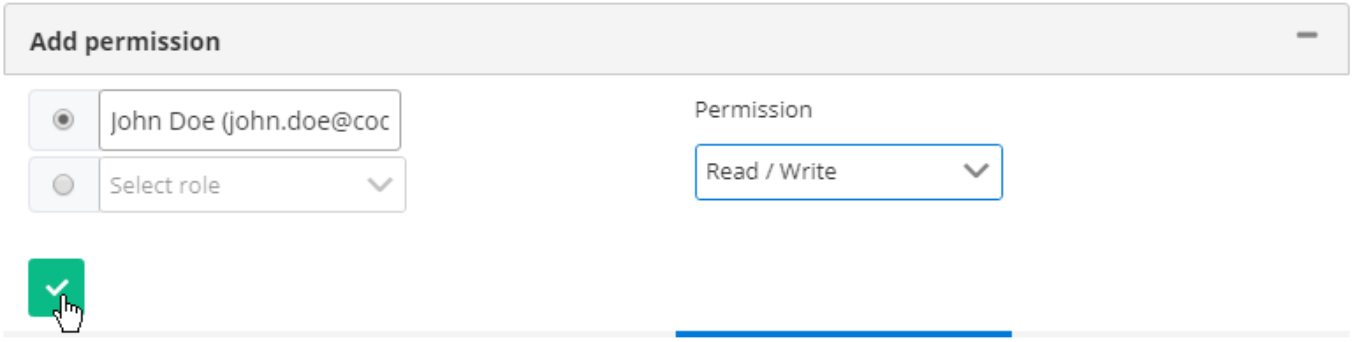


The existing Permissions are listed at the bottom of the dialog.

Adding a permission

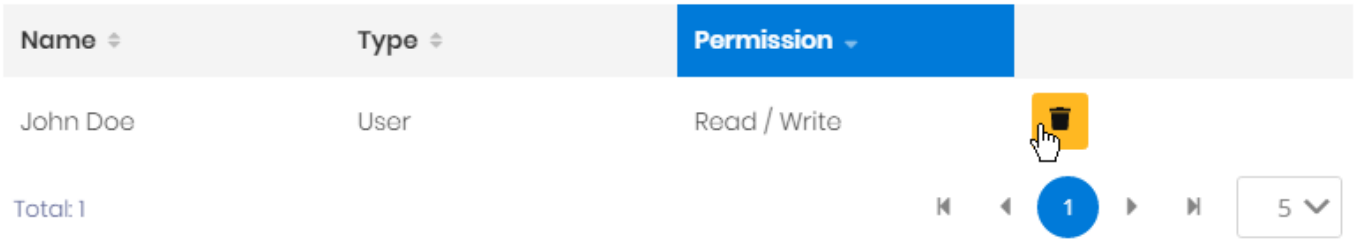
If you want to override the default permissions, fill-in the *Add permission* form :

1. select a User or Role
2. select the desired Permission
3. click the validation button :



Removing Permissions

If you want to remove a Permission, click the *Remove* (trash) icon in front of it in the list :



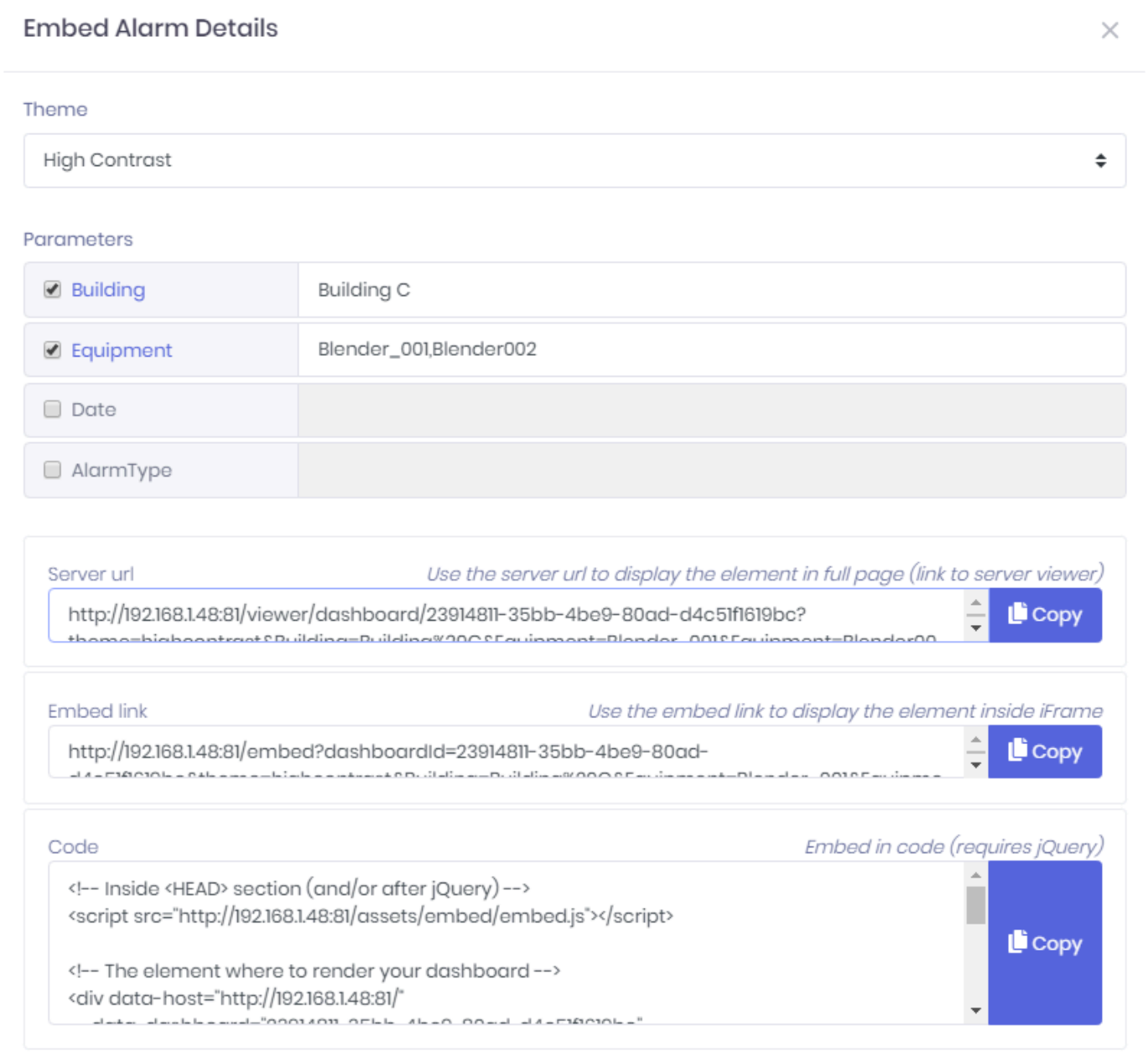
Last modified: Oct 25, 2019

3.4.7. Linking / Embedding Dashboards

Get link / embed

When opening a Dashboard from its URL, or when embedding it into another tool, you can pass options to control how the Dashboard is displayed.

From the Dashboard's action menu, select *Get link / embed* to open the corresponding dialog :



This dialog generates URL's and code to link or embed the Dashboard.

Options

Select the below options to update the URL and code :

Theme

You may choose to override the Dashboard's Theme.
Select the desired Theme from the list.

Parameters

You may choose to override the Dashboard's Parameter's default values.
Check the Parameters you wish to override, and fill-inf the desired values.

Results

After configuring the desired options, you can *Copy* the results from the text boxes at the bottom of the dialog :

Server URL

This will display the Dashboard with the same visuals as when normally viewing with Alpana Server's dashboard viewer.

Embed link

This is a clutter-free version of the dashboard viewer : there is no top bar, no controls, no logo.
This is useful for embedding because the Dashboard will fit better in smaller spaces, and because the controls (like Pin Widget) are usually not meaningful in this context.

Code

This is for embedding inside another web page that uses jQuery.

> ✻ Note : the containing element needs to specify a size, otherwise the Alpana div will have a 0px size.

> ✻ Important : when embedding, you will need to let Alpana whitelist your site for CORS requests.
> Otherwise, Alpana content will not appear.

Last modified: Oct 25, 2019

3.5. Widgets

Widgets are basic bricks of construction for Dashboards..
They are created using Alpana Designer.
“Publishing a Widget” means pushing it to Alpana Server to make it available to other users on the platform.

Publishing Widgets can be useful for several reasons :

- Re-use : Published Widgets are a library of visualizations that can be downloaded to compose new Dashboards.
- Embed : When embedding inside another application, you may need a single Widget instead of a full Dashboard.

This section describes publishing, viewing and managing Widgets.

To access Widget management, use the *Widgets* menu :

Last modified: Oct 28, 2019

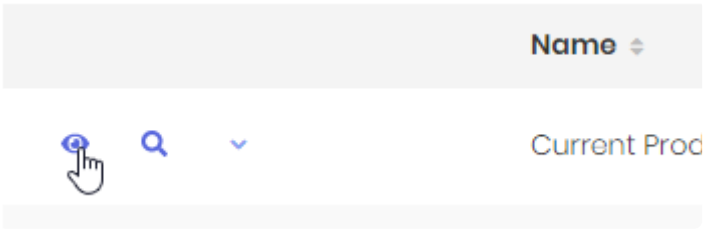
3.5.1. Viewing Widgets

From the list of Widgets, you can open any widget for viewing using any of the two following options :

- *Open* in a new page
- *Preview* in the current page

Opening to view in new page

Click the eye button in front of the Dashboard to open the full-page viewer in a new tab of your web browser :

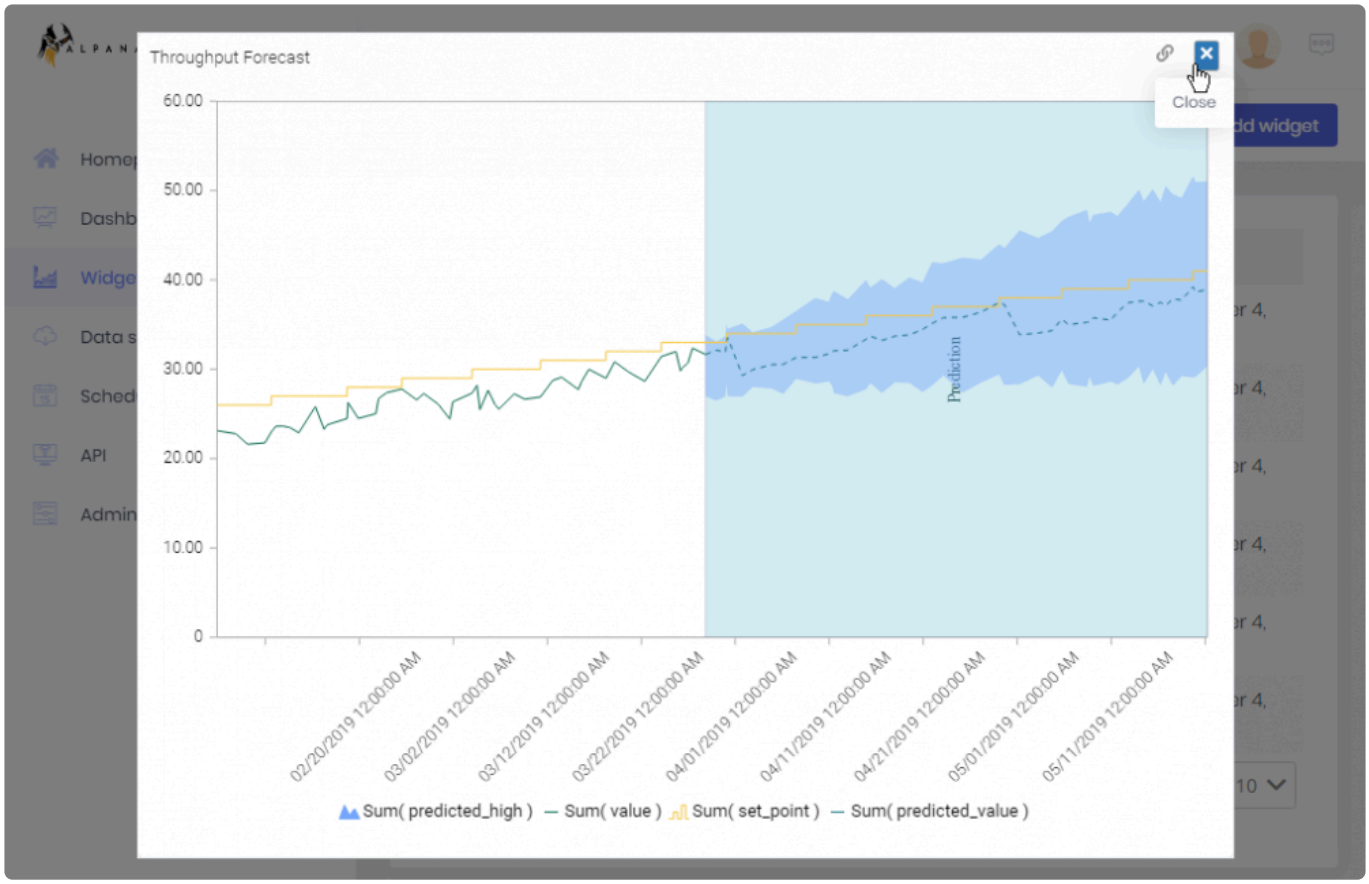


This is the preferred method, as it will include all viewing features.
This is what we will use in the next chapters.

Previewing in the current page

Click the magnifying glass button in front of the Widget to open the preview in the same page :

You can close the preview by clicking the cross at the top right :



Pinning Widgets

Widgets can be configured to show a "*Pin*" control.
This control allows to Pin the Widget to a Homepage.

To learn how to use Homepages and Widget Pinning, please see the dedicated chapter.

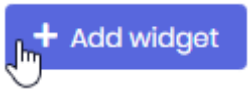
Last modified: Oct 28, 2019

3.5.2. Managing / Publishing Widgets

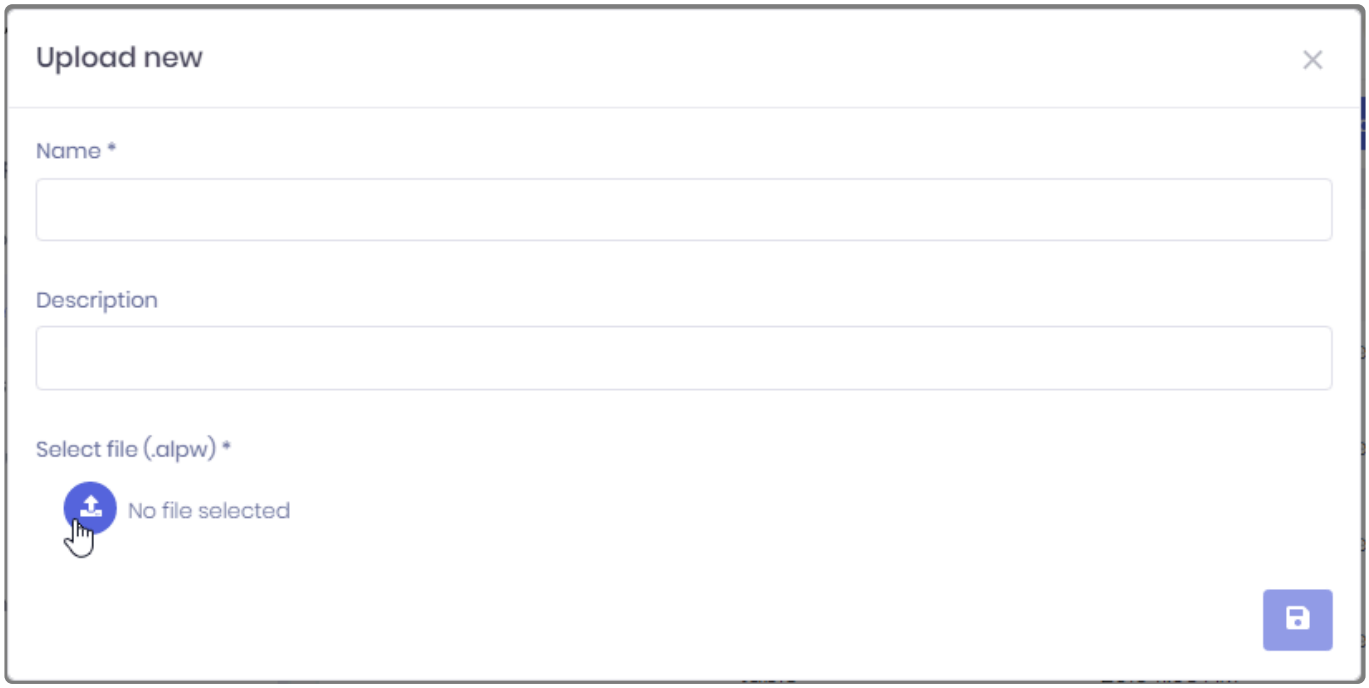
Publishing from Alpana Server

Widgets can be published from Alpana Designer.
Alternately, you can publish a widget@.alpw@ file directly to the Server.

Click the *Add widget* button :



This opens the *Upload New* dialog :

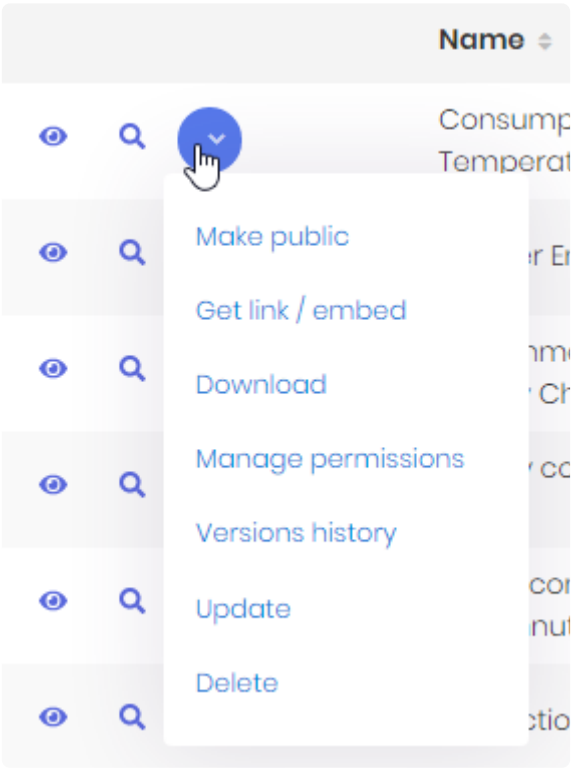


Click the *Select File* button to browse your filesystem for a widget `.alpw` file.
Fill-in a *Name* and optionally a *Description*.

When you click the *Save* button, the Widget is published.

Actions menu

Each published Widget has a set of actions available.
To list those actions, open the widget menu next to the Widget's name :



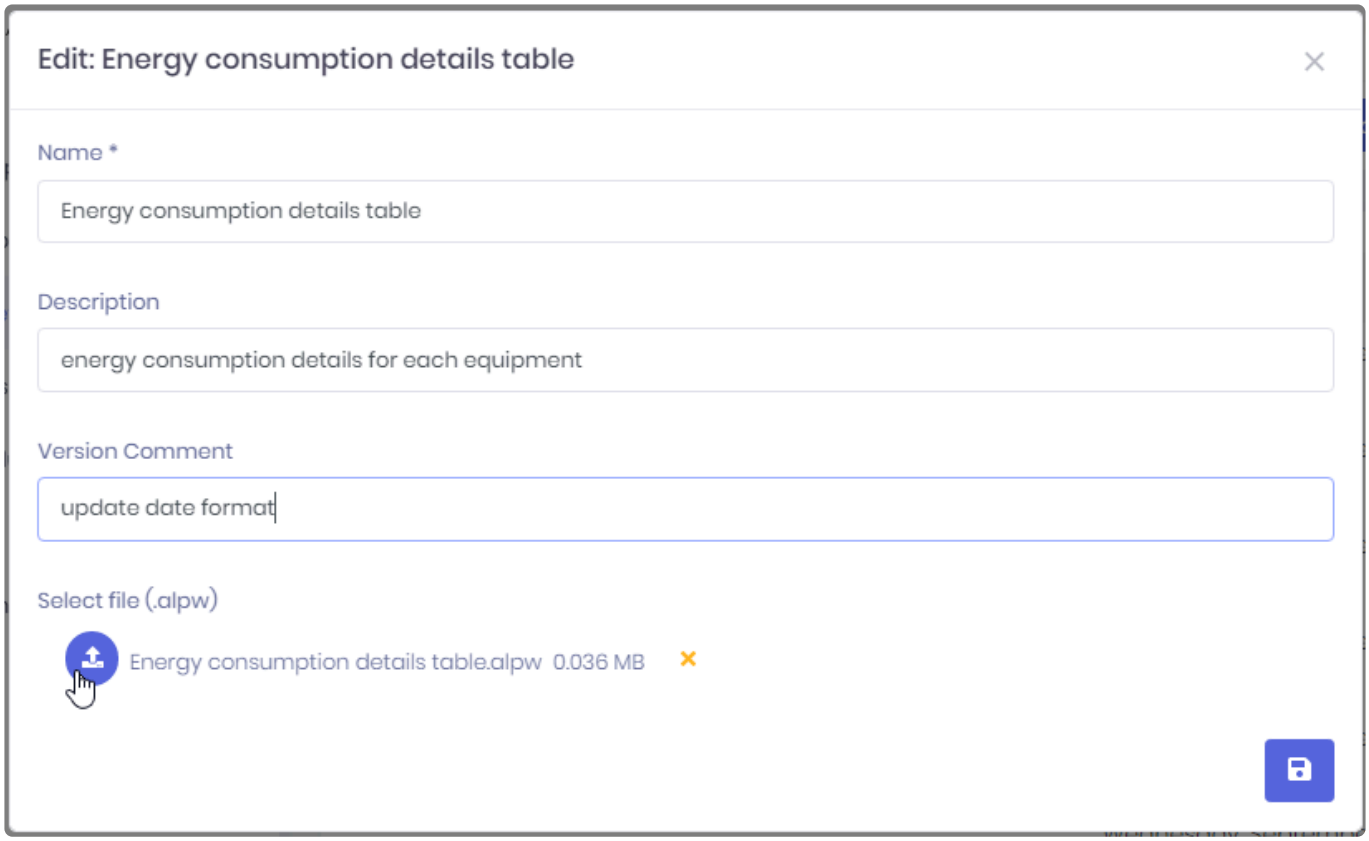
Some of those actions are detailed in the corresponding chapter :

- *Make public* / *Make private*
- *Get link / embed*
- *Download*
- *Manage permissions*
- *Versions history*
- *Update*
- *Delete*

Update

This opens the *Edit* dialog to update the Widget's name and description.

You can also upload a new widget `.alpw` file to replace the current one.
If you do so, this will create a new version, so you can input a *Version Comment*.



Download

This downloads the widget `.alpw` file.
This allows for example to import it in Alpana Designer.

Delete

This deletes the selected Widget.

Make public / Make private

(see corresponding chapter for Dashboards)

Manage permissions

(see corresponding chapter for Dashboards)

Get link / embed

(see corresponding chapter)

Versions history

(see corresponding chapter)

Last modified: Oct 28, 2019

3.5.3. Widget Version History

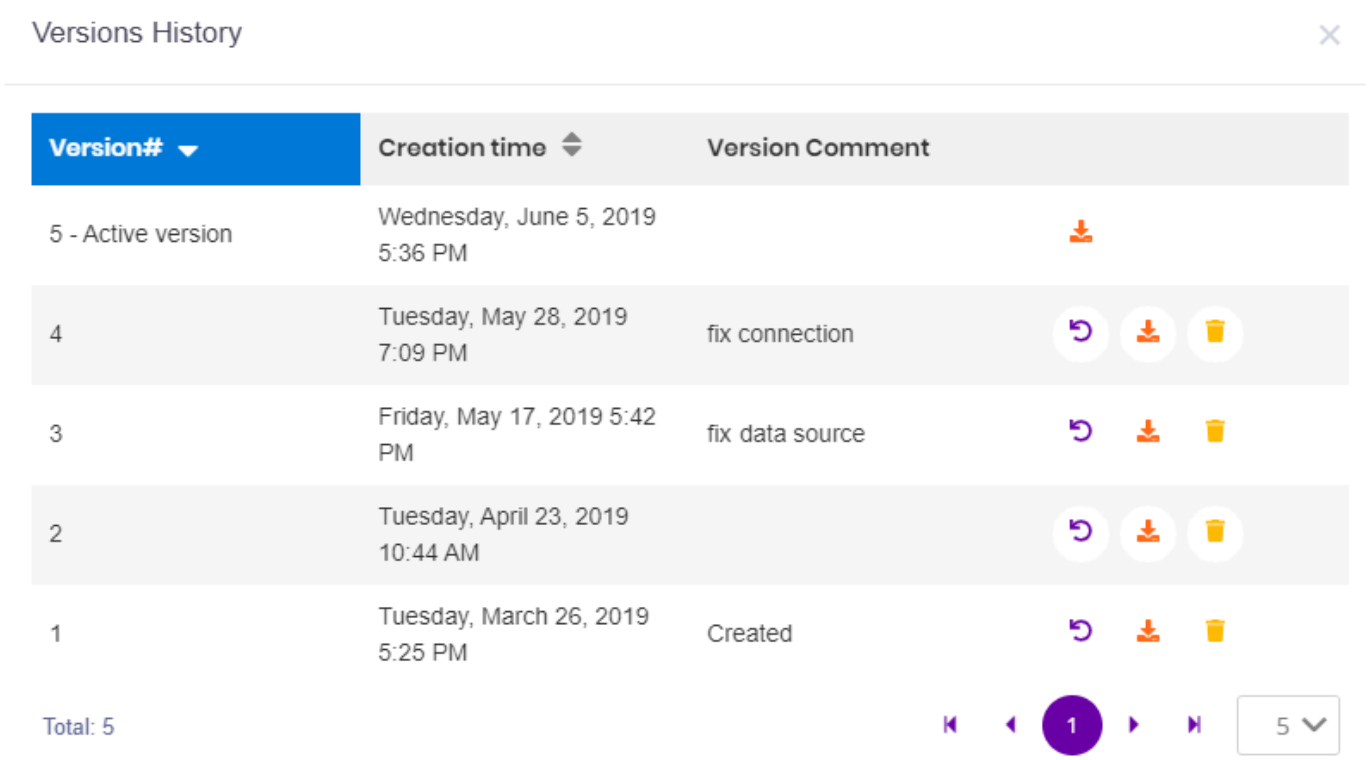
Every time you over-write an existing Widget, the previous version in saved also.
The Version History allows you to see the changes applied to a Widget, and restore a previous state.

Opening the Versions History

From each Widget action menu, select *Versions history* to open the dialog.

Viewing Version History

The Versions History dialog shows a paginated list of Versions :



Each Version displays :

- a Version Number
- its Creation Time
- an optional Version Comment

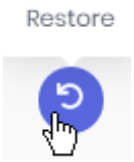
You can sort the list by version number and creation time bu clicking on the corresponding column headers.

You can browse versions pages and choose a page size at the bottom of the dialog.

Restoring an older Version

By default, the Active Version of a Widget is the latest published.
However, you can manually restore any previous Version.

Click the _ Restore_ button in front of the selected Version :

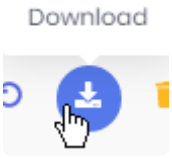


The *Active version* indicator will move back to that older version.

So if you want to restore the latest Version again (or any other Version), just do the same process again.

Downloading an older Version

Click the *Download* button to download the corresponding Version :



This will download the corresponding widget `.alpw` file.

This is useful for example for studying a previous version, and maybe turning it into a separate Widget.

Deleting an older version

You can delete any previous version.
Click the *Delete* button in front of the corresponding Version :

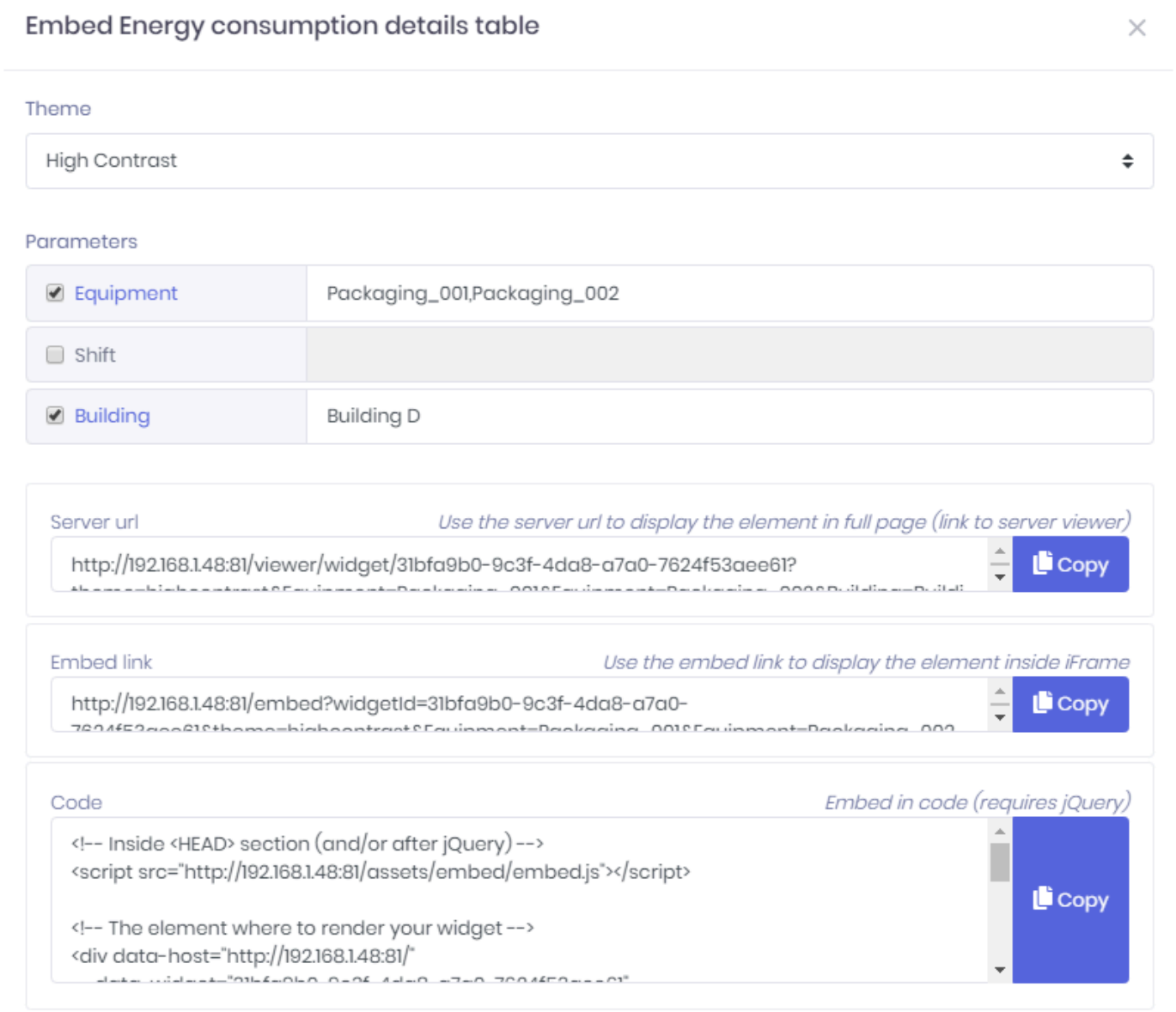
Last modified: Oct 28, 2019

3.5.4. Linking / Embedding Widgets

Get link / embed

When opening a Widget from its URL, or when embedding it into another tool, you can pass options to control how the Widget is displayed.

From the Widget's action menu, select *Get link / embed* to open the corresponding dialog :



This dialog generates URL's and code to link or embed the Widget.

Options

Select the below options to update the URL and code :

Theme

You may choose to override the Widget's Theme.
Select the desired Theme from the list.

Parameters

You may choose to override the Widget's Parameter's default values.
Check the Parameters you wish to override, and fill-inf the desired values.

Results

After configuring the desired options, you can *Copy* the results from the text boxes at the bottom of the dialog :

Server URL

This will display the Widget with the same visuals as when normally viewing with Alpana Server's dashboard viewer.

Embed link

This is a clutter-free version of the dashboard viewer : there is no top bar, no controls, no logo.
This is useful for embedding because the Widget will fit better in smaller spaces, and because the controls (like Pin Widget) are usually not meaningful in this context.

Code

This is for embedding inside another web page that uses jQuery.

> ✻ Note : the containing element needs to specify a size, otherwise the Alpana div will have a 0px size.

> ✻ Important : when embedding, you will need to let Alpana whitelist your site for CORS requests.
> Otherwise, Alpana content will not appear.

Last modified: Oct 28, 2019

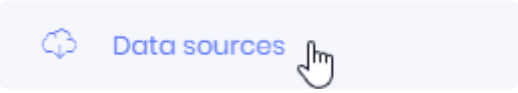
3.6. Data Sources

Data Sources are basic bricks of construction for Dashboards..
They are created using Alpana Designer.
“Publishing a Data Source” means pushing it to Alpana Server to make it available to other users on the platform.

Publishing Data Sources can be useful for re-use : published Data Sources are a library of prepared queries that can be downloaded to compose new Dashboards.

This section describes publishing, viewing and managing Data Sources.

To access Data Source management, use the *Data Sources* menu :



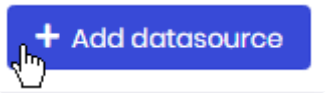
Last modified: Oct 28, 2019

3.6.1. Managing / Publishing Data Sources

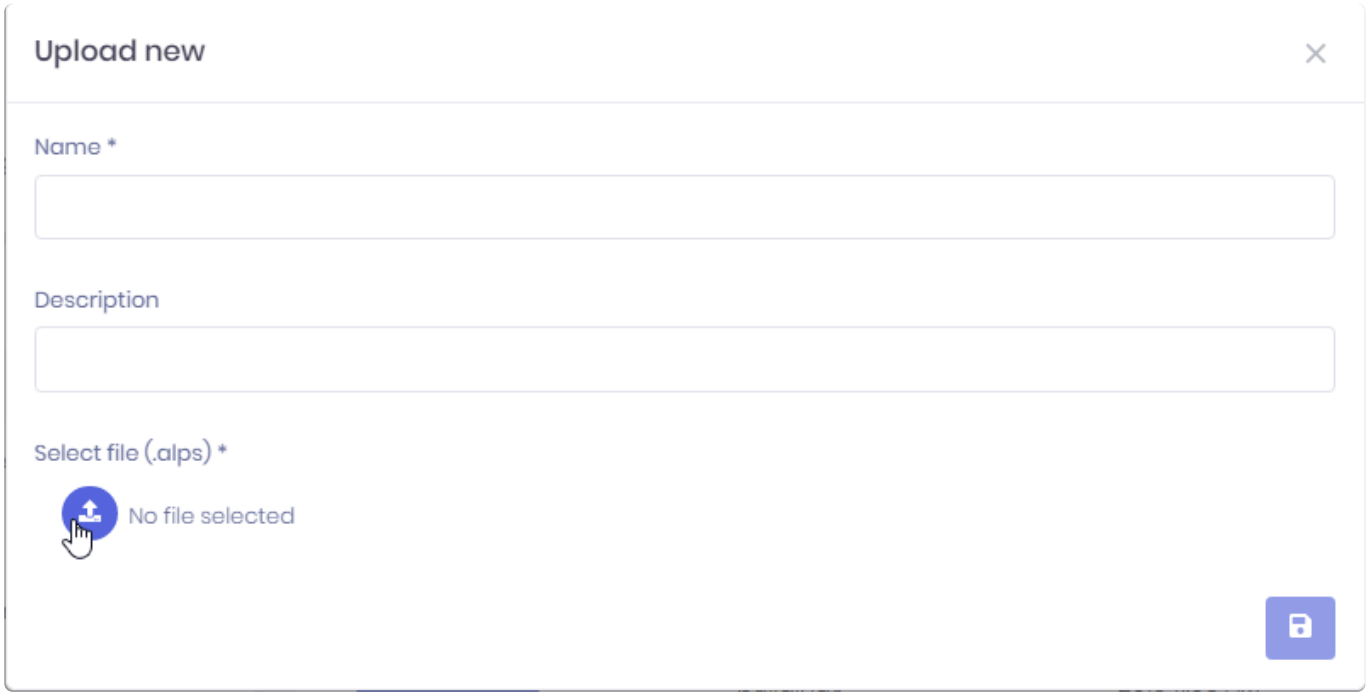
Publishing from Alpana Server

Data Sources can be published from Alpana Designer.
Alternately, you can publish a Data Source `.alps` file directly to the Server.

Click the *Add datasource* button :



This opens the *Upload New* dialog :

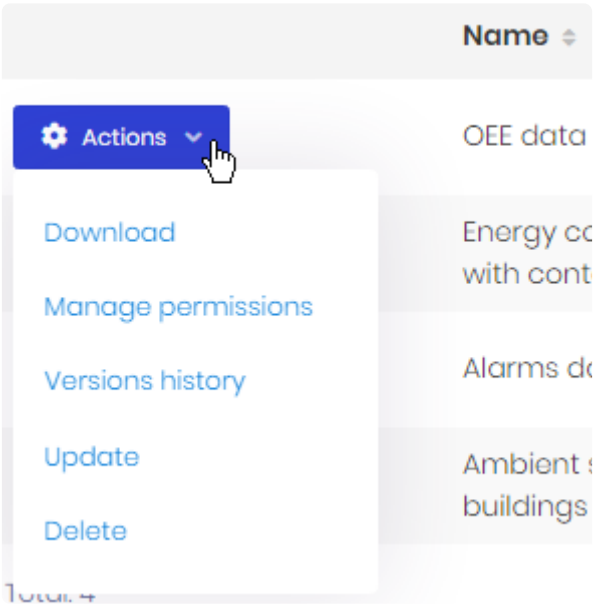


Click the *Select File* button to browse your filesystem for a datasource `.alps` file.
Fill-in a *Name* and optionally a *Description*.

When you click the *Save* button, the DataSource is published.

Actions menu

Each published DataSource has a set of actions available.
To list those actions, open the datasource menu next to the DataSource's name :



Some of those actions are detailed in the corresponding chapter :

- *Download*
- *Manage permissions*
- *Versions history*
- *Update*
- *Delete*

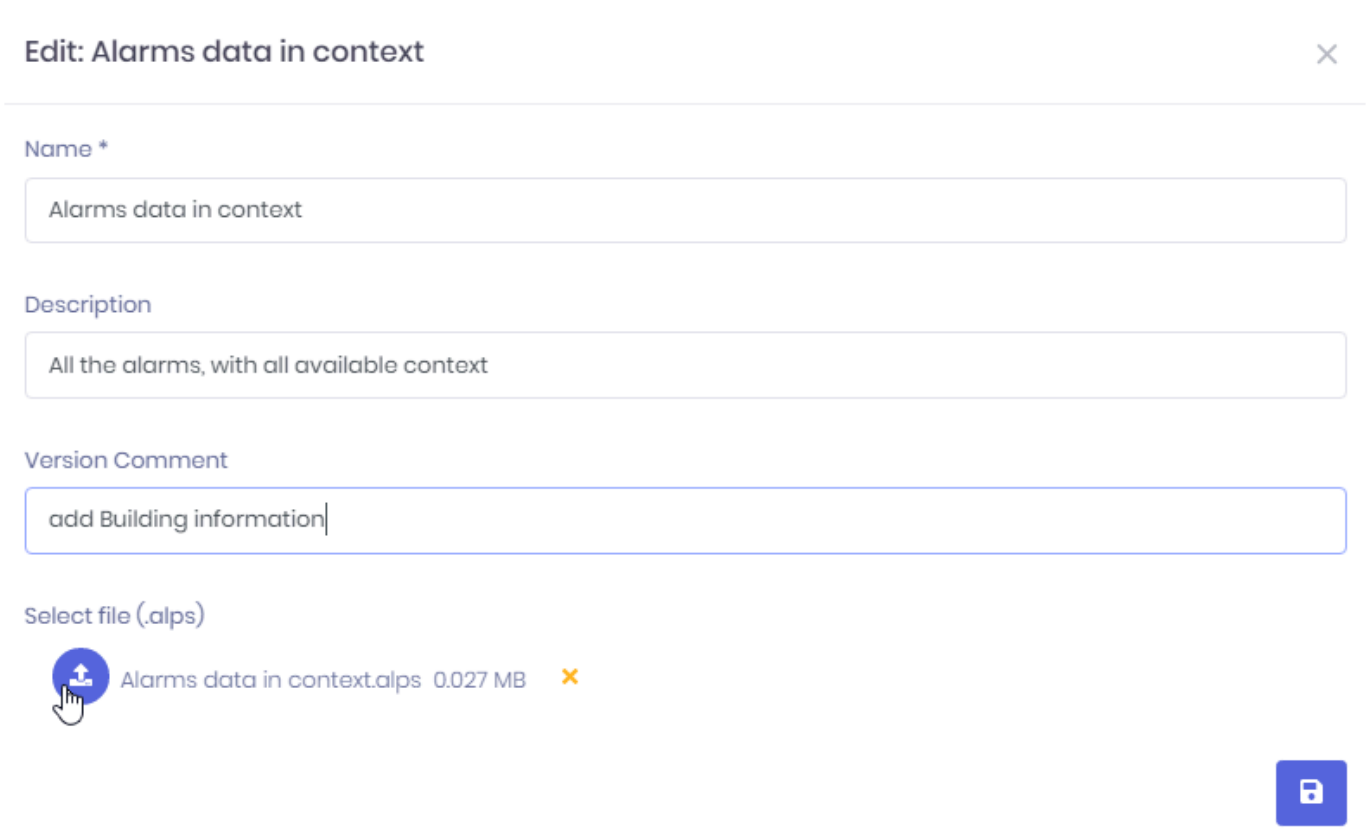
Download

This downloads the datasource `.alps` file.
This allows for example to import it in Alpana Designer.

Update

This opens the *Edit* dialog to update the DataSource's name and description.

You can also upload a new widget `.alps` file to replace the current one.
If you do so, this will create a new version, so you can input a *Version Comment*.



Delete

This deletes the selected Widget.

Manage permissions

(see corresponding chapter)

Versions history

(see corresponding chapter)

Last modified: Oct 28, 2019

3.6.2. Data Sources Version History

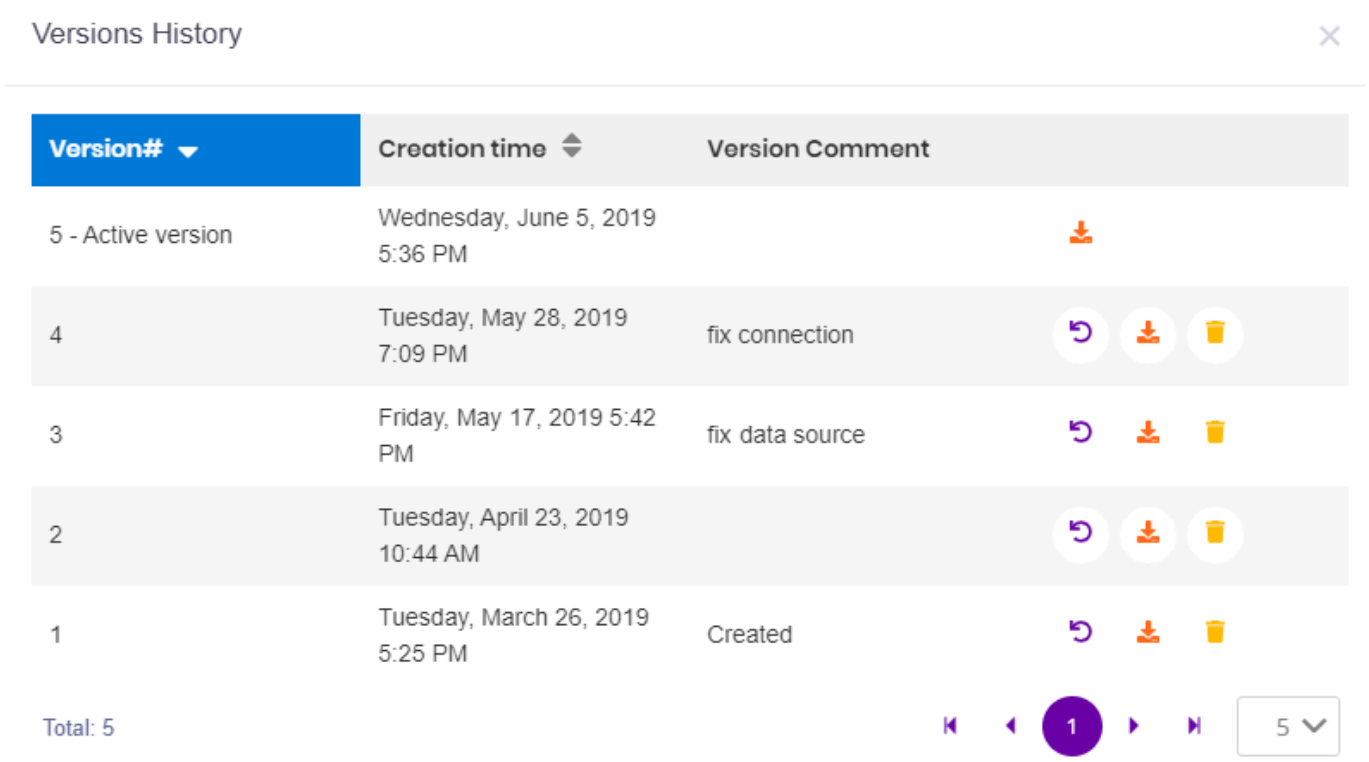
Every time you over-write an existing Data Source, the previous version in saved also.
The Version History allows you to see the changes applied to a Data Source, and restore a previous state.

Opening the Versions History

From each Data Source action menu, select *Versions history* to open the dialog.

Viewing Version History

The Versions History dialog shows a paginated list of Versions :



Each Version displays :

- a Version Number
- its Creation Time
- an optional Version Comment

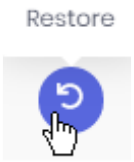
You can sort the list by version number and creation time bu clicking on the corresponding column headers.

You can browse versions pages and choose a page size at the bottom of the dialog.

Restoring an older Version

By default, the Active Version of a Data Source is the latest published.
However, you can manually restore any previous Version.

Click the _ Restore_ button in front of the selected Version :

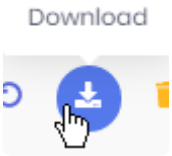


The *Active version* indicator will move back to that older version.

So if you want to restore the latest Version again (or any other Version), just do the same process again.

Downloading an older Version

Click the *Download* button to download the corresponding Version :



This will download the corresponding Data Source `.alps` file.

This is useful for example for studying a previous version, and maybe turning it into a separate Data Source.

Deleting an older version

You can delete any previous version.
Click the *Delete* button in front of the corresponding Version :

Last modified: Oct 28, 2019

3.7. Schedule Management

Alpana Server integrates a schedule management that allows to automate tasks.

This chapter describes those Schedules and their usage.

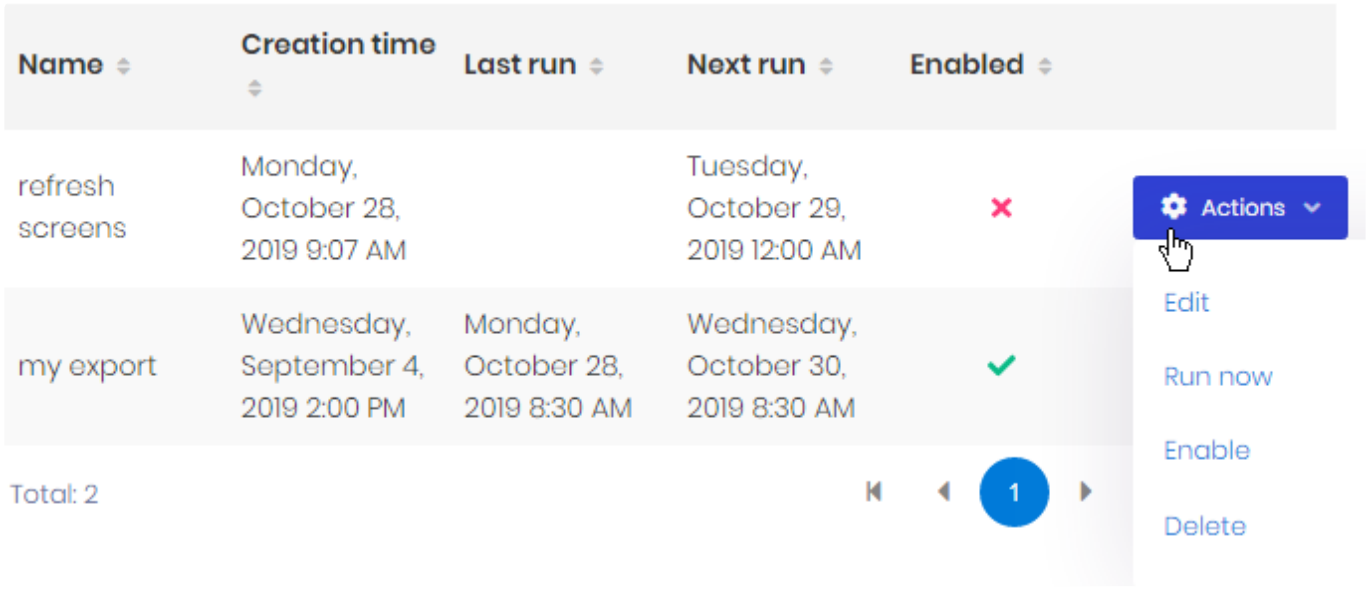
To access Schedule management, use the *Schedules* menu :

Last modified: Oct 28, 2019

3.7.1. Schedules management

Actions menu

Each Schedule has a set of actions available.
To list those actions, open the schedule menu next to the Schedule's name :



Some of those actions are detailed in the corresponding chapter :

- *Edit*
- *Run now*
- *Enable / Disable*
- *Delete*

Edit

This opens the *Edit* dialog to update the Schedule.
This dialog is detailed in the chapter dedicated to Creating a Schedule.

Run now

This forces the Action to run now, instead of waiting for the Trigger to happen.

Enable / Disable

This allows to enable or disable the Schedule.
The Schedule will only run if it is Enabled.

Delete

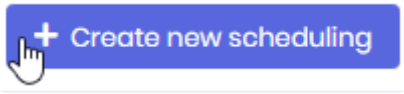
This deletes the selected Schedule.

Last modified: Oct 28, 2019

3.7.2. Creating a Schedule

This chapter describes creating a new Schedule or editing an existing Schedule.

To create a new Schedule, click the *Create new scheduling* button :



A Schedule is composed of 4 types of properties that correspond to the 4 steps of creating the Schedule :

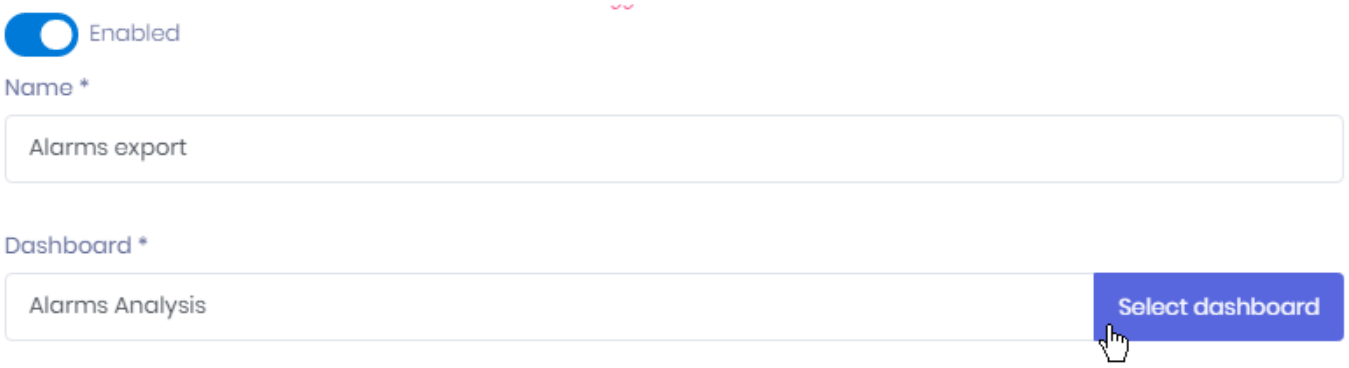
1. *Main* : the source of the Schedule (which Dashboard)
2. *Trigger* : the event that will trigger the Schedule
3. *Actions* : what the Schedule will do
4. *Targets* : who will receive the result

To navigate across the steps, you can use the *Previous* / *Next* buttons :

…, or you can click on the corresponding step :

Last modified: Oct 28, 2019

3.7.2.1. Main



Enabled

This allows to enable the Schedule as soon as it will be created.

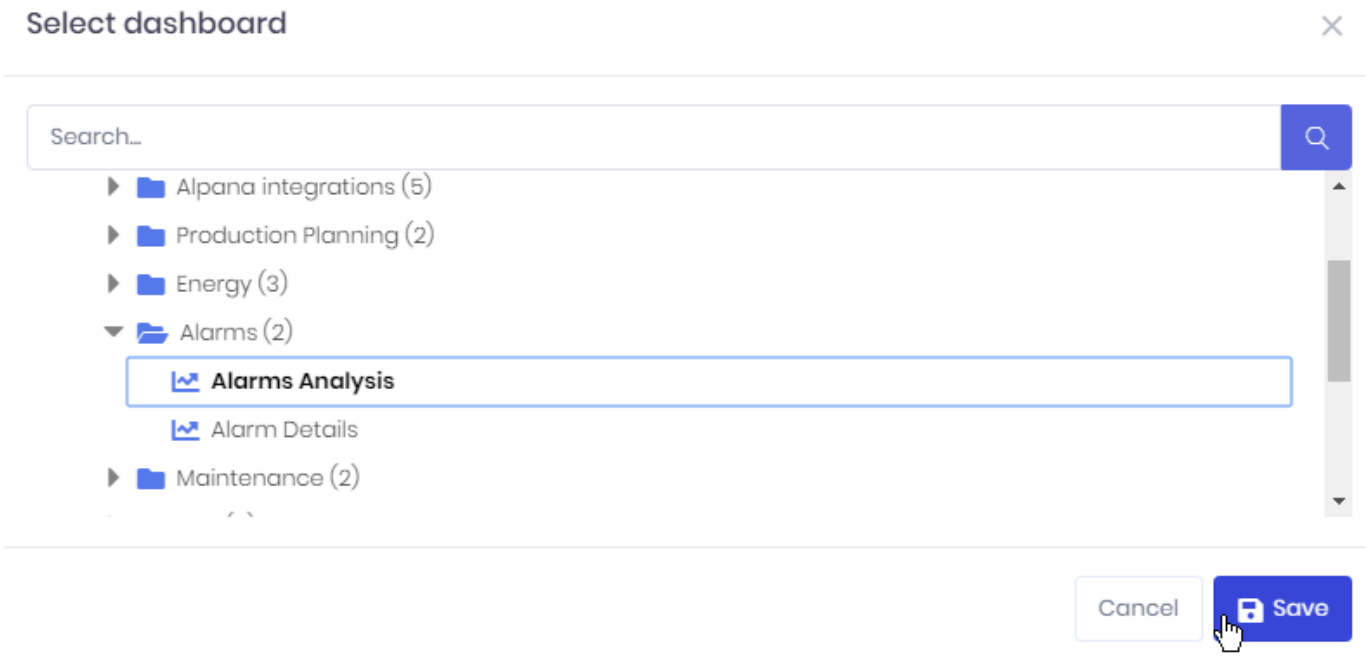
Name

The Schedule *Name* is used for identifying it in the list.
It can also be used as a placeholder inside the e-mail sent to users.

Dashboard

This is the dashboard that will be used as a Trigger (if using the “Data condition” mode) and as an attachment Action (when exporting the Dashboard to send as a e-mail).

If you opened the “Create Schedule” dialog from the Dashboards action menu, then the Dashboard is already selected.
Otherwise, click the *Select dashboard* button.
This opens the *Select dashboard* dialog to let you find the desired Dashboard :



Last modified: Oct 28, 2019

3.7.2.2. Trigger

The Trigger defines the event that will trigger the Action.

There are two types of Triggers :

- *Timer* : the event happens on a regular time-based schedule
- *Data condition* : the event happens when a condition is met

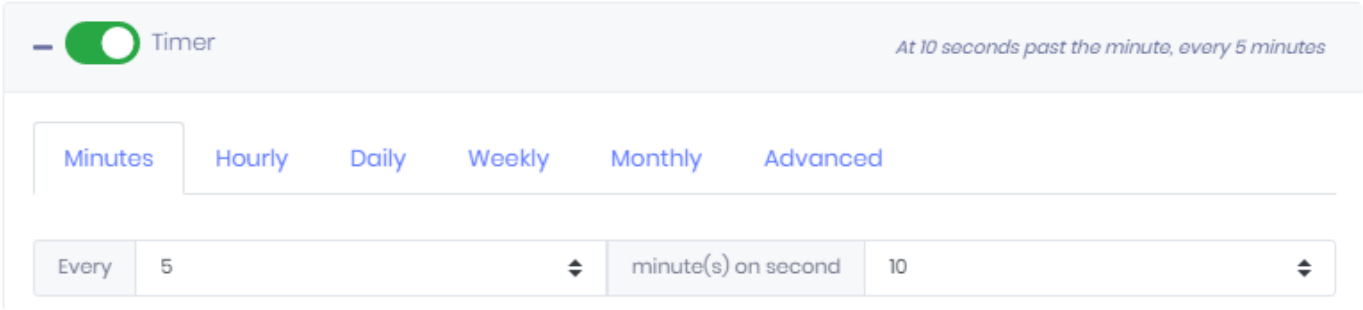
Last modified: Oct 28, 2019

3.7.2.2.1. Timer

Timed triggers can be defined in several ways depending on the desired time scale.
You can select a time scale from the corresponding tabs :

Minutes

Select a frequency in minutes :



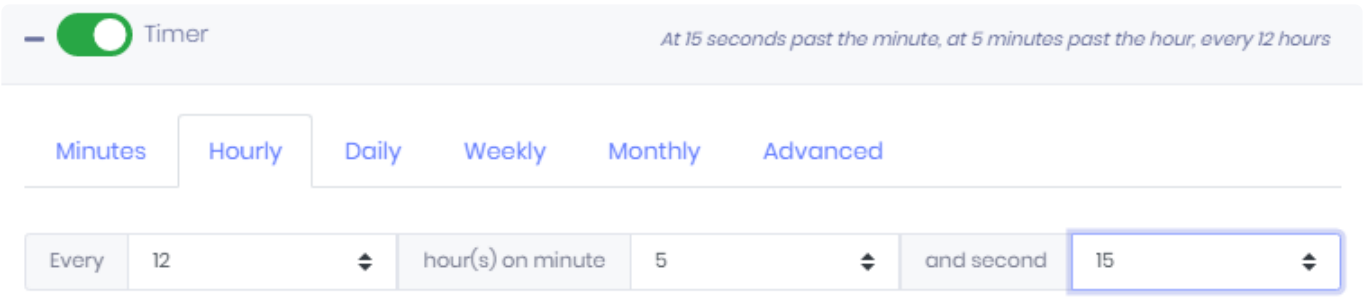
You can also specify a “second”. This means that the trigger will happen when the second is equal to the selected second.

Example :
I configured “Every 5 minutes on second 10”.
This means the trigger will happen at :

- 11:00:10
- 11:05:10
- 11:10:10
- 11:15:10
- 11:20:10
- …

Hourly

Select a frequency in hours :



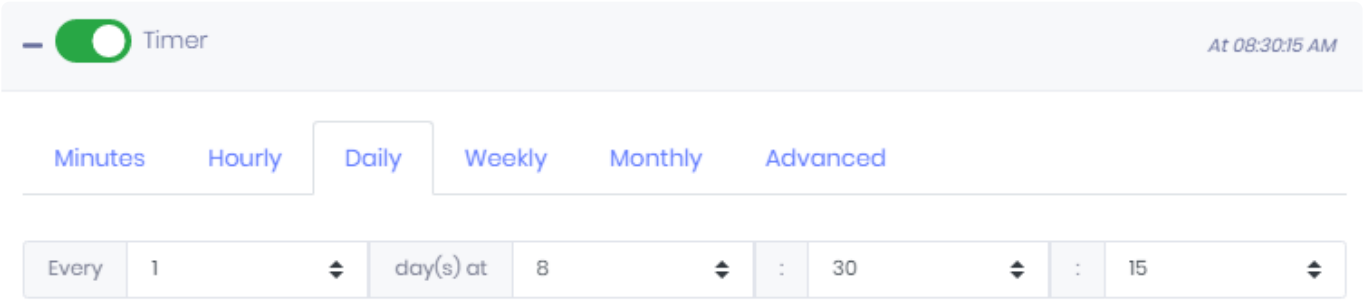
You can also specify a “minute” and a “second”. This means that the trigger will happen when the minute:second is equal to the selected minute:second.

Example :
I configured “Every 12 hours on minute 5 and second 15”.
This means the trigger will happen at :

- 2019-10-28 at 11:05:15
- 2019-10-28 at 23:05:15
- 2019-10-29 at 11:05:15
- 2019-10-29 at 23:05:15
- …

Daily

Select a frequency in days :



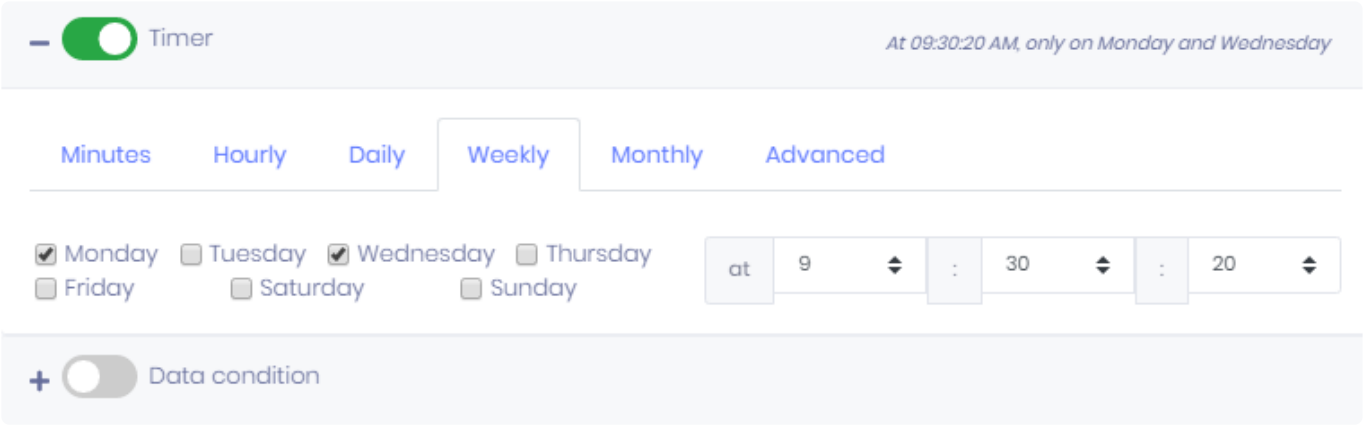
You can also specify a “hour”, a “minute” and a “second”. This means that the trigger will happen when the hour:minute:second is equal to the selected hour:minute:second.

Example :
I configured “Every 1 day on hour 8 on minute 30 and second 15”.
This means the trigger will happen at :

- 2019-10-01 at 08:30:15
- 2019-11-01 at 08:30:15
- 2019-12-01 at 08:30:15
- …

Weekly

Select desired days of the week :



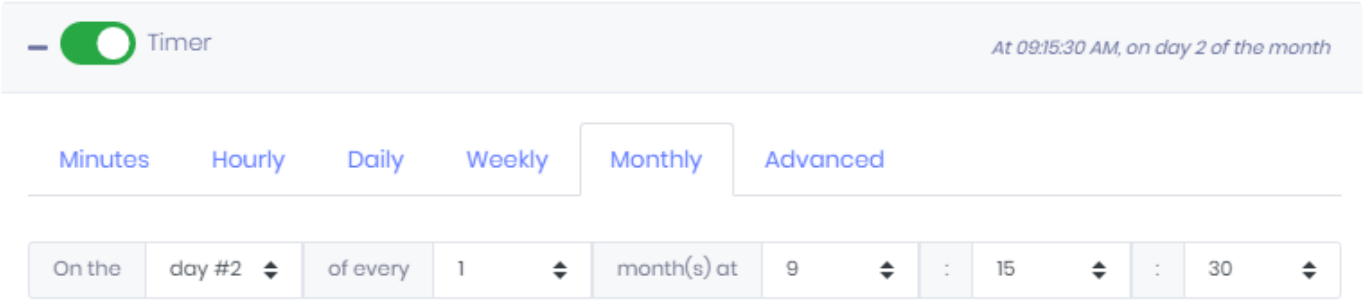
You can also specify a “hour”, a “minute” and a “second”. This means that the trigger will happen when the hour:minute:second is equal to the selected hour:minute:second.

Example :
I configured “Every Monday and Wednesday on hour 9 on minute 30 and second 20”.
This means the trigger will happen at :

- Monday 2019-10-07 at 09:30:20
- Wednesday 2019-10-09 at 09:30:20
- Monday 2019-10-14 at 09:30:20
- Wednesday 2019-10-16 at 09:30:20
- …

Monthly

Select desired day of the month :



You can also specify monthly frequency, a “hour”, a “minute” and a “second”. This means that the trigger will happen every X months when the hour:minute:second is equal to the selected hour:minute:second. Instead of a day number, you can also select “Last day”, which will always be the last day of the month.

Example :
I configured “On day #2, every 1 month on hour 9 on minute 30 and second 20”.
This means the trigger will happen at :

- 2019-10-02 at 09:15:30
- 2019-11-02 at 09:15:30
- 2019-12-02 at 09:15:30
- 2020-01-02 at 09:15:30
- …

Advanced

For more complex trigger expressions, you can use CRON expressions

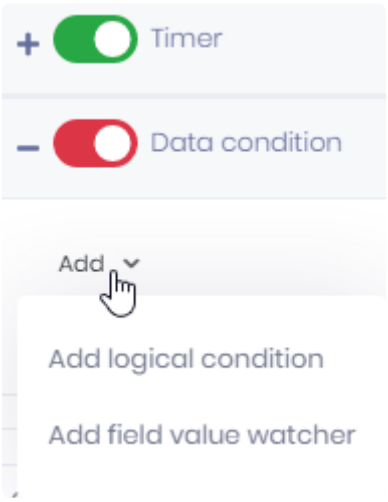
The supported expressions are documented here : https://www.quartz-scheduler.net/documentation/quartz-3.x/tutorial/crontrigger.html

Data condition

> Note : to define a *Data condition* trigger, you must also set a *Timer*.
> This is needed to define when the condition will be checked.

To add a Data Condition, select the desired type :

- *Add logical condition* : this allows to compare a widget field value to a target for example
- *Add field value watcher* : this allows to follow a field value over time



Logical Condition

Field value watcher

==========

condition add logical condition add field value watcher configure criteria for combining conditions list conditions remove condition logical condition select widget from dashboard select truth criteria configure

comparison with Field, Operator, Const configure comparison with Field, Operator, Field combine comparisons with boolean operators : add/remove branches, add child comparison or boolean operator filter condition with Field, Operator, Const filter condition with Field, Operator, Field field value watcher select widget from dashboard select field from widget watch value changes watch value does not change configure deadband with widget date/time field configure change thresholds low/high, in absolute or percentage of value filter condition with Field, Operator, Const filter condition with Field, Operator, Field

Last modified: Oct 28, 2019

3.7.2.2.2. Data condition

Note : to define a *Data condition* trigger, you must also set a *Timer*.
This is needed to define when the condition will be checked.

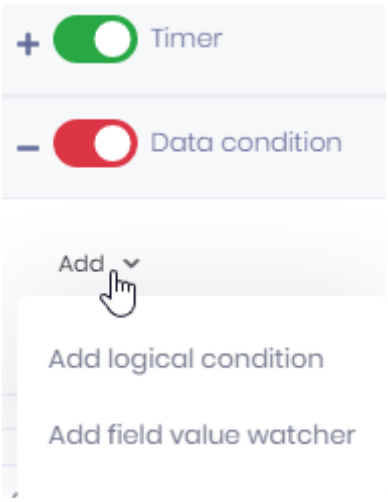
Last modified: Mar 26, 2020

3.7.2.2.2.1. Managing conditions

Adding a condition

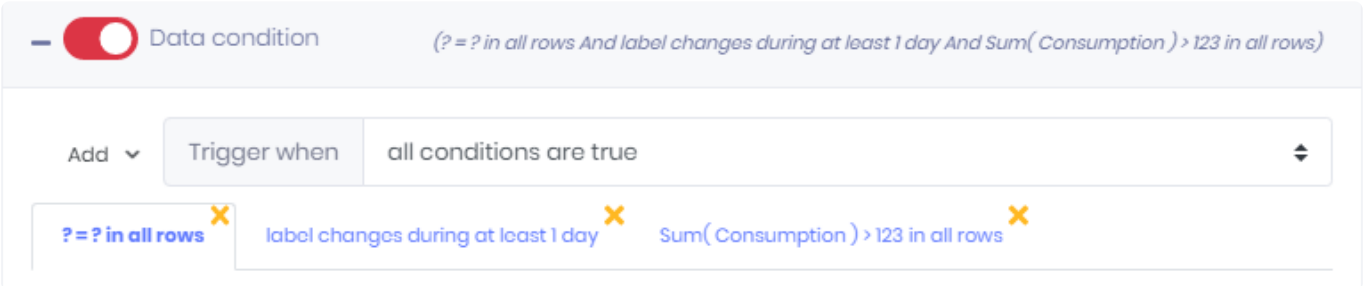
To add a Data Condition, select the desired type :

- *Add logical condition* : this allows to compare a widget field value to a target for example
- *Add field value watcher* : this allows to follow a field value over time



Combining conditions

When several Conditions are combined, you can browse them using tabs, and delete them using the cross icon on the corresponding tab :



You can also define how the Conditions will be combined.
Select a rule from the *Trigger when* list :

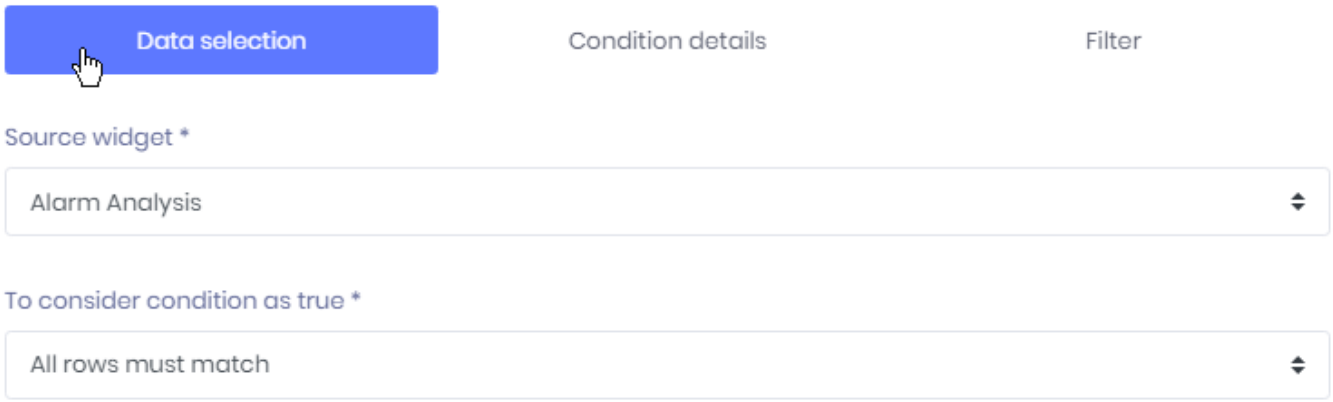
- *Trigger when all conditions are true*
- *Trigger when at least one condition is true*
- *Trigger when non conditions are true*
- *Trigger when exactly one condition is true*

Last modified: Mar 26, 2020

3.7.2.2.2.2. Logical condition

Logical conditions have three tabs to configure :

Data selection



This is where you define which Widget will be used to get the data : the *Source Widget*.
The list contains all Widgets of the selected Dashboard.

You can also define the rule to consider the condition to be true :

- *All rows must match*
- *At least one row must match*
- *No rows must match*
- *First row must match*
- *Last row must match*
- *A minimum number of rows must match* (you must also configure the minimum number of rows to match)
- *A minimum number of rows must match, without break* (you must also configure the minimum number of rows to match)

This means that when the Widget query returns several rows of data, you are able to define a truth rule more precisely.

Note : on widgets that aggregate data (for example with fields configured as `SUM(myField)`), then the result is only a single row of data, aggregated over all the widget.

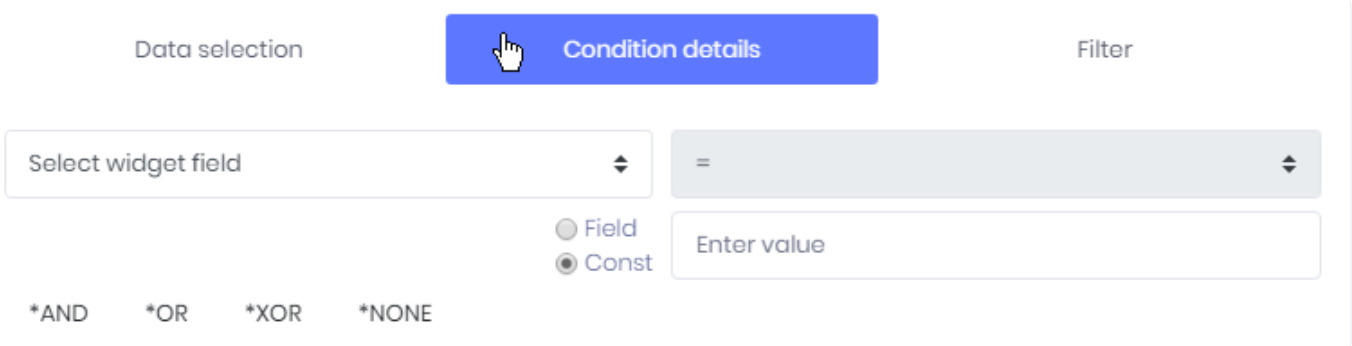
Condition details

This is where you configure the condition itself.
The condition is composed of Comparisons : boolean expressions that compare Widget fields to another

value.

Configuring a Comparison

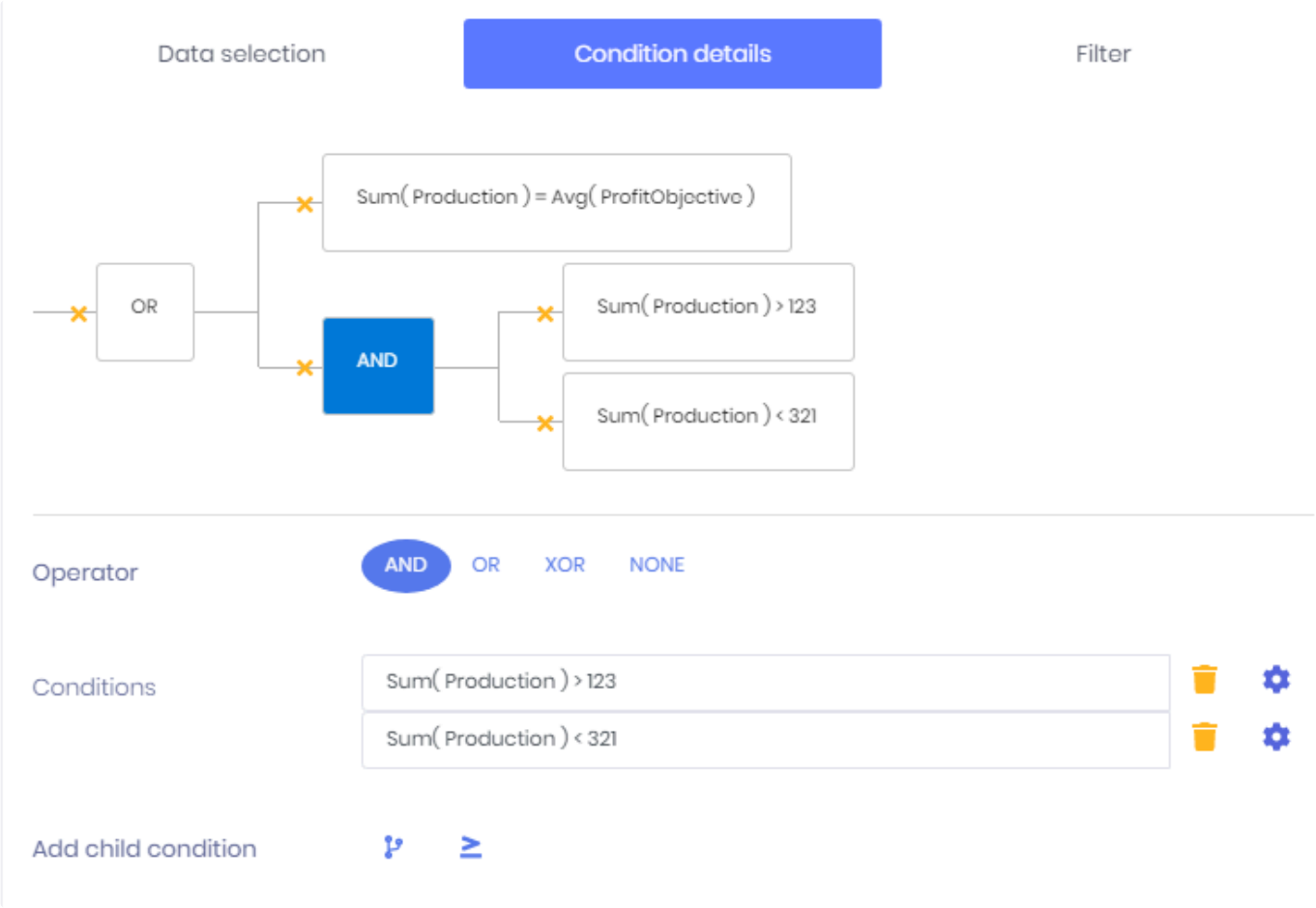
To configure a Comparison, you need to configure :



- a Widget field (belonging to the Widget selected in the *Data selection*)
- a boolean operator (=, <>, >, >=, <, <=)
- a reference value. This can be a constant (*Const*) that you enter manually now, or another *Field* from the Widget (like a target or threshold value)

Combining Comparisons

Optionally, you can use logical Operators (`AND`, `OR`, `XOR`, `NONE`) to combine several Comparisons together :



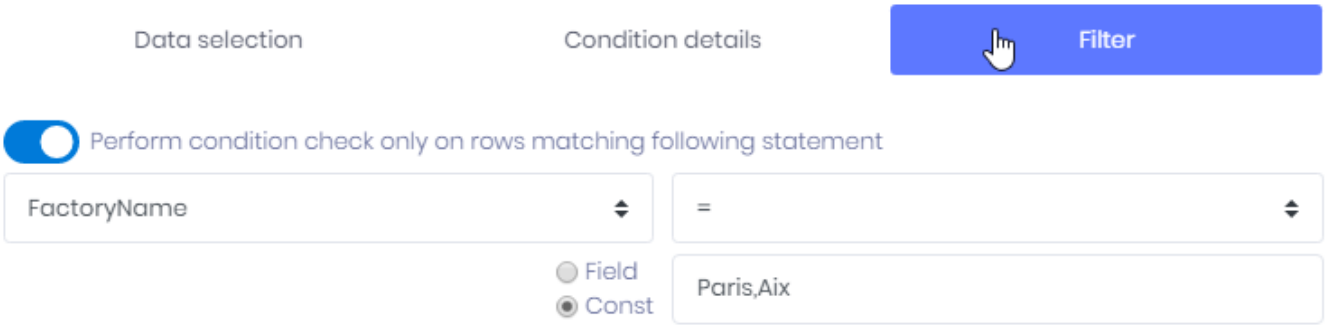
The Comparisons are linked together by Operators inside a tree.
To delete a node of the tree, click the corresponding cross icon.
To configure a node, click on it.

On an Operator, you can :

- modify the Operator to select a different Operator
- modify or delete children Conditions
- add a child Conditions, which can be another Operator or a Comparison

Filter

You can also filter the result to only accept some data :



You can create a filter on a string field for example.

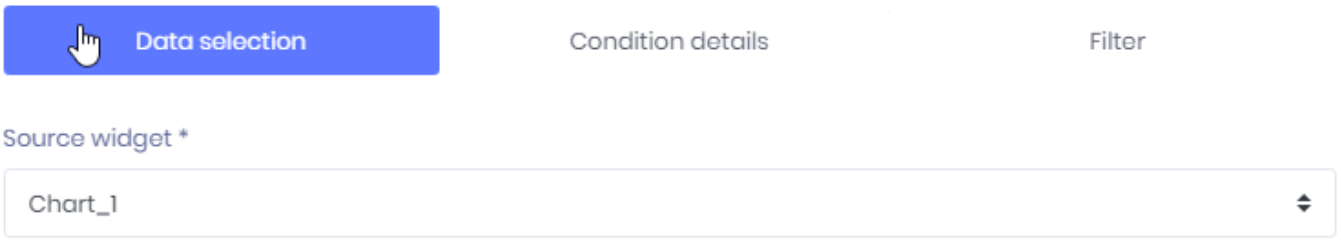
Last modified: Mar 30, 2020

3.7.2.2.2.3. Field value watcher

The Field Value Watcher will look at data changes over time to detect when a KPI goes out of a normal state.

Field value watchers have three tabs to configure :

Data selection

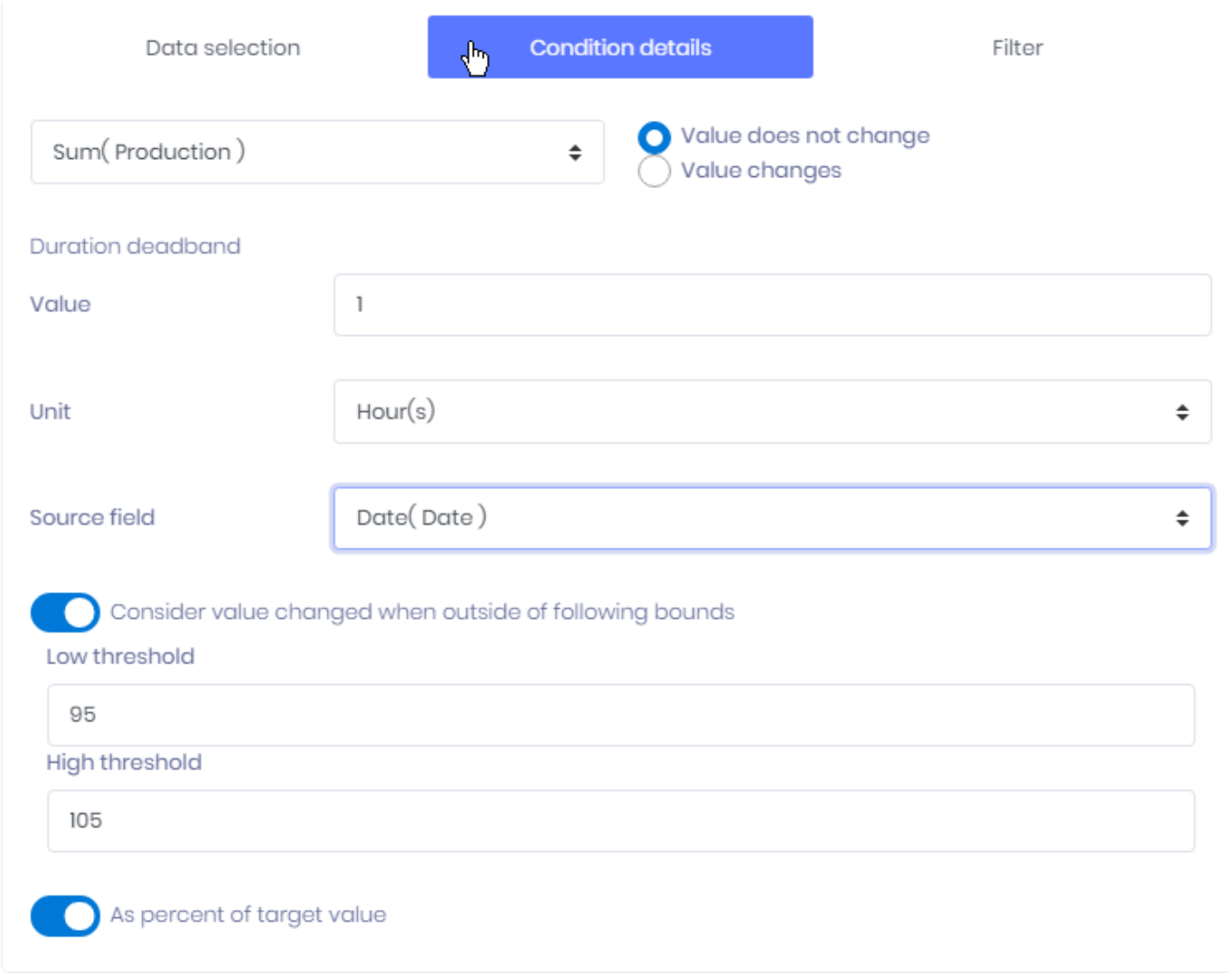


This is where you define which Widget will be used to get the data : the *Source Widget*.
The list contains all Widgets of the selected Dashboard.

> The Field Value Watcher requires a Date field, so you must have at least one in your widget.

Condition details

This is where you configure the condition itself.



Select field to watch

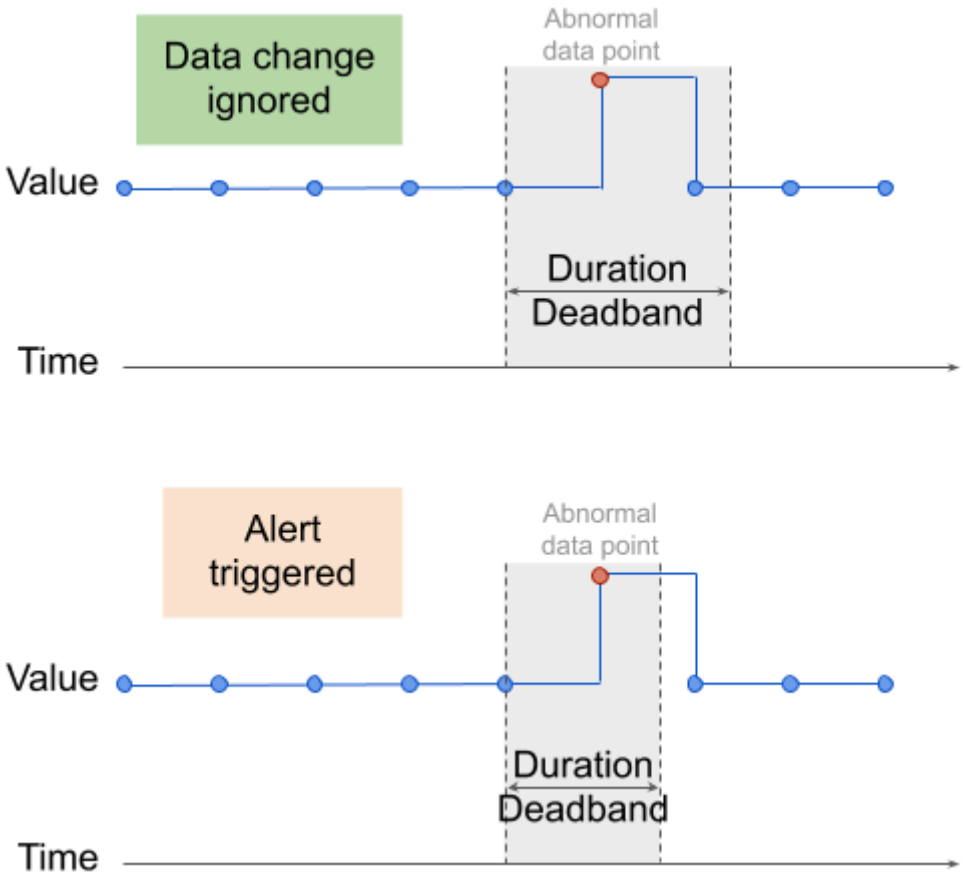
Select which measure field from the data of the Widget you need to watch.

Value does not change / Value changes

Select if you want to be alerted when the value stays the same, or when the value changes.

Duration deadband

Allows to define a deadband to ignore changes that happen too fast.
This requires a date field from the Widget.



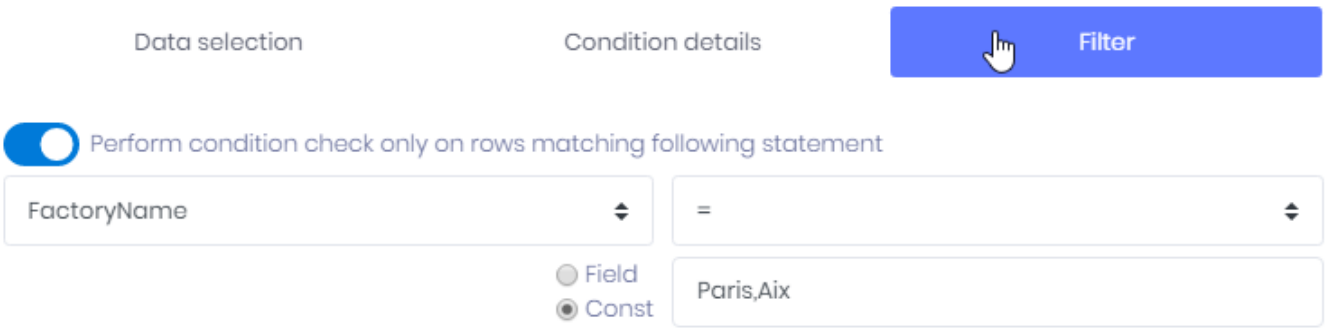
If the duration between two consecutive constant values is always smaller than the Deadband, then the abnormal values are ignored.

Consider value changed when outside of following bound

This allows to define a value interval to consider that the value really changed.
This can be expressed in absolute value, or *As percent of the target value*.

Filter

You can also filter the result to only accept some data :



You can create a filter on a string field for example.

Last modified: Mar 26, 2020

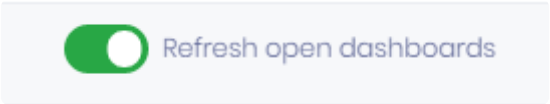
3.7.2.3. Actions

The Action defines what will be executed when the Trigger happens.

There are two types of Actions :

Refresh open dashboards

This will force open the Dashboards of the *Target* users to refresh.

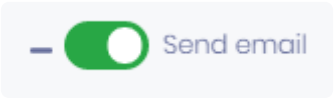


This is useful for implementing **even-based refresh** :
You have a Dashboard on a TV screen logged with `UserTV`.
You want to refresh the data in the Dashboard only when a new batch has been completed.
You set up :

- a *Trigger* that is based on a *Data Condition* corresponding to “batch was completed”.
- an *Action* to *Refresh open dashboards*
- as *Target*, User `UserTV`

Send email

This will send an e-mail to the *Targets* with an export of the Dashboard as attachment :



Attachment format

Select the desired file format for the attachment :

- *Image*
- *PDF*
- *Excel*

The Dashboard will be exported to the corresponding format and attached to the e-mail.

Layout Configuration

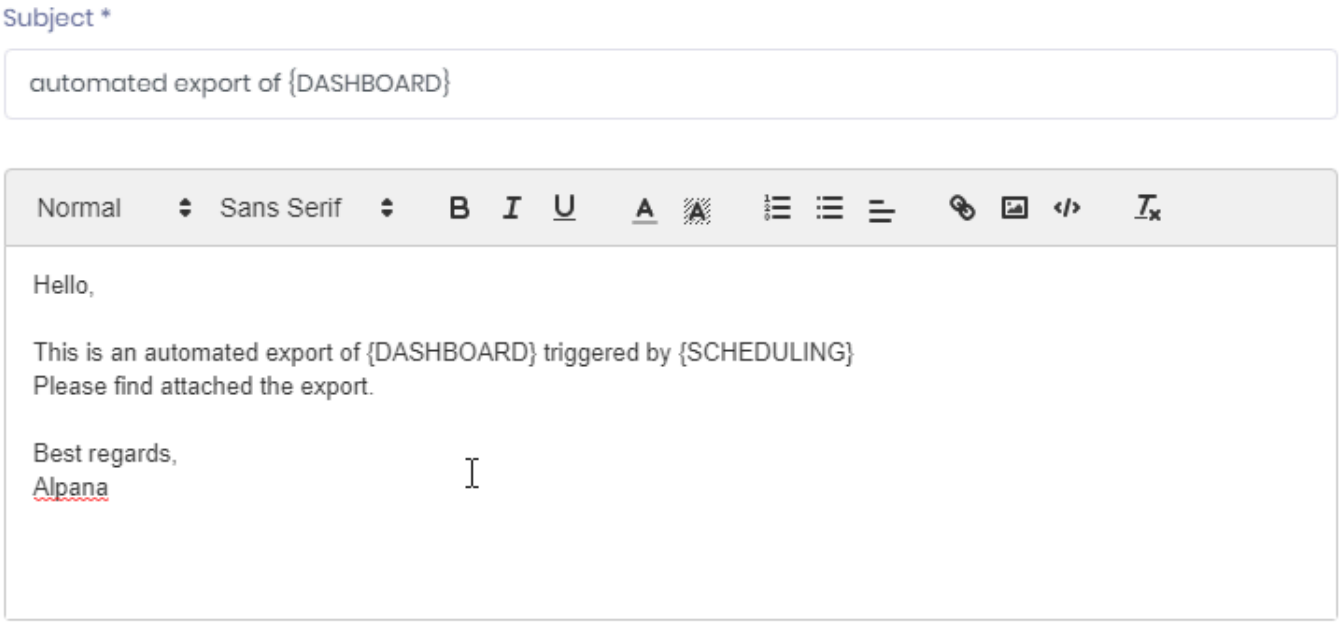
Image and *PDF* exports require to choose a Layout.

Subject

Write the subject field of the e-mail.

Content

Write the content of the e-mail using a rich text editor.



Placeholders

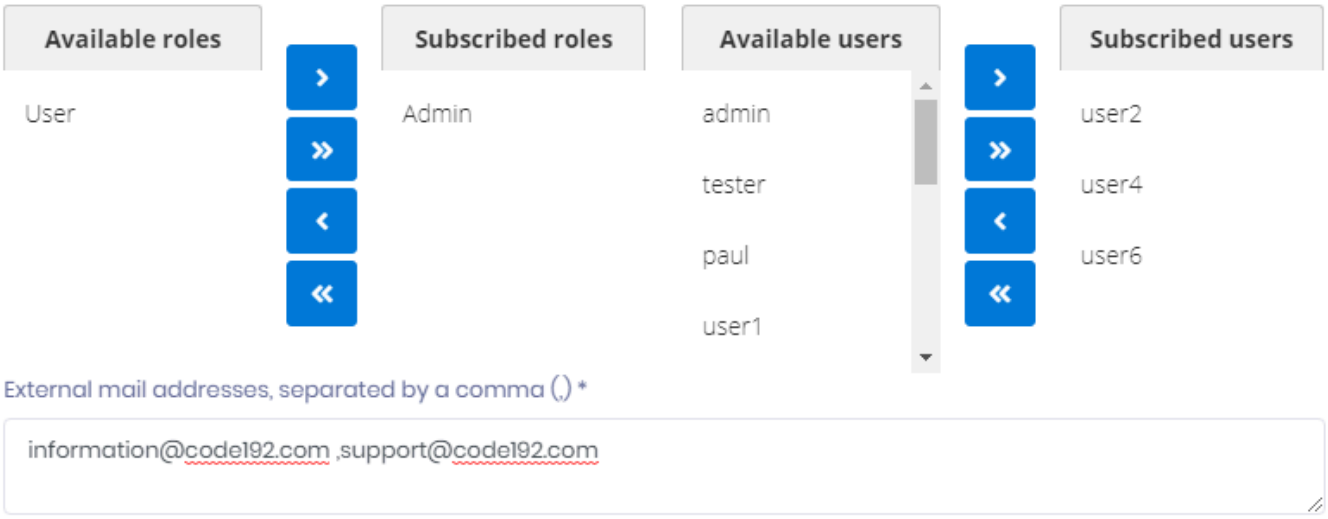
In the content and Subject, you can use some placeholders for displaying some variables values :

```
{SCHEDULING} : the name of the Schedule that is running
{DASHBOARD} : the name of the Dashboard exported
```

Last modified: Jun 17, 2020

3.7.2.4. Targets

The *Targets* are the intended recipients of the Schedule : the *Action* will be performed for them only.
You can see this as a mailing list for example.



Adding/removing Targets

To add an item (Role, User) from the list of Available items, select it and click the **>** icon.
To add all items from the list of Available items, click the **>>** icon.
To add an item from the list of Subscribed items, select it and click the **<** icon.
To add all items from the list of Subscribed items, click the **<<** icon.

Roles

You can target a list of Roles.
All users who have this Role will be a Target.

This is the most efficient way to maintain the Targets, because you won't have to edit the Schedule anymore, only the Role.
For example, if you want to add or remove users as Targets of this Schedule, you can just add or remove the dedicated Role to them.

Users

You can target a list of Users.
All individually selected users will be a Target.

External e-mails

You can target a list of e-mails.
This allows to send e-mails to users who are not registered Users on Alpana Server.

To include several e-mails, use the comma ， character.

This is the least efficient way to manage it, because you will have to edit the Schedule every time the e-mail changes.
Also, you won’t be able to force a Refresh as an *Action*.

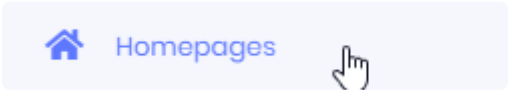
Last modified: Oct 28, 2019

3.8. Homepages and Pinning Widgets

Homepages are visual bookmarks of Widgets that can come from any Dashboard.
The action of selecting a Widget for adding it to a Homepage is called “Pinning the Widget to the Homepage”.
Your default Homepage is the first page that is displayed when you connect to Alpana Server.

This is useful for easily recomposing Dashboards based on existing visualizations and monitoring KPIs easily as soon as you connect.

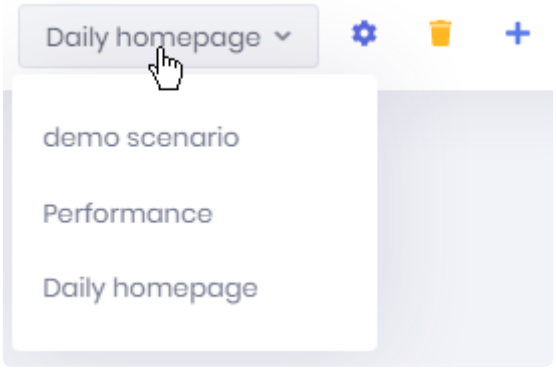
To access Homepages, use the *Homepages* menu :



Last modified: Oct 28, 2019

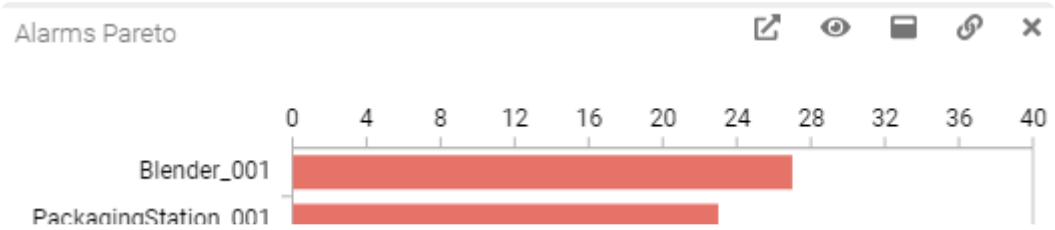
3.8.1. Viewing Homepages

By default, your *Default* Homepage is displayed.
To select a different Homepage, use the list :



Widget controls

Each widget has a set of controls attached :



Open Dashboard

This button allows you to open in a new tab the Dashboard that originally contained this Widget :

This allows to see the Widget in the context of its full Dashboard and get more context information where it came from.

Open in new tab

This button opens the selected Widget as a full page in a new tab :

Maximize

This button opens the selected Widget as maximized in the current page :

Embed

This button opens the *Get Link / Embed* dialog for the corresponding Widget :

Last modified: Oct 28, 2019

3.8.2. Creating Homepages

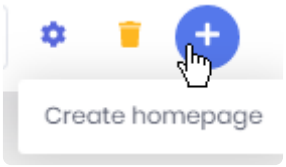
To create a new Homepage, you can either :

- add an empty Homepage from the *Homepages*
- create a new Homepage on the fly when Pinning a Widget from a Dashboard

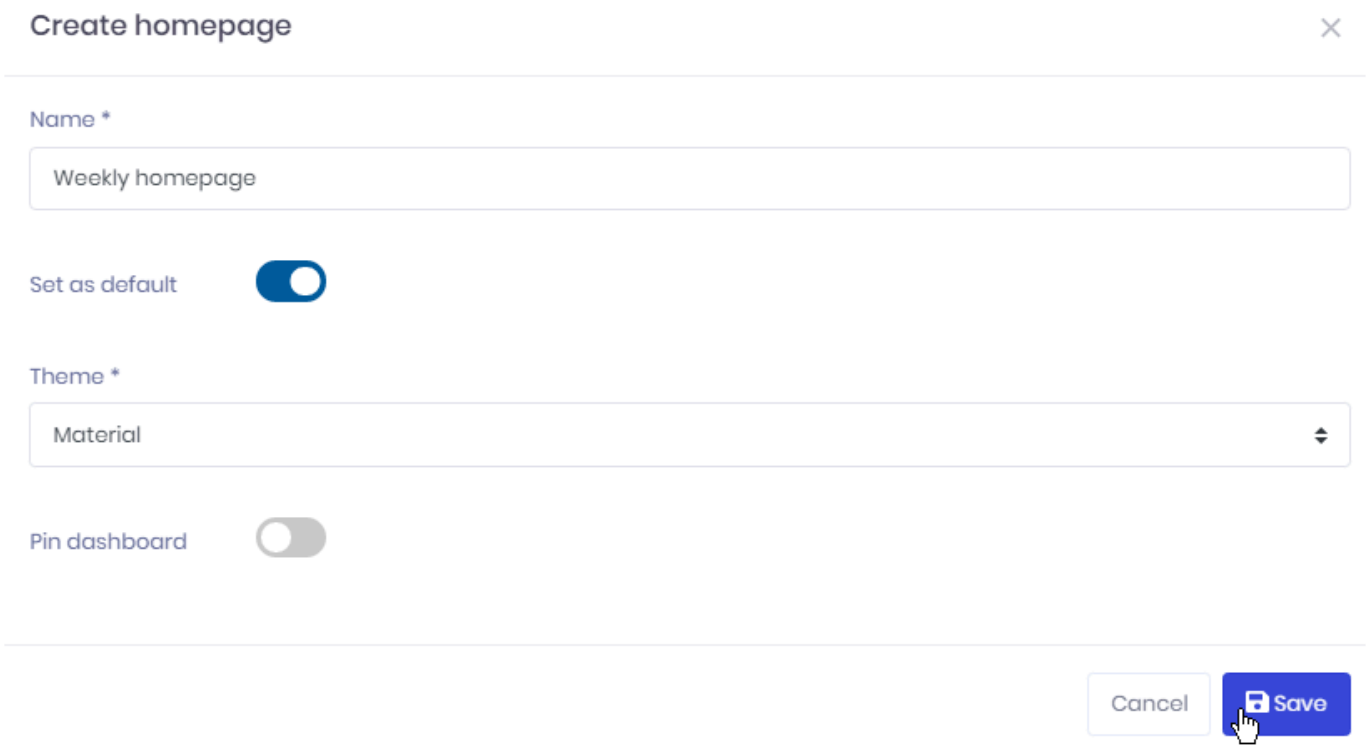
Create Homepage

Create an empty Homepage

Click the *Create homepage* button to add a new empty homepage :



Fill-in the desired *Name* and *Theme*, optionally, select this to be your Default Homepage (opened first) :



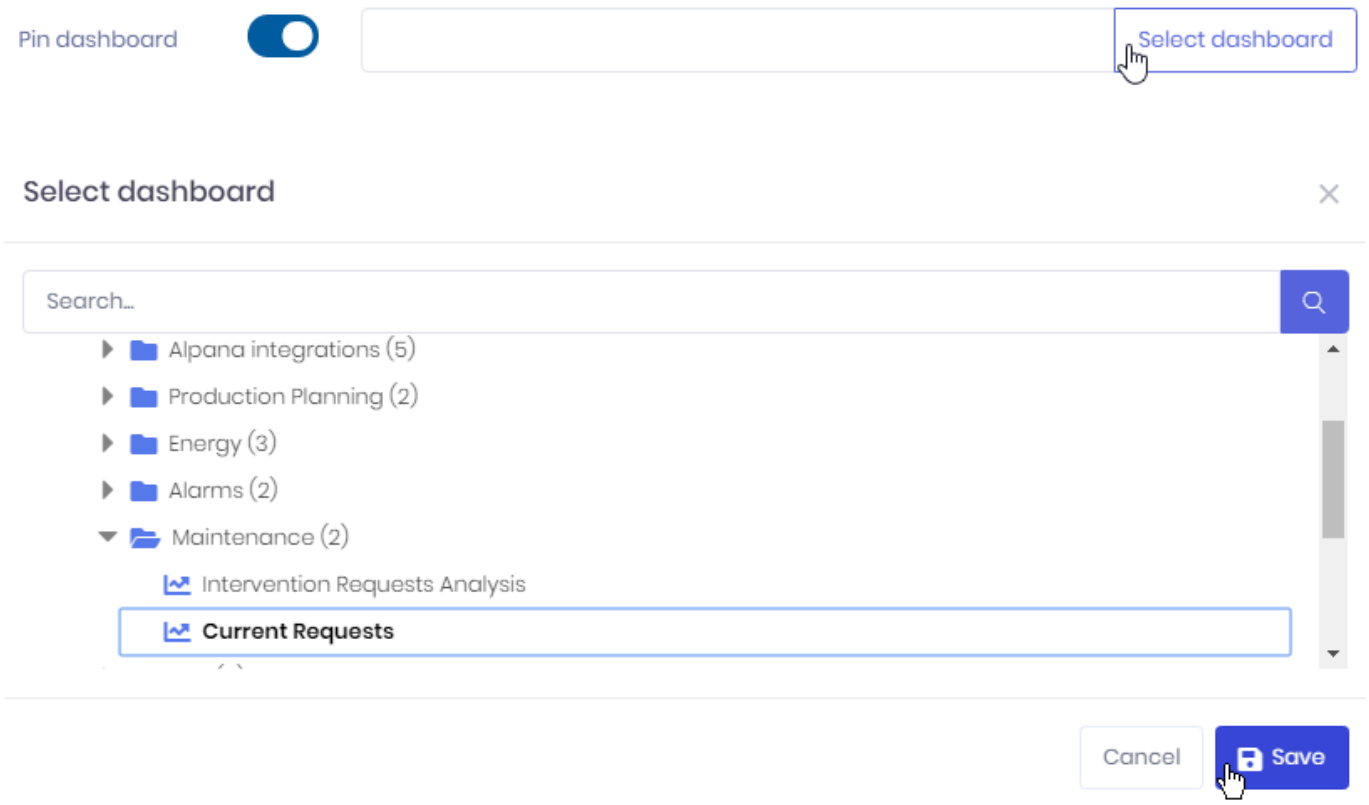
Click *Save*, and your Homepage is ready to be used for Pinning Widgets to it.

Pin dashboard

Instead of an empty Homepage that could be used for pinning Widgets, you can choose to pin a fill

Dashboard.

Check *Pin dashboard* and click *Select dashboard* to browse for the desired Dashboard :

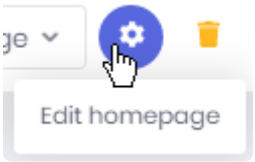


The selected Dashboard will be pinned entirely to the Homepage.

When a Homepage contains a Pinned Dashboard, it cannot have any other content (like Pinned Widgets)

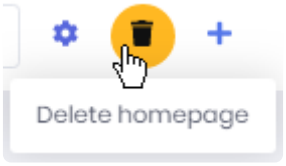
Edit homepage

Once created, the Homepage can be edited.
Click the *Edit homepage* button to open the dialog again and change the *Name*, *Theme* and *Default* status of the Homepage :



Delete homepage

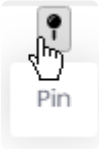
Once created, the Homepage can be deleted.
Click the *Delete homepage* button to delete it :



Last modified: Oct 28, 2019

3.8.3. Pinning Widgets

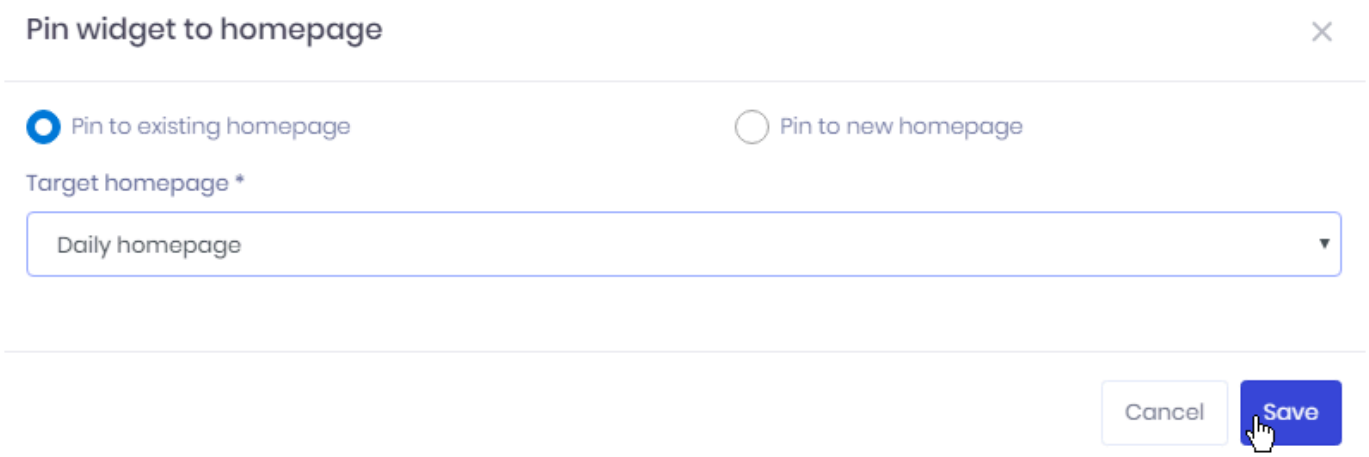
Widgets can be configured to show a “*Pin*” control.
This control allows to Pin the Widget to a Homepage.



This opens the *Pin widget to homepage* dialog where you can :

Pin to existing homepage

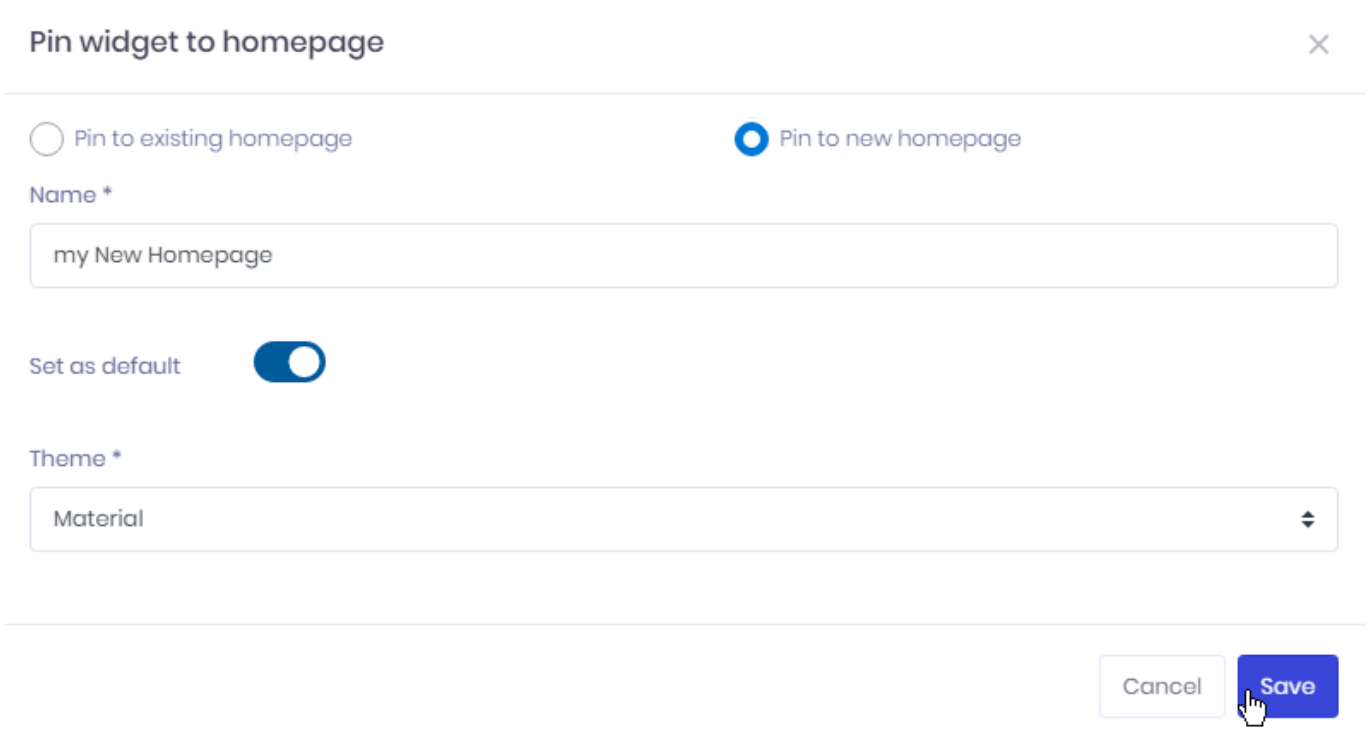
This sends the widget to an existing Homepage :



Select the desired Homepage from the list, and click *Save*.

Pin to new homepage

This creates a new Homepage and sends the Widgets there :



Give a *Name* to this new Homepage, select a *Theme*, and decide if you want it as your Default Homepage.

Last modified: Oct 28, 2019

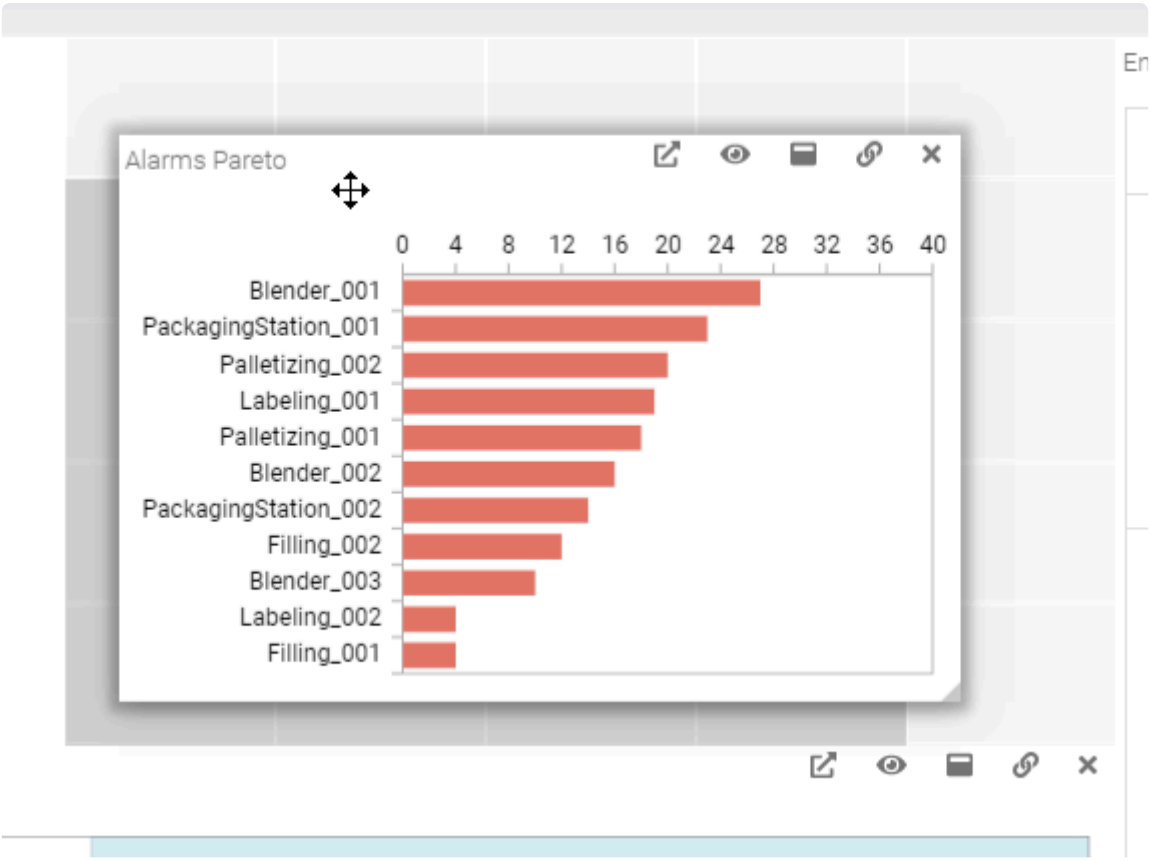
3.8.4. Editing the Homepage Layout

After Pinning Widgets to the Homepage, you can manually arrange the layout.

> Note : If a full Dashboard is pinned, the layout cannot be modified.

Moving a Widget

To move a Widget to the desired place, grab it by the title and drag it :

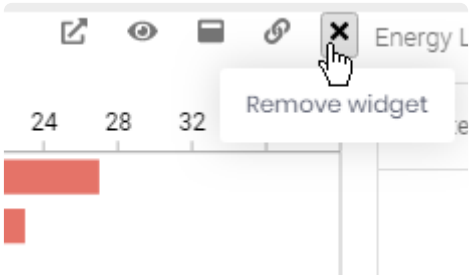


Resizing a Widget

To resize a Widget, use the sides and corners :

Removing a Widget

If you wish to remove the Widget from the Homepage, click the cross icon in the controls of the Widget :



Last modified: Oct 28, 2019

3.9. Runtime Connections

(new in Alpana 2020 R1)

Alpana Runtime Connections are tables of data hosted within Alpana Server.
Dashboards can connect to this data to display it inside widgets (see Alpana Runtime Connection from the Designer).
Dashboards can present data input forms to allow users to fill these tables.
The table's schema and data can be configured in Alpana Server.

A typical use-case is to replace Excel-based data input with a solution that is scalable, secure, and compatible with the usage within a web server.

To access Runtime Connections management within Alpana Server, use the *Runtime Connections* menu :

Last modified: Jun 16, 2020

3.9.1. Folders

Runtime Connections are organized in Folders.
Folders can contain other Folders.

This Chapter documents the management of Folders.

Creating Folders

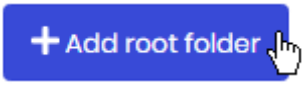
> When you install Alpana Server or create a new Tenant, no Folders exist by default.
> You will need to create at least one Folder for creating a Runtime Connection.

There are two types of Folders :

1. *root Folders* are at the root of the tree structure.
2. *sub Folders* are nested under root Folders or other sub Folders

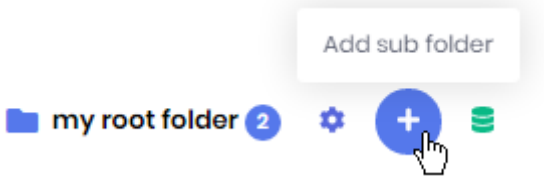
Creating a root Folder

To create a root Folder, click *Add root Folder* :



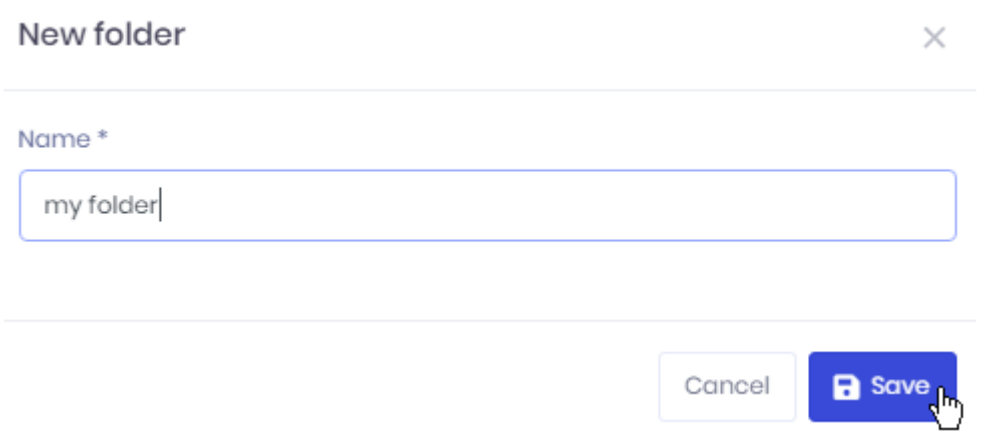
Creating a sub Folder

To create a sub Folder, click on the desired parent Folder, then click the *Add sub Folder* icon in front of it :



New Folder menu

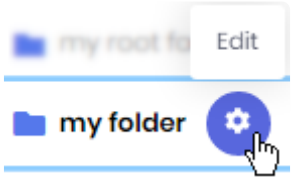
When creating a Folder, fill in the desired name and click *Save* :



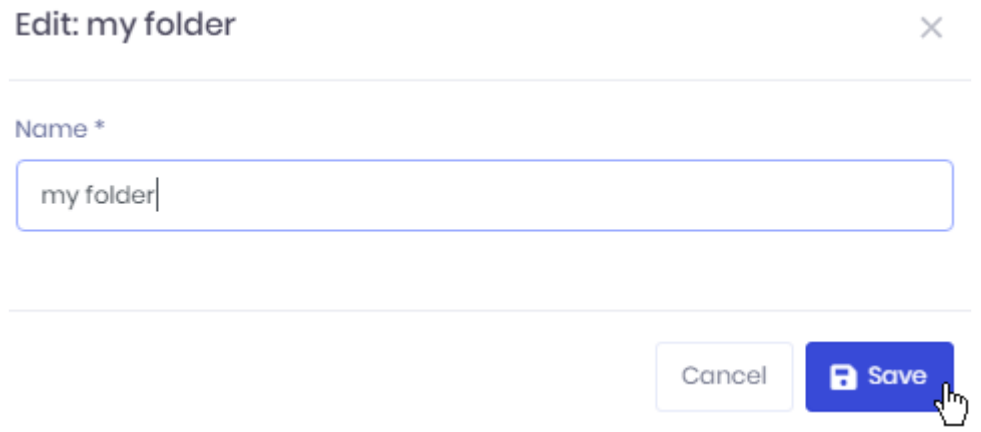
Managing Folders

Editing a Folder

To Edit a Folder, click on it, then click the *Edit* icon in front of it :



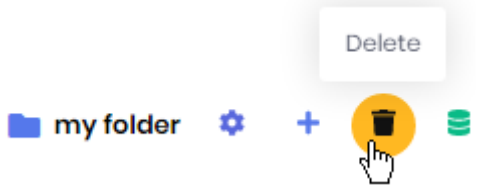
This allows you to change the Folder name :



Deleting an empty Folder

> A Folder cannot be deleted while it still contains Runtime Connections, and while any of its Sub Folders contains Runtime Connections.

To Delete a Folder, click on it, then click the *Delete* icon in front of it :



Moving a Folder

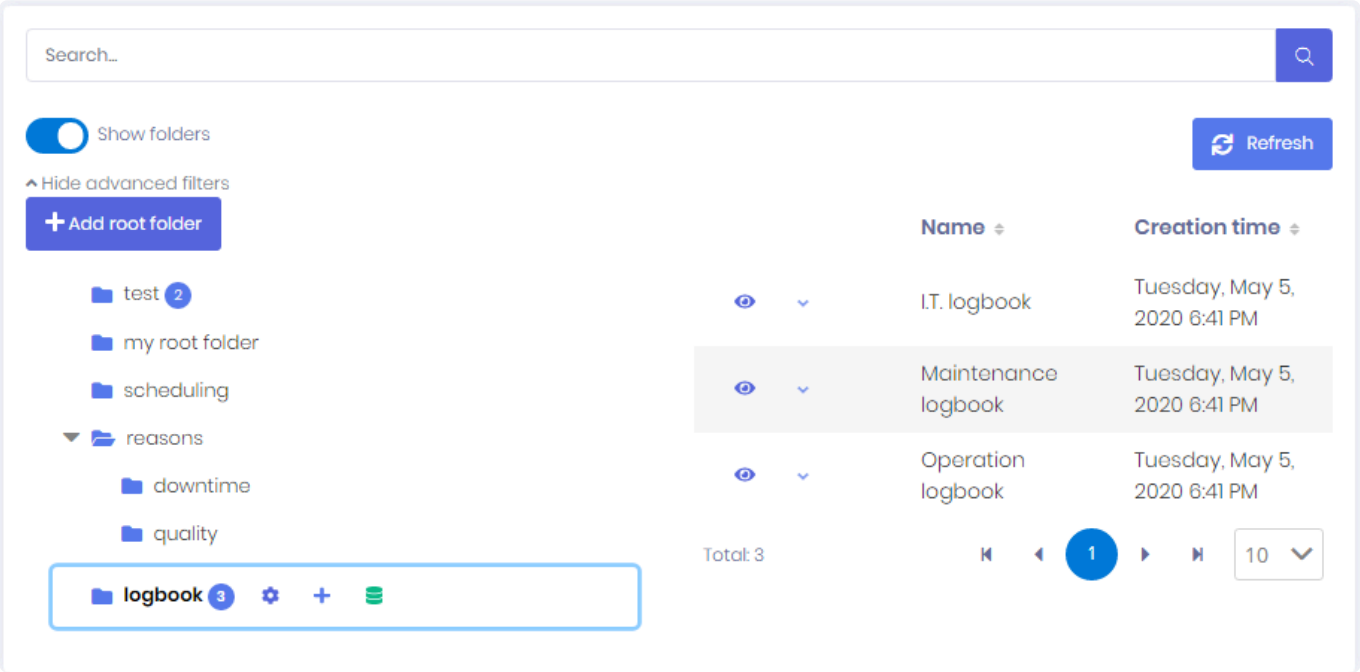
To move a Folder, drag it and drop it onto another Folder.

Last modified: Jun 15, 2020

3.9.2. Browsing the Runtime Connections

The Runtime Connections list is composed of :

- a Search bar at the top
- the list of Folders on the left
- the list of Runtime Connections from the selected Folder on the right



Browsing Folders and listing Runtime Connections

Click on a Folder to select it.
The Runtime Connections from that Folder will be listed on the right.

The list is paginated and the page size can be configured at the bottom.

The list can be sorted by Name or by Creation Time.

Filtering the list of Runtime Connections

The Search bar allows to filter Runtime Connections by name.
To clear the filter, just use an empty search.

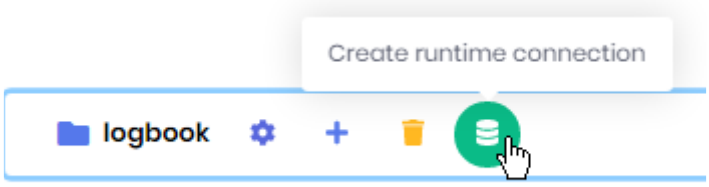
Last modified: May 05, 2020

3.9.3. Creating a Runtime Connection table schema

This chapter describes creating the schema for a Runtime Connection table.
This is similar to creating a template form for the user input, or defining a schema for the hosted table.

Creating a new Runtime Connection

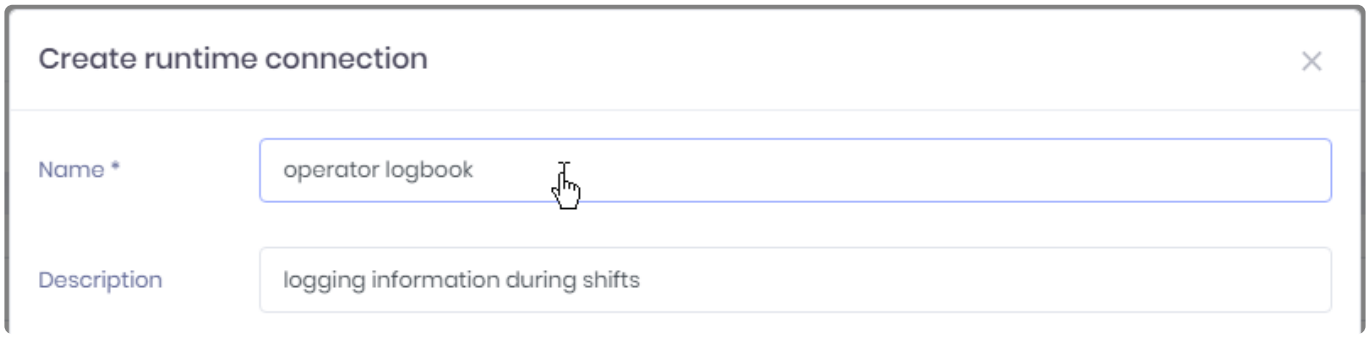
Select a folder and click the *Create Runtime Connection* icon :



The *Create Runtime Connection* dialog appears, and lets you configure the name, schema and options of the table.
See below for the detailed properties

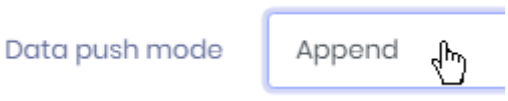
Name and *Description*

The *Name* (compulsory) will be used to identify this Runtime Connection.
The *Description* allows you to optionally add some complementary information.



Behavior when adding data

Data push mode



The *Data push mode* defines how new data will be added to the table :

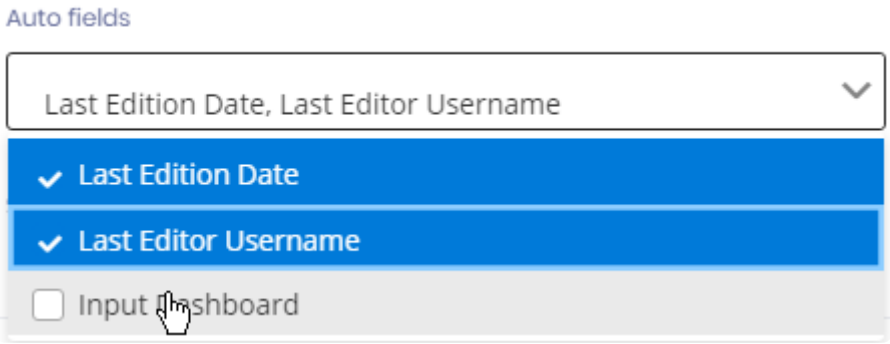
- *Append* : each a new entry will be appended to the existing table. The first (oldest) entries will be removed if they exceed the *Max rows count* parameter.
- *Replace* : each a new entry will completely replace the previous data in the table : all previous rows are removed.

Note : With the *Replace* mode, you can add multiple rows at the same time if you push data using the web API.

Max rows count

This defines a limit to the table size.

Auto fields

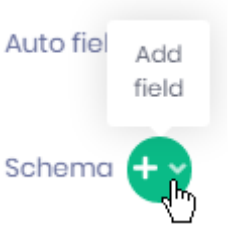


Some fields can optionally be included in the data and filled automatically by Alpana :

- *Last Edition Date* : will always contain the date/time when the row was last created or edited
- *Last Editor Username* : will contain the username of the last User who created or edited the row
- *Input Dashboard* : if the row was created from a Dashboard directly, this will contain the Dashboard name

Adding/Removing fields to the schema

Adding a field



To add a new field to the schema, click the *Add Field* button, and select the desired field type in the dropdown menu (see below for details on each field type) :

- *String*
- *Number*
- *Boolean*
- *Date/Time*
- *List of choices*

Editing a field

After adding a field, a new foldable section is created accordingly.
You can click on the arrow icon to fold/unfold the section :

The section will contain properties detailed below.

Field name

Each field must have a name.

Removing a field

To remove a field, click the trash icon in the section header :

Field configuration properties

Each field can be configured with some properties listed below.
These properties and their meaning depend on the field type.

String

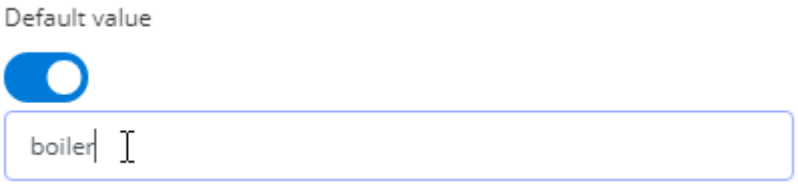
A *String* field will contain a free-form text value.

Allow null

When a field is configured to *Allow null* values, then the user can omit filling-in the field in the form, and the data row will contain a `NULL` value for this field.
If this is configured to be false, then the field is compulsory, and the user must input a value.

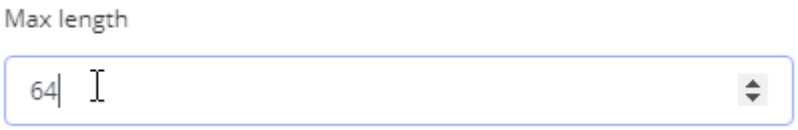
Default value

A *Default value* can pre-filled when opening the input form :



Max length

The limit on the length of the input string can be configured :



Number

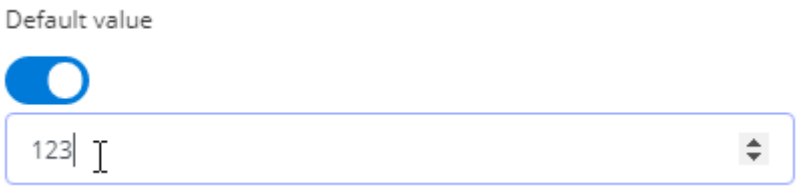
A *Number* field will contain a real or integer number.

Allow null

When a field is configured to *Allow null* values, then the user can omit filling-in the field in the form, and the data row will contain a `NULL` value for this field.
If this is configured to be false, then the field is compulsory, and the user must input a value.

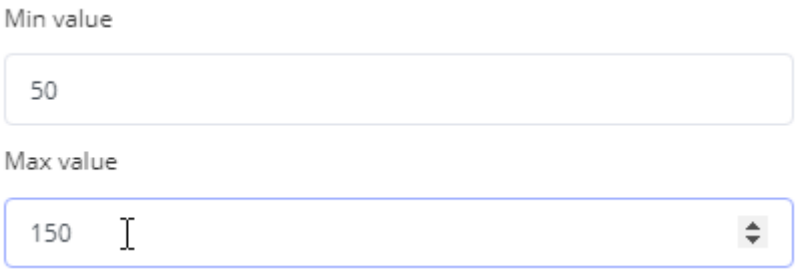
Default value

A *Default value* can pre-filled when opening the input form :



Min value / Max value

A minimum and/or maximum value can be enforced for the number input :



Boolean

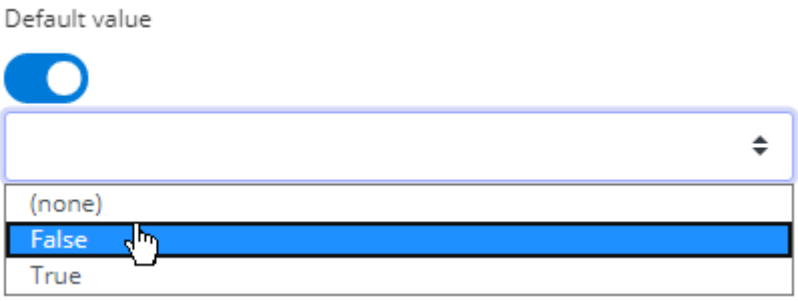
A *Boolean* field will contain a true/false value.

Allow null

When a field is configured to *Allow null* values, then the user can omit filling-in the field in the form, and the data row will contain a `NULL` value for this field.
If this is configured to be false, then the field is compulsory, and the user must input a value.

Default value

A *Default value* can pre-filled when opening the input form :



Date/Time

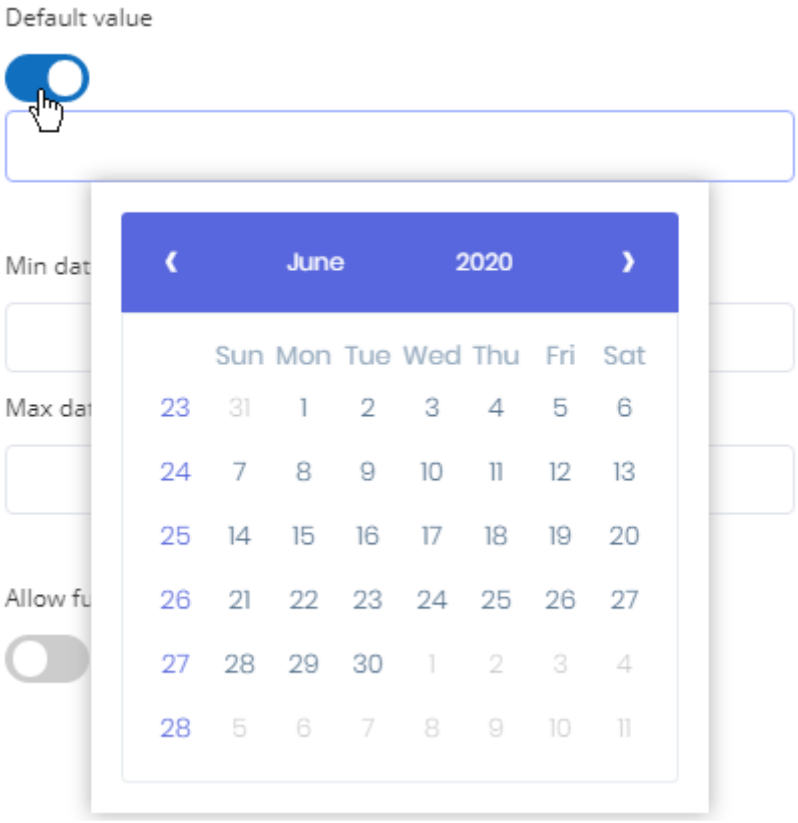
A *Date/Time* field will contain a timestamp composed of a date and a time.

Allow null

When a field is configured to *Allow null* values, then the user can omit filling-in the field in the form, and the data row will contain a `NULL` value for this field.
If this is configured to be false, then the field is compulsory, and the user must input a value.

Default value

A *Default value* can pre-filled when opening the input form, and configured by selecting from a calendar :



Min date / Max date

A minimum and/or maximum date can be enforced for the input.
This means that the user will not be able to input dates before or after these limits :

Allow future

This parameter decides if it is allowed to select dates from the future or not :

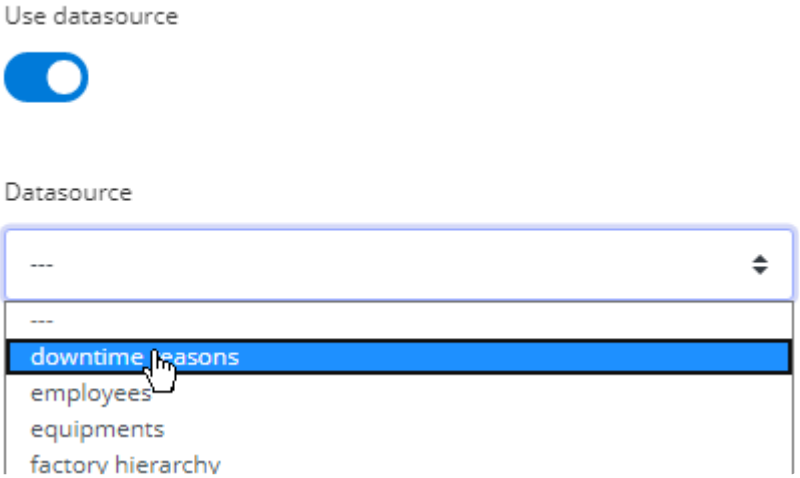
List of choices

A *List of choices* field will contain a list of possible values from which the user can select.
The list of values can be entered in free-form during the configuration, or use a published Data Source to point to data from a database.

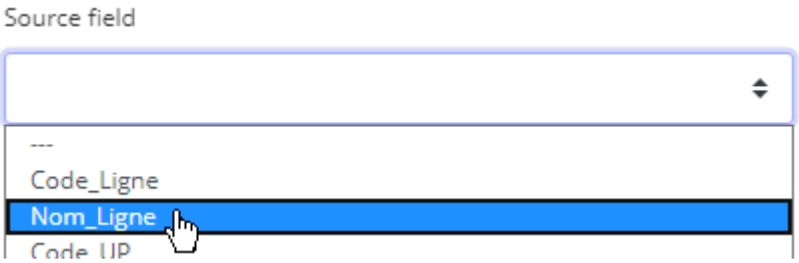
***Use datasource* : getting the list from a Data Source**

When this is true, the list of possible values proposed to the user during input will be collected from the field of a published Data Source.

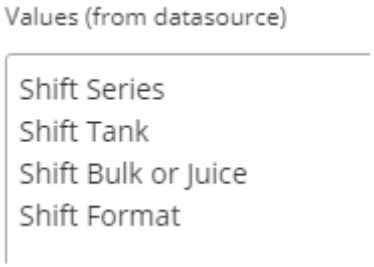
You must select a *Datasource* from the published Data Sources :



Then select a field from that Data Source, which will be use to retrieve the list of proposed values for input :



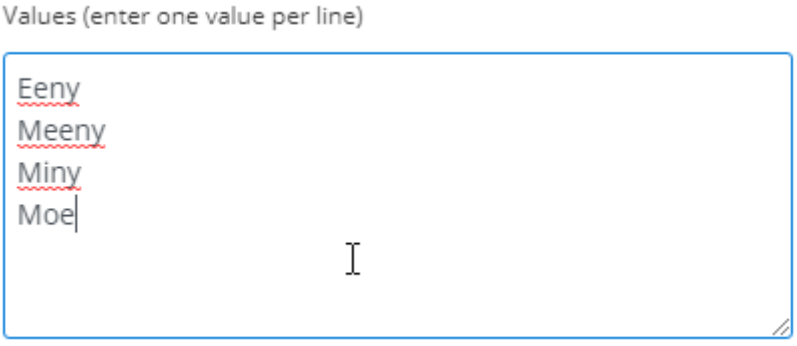
Finally, you can view the current list of *Values* that would be proposed to the user :



Note : this query will be executed every time the input form is opened by a user, so that the list of values will always remain current to the data.

Don't *Use datasource* : using a custom list

When the *Use datasource* option is set to false, the list of choices proposed to the user can be configured using the multi-line text input called *Values* :



Each separate line of text defines a separate choice proposed for entry.

Default value

A *Default value* can pre-filled when opening the input form, and selected from the available choices :

Allow null

When a field is configured to *Allow null* values, then the user can omit filling-in the field in the form, and the data row will contain a `NULL` value for this field.
If this is configured to be false, then the field is compulsory, and the user must input a value.

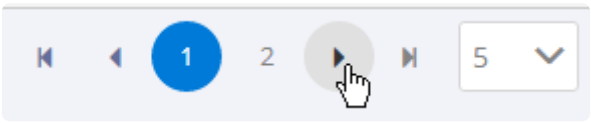
Last modified: Jun 15, 2020

3.9.4. Inputting and Managing data

The *Show Data* button opens the Runtime Connection table in a full web page to let you view current data, and add/update/delete entries :

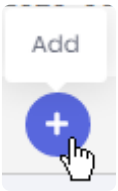
Viewing data

The page displays the content of the Runtime Connection in a paginated table.
Use the controls to change page if necessary.



Adding data

To add a single entry of data, click the *Add* button at the bottom of the table :



This opens the dialog to the full input form, to let you input a new data row.

Note : if the update mode is set to *Replace*, then this view only allows to enter a single data row. To enter multiple rows, you need to use the API.

Updating data

At the end of each row, the *Update* button allows to update the data of the entry :

This opens the dialog to the full input form, to let you update the data row.

Deleting data

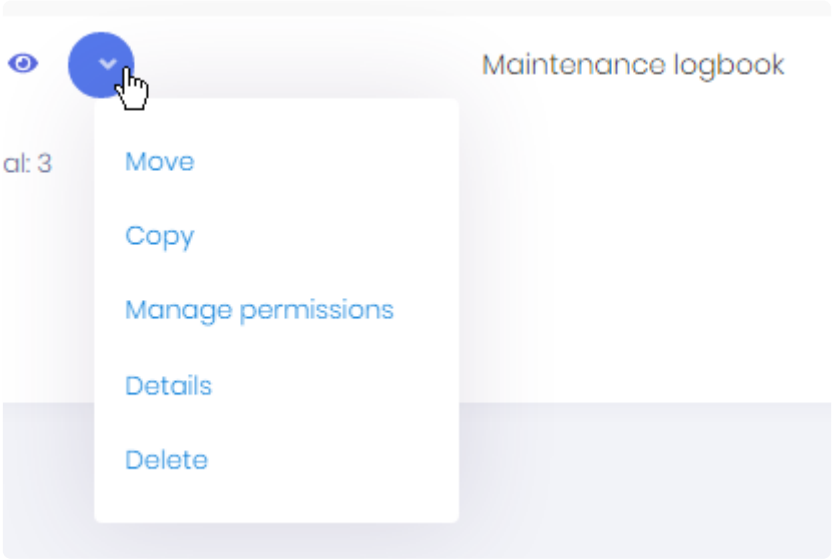
At the end of each row, the *Delete* button allows to delete the data row :

> ! This deletes the data irrevocably.

Last modified: Jun 16, 2020

3.9.5. Runtime Connections Management

Each Runtime Connection has a set of actions available.
To list those actions, open the runtime connection menu next to the Runtime Connection's name :

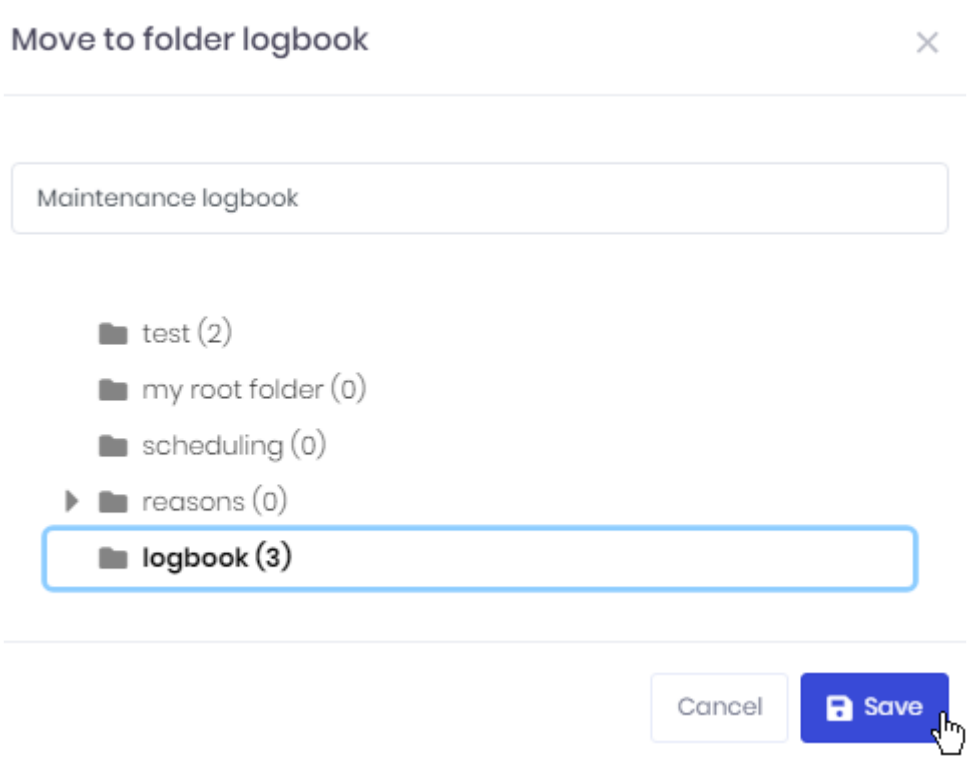


Some of those actions are detailed in the corresponding chapter :

- *Move*
- *Copy*
- *Manage permissions*
- *Details*
- *Delete*

Move

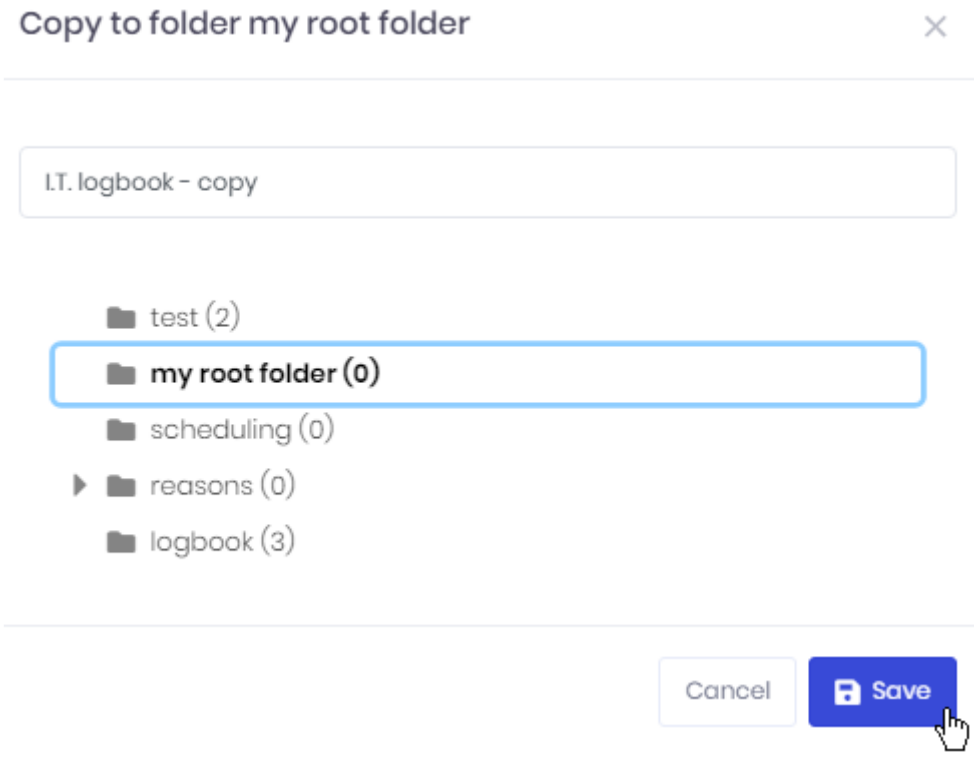
This opens the *Move* dialog which allows to select a destination Folder for moving this Runtime Connection to :



Copy

> *Copy* allows to copy the schema of the Runtime Connection, but data is not copied.

This opens the *Copy* dialog which allows to select a destination Folder for copying the **schema** of this Runtime Connection to :



Manage permissions

This allows to manage access to the Runtime Connection.
(see corresponding chapter)

Details

This allows to edit the schema of the Runtime Connection.
See Creating a Runtime Connection table schema for details.

Delete

This deletes the selected Runtime Connection.

> ! Important : the schema and data are lost.

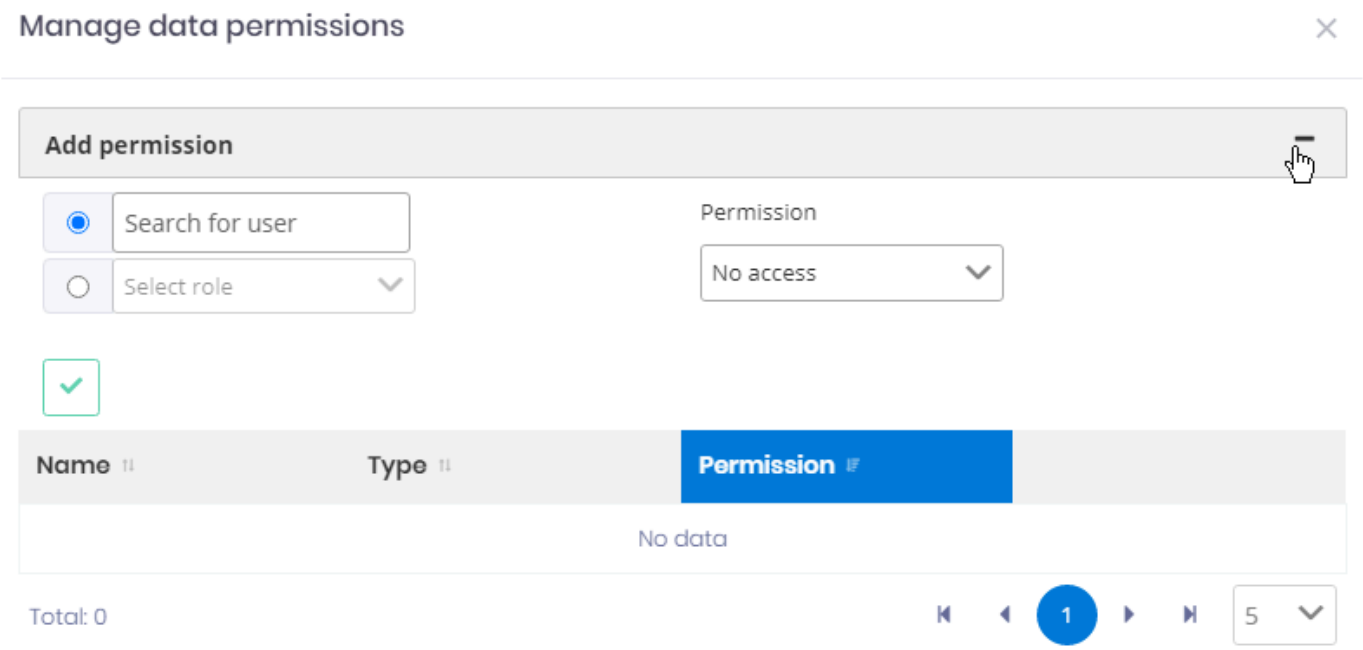
Last modified: Jun 16, 2020

3.9.6. Managing Permissions of Runtime Connections

Manage permissions

> ! When managing permissions, be careful not to remove access to your own user.
> If this happens, you can work around it by : creating a new Role with all permissions, assigning this as the only Role to a User, and letting that User remove the Permissions restriction.

From the Runtime Connection's action menu, you can manage its Permissions.
Select *Manage permissions* to open the corresponding dialog :

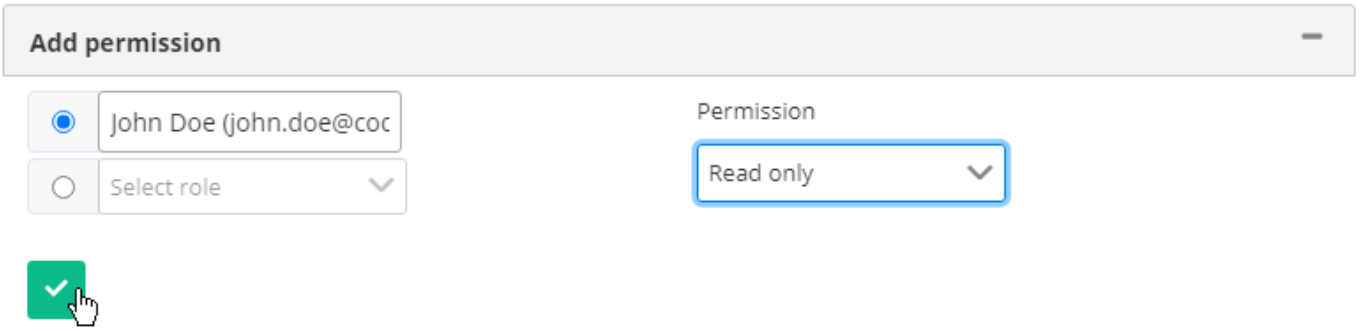


The existing Permissions are listed at the bottom of the dialog.

Adding a permission

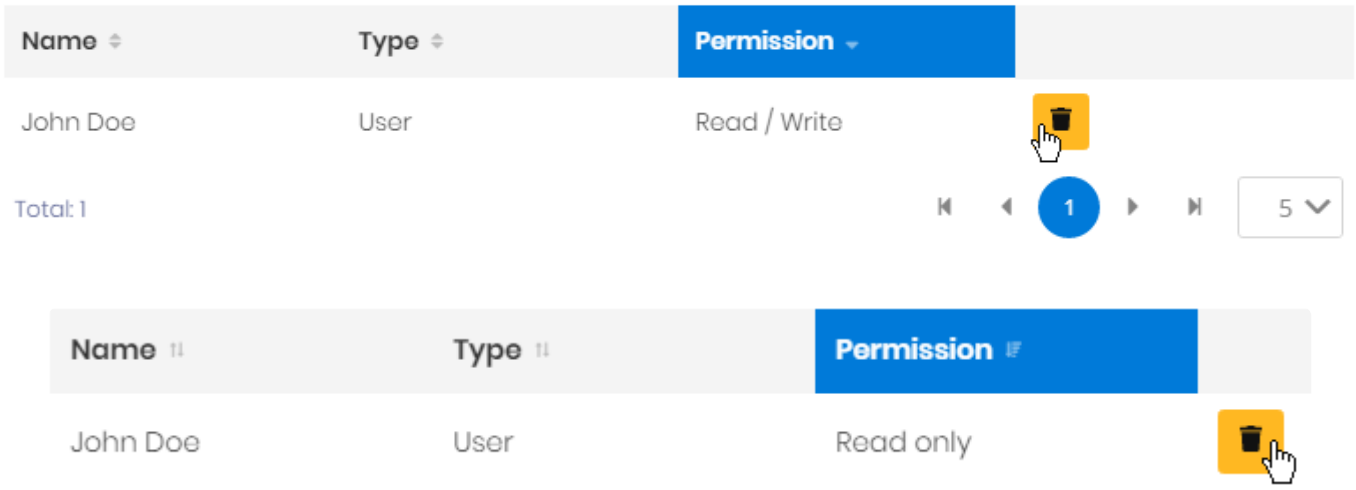
If you want to override the default permissions, fill-in the *Add permission* form :

1. select a User or Role
2. select the desired Permission
3. click the validation button :



Removing Permissions

If you want to remove a Permission, click the *Remove* (trash) icon in front of it in the list :



Available permissions

Possible permissions are :

- *No Access* : user has no permission to the Runtime Connection
- *Read Only* : user has only the permission to read data from the Runtime Connection
- *Read/Insert* : user has the permission to read data and insert data to the Runtime Connection
- *Full* : user has all permissions to the Runtime Connection : Read, Insert, Update, Delete

Last modified: Jun 19, 2020

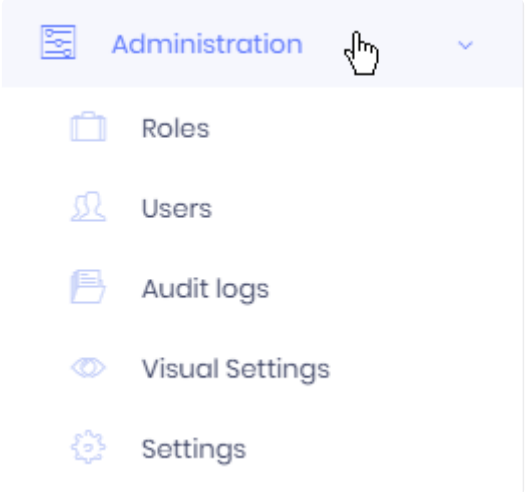
3.10. Administration

Administrators can use the web interface from Alpana Server to manage their users and configuration.

Note that the Host and each Tenant both contain useful administration activities :

- on the Host, you can manage the Tenants themselves and all system-wide settings like the license, the e-mail settings, etc
- on each Tenant, you can manage the Tenant's objects and properties like users, theme, etc

To access Administration options, use the *Administration* menu and its sub menus :



Last modified: Oct 28, 2019

3.10.1. License Management

License is managed at the Host level for all Tenants.
Login to the Host of your Alpana Server and navigate to *Administration* > *Settings*, tab *License Management*.

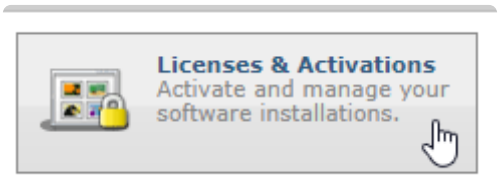
See the below chapters to manage your license.

Last modified: Dec 26, 2019

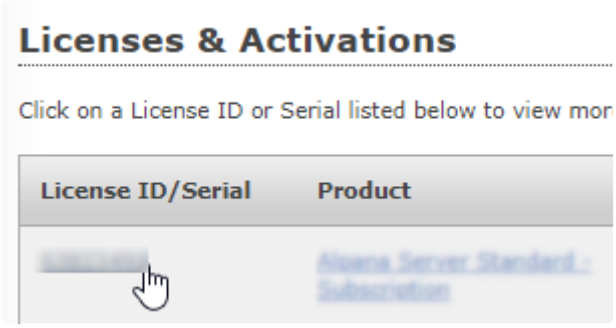
3.10.1.1. License Activation

Browsing your license portal

Login to the web License Portal.
Navigate to *Licenses & Activations* :

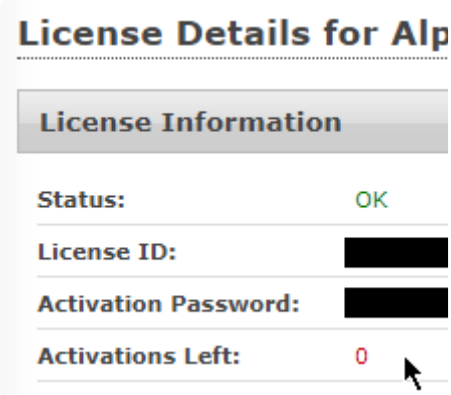


Under *Licenses & Activations*, click the *License ID* you want to activate :



Important note

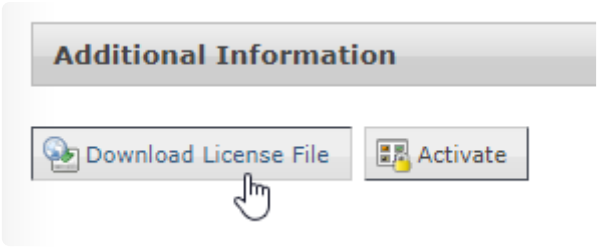
> ! Before starting the license activation, please make sure that you have enough Activations left !
> If your *Activations Left* is 0, then your license is probably used elsewhere. Please contact your Software Distributor for support.



> ! To continue, the *Activate* button should be clickable.

Getting the License File

At the bottom, click the button *Download License File* :

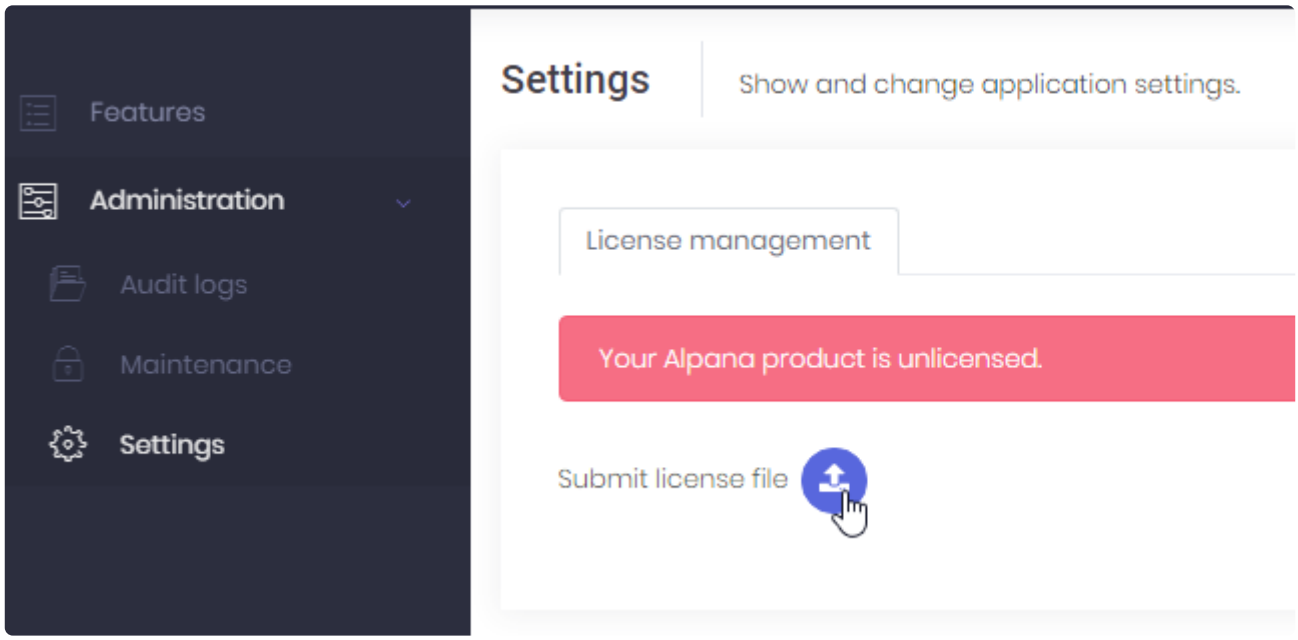


⇒ A file `AlpanaServer.lic` is downloaded.
Move this file to your Alpana Server installation.

> You can leave this web page open, it will be used below to complete the Activation.

Activating the License File

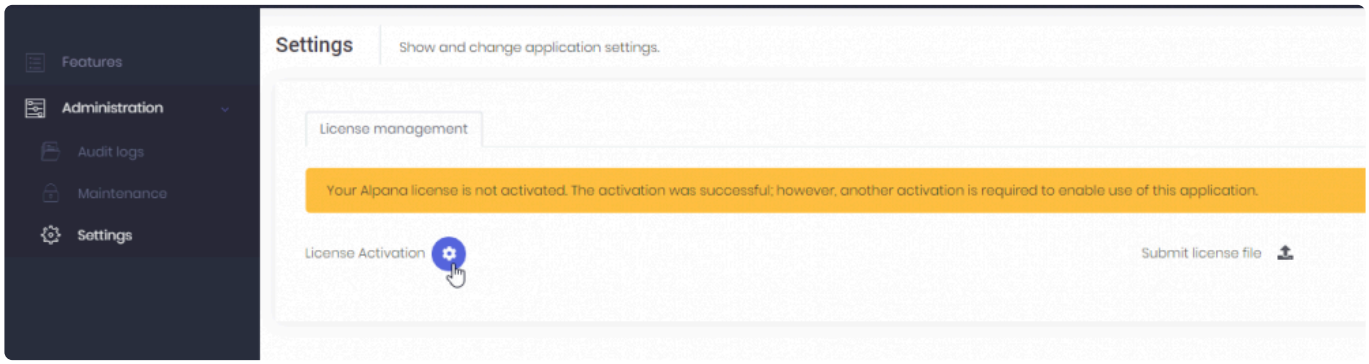
On the Host of your Alpana Server, navigate to *Administration* > *Settings*, tab *License Management*.
Click *Submit License File* and upload the `AlpanaServer.lic` file :



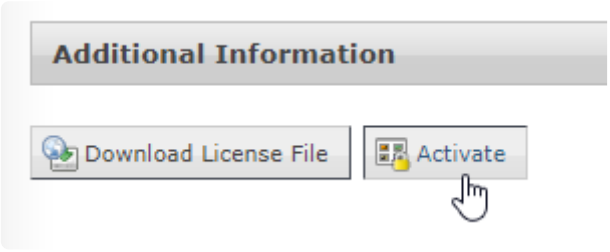
⇒ The first step of license activation is complete, but your license is not yet activated.

Activating the License

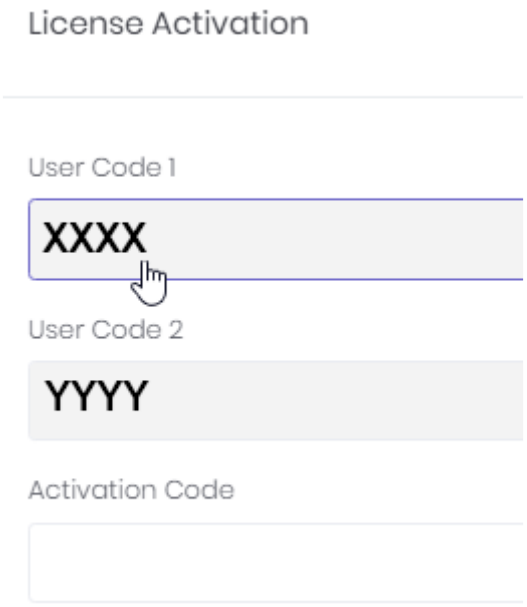
On the Host of your Alpana Server, click *License Activation* and leave the pop-up window open :



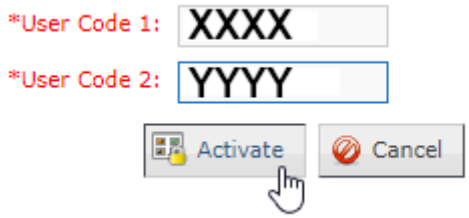
On the web License Portal, click *Activate* and leave the pop-up window open :



On the Host of your Alpana Server, copy the *User Code 1* and *User Code2* :

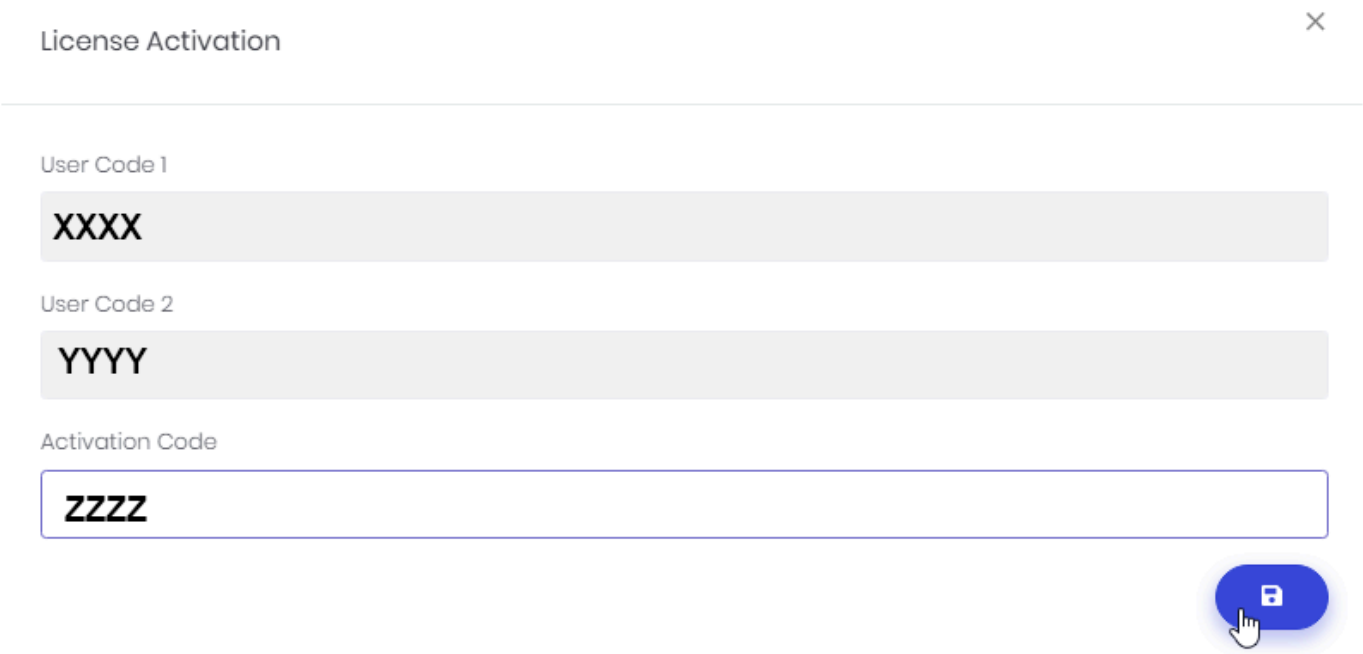


…and paste them in the correct fields in the web License Portal, and click *Activate* :



⇒ As a result, the web License Portal displays an *Activation Code* :

Copy the *Activation Code* and in the Host of Alpana Server, paste it and click to save :



⇒ The License should be activated.

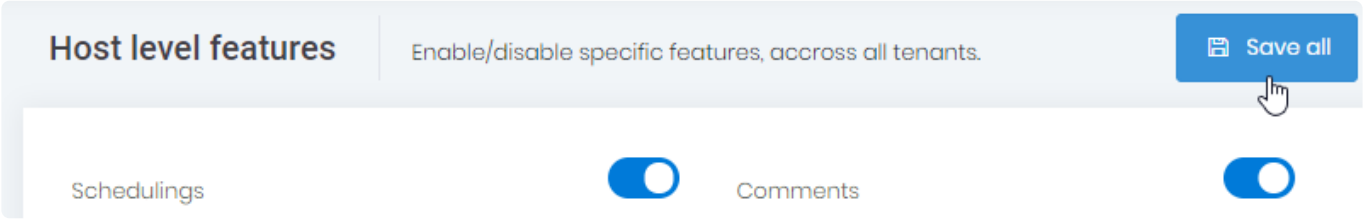
If the License Activation was successful, you may delete the `AlpanaServer.lic` file. It cannot be used later, and keeping it would only create confusion during a later activation.

Last modified: Dec 26, 2019

3.10.1.2. Host-level Features

Depending on your license type, some features may be activated at the Host level.
Login to the Host and browse to *Features* :

Check the required Features and click *Save* :



> ! Important : After changing the allocated features, you must restart the application (from IIS Manager).

Then, in order for Tenants to benefit from the features, you must allocate them.

Last modified: Dec 20, 2019

3.10.1.3. License Deactivation

Why Deactivate

You may want to return the license from a Host in order to decommission the Host and use the license on another Host.

For example, before moving the installation to a new hardware, **you must de-activate**.

Note of warning

Deactivate is not the same as *Clear license*.
This page is documenting Deactivation only.

! **Do not** use *Clear license data* unless instructed by Support.

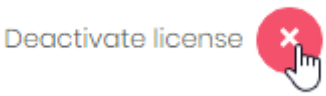
How to Deactivate a License

! Warning : This will remove the license from the current Alpana Server installation.
Until a new license is activated, you users won't be able to login or view dashboards.

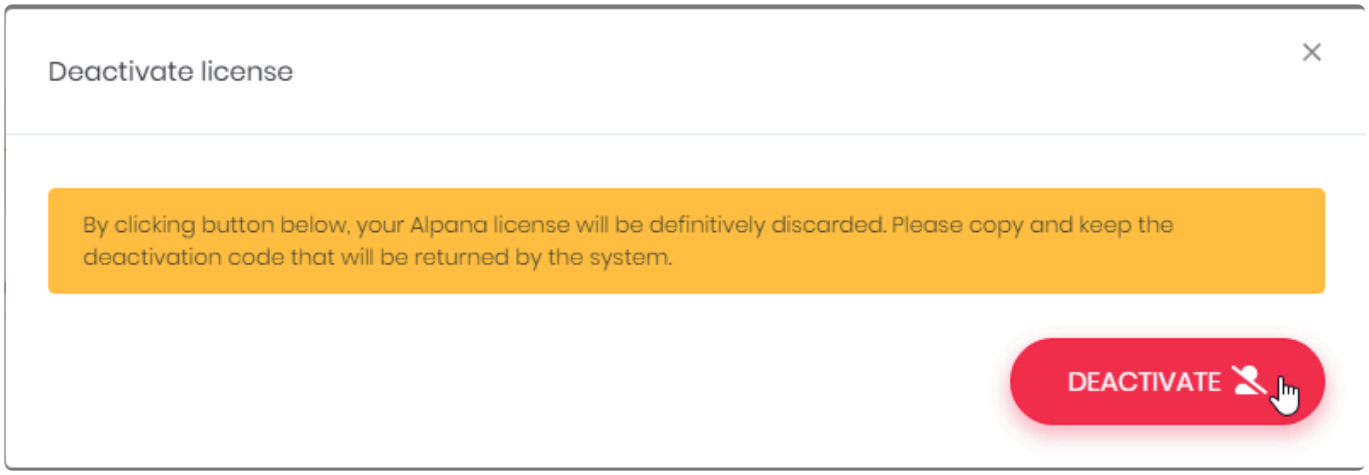
Note : You may at a later time activate again a license on this installation if needed.

On the Host of your Alpana Server, navigate to *Administration* > *Settings*, tab *License Management*.

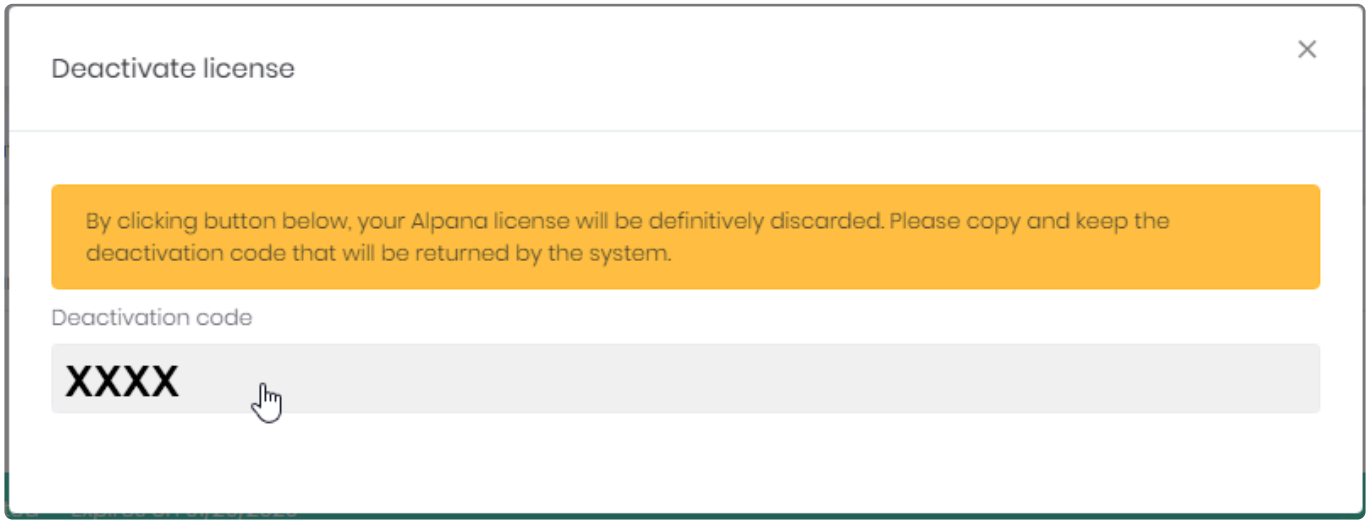
Click *Deactivate License* :



Read the warning and confirm the deactivation :



Please copy and keep the *Deactivation code*



> ! If you fail to keep your *Deactivation code*, you may not be able to re-use your license at a later time.

Contact Support

Contact Support with your *License ID* and *Deactivation Code* to allow to free your license.

Last modified: Jan 28, 2020

3.10.2. Tenant Management

Note : For a better understanding of Tenants, you can see the Multitenancy chapter.

Tenants are managed at the Host level (that's the main purpose of the Host).
By default, only one Tenant exists and is called *Default*.
To manage Tenants, see the below chapters.

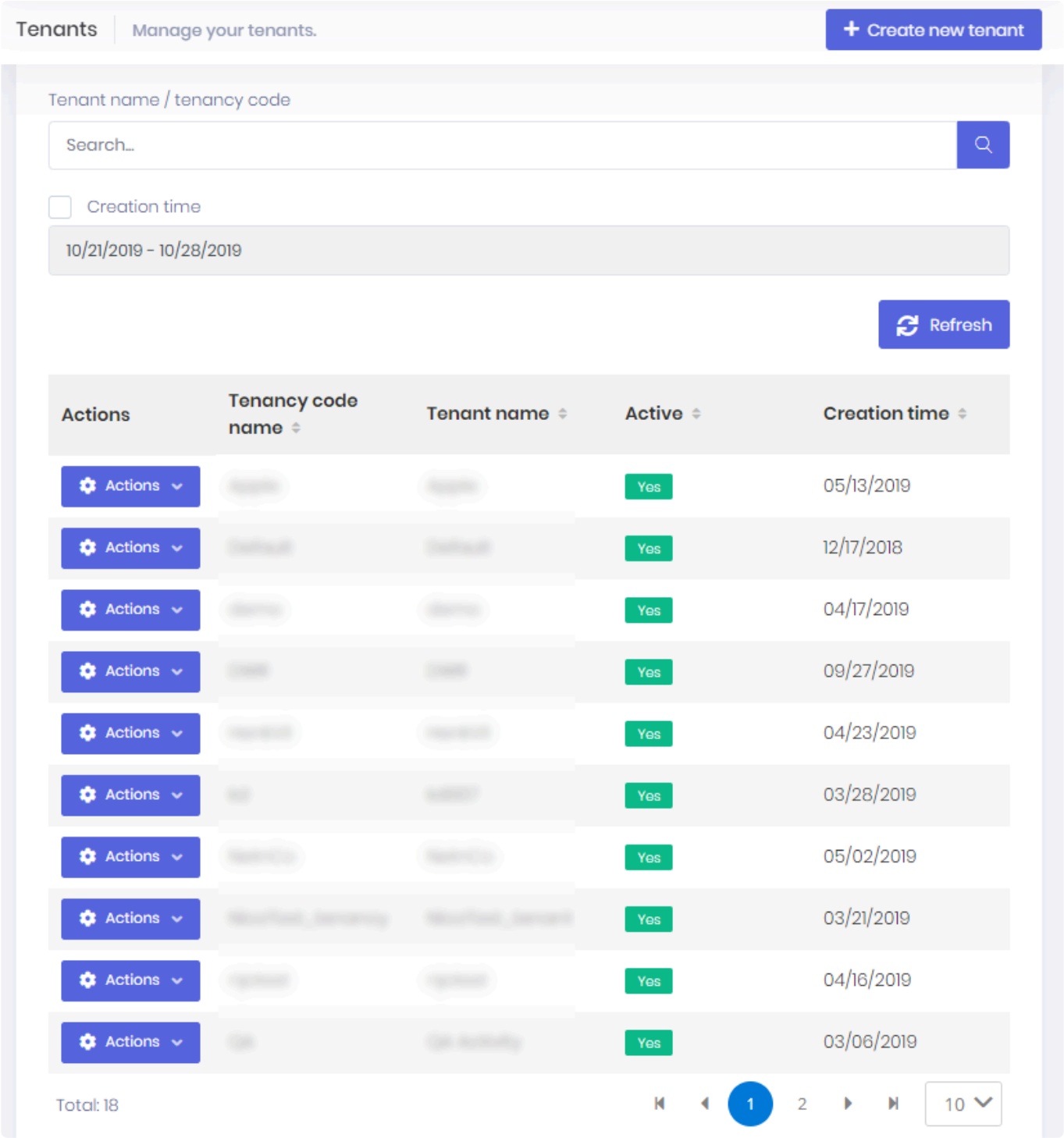
To access Tenants management, use the *Tenants* menu of the Host :

Last modified: Oct 28, 2019

3.10.2.1. Browsing the Tenants

Tenants list

The Tenants list shows all Tenants :



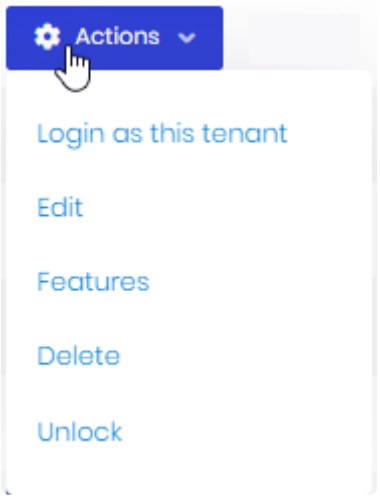
Tenants can be searched by name and/or creation time by using the search bar at the top.

The paginated list can be sorted by :

- *Tenancy code name*
- *Tenant name*
- *Active* status
- *Creation time*

Actions menu

Each Tenant has an *Actions* menu that allows to configure it :

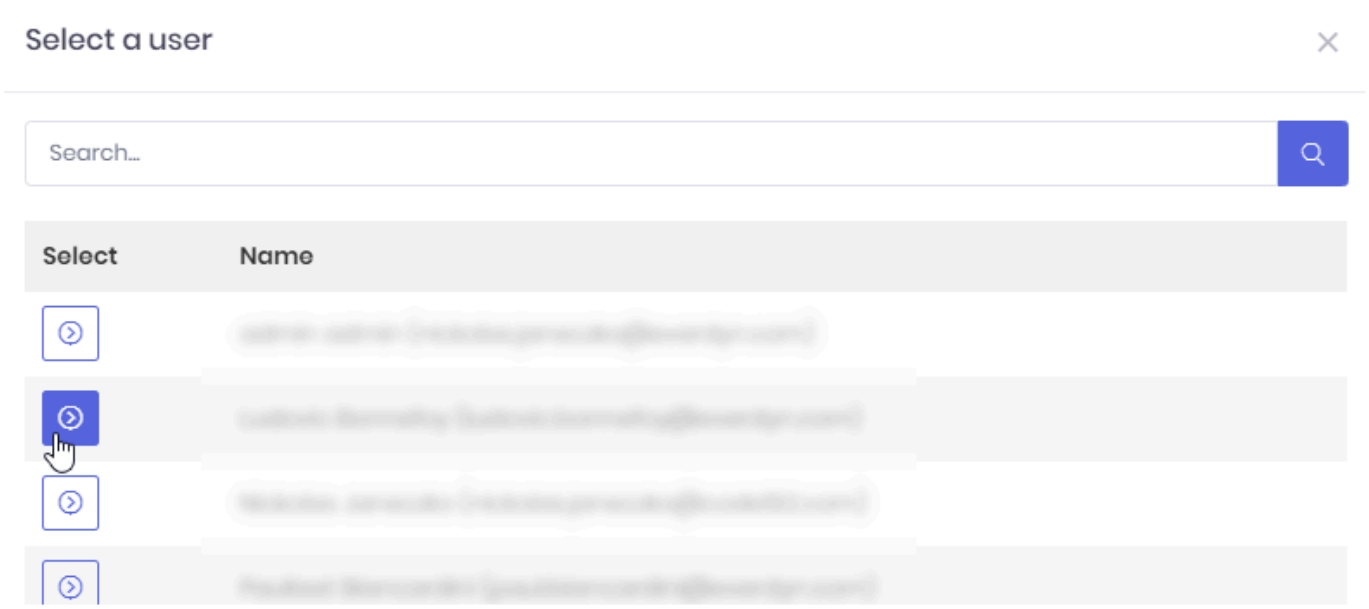


Some of those actions are detailed in the corresponding chapter :

- *Login as this tenant*
- *Edit*
- *Features*
- *Delete*
- *Unlock*

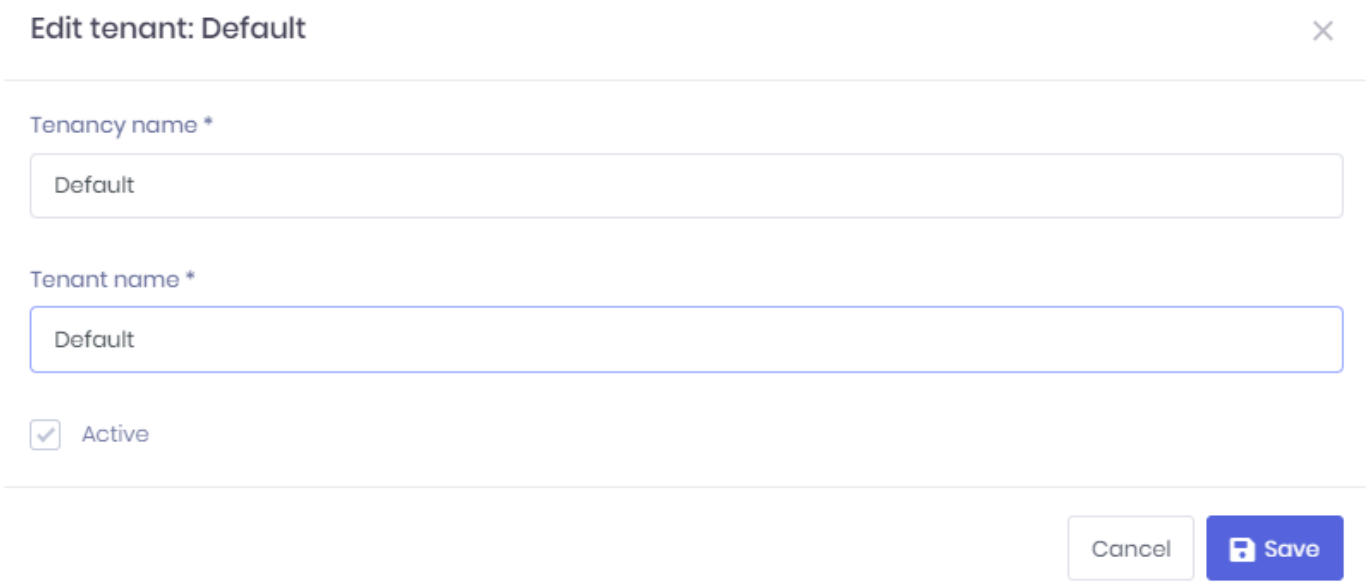
Login as this tenant

This allows to impersonate a User from this Tenant and log in as them.
The *Select a user* dialog opens and lets you select a User from that Tenant :



Edit

This allows to modify the Tenants properties that were selected when creating it.
The *Edit tenant* dialog opens and lets you rename the Tenant and set it *Active* or not :



Features

(see corresponding chapter)

Unlock

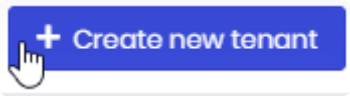
When the `admin` user of the Tenant has been locked out (too many incorrect login attempts), and admin on the Host can unlock it through this action.

Last modified: Oct 28, 2019

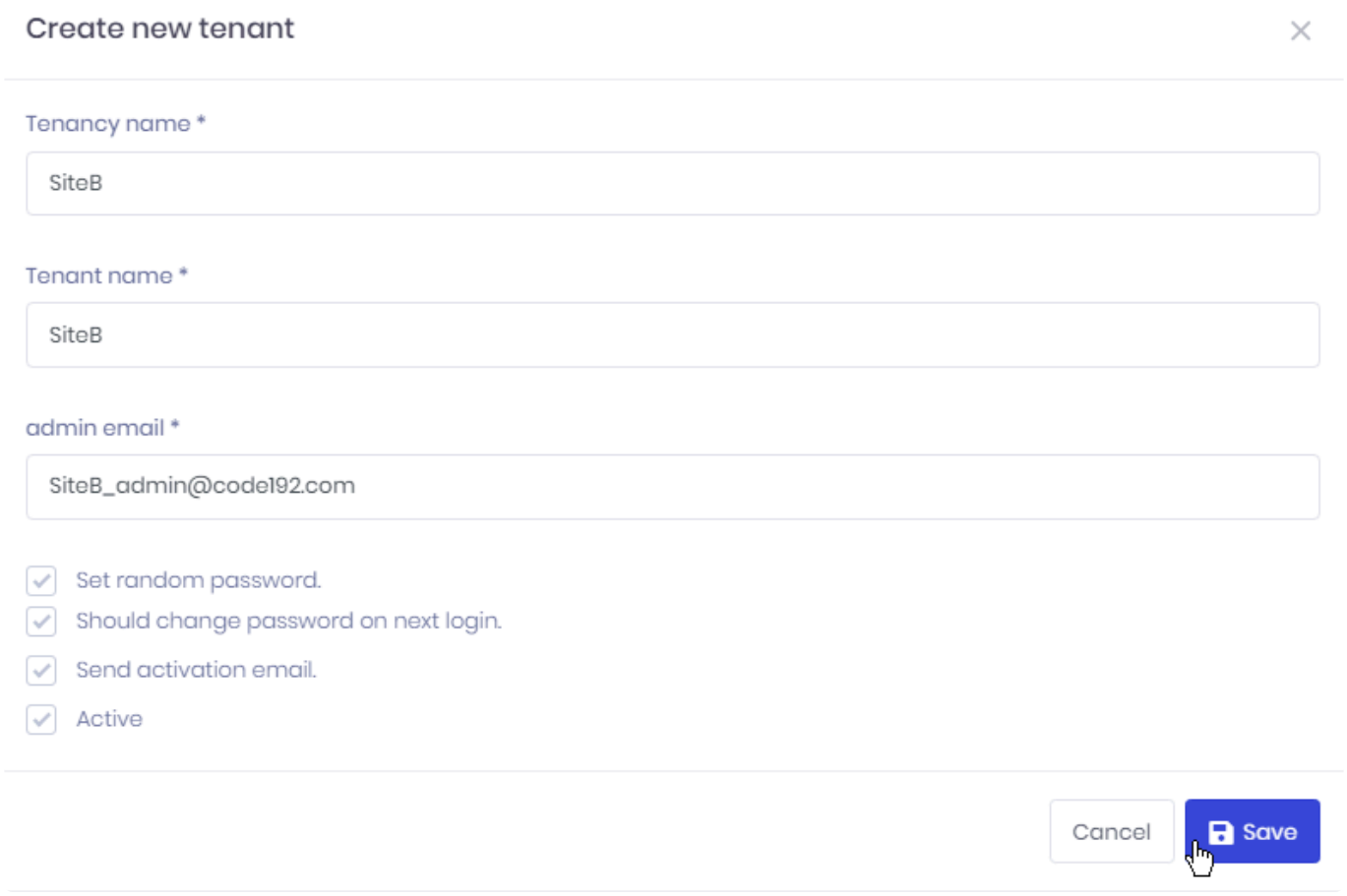
3.10.2.2. Creating a Tenant

By default, 1 Tenant is already created after installing. It is called `Default`.

To create a Tenant, click *Create new tenant* :



This opens the *Create new tenant* dialog_ :



Fill-in the required information :

- *Tenancy name* / *Tenant name* : the name of the Tenant.
- *Set random password* : if you don't have an e-mail server configured, un-check this option and specify a password explicitly
- *Should change password on next login* : force the admin user to change password when logging in next time
- *Send activation e-mail* : send an activation e-mail with the random password and a link to login
- *Active* : only an active user can login

! After creating a new Tenant, the default Roles (Admin, User, …) may not have all required permissions checked. Please make sure that you check all the permissions of the static Roles.

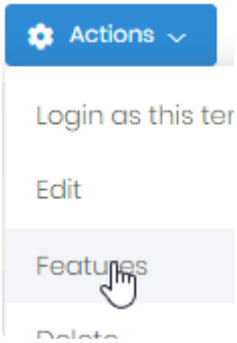
Last modified: Jan 08, 2020

3.10.2.3. Tenant Features

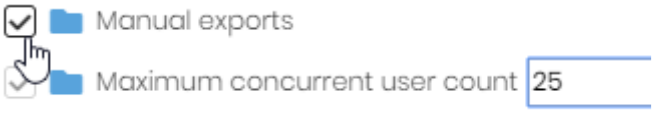
Depending on your license type, some features may be activated at the Host level.
In addition, your license contains some limited resources (number of users, dashboards, etc)

Then, in order to benefit from those features or resources, they must be allocated to each Tenant as desired :

On the Host, Navigate to *Tenants* management.
Then on the desired Tenant, select *Action* > *Features* :



In the *Features* window, check which Features to allocate, including resources, and click *Save* :



Last modified: Jun 18, 2019

3.10.3. Users, Roles and Permissions

Access to Alpana Server is secured through user accounts.
Users login to Alpana Server and can access content according to their permissions.
Administrators can manage users permissions by using Roles.

This section details the management of Users, Roles and Permissions

Last modified: Oct 29, 2019

3.10.3.1. Permissions descriptions

User and Roles can be assigned Permissions to Alpana Server.
Below is the list of existing permissions and their description.

Host permissions

On the Host, the possible Permissions are as follows :

- *License management* : User can manage the license (activate, deactivate, enable/disable features)
- *Pages* : the web Pages that the User can access on the Host :
  - *Administration* : User can access the Host Administration pages
    - *Audit Logs* : User can read the Audit logs
    - *Maintenance* : User can access the Maintenance page
    - *Roles* : User can access the Host's Roles page
      - *Creating a new role* : User can create a new Role
      - *Deleting role* : User can delete any Role
      - *Editing role* : User can edit any Role
    - *Settings* : User can access the Host Settings page
    - *Users* : User can access the Users page to view existing users of the Host
      - *Changing permissions* : User can change Permissions of other Host users
      - *Creating a new user* : User can create other users
      - *Deleting user* : User can delete users
      - *Editing user* : User can edit other users' properties
      - *Login for users* : User can impersonate other users of this Tenant to login as them
      - *Unlock* : User can unlock users that have been locked following too many incorrect login attempts
    - *Visual settings* : User can access the Visual Settings page
  - *Tenants* : User can access the Tenants page
    - *Changing features* : User can assign/remove features to any Tenants
    - *Creating new tenant* : User can create Tenants
    - *Deleting tenant* : User can delete Tenants
    - *Editing tenant* : User can edit Tenants' properties
    - *Login for tenants* : User can impersonate Tenants users to login on any Tenant

Tenant permissions

On each Tenant, the possible Permissions are as follows :

- *API* :
  - *Manage API* : (permission not used)
  - *Use API* : User can access the API snippets page
- *Pages* : the web Pages that the User can access on the Tenant :

    - *Administration* : User can access the Tenant Administration pages
        - *Audit Logs* : User can read the Audit logs
        - *Roles* : User can access the Tenant's Roles page
            - *Creating a new role* : User can create a new Role
            - *Deleting role* : User can delete any Role
            - *Editing role* : User can edit any Role
        - *Settings* : User can access the Tenant Settings page
        - *Users* : User can access the Users page to view existing users of the Tenant
            - *Changing permissions* : User can change Permissions of other users of the Tenant
            - *Creating a new user* : User can create other users
            - *Deleting user* : User can delete users
            - *Editing user* : User can edit other users' properties
            - *Login for users* : User can impersonate other users of this Tenant to login as them
            - *Unlock* : User can unlock users that have been locked following too many incorrect login attempts
        - *Visual settings* : User can access the Visual Settings page
    - *Categories* : User can view Categories
        - *Manage Categories* : User can create/edit/delete Categories
    - *Dashboard Viewer* : User can view Dashboards
        - *Comments* : User can access the Comments panel of Dashboards
        - *Embedding* : User can access the Get Link / Embed page of Dashboards
        - *Export Dashboards* : User can manually export Dashboards to files
    - *Dashboards* : User can access the list of Dashboards
        - *Download dashboards* : User can download published Dashboards
        - *Manage dashboards* : User can move/copy/delete Dashboards
    - *Datasources* : User can access the list of Data Sources
        - *Download datasources* : User can download published Data Sources
        - *Manage datasources* : User can create/delete Data Sources
    - *Homepages* : User can view their Homepages
        - *Manage homepages* : User can create/edit/delete their Homepages
    - *Manage entities permissions* : User can manage individual permissions of Dashboards, Widgets, Data Sources
    - Manage entities versions history_ : User can manage the version history of Dashboards, Widgets, Data Sources
    - *Schedulings* : User can view the Schedules page
        - *Manage schedulings* : User can create/edit/run/delete Schedules
    - *Widgets* : User can access the list of Widgets
        - *Download Widgets* User can download published Widgets
        - *Manage widgets* : User can manage widgets
- *Schedulings* :
    - *Subscribe to schedulings* : User can be added as target to a Schedule

Last modified: Oct 29, 2019

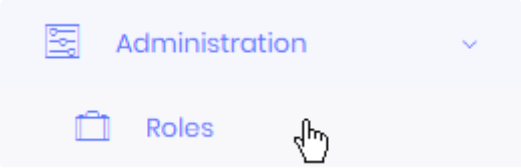
3.10.3.2. Roles

Roles allow to easily manage Users and their permissions :

- you can assign multiple Roles to a User
- adding/removing Permissions is just adding/removing the Role to the User

Note : Additionally, Roles can be used as mailing lists for Schedules.

To manage Roles, go to the *Administration* > *Roles* page :



Assigning Roles

To assign Roles to Users, you must edit the Users, see the corresponding chapter.

Some roles are assigned by default thanks to the *Default* property (see below).

Default Roles

Some Roles exist by default in Alpana Server.
They cannot be deleted.

Host default Roles

When installing Alpana Server, some default Roles exist on the Host :

Admin

> ! Be very careful when removing permissions from the *Admin* Role.
> Make sure that you still have at least some users with enough permissions to restore access to features.
> If you want to create administrators with restricted access, you can create dedicated Roles.

The *Admin* Role is added by default to new users of the Host.
By default, it gives full permissions.

Tenants default Roles

When creating a Tenant, some default Roles are created on the Tenant :

Admin

> ! Be very careful when removing permissions from the *Admin* Role.
> Make sure that you still have at least some users with enough permissions to restore access to features.
> If you want to create administrators with restricted access, you can create dedicated Roles.

Users with the *Admin* Role have **read/write access** to all objects by default (Dashboards, Data Sources, Widgets).
These permissions can then be overridden at the object level.

User

The *User* Role is added by default to new users of the Host.
Users with the *User* Role have **read access** to all objects by default (Dashboards, Data Sources, Widgets).
These permissions can then be overridden at the object level.

Guest

Users with the *Guest* Role have **no access** to any objects by default (Dashboards, Data Sources, Widgets).
These permissions can then be overridden at the object level.

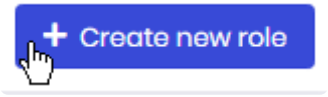
Example : If UserX is Guest and you add permission “Dashboard Viewer” to this UserX, then they will be able to view every dashboard where you explicitly add permission “Read” for this UserX, and no other Dashboard.

Reserved Connection

Users with the *Reserved Connection* Role are **guaranteed that they can login**, because no other concurrent user can take their token for logging in.
This means that they automatically and permanently count as users inside the *Maximum concurrent access count* license property of the Tenant.

Creating a Role

To create a new Role, click *Create new role* :

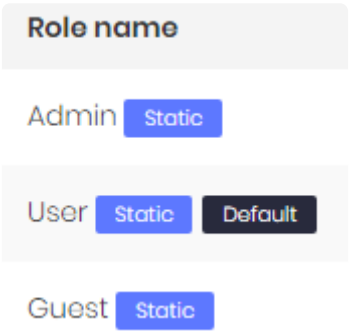


Then fill-in the Role form (see below *Edit Role*).

Viewing Roles

Role attributes

Some Role attributes are displayed as a tag in the list :

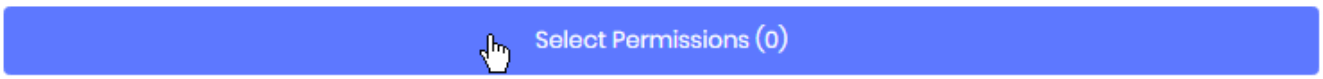


- *Static* : this Role exists by default and cannot be deleted (see above)
- *Default* : this Role is assigned to new Users by default

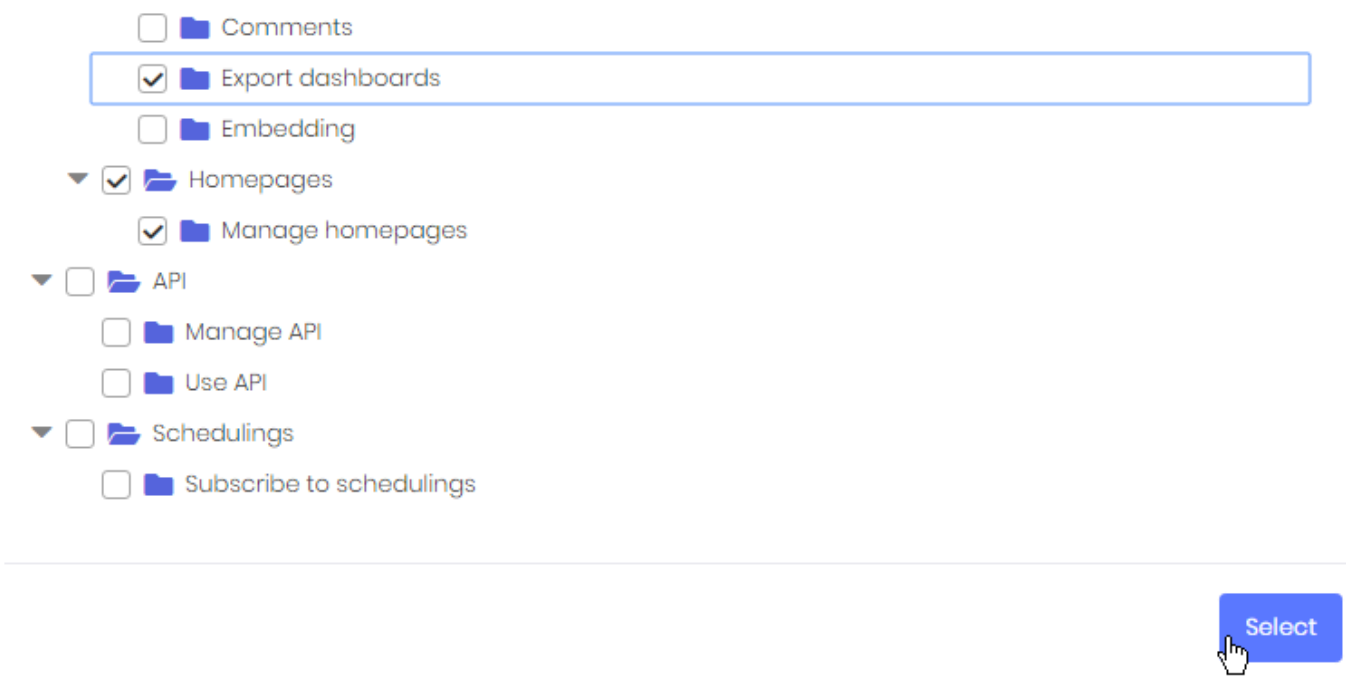
Filtering the list

You can filter Roles by which permissions are active :

Click *Select permissions* in the top bar :

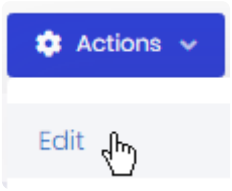


Then select the required Permissions and click *Select* :



Editing a Role

To Edit a Role, click the *Actions* > *Edit* button :

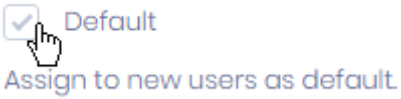


Role name

Use a descriptive role name.

Default

Check this option if you want to Role to be assigned to all new Users by default :

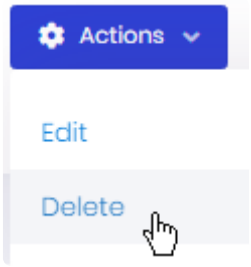


Permissions

Check the desired permissions, as described in the corresponding chapter.

Deleting a role

To delete a Role, click *Action* > *Delete* :



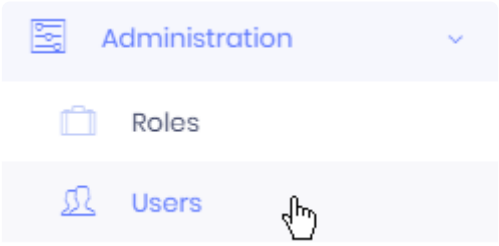
Note : The Roles that exist by default in Alpana Server cannot be deleted.

Last modified: Aug 26, 2020

3.10.3.3. Users

A User represents an access to Alpana Server, with its associated Roles and Permissions.
Users can manage some of their own properties by editing their Profile.

To manage Users, go to the *Administration* > *Users* page :



> Note : Users at the Host level are different from Users at the Tenant's level.
> In the same way, Users are managed separately on each Tenant (they could have the same name but are still managed as different Users)

Default Users

Some Users exist by default in Alpana Server.
They cannot be deleted.

Host default User

When installing Alpana Server, an administrator user is created with some default credentials :
username : `admin`

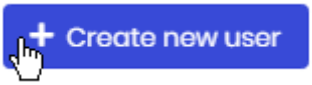
Tenants default User

When creating a Tenant, an administrator user is created with some default credentials :
username : `admin`

Creating a User

Manual User creation

To create a new User, click *Create new user* :



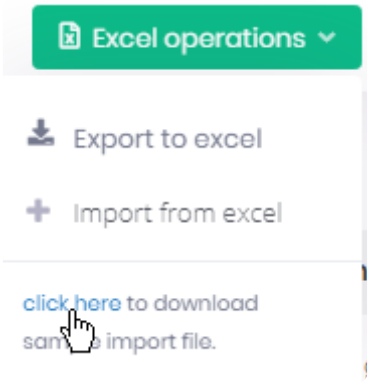
Then fill-in the User form (see below *Edit User*).

Mass import of Users

Instead of manually creating Users one by one, you may wish to import many Users at once.
Alpana Server allows to import a list of Users from an Excel file.

Download a sample Excel file

To prepare your User import, you can download a sample Excel file.
Click *Excel operations* > *click here to download sample import file* :



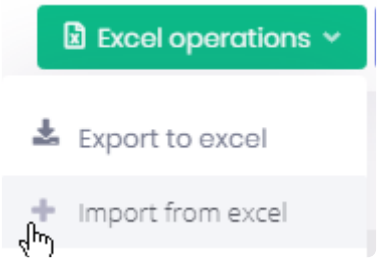
This downloads an Excel file called `ImportUsersSampleFile.xlsx` inside your web browser.
This file contains the correct field names and some sample import.

Prepare your Excel file

Delete the sample import and add your own users' informations.
To assign multiple Roles, write the desired list of Roles separated by commas `,` .

Import Users from Excel

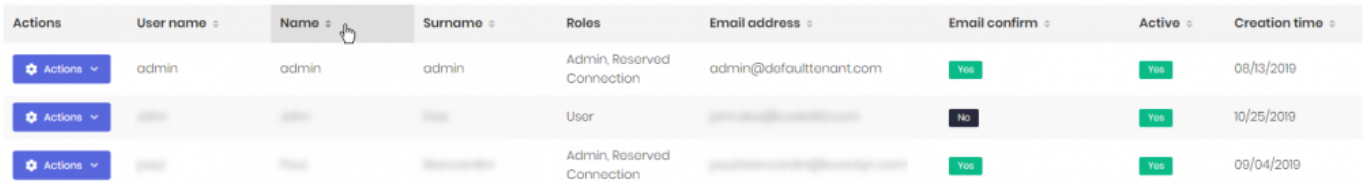
Click *Excel operations* > *Import from excel* :



Viewing Users

List

The list of Users are presented in a paginated table displaying their main properties, which all can be used to sort the list :



Filtering the list

The top bar also contains filters and advanced filters :

Searching names

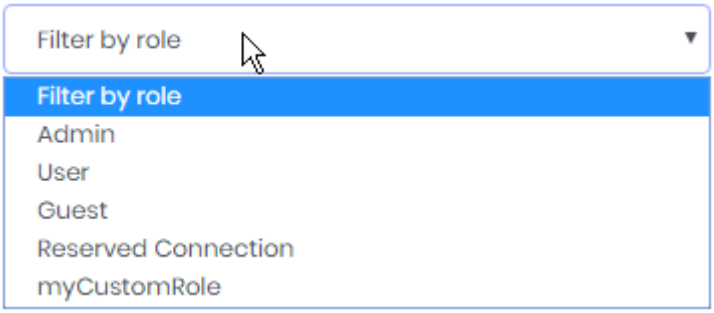
Using the *Search…* bar, you can search users by username, name or e-mail :

Locked users

You can search for Users who are currently *Locked*.
This can let you as an admin to unlock them quickly.

Filter Roles

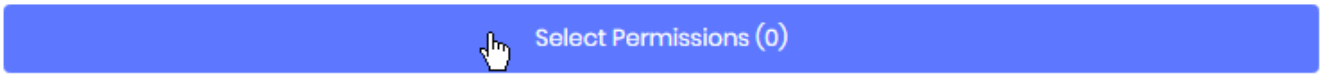
You can filter the list by Roles :



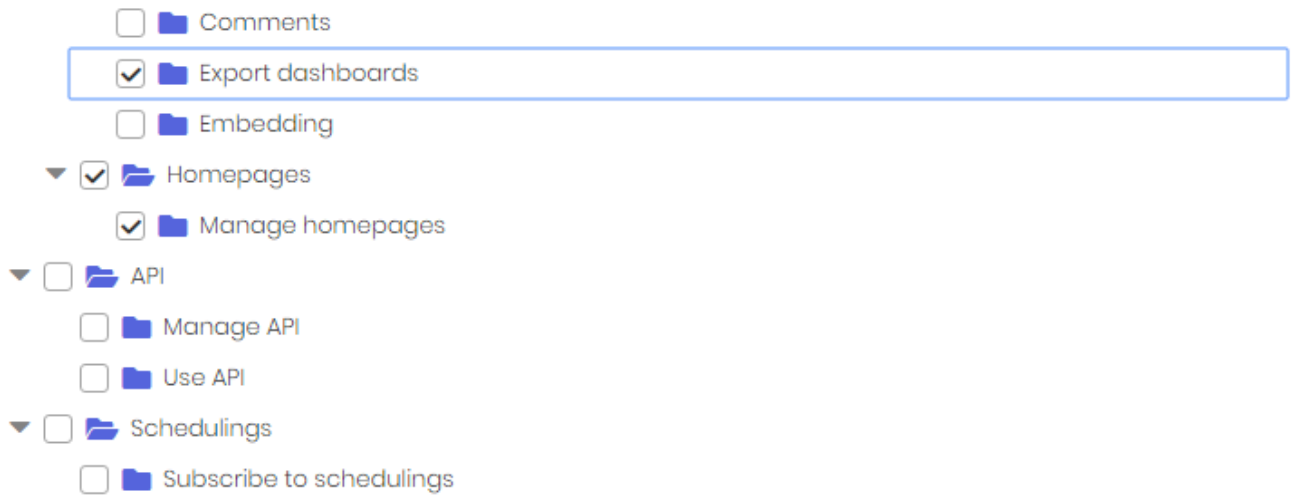
Filter by Permissions

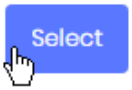
You can filter Users by which permissions are active :

Click *Select permissions* in the top bar :



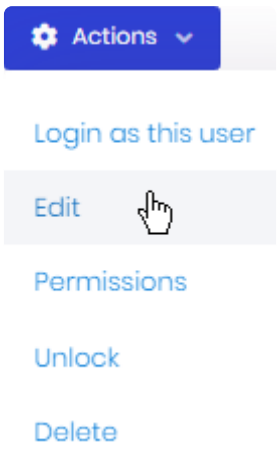
Then select the required Permissions and click *Select* :



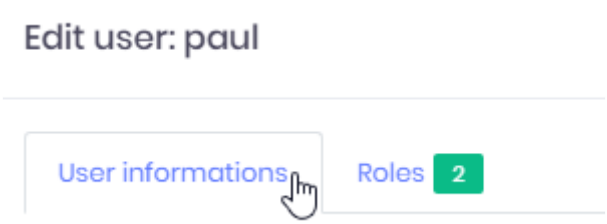


Editing a User

To Edit a User, click the *Actions* > *Edit* button :



User properties contain two tabs :



- *User informations* : all the User's properties, see below
- *Roles* : the Roles assigned to the User

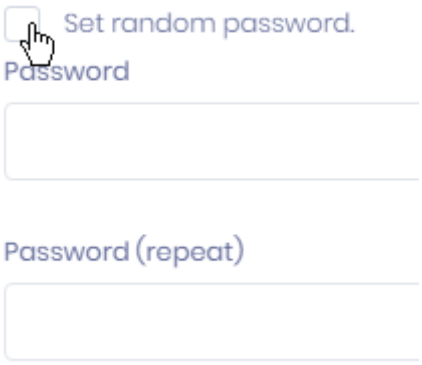
Name and profile properties

User profiles have the following properties to describe them :

- *Name* : the first/given/forename
- *Surname* : the last/family/surname
- *Email address* : a valid e-mail address
- *Phone number*
- *User name* : the name the User will use for logging in

Password

You can *Set random password* or specify a password explicitly yourself :



The option *Change password on next login* forces the user to change their password the next time they will log in :

Active / Activation e-mail

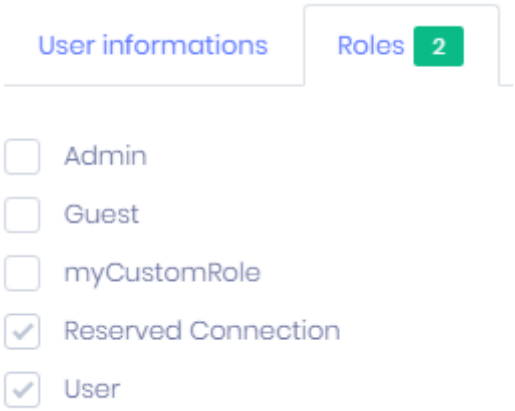
You can select if the User is already *Active* and/or if they should receive an *Activation email* that will confirm their account creation :

Lockout enabled

A User goes on *Lockout* when they have tried to connect with an incorrect password too many times in a row.
This is a security measure that can be disabled individually for users.

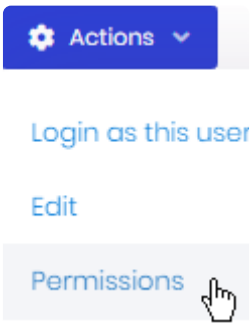
Assigning *Roles*

From the *Roles* tab, you can assign the desired roles to the User :



Overriding Permissions

Users' permissions can be overridden from their Role.
Click *Action* >_ Permissions_ :

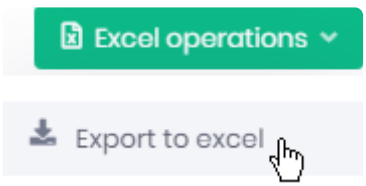


Check the desired permissions, as described in the corresponding chapter.

Exporting Users lists

For administrative purposes, it can be useful to export the list of Users.
Alpana Server lets you export the list of created Users to an Excel file.

Click *Excel operations* > *Export to Excel* :

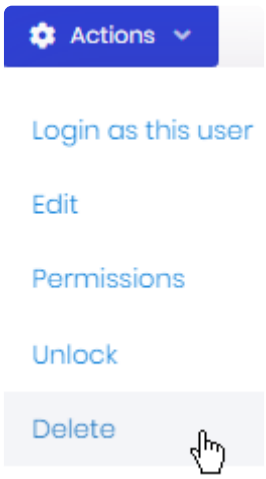


Your web browser will download the Excel file with the full list of Users and their properties :

- Name
- Surname
- User name
- Phone number
- Email address
- Email confirm
- Roles
- Active
- Creation time

Deleting a User

To delete a User, click *Action* > *Delete* :

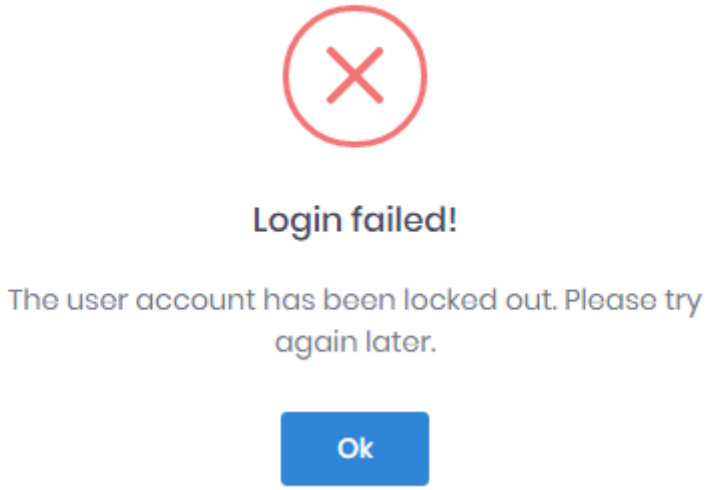


Note : The Users that exist by default in Alpana Server cannot be deleted.

Last modified: Oct 29, 2019

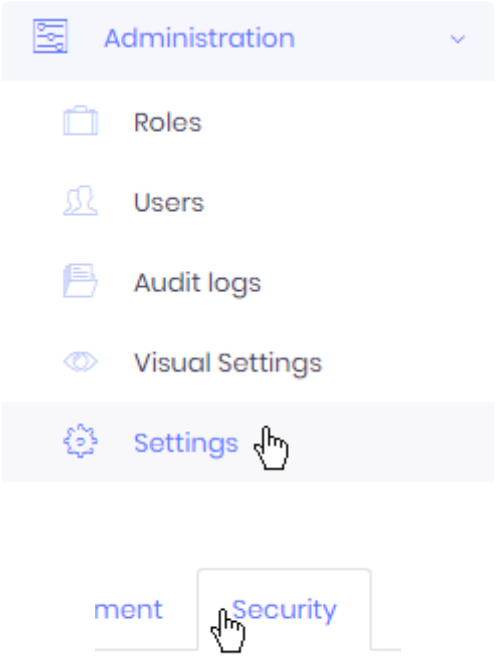
3.10.3.4. User Lockout

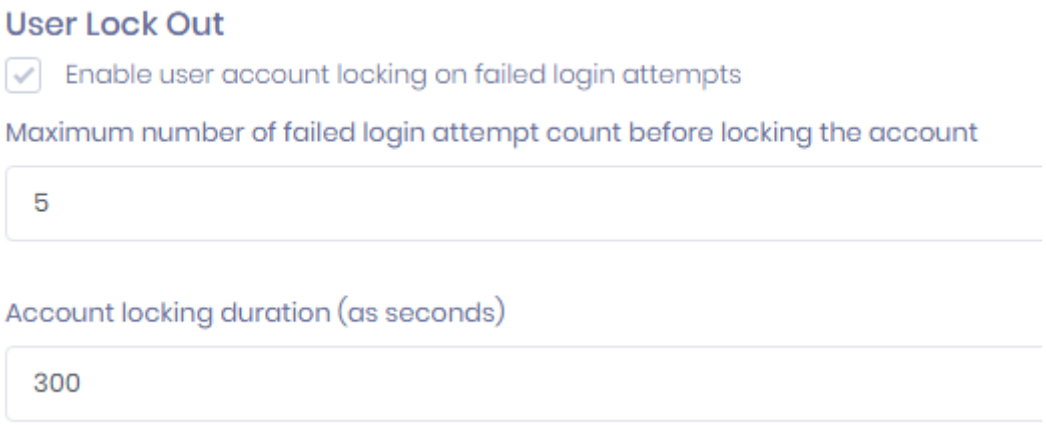
A User goes on *Lockout* when they have tried to connect with an incorrect password too many times in a row.
This is a security measure that can be configured globally and disabled individually for users.



Configuring Lockout

Lockout is configured at the Tenant level under *Administration* > *Settings*, in tab *Security* :





Enable user account locking on failed login attempts

This setting enables or disables the feature.

Maximum number of failed login attempt count before locking the account

This setting defines how many times **in a row** a user must fail to login (incorrect password) to trigger the Lockout.

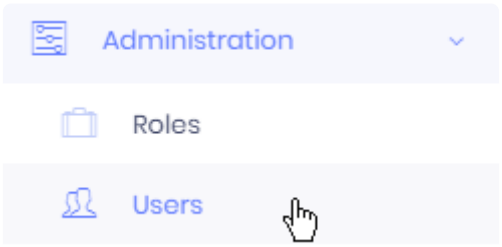
Account locking duration

Number of seconds that the account will stay Locked.
After this duration, the User can try again to login.

How to Unlock a User

When a User has been locked, they must wait for the timeout to complete.
They can also ask an administrator to unlock them.

To manage Users, go to the *Administration* > *Users* page :



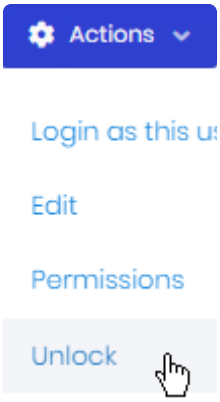
Finding Users on Lockout

You can search for Users who are currently *Locked*.

In advanced filters, check *Only locked users* :

Unlocking a User

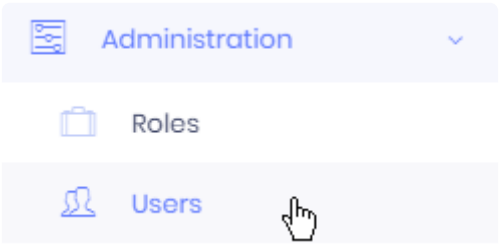
When you have found the locked User, click *Aciton* > *Unlock* :



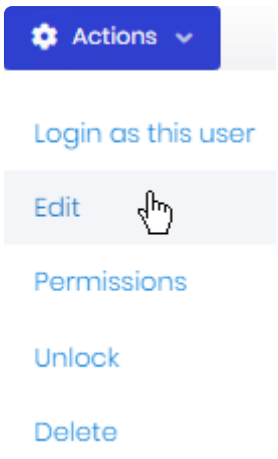
Exceptions

You can exclude some users from the Lockout : those Users can attempt to login as many times as they want and will never be locked out.

To manage Users, go to the *Administration* > *Users* page :



To Edit a User, click the *Actions* > *Edit* button :



Lockout enabled

Un-check *Lockout enabled* create an exception for this User :

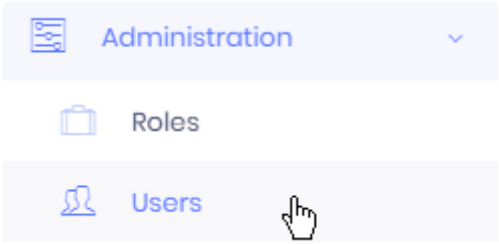
Last modified: Oct 29, 2019

3.10.3.5. User Impersonation

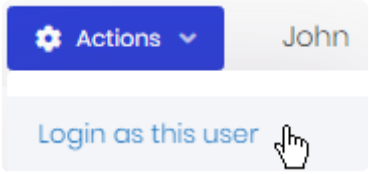
Administrators can impersonate other Users.
This is useful for support.

Impersonating a User

To impersonate another User from the same Tenant (or Host), go to the *Administration* > *Users* page :



Find the desired User, and click *Actions* > *Login as this user* :



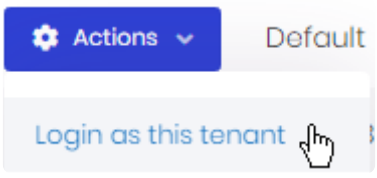
This will log you as the target User.

Impersonating a User from a specific Tenant

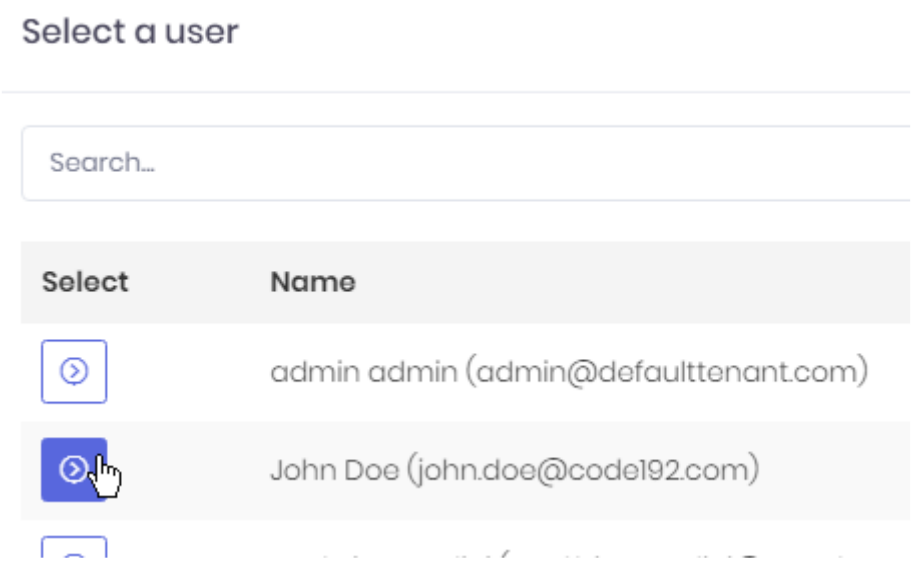
From the Host, you can impersonate Users from Tenants.

Go to the *Tenants* page :

Find the desired Tenant and click *Actions* > *Login as this tenant* :



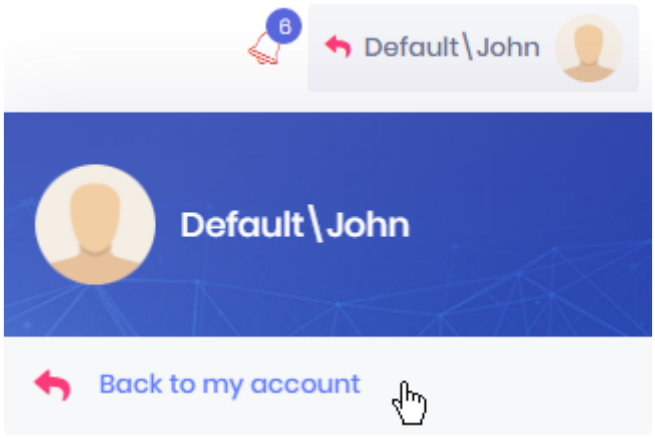
A dialog opens with the list of Users from that Tenant.
Find the desired User of that Tenant, and click the *Select* button :



This will log you as the target User on the target Tenant.

Returning to your account

After impersonating a User, you can return to your own account without logging out.
Click the User's profile button and select *Back to my account* :

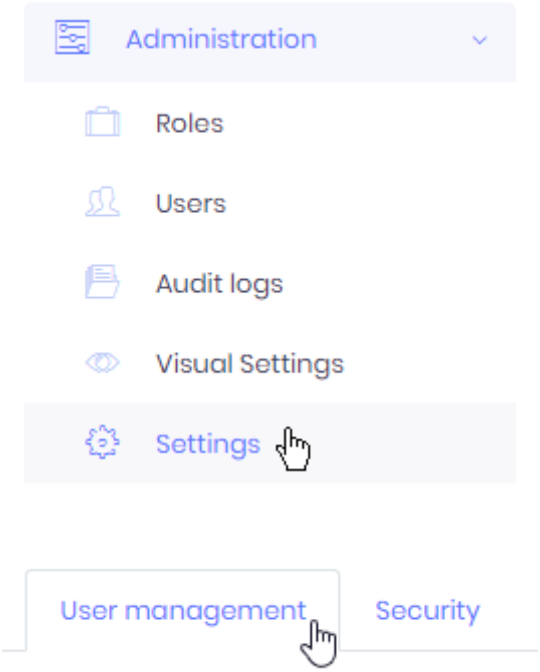


Last modified: Oct 29, 2019

3.10.3.6. User Settings

As administrator of the Host or of a Tenant, you can configure user account and login features on a global level.

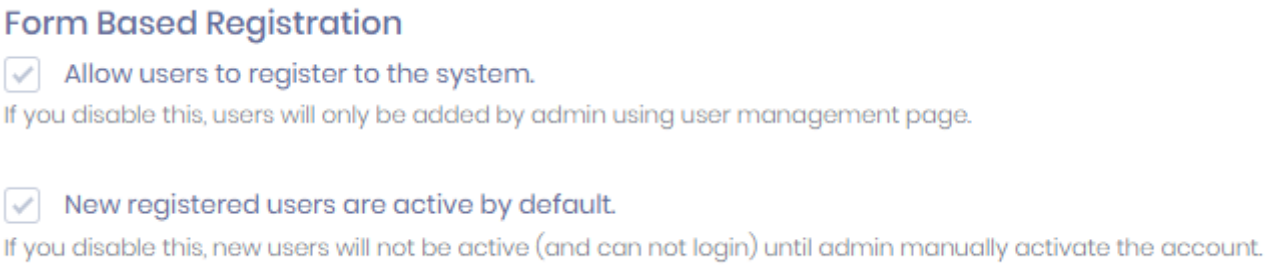
To manage user account and login features of a Host or Tenant, go to the corresponding *Administration* > *Settings* and browse to the tabs *User management* and *Security* :



User management

Allowing users to Sign Up

You can allow users to create their own account :



- *Allow users to register to the system* : users will be able to create a new account by filling-out a form on the Login page
- *New registered users are active by default* : users will be active immediately without administrator intervention

Inform of cookie use

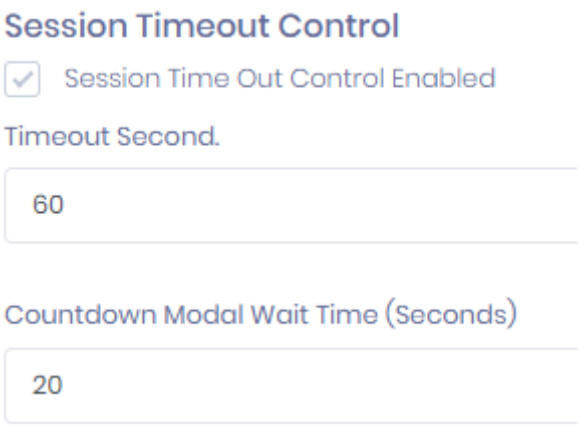
You can inform users that the Alpana Server website uses cookies :

Users will see an information message when they login, which they can validate by clicking “Got it!” :

Session timeout

You can force users to logout on inactivity.
This is useful if you want to limit the un-necessary use of concurrent licenses.

Check *Session Time Out Control enabled* :



- *Timeout Second* : how many seconds to wait for inactivity and start displaying the inactivity countdown message
- *Countdown Modal Wait Time (Seconds)* : how many seconds to display the inactivity countdown message before actually forcing the user to logout

When the User has been inactive for *Timeout Second*, the below inactivity countdown message is displayed.
If the User moves the mouse or keyboard, the message disappears and the *Timeout Second* is reset. Otherwise, the user is logged out.

Note : Changes to this setting don't apply to users that are currently logged-in. To apply the change, users must logout, then login again.

LDAP

You can let users from an Active Directory domain connect to the Tenant.

Pre-Requisite

For this option to appear, you must first enable the "LDAP" Feature at the Host level, then assign it to the desired Tenant.
Finally, **you must restart the Alpana Server app** to apply the change (restart the BackEnd website).

Configuring

On the desired tenant, check the option *Enable LDAP Authentication*, then fill-in the form :

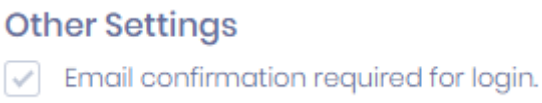
- *Domain name* : Active Directory domain
- *User name* : user on the ACtive Directory with permissions to get the user list
- *Password* for that User.

Using

When configured, users can login using their Active Directory username and password, and the Alpana account will automatically be created for them, like a new User.

Email confirmation

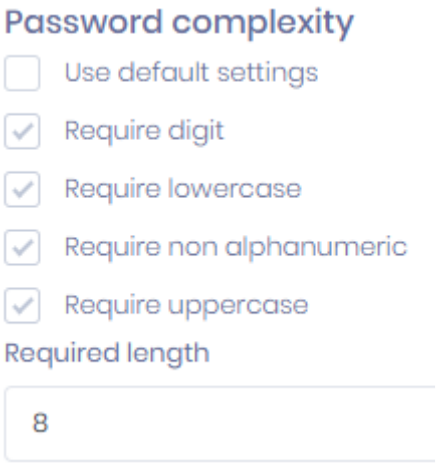
You can require users to confirm their e-mail address before they can login for the first time :



Security

Password complexity

You can force user to include a minimal complexity in their choice of password :



Un-check *User default settings* and check the required character sets and password length.

User Lock Out

You can configure Alpana Server to lock user accounts for a while when they have entered a wrong password several times.
See User Lockout

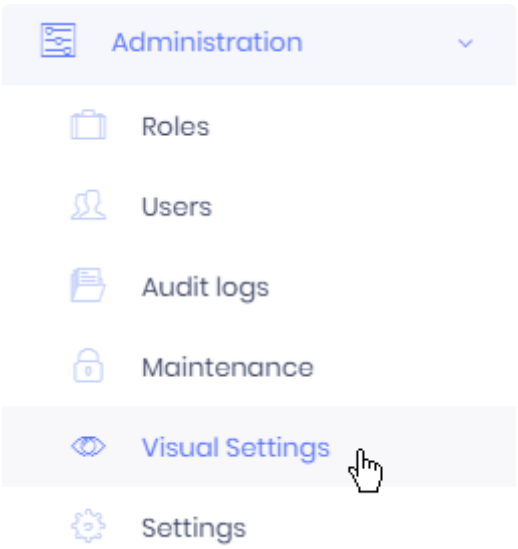
Last modified: Jun 19, 2020

3.10.4. Theme and Appearance

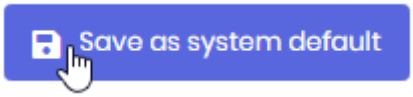
The visual theme of the Alpana Server web interface can be configured in many ways.
You can configure a different look and feel for each Tenant and the Host separately.

Theme configuration

Go to *Administration* > *Visual Settings* :

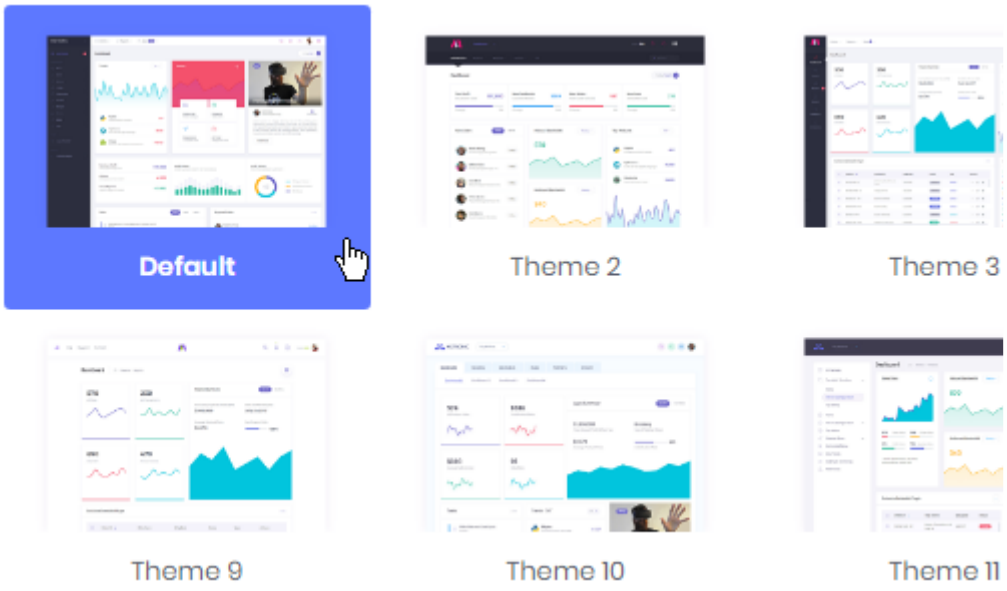


Make the required changes, then click *Save as system default* :



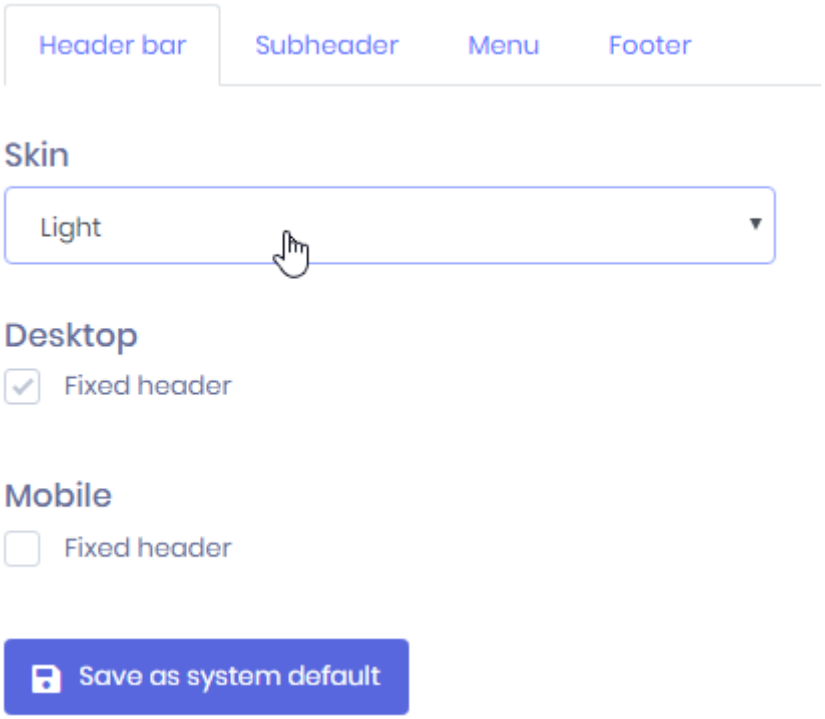
Select a Theme

To select a Theme, click on the corresponding thumbnail :



Configure a Theme

To configure the selected Theme, use the options at the bottom of the page :

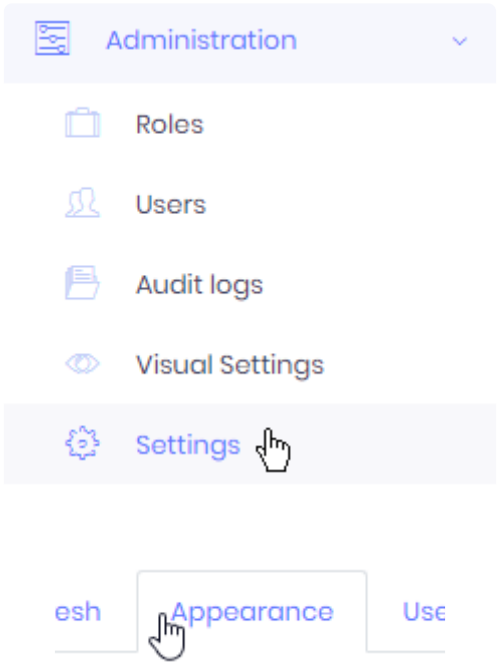


Depending on the selected Theme, you may configure many properties, like menu behavior, header colors, etc.

Custom Logo

Instead of using the default Alpana logo, you may use your own.

Go to *Administration* > *Settings*, in the *Appearance* tab :



Customize logo

Under *Logo*, click *Choose file* and browse your filesystem :

You must select a `JPG` / `JPEG` / `PNG` / `GIF` file with a maximum of 30KB size and 139×35 pixel resolution.

Then click *Upload*.

Remove custom logo

to remove the custom logo and recover the original Alpana logo, click *Clear*.

Overriding the Theme CSS and colors

You can override the Theme CSS to include your own colors.
This is described in the next chapter How to Use a Custom CSS

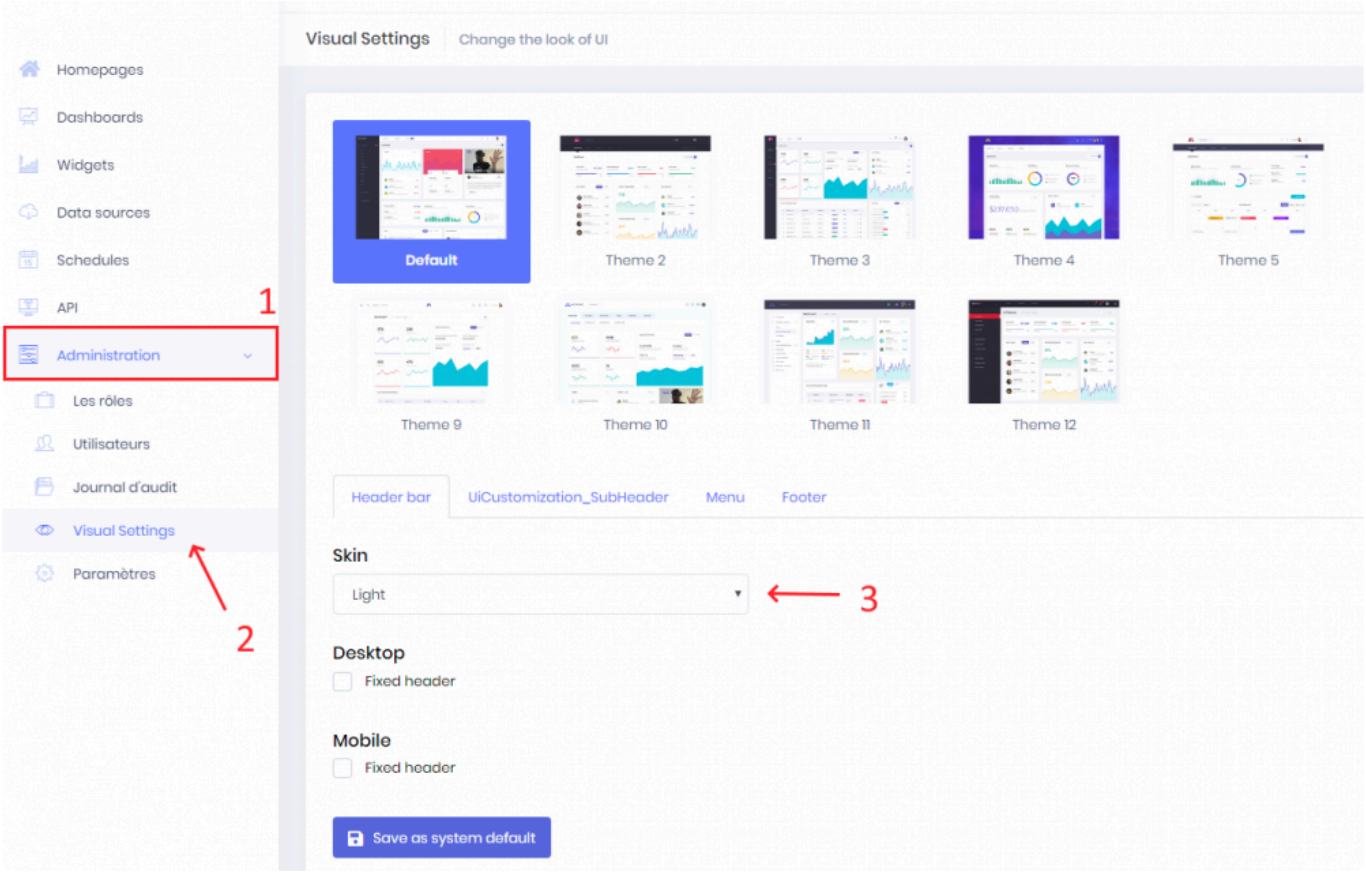
Last modified: Oct 29, 2019

3.10.4.1. How to Use a Custom CSS

Applying a Custom CSS

To apply your Custom CSS theme to Alpana Server please follow the below steps:

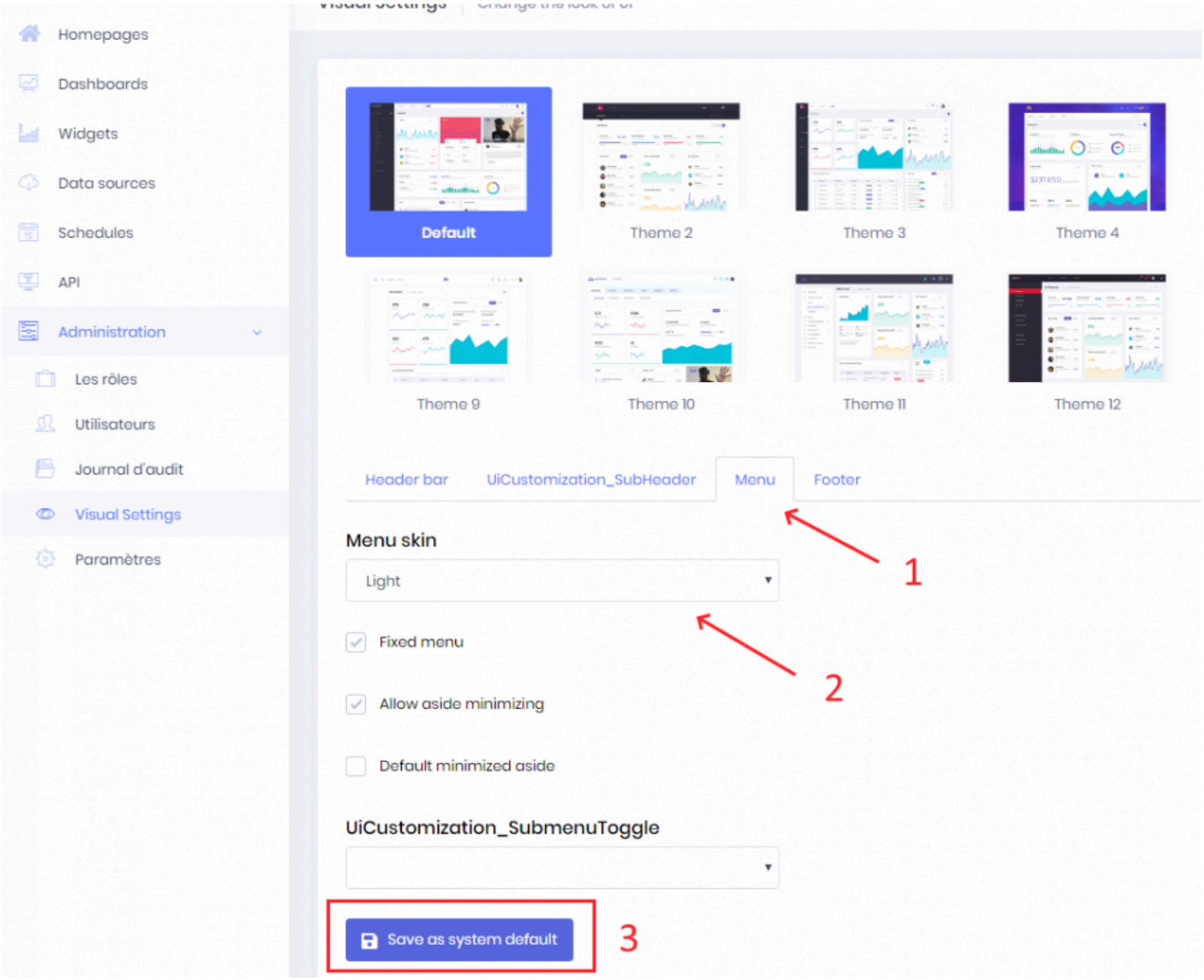
Prepare the Theme



First, you need to go to the Visual Settings Page by clicking on Administration (Step 1 on screenshot), then click to Visual Settings (Step 2 on the screenshot).

Then, Select the Light Skin in the Skin Combobox. (Step 3 on the screenshot)

Once done, click on the menu tab like the step 1 on the following screenshot:



Then, select Light in the Menu Skin Combobox like Step 2 on the screenshot.

Once done, please ensure the DEFAULT theme is selected like on the screenshot and click on the “Save as system default” button which is Step 3 on the screenshot.

Create the CSS file

After this; you will need to prepare your custom CSS file.

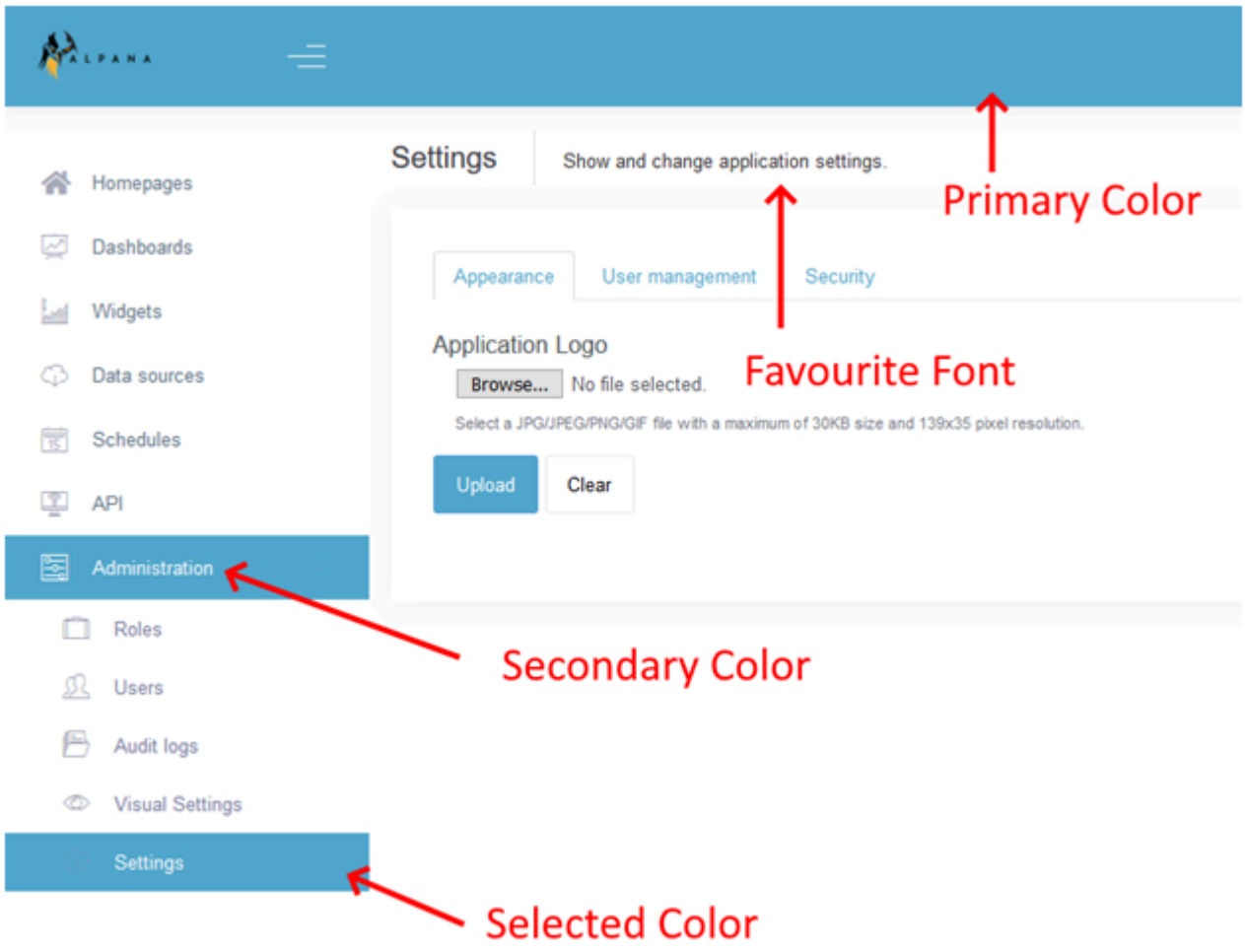
To help you with this step, we prepared a template file, please download the following file: Alpana Server Custom CSS Placeholders File

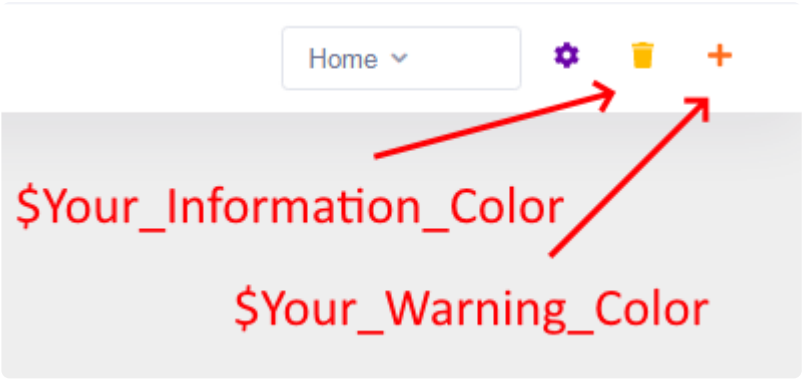
Then, you need to edit this file with your favorite text editor by changing the following placeholders by your custom values:

```
$Your_Primary_Color
$Your_Secondary_Color
$Your_Selected_Item_Color
$Your_favourite_font
$Your_Information_Color
$Your_Warning_Color
```

```
$Your_Error_Color
```

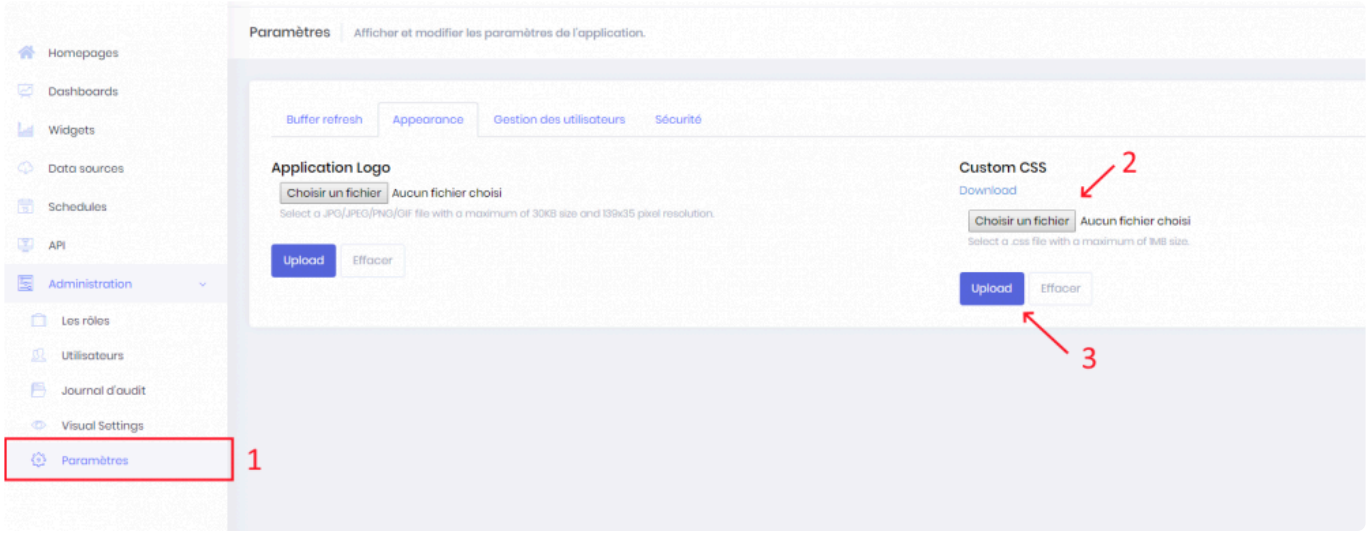
Please take a look at the placeholders description in the following screenshots:





Upload the CSS

When it’s done, you need to go to the Parameter Section shown in the screenshot below by clicking on Parameter (Step 1)

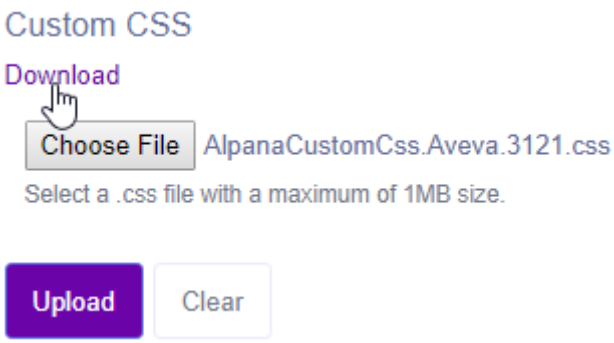


Then click on the Browse button (Step 2). From there, you need to select on your computer the previous edited file with your custom values and click “open” from the dialog box.

Finally click on the Upload button (Step 3) and you should have your selected theme applied.

Modifying the existing CSS

Once you have applied a Custom CSS, you can download it by clicking *Download* :



This allows you to modify it and re-upload it.

Removing the Custom CSS

If you wish to remove the custom CSS, click *Clear* :

Custom CSS

Download

Choose File AlpanaCustomCss.Aveva.3121.css

Select a .css file with a maximum of 1MB size.

Upload Clear

Last modified: Oct 29, 2019

3.10.5. Host-level Settings

Some Alpana Server settings are managed at the Host level only, and impact all Tenants.
The following chapters detail how to configure those settings.

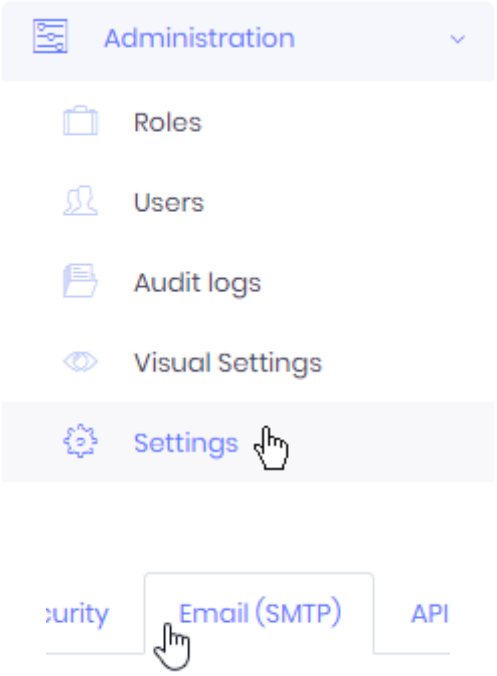
Last modified: Oct 31, 2019

3.10.5.1. Email (SMTP)

Alpana can send e-mails to users, and requires to configure the e-mail sending for the following usage :

- user activation
- recovering forgotten passwords
- sending exports through Schedules
- receiving notifications

To manage e-mail settings, go to the Host, in *Administration* > *Settings* and browse to the tab *Email (SMTP)* :

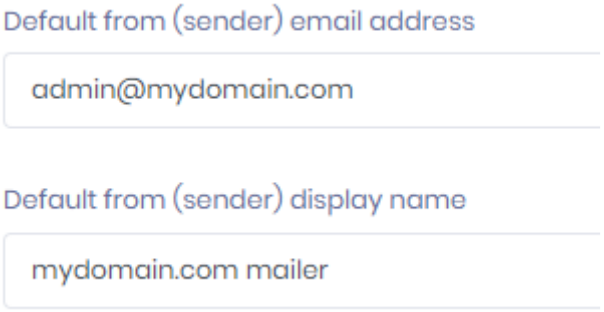


Confiigure E-mail sending

Configure Sender

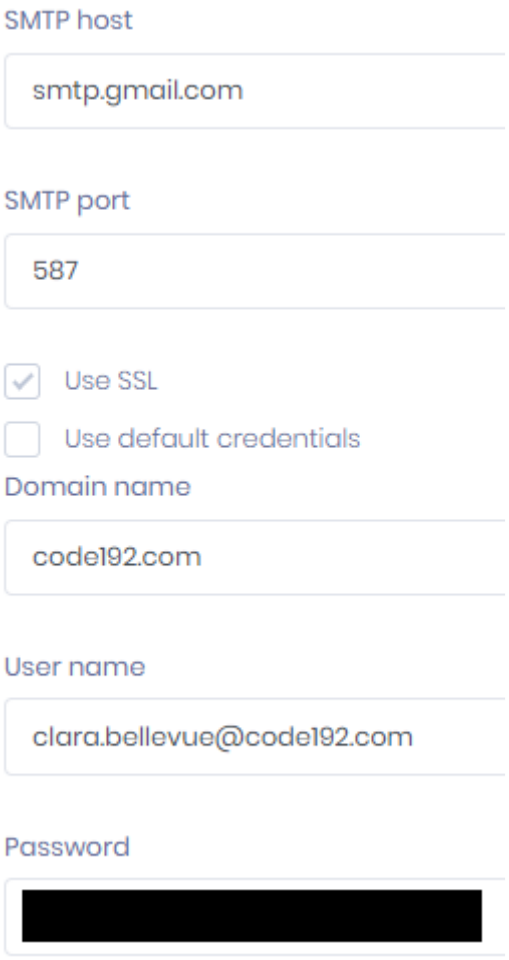
You can configure which e-mail address and display name will be used for sending e-mails :

- *Default from (sender) email address* : the email address that wil be used for sending e-mails on behalf of Alpana Server
- *Default from (sender) display name* : the full name that will be displayed on e-mails originating from Alpana Server



Configure SMTP Server

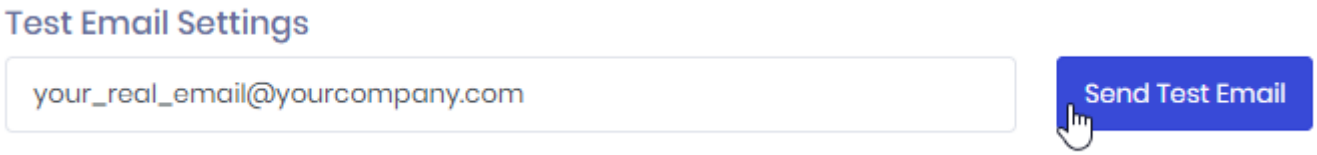
You must configure a SMTP Server with ability to send e-mails on behalf of Alpana Server :



- *SMTP host* : the hostname where the SMTP server is running
- *SMTP port* : the port where the SMTP server is running
- *Use SSL* : check if the communication with the SMTP server is encrypted
- There are two alternative authentication options :
  - check *Use default credentials* : let Alpana Server connect anonymously to the SMTP server. Some SMTP server configurations don't allow this.
  - uncheck *Use default credentials* : fill-in a *Domain name*, *User name*, *Password* to let Alpana Server authenticate to the SMTP server

Test E-mail sending

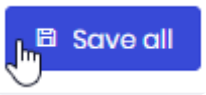
To verify that the configuration works, you can tell Alpana Server to send a test e-mail to your e-mail address :



If you receive the e-mail correctly, then the configuration is valid.

Save settings

Once the configuration is good, you must save it by clicking the *Save all* button on the top right :



Last modified: Nov 04, 2019

3.10.5.2. API Origins Whitelist

Why

Querying or embedding a content from Alpana into another website requires to be allowed from Alpana. This is called Cross-origin resource sharing , see also Same-origin policy.

If you fail to do so, you won't be able to query or embed Alpana content from your application, and you will get a typical CORS error like :

```
Access to XMLHttpRequest at 'http://xxx.xxx.xxx.xxx:xxx' from origin 'http://yyy.yyy.yyy.yyy:yyy' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.
```

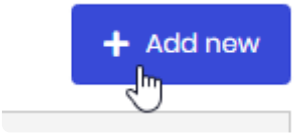
This is a normal thing, for security purposes.
For example, the https://www.google.com/ URL currently refuses to be embedded inside an iframe of another website.

In Alpana, you can create an exception, to allow a specific "foreign website" to access Alpana content.

> If you are making queries from Javascript code in a local HTML file open in your browser, you won't be able to make a query and your browser will probably block you.
> If you really just want to test that your code works, then your browser may let you disable security for this.
> Check your browser documentation or internet (e.g. `Chrome.exe --disable-web-security`)

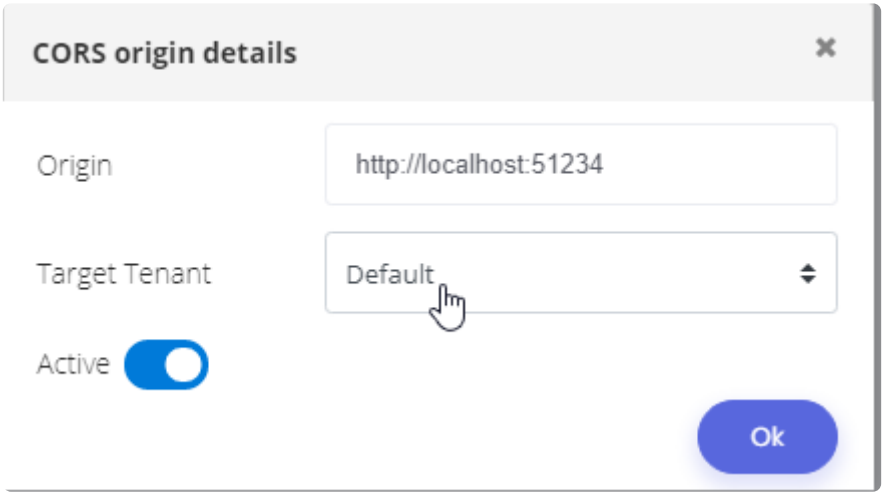
Creating an exception

On the Host of the Alpana Server; go under *Administration* > *Settings* > *API Origins Whitelist* and click *Add New* :



Configuration

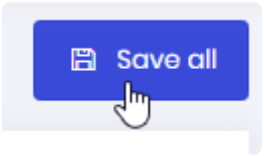
In the configuration window, fill in the details :



- *Origin* : this is the URL of your application. The format is `(protocol)://(site):(port)` with no trailing slash `/`. Examples : `https://www.google.com:80`, `http://123.456.789:81`
- *Target Tenant* : the tenant that will accept the queries or embedding
- *Active* : the exception is Active or not

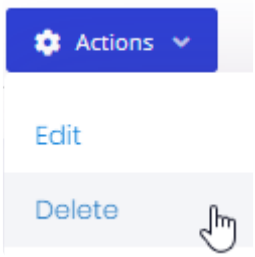
Click *OK*.

At the top of *Administration* > *Settings* > *API Origins Whitelist*, don't forget to click *Save All* :



Deleting an exception

To delete the desired exception from the list, click *Action* > *Delete* in front of it :

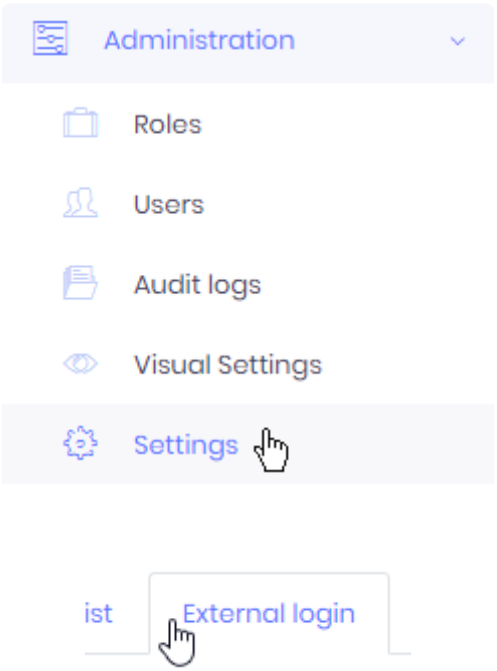


Last modified: Oct 31, 2019

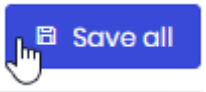
3.10.5.3. External Login

You can let users login using external authentication authorities, through SSO.

To configure these external login configurations, go to the Host, in *Administration* > *Settings* and browse to the tab *External login* :

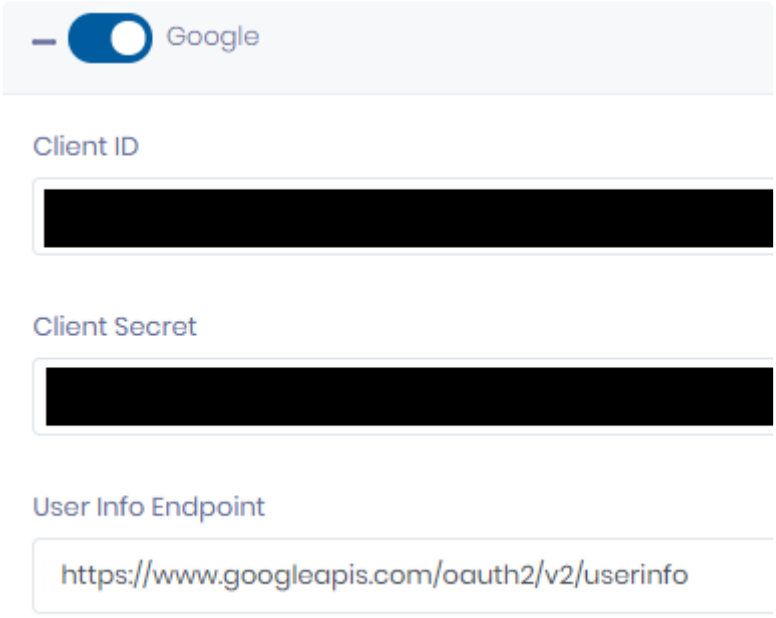


When configured, click *Save all* at the top right :



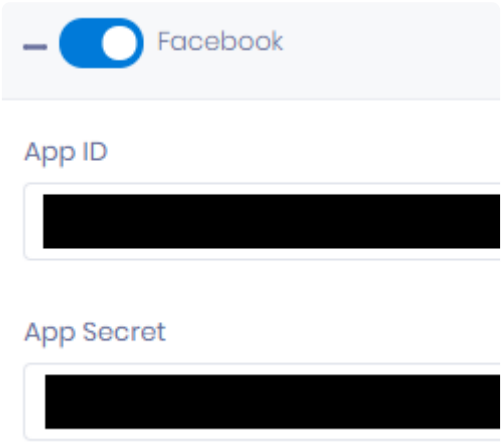
Google

To use authentication through Google, fill-in the corresponding form :



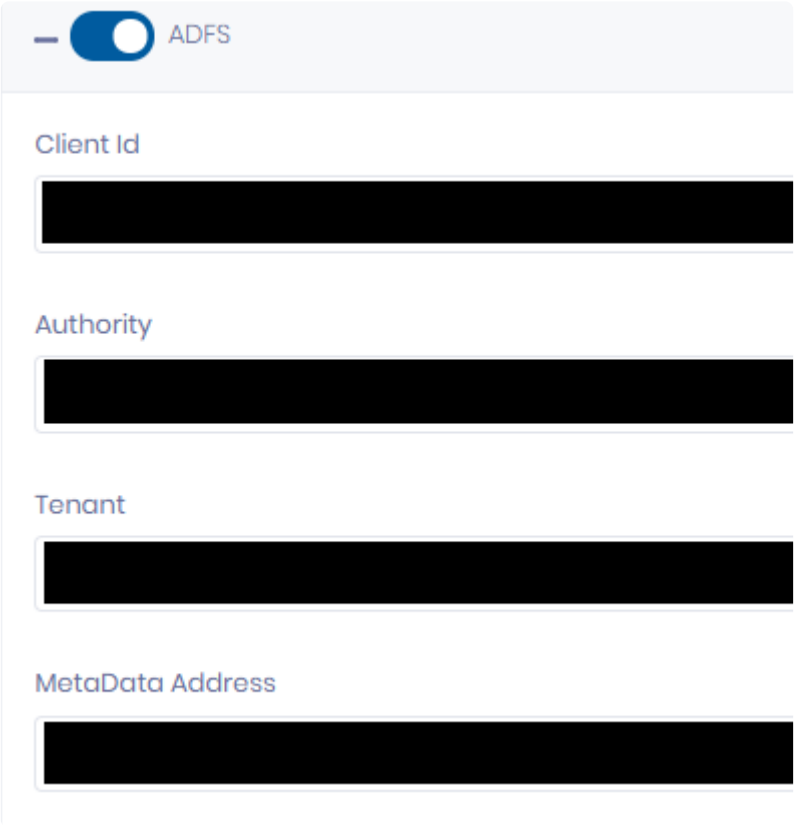
Facebook

To use authentication through Facebook, fill-in the corresponding form :



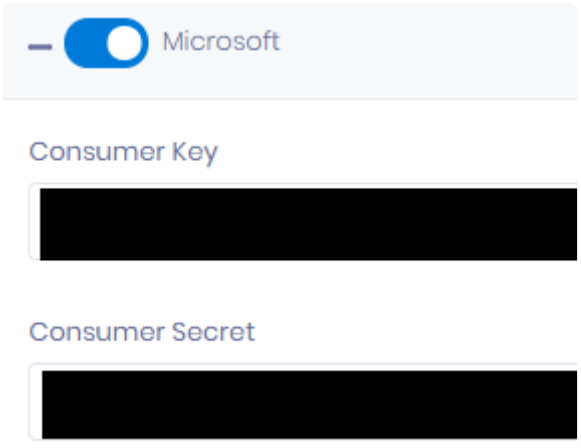
ADFS

To use authentication through Active Directory with WsFederation (ADFS), fill-in the corresponding form :



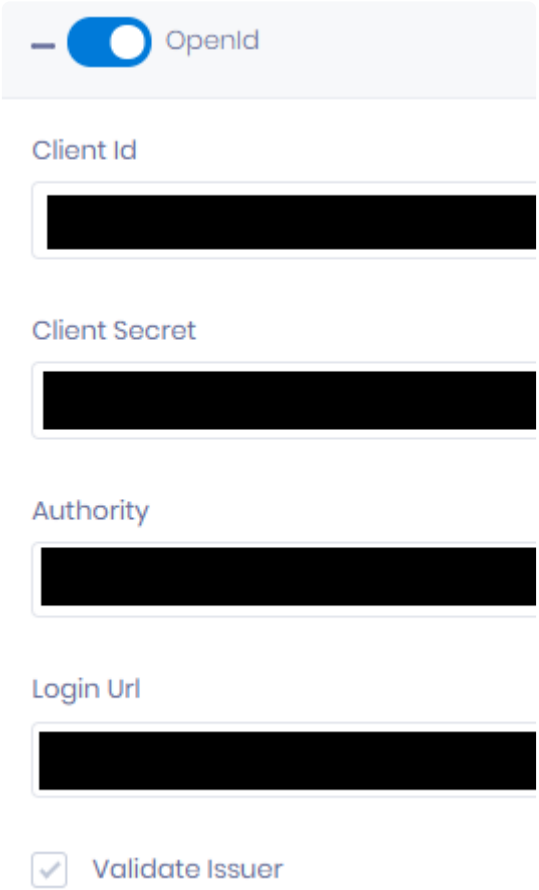
Microsoft

To use authentication with a Microsoft identity account, fill-in the corresponding form :



OpenId

To use authentication through OpenId, fill-in the corresponding form :



Last modified: Oct 31, 2019

3.10.5.4. Concurrent Login

By default, each specific User of the Tenants can be logged-in from multiple browser simultaneously.
As an administrator, you can change this behavior.

Go to the Host, in *Administration* > *Settings* and browse to the tab *Security*.
Then check the *Only one concurrent login per user* option.

Now, when a User logs in on a new browser, the other open sessions will be closed automatically.

Last modified: Jun 19, 2020

3.10.6. Background Jobs

(new in version 2020 R1)

Introduction

Long-running tasks are managed by a Background Jobs manager : manual dashboard exports, manual data import from files, etc
If the job fails, it is retried once.
Results are communicated through the Notifications.
This allows a better handling of simultaneous exports

> Job data is retained for 1 week. After that it becomes deleted, and you cannot download your old PDF export for example.

Jobs Overview

On each tenant, there is an administration interface to ti the Jobs.
Go to *Administration* > *Background Jobs* to view :

- *Execution History*
- *Processing Jobs* that are currently running
- *Enqueued Jobs* that are ready to run
- *Next Executions* which are upcoming

Hangfire dashboard

For advanced administration/debugging and support, the internal background job manager exposes a dashboard.
It is called “Hangfire”, and the dashboard is available on the web page of the Back-End website of Alpana Server.

To access it, simply use your web browser to navigate on the Back-End website of Alpana Server.

Login with your **Host** credentials, and use the Hangfire web page to view internals of the Background Jobs.

Last modified: Jun 19, 2020

3.11. API

Alpana Server exposes a web API that exposes many actions and properties.
The following chapters document how to use this API.

Last modified: Jul 23, 2019

3.11.1. General API help and troubleshooting

Permissions

To work with the API, the user must have the required permissions on the Tenant :

- API permissions
- desired object permissions : for example, for API queries on Schedules, the user must have permissions on Schedules

API location

The API responds on the Back-End website (Alpana3App).

Resources

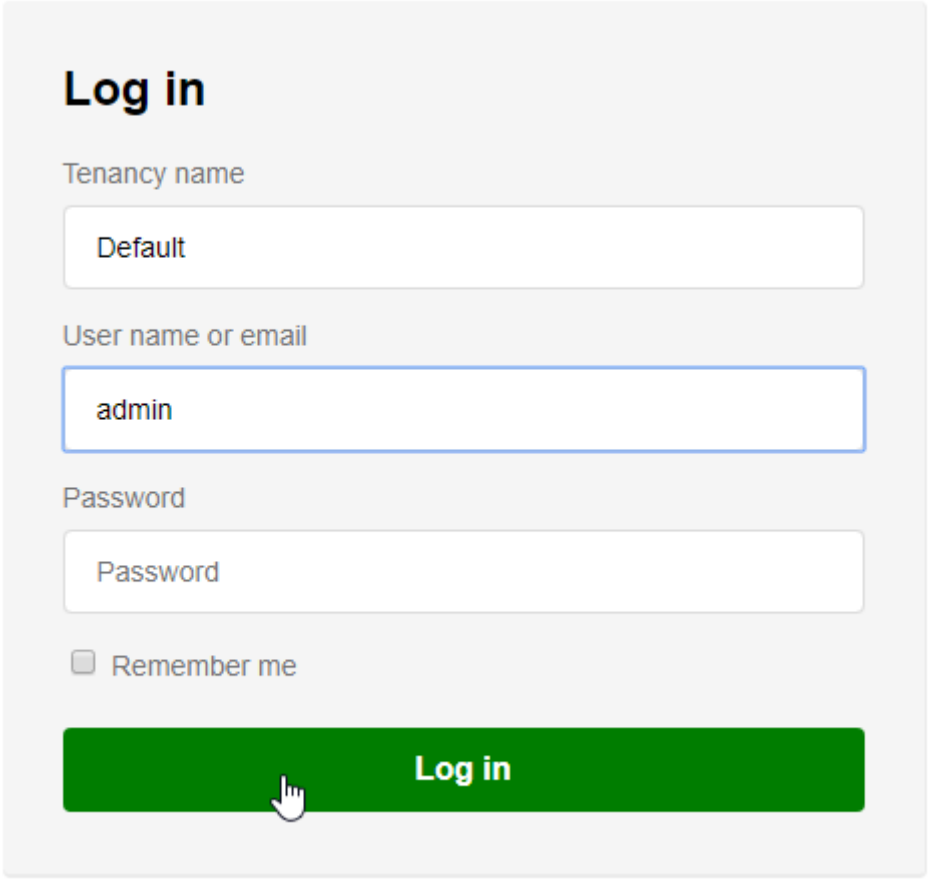
API Code snippets on Alpana Server

On your Alpana Server tenants, users with permissions to the API have access to auto-generated code snippets to help you make common requests.
So use this utility, login to the desired tenant (to the Front-End) and navigate to menu *API* :

Swagger UI : Automatic documentation and test

Swagger UI allows you to access the full list of API endpoints, along with some auto-generated documentation, and an interface for testing queries.
To access it, simply use your web browser to navigate on the Back-End website of Alpana Server.

Login with your tenant credentials, and use the Swagger web page to browse endpoints, unfold their documentation, and try them from the browser.



Testing with Postman

You can easily test your API queries with a third-party tool called Postman .

This allows you to configure all parameters of the queries, and generate code in your programming language.

Common issues

If your application is a web application running in a browser, it will need to be allowed to make queries to the API. Otherwise, you will encounter CORS errors.
In order to do this, see API Origins Whitelist.

Last modified: Jul 23, 2019

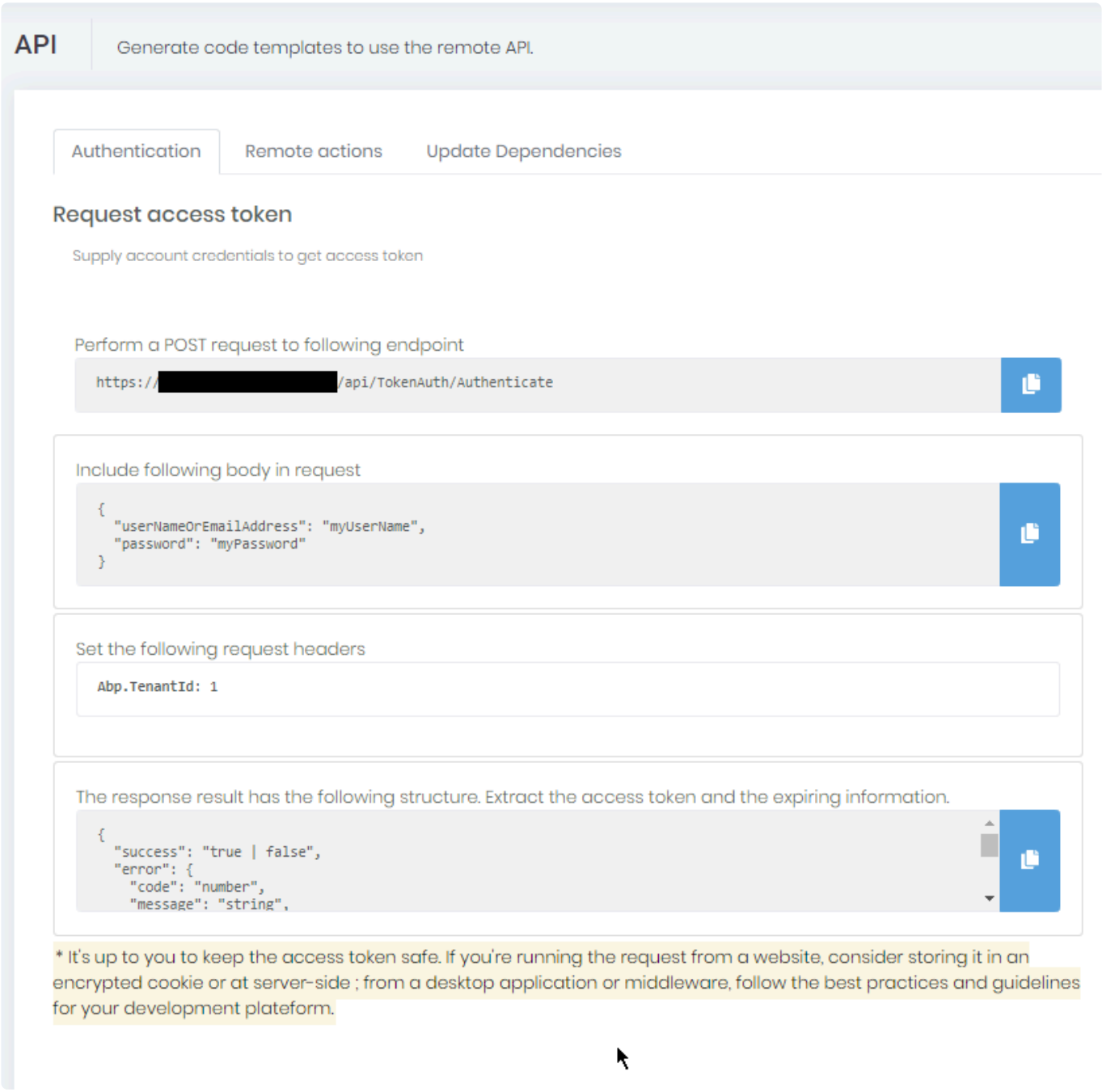
3.11.2. Authentication to the API

Getting code snippets for authentication

Getting an authentication token

Authentification to the API is documented in each Tenant under the menu *API* > *Authentication*

Perform the documented query with a valid username/password combination in order to get an **authentication token**.



Token validity

The access token lifetime is 7 days.
The refresh token lifetime is 1 year.

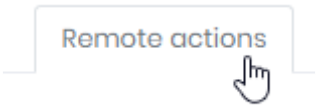
Important note on Security

If you need to perform authenticated queries programmatically from your application, you will need to consider the best practices for handling these credentials safely, without storing passwords for example.

Last modified: Aug 26, 2020

3.11.3. Remote Actions

Remote Actions are some commonly used API actions that trigger actions to clients.
You can find the corresponding snippets by navigating to *API* > *Remote actions*.



Last modified: Jul 23, 2019

3.11.3.1. Refreshing / Updating Filters

Scenarios

Refreshing

You want to remotely force a refresh on any widget from any dashboard currently open in a browser.
This can be because you want event-based refresh.
This can be because you want very fast refresh (almost “live”) on some widgets.

You may want to filter target users who see this refresh.
This can be because you only want to force a refresh on the TV screen on the plant floor, and not disturb other users.

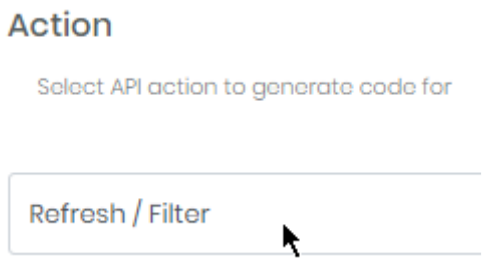
Updating Filters

You want to remotely send Parameters values inside any dashboard currently open in a browser.
This can be because you embedded the Alpana dashboard inside your application, and you want to send filter values from your application.

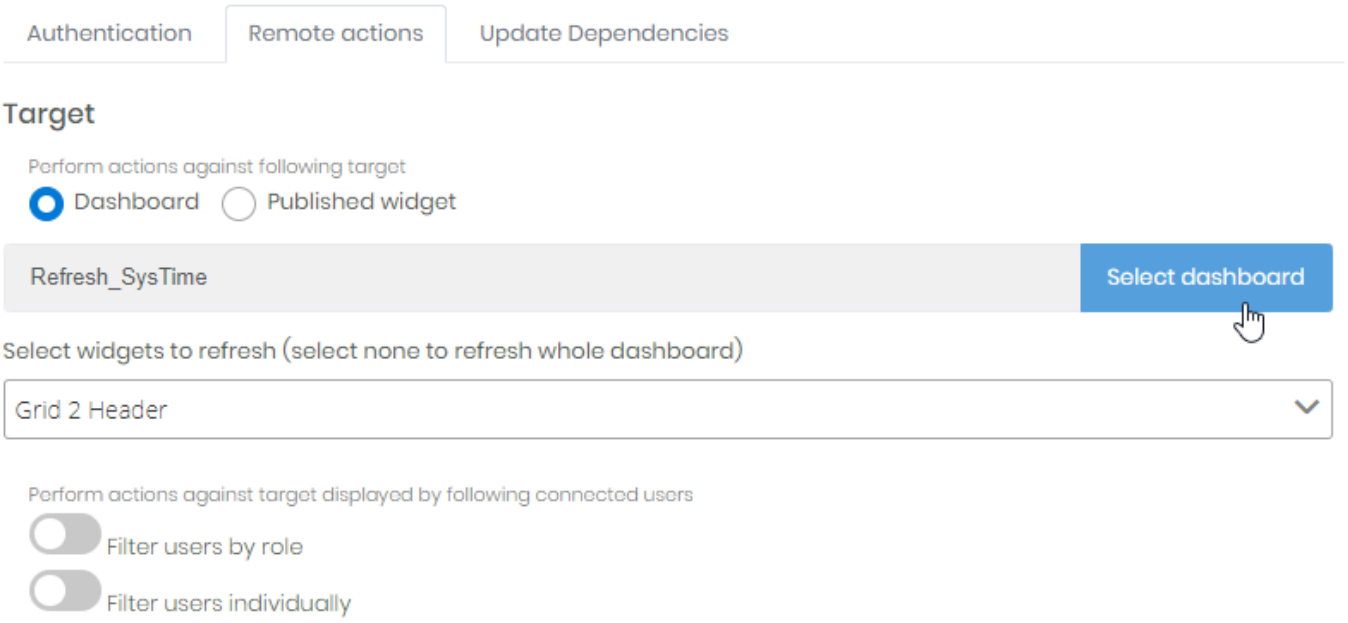
You may want to filter target users who see this filtering.
This can be because you want only the user currently browsing the application to see the filtering.

Code snippets

You can find code snippets by navigating to *API* > *Remote Actions* and selecting Action *Refresh / Filter* :



Fill in the desired information as *Target* on the left pane (like the desired Dashboard, Parameters, etc) :



Then copy to your application the code snippets generated on the right.

Last modified: Jul 23, 2019

3.11.3.2. Real Time Widgets Data

Scenario

You want to instantly send values inside a Widget, but you don't want to configure this Widget with a Data Source.
For example, the data does not exist in storage, but your application holds it and sends it *live* into the published Widget.

Example :
You have a Circular Gauge widget that is published on Alpana Server, but configured with no Data Source.
You have an ArchestrA object with some UDA that is not stored to Historian.
The object has a script that runs periodically every 3s to send the value from the UDA into the Circular Gauge widgets.
This value is immediately displayed to users who have the dashboard open at that time.

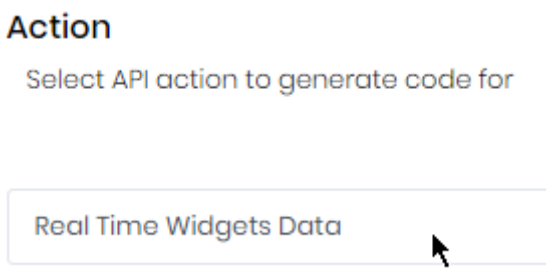
Preparing the target Widget

First, you need to configure the Widget.
See Binding to Real-Time Data to configure the Widget to accept the real-time data.
In this configuration, you define the Widget graphical properties and which fields will receive data. You don't actually connect the Widget to a Data Source.

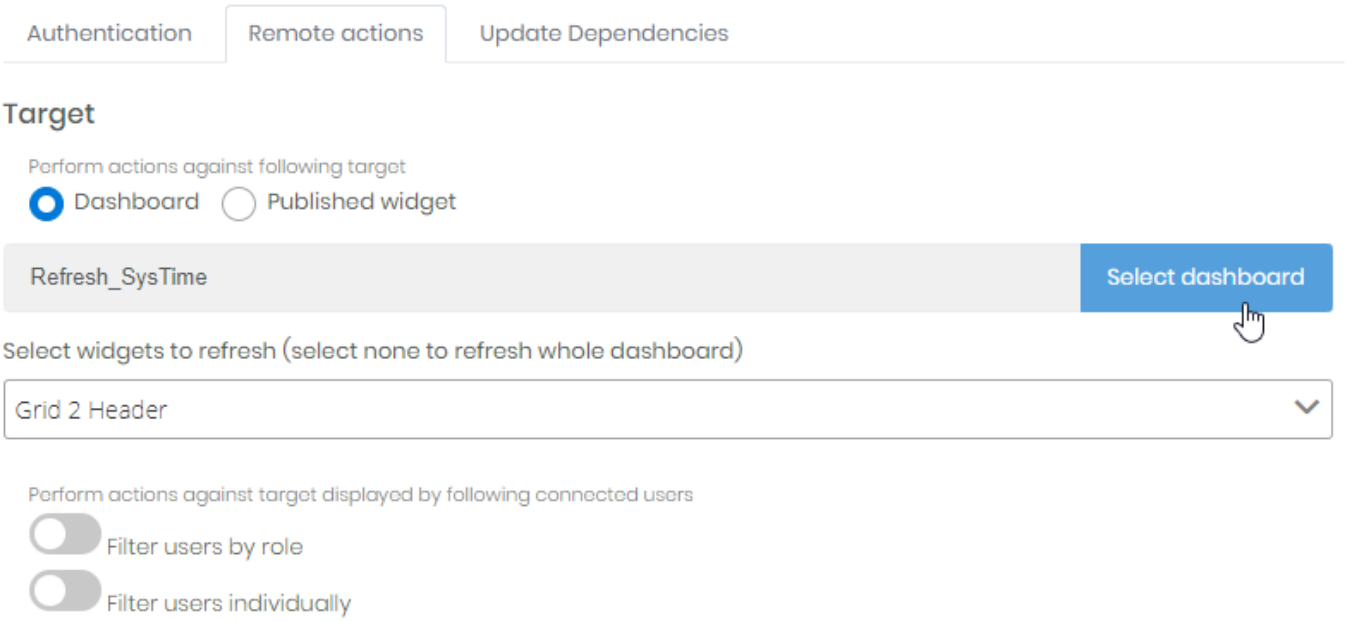
Generating Code snippets

Finally, you need to send data through the API.

You can find code snippets by navigating to *API* > *Remote Actions* and selecting Action *Real Time Widgets Data* :



Fill in the desired information as *Target* on the left pane (like the desired Dashboard, Parameters, etc) :



Then copy to your application the code snippets generated on the right.

Replace the `"values"` with the desired values.

Last modified: Apr 24, 2020

3.11.4. Client-Side Hooks

Alpana provides hooks to let you implement custom behavior when you embed inside your application. You can subscribe to be notified of the event in Javascript, and get event properties.

> ! Authentication is not explained here, and should be managed by your application in a secure manner.

First, embed the Dashboard in your application by using the “Embed in code (use jQuery)” code snippet.

Then in your code, you can do something like :

```
window.addEventListener('message', function (e) {
        if (e.data && e.data.name) {
                switch (e.data.name) {
                        case 'onParametersChanged':
                                const parameters = e.data.arguments;
                                if (parameters && parameters.length) {
                                        alert(parameters[0].name
                                                + ' value changed : '
                                                + (parameters[0].values && par
ameters[0].values.length ? parameters[0].values.join() : 'null'));
                                }
                                break;
                }
        }
}, false);
```

Last modified: Oct 31, 2019

3.12. Maintenance and Migration

This chapter describes commonly needed maintenance operations.

Last modified: Jun 17, 2019

3.12.1. Start / Stop IIS websites

Alpana Server deploys websites in Microsoft IIS :

- `Alpana3App` : the back-end site
- `Alpana3Client` : the front-end site

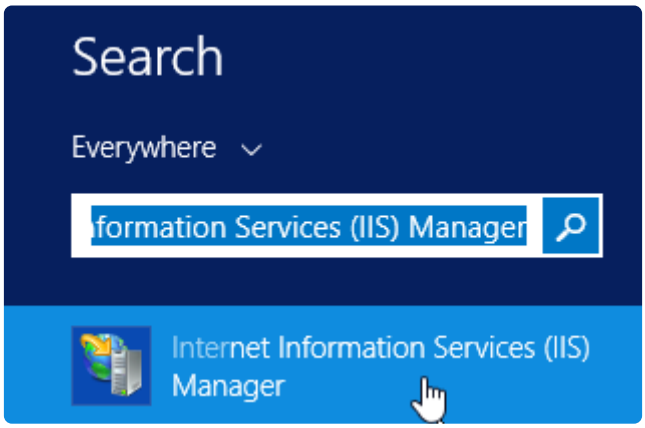
You can start/stop/restart those websites by using the IIS management tools.

When the machine restarts, the started/stopped state of the sites is maintained by IIS.

IIS Management Console

To open IIS Manager, run the program `inetmgr`

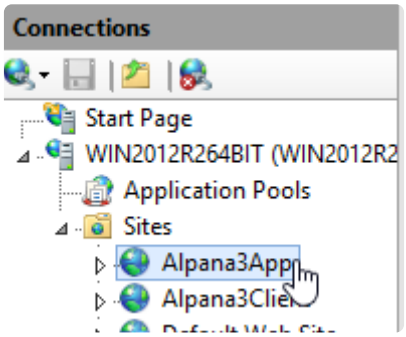
Alternately, you can search for the program name “Internet Information Services (IIS) Manager” :

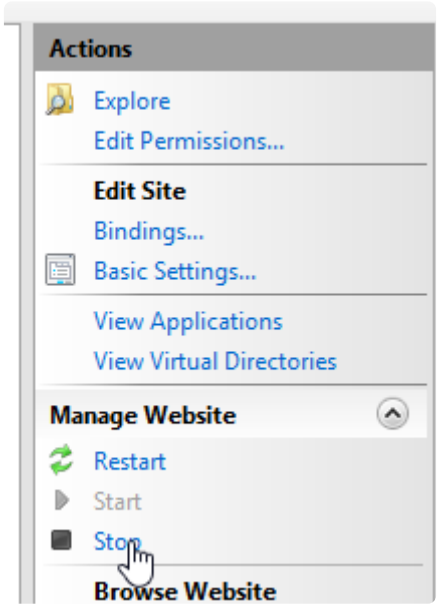


Starting/Stopping/Restarting websites

The websites may be started from the Alpana Server installer, as a final option (default).

If you wish to start/stop/restart the websites manually, open IIS Manager, click on the website and click *Start* (or *Stop* or *Restart*) :

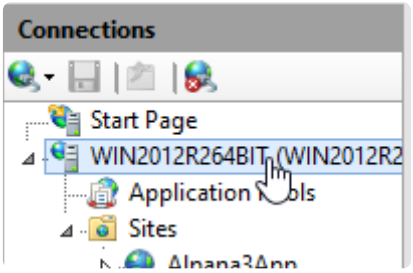


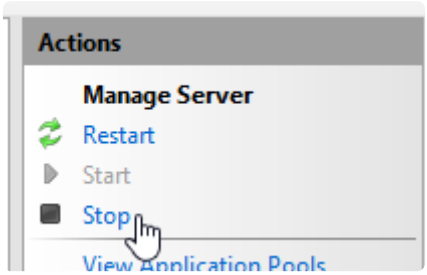


Note : even after stopping the `Alpana3App`, the application `code192.Alpana.Server.Web.Host.exe` may still be left running in memory.
To completely stop it, it may be required to stop IIS altogether (see below).

Stopping/Restarting IIS

To stop IIS completely, select the server name, and click *Stop*.
Wait that the animation progress stops.
Click *Start* to start IIS again.





Last modified: Dec 26, 2019

3.12.2. Backup

Introduction

Two elements of Alpana Server must be backed up :

- the Alpana Server database, named `AlpanaDb` by default
- the dashboards folder located in the installation folder, by default `C:\inetpub\alpana3\`, inside `\app\App_Data\Published\`

Backup

> ✻ To perform a backup, it is recommended to stop the websites first in order to ensure consistency between the database and the folders. (e.g. : a dashboard is uploaded between the database backup and the folder backup)
> If this is not desired, one should at least backup the folder **after** the database.

Database backup

Locate the Alpana Server database, named `AlpanaDb` by default.
Use standard Microsoft SQL procedure to perform a full backup of the Alpana Server database.

> ✻ The Alpana Server database is created by default with recovery mode *Simple*.
> If you wish for a different setting, you are free to change it.

Folder backup

Locate the installation folder, by default `C:\inetpub\alpana3\`.
Copy and compress the `\app\App_Data\Published\` folder.

Restore

> ✻ Before deleting any item, make sure you make a backup first.

> ! Restoring requires to stop the websites, so this will incur downtime to your users.

Stop the websites

Open the IIS Management Console with : Windows > Run… `inetmgr`.
Select the two Alpana Server websites and click *Stop* for each of them.

Restore the folder

Locate the installation folder, by default `C:\inetpub\alpana3\`.

Replace the `\app\App_Data\Published\` folder by the backup.

Restore the database

Use standard Microsoft SQL procedure to perform a full backup of the Alpana Server database.

> ! Use the same database name as previously on this installation.

Start the websites

Open the IIS Management Console with : Windows > Run… `inetmgr`.
Select the two Alpana Server websites and click *Start* for each of them.

Last modified: Jul 11, 2019

3.12.3. Migration

Purpose

Case 1 : Development → Staging → Production

You have several environments for your development cycle.
When development has been tested on one environment, you want to migrate the content (dashboards, users, etc) towards the Production environment.

Case 2 : Moving to another machine

You need to move your Alpana Server installation to a different machine (e.g. for architecture reasons).

> In this case, it is possible to move the license by de-activating it on the first machine and using it later on the target machine.
> Check with your support what is the most appropriate and see License Deactivation.

Migration Procedure

To migrate from *MachineX* to *MachineY*, follow the below procedure :

1. Alpana Server must be installed on *MachineY*. If you are using a separate license, then the *MachineY* license must be activated too (though this can be done at a later time)
2. Perform a backup on *MachineX*
3. Perform a restore on *MachineY*

Migrated content

Note : all content from *MachineX* will be migrated :

- Users and their passwords
- Dashboards, Widgets, Data Sources
- Roles, Permissions, …
- Server Settings, …

Possible Issues

When migrating, some Dashboards may become unable to access their data.
This commonly happens when data is not accessible from the *MachineY* using the same Connection.
Example : database is not accessible on the new network.
The fix is then to open the Dashboard and edit the Connection accordingly.

Error logs should help you find and fix the issue.

Last modified: Jun 17, 2019

3.13. Troubleshooting

This chapters explains how to find error logs and how to fix common issues.

Last modified: Jul 25, 2019

3.13.1. Installation Troubleshooting

Generating install logs

In some cases, the installation of Alpana Server may fail and roll back the changes.
To get more information, you can start the installer with error output by following the below procedure.

Start cmd

Run Windows command prompt “cmd” **as administrator** and navigate to the folder where the installer `.msi` file resides (for example here `code192.Alpana.Server.Setup_3010.msi`).

Start installer with log output

From the command prompt, start the following command :

```
msiexec /l*vx Alpana_install.log /i code192.Alpana.Server.Setup.msi
```

Replace `Alpana_install.log` with the desired log file name that will be created in the folder.
Replace `code192.Alpana.Server.Setup.msi` with the file name of your Alpana Server installer.

Get the Alpana Server log

Even if the installation fails and rolls back, the installer will usually leave behind an error log in the destination installation folder.
This `Logs.txt` file will contain additional useful information for support.

Get Windows Events

Look in Windows Event logs for an error happening during installation.
For example, sometimes SQL will log a useful error log there.

Send logs

When the installation has completed, you can send the zipped `Alpana_install.log` and `Logs.txt` files, and any Windows Event logs to your Alpana Support.
Don’t forget to include any relevant information about your installation (Windows/SQL/IIS versions ? What pre-requisites where configured and installed ? What configuration choices did you make during installation ? …)

Post-installation issues

Website startup issues

If the Front-End doesn't seem to start, try to open the URL of the Back-End directly in your web browser.

- If you see a login page on the back-end, then the Front-End should start, see error logs and contact Support.
- If instead you see a "Error HTTP 500" message or similar, then you are probably missing some pre-requisites on IIS.

When changing the configuration to IIS, you must repair the AspNet Core component :

1. update IIS configuration
2. go to Windows Control Panel and for each component of AspNet Core, perform a "Repair". Usually, this is done through right-click > Change, then select *Repair*.

License activation issues

When uploading the license file, if nothing seems to happen, here is a possible solution :
Check the host Audit Logs.
If a warning is present at that time, check the details.
If the details contain a message like "access denied" with a file location, then you may have permission issues in the IIS website folders.
Make sure that IIS and the Windows user running Alpana (`IIS APPPOOL\Alpana3App`) have permissions to create files in sub-folders of the installation folder.

Last modified: Jun 19, 2020

3.13.2. Dashboard error logs

What errors are logged

Published Dashboards and Widgets can generate errors when viewed.
Typically, this will show a *Data retrieval error* message in place of the Widget's visuals :

This means that the Widget was unable to get the required data.

These errors can have several explanations :

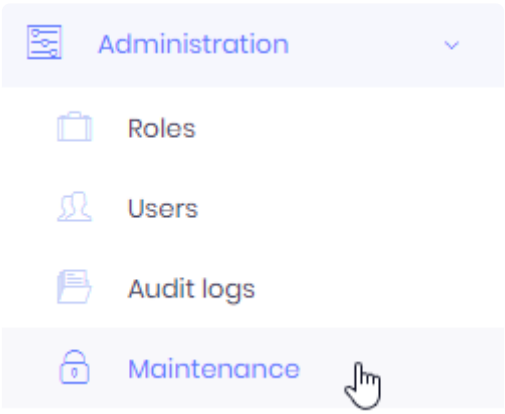
- data not available
- SQL error
- wrong query
- …

When viewed from Alpana Designer or Preview, you can access the full error message, see the relevant help page.

When viewed from Alpana Server, you can access the full error message from the web interface, or inside a file, see below :

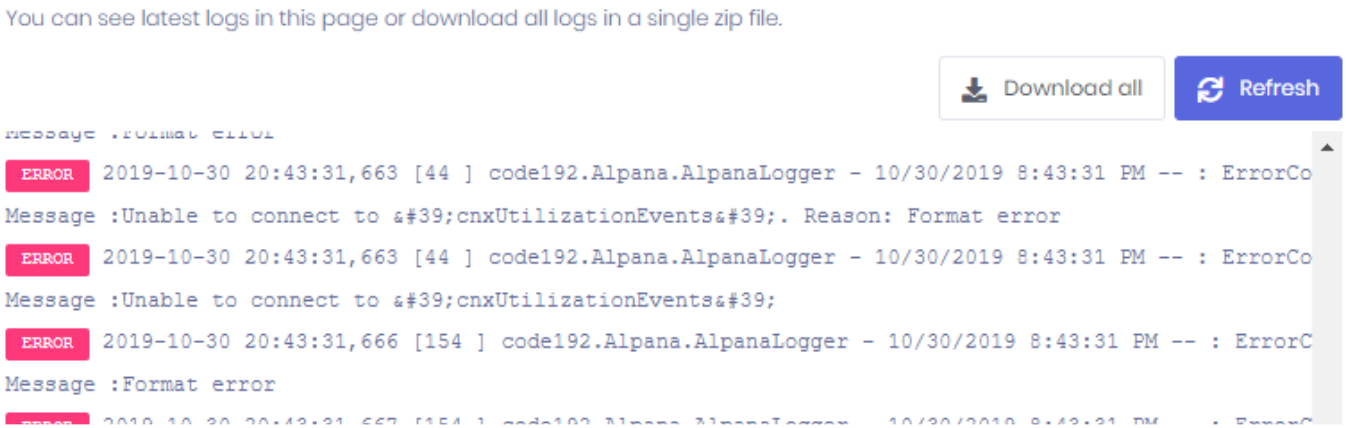
From the Host

From the Host, you can view the full error logs.
Dashboard logs are located in *Administration* > *Maintenance* under the tab *Website logs* :



The recent logs are presented in a full list with :

- log level : `ERROR` / `WARN` / `INFO`
- timestamp : time of the event
- source : where the log came from
- message : the full error message, which can span several rows



You can click *Refresh* to view the latest logs at the bottom of the page.

You can click *Download all* to export the full logs as text inside a compresses `.zip` file.

Alpana Server Error Log Files

If for some reason you are unable to access the error logs from the web interface of the Front-End (maybe you are unable to log in), you can find the logs as text files in the following folder :
(under your installation folder, by default `C:\inetpub\alpana3\`) `.\app\App_Data\Logs\`

Last modified: Oct 31, 2019

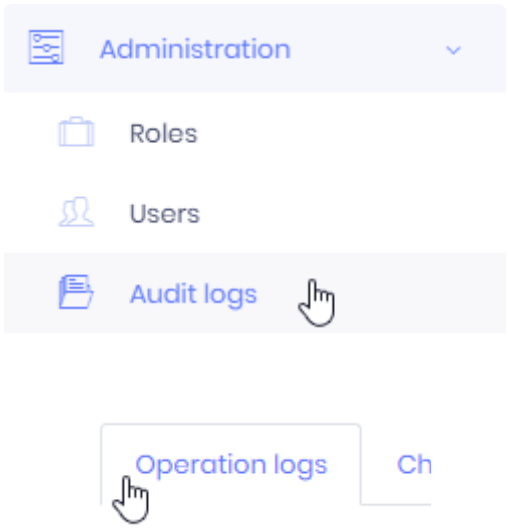
3.13.3. Server usage logs

From the Administration menu of Alpana Server, logs are available, both at the Host level and at the Tenant level.

Operation logs

The Operation Logs list all operations performed on the application : authenticating, changing settings, etc

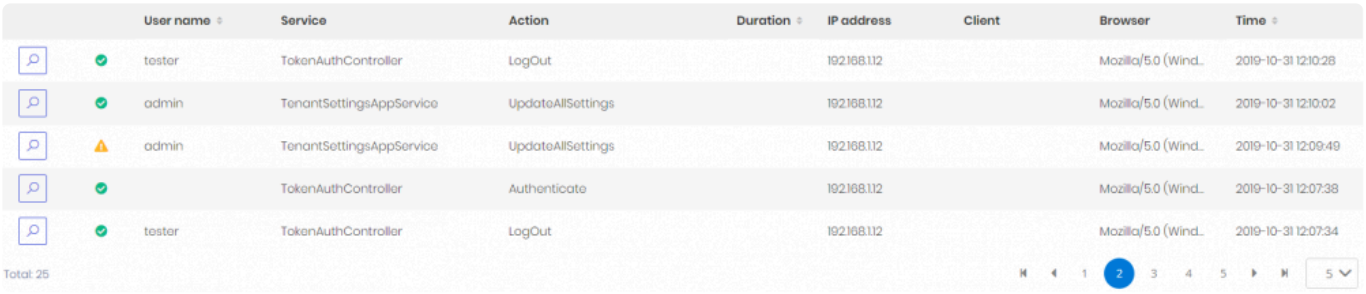
Operations Logs are located in *Administration* > *Audit logs* under the tab *Operation logs* :



Viewing Operation Logs

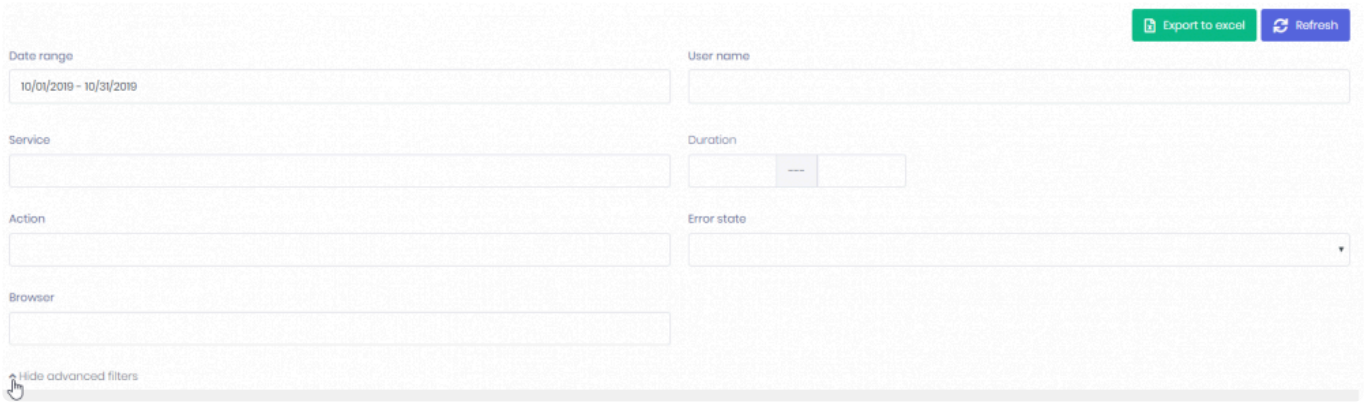
Logs are presented in a list with properties like :

- Log level : a symbol showing the level of importance of the log (information, warning, error)
- *User name* : user authenticated for the operation
- *Service* : which application service was responsible for handling the operation
- *Action* : which Action was performed or attempted
- *IP address* : the IP address of the request
- *Browser* : the User Agent of the request
- *Time* : when the operation happened



Filtering logs

These properties can be used for filtering the logs : select a *Date range* and/or another property to filter it (including *Advanced filters*), then click *Refresh* to refresh the view including the filters :



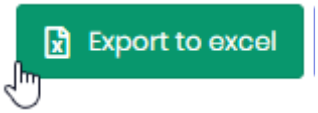
Viewing details

You can click the magnifying glass icon in front of a log entry to view details :

This opens a details window with the same information as in the log list, and with more details, like the action *Parameters* and *Error State* which can help understand what is happening.

Exporting logs

You can export the filtered logs to Excel by clicking *Export to Excel* :

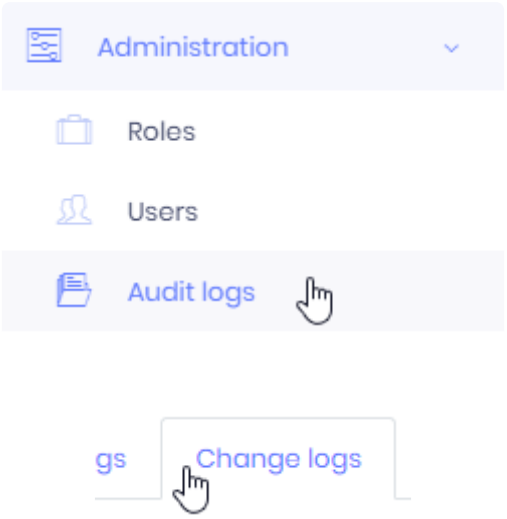


Change logs

Lists all changes to “entities” in the database :

- on Tenants, this is : published Dashboards, published Widgets, published DataSource, …
- on the Host, this is the Tenants

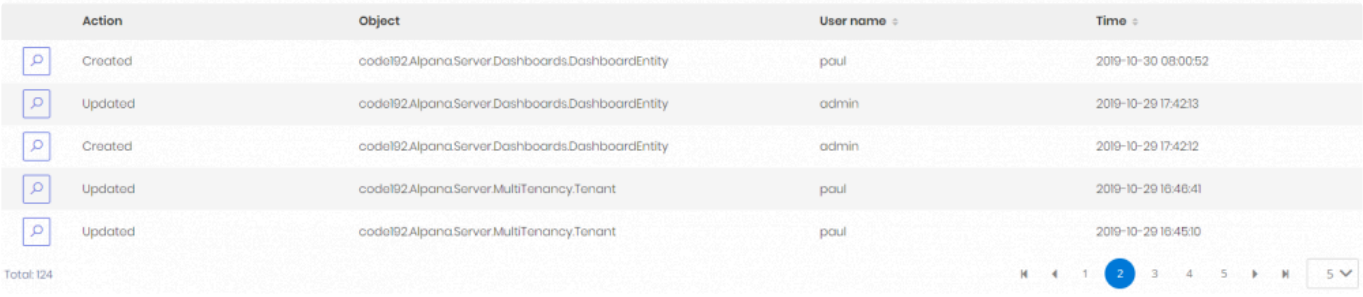
Change Logs are located in *Administration* > *Audit logs* under the tab *Change logs* :



Viewing Change Logs

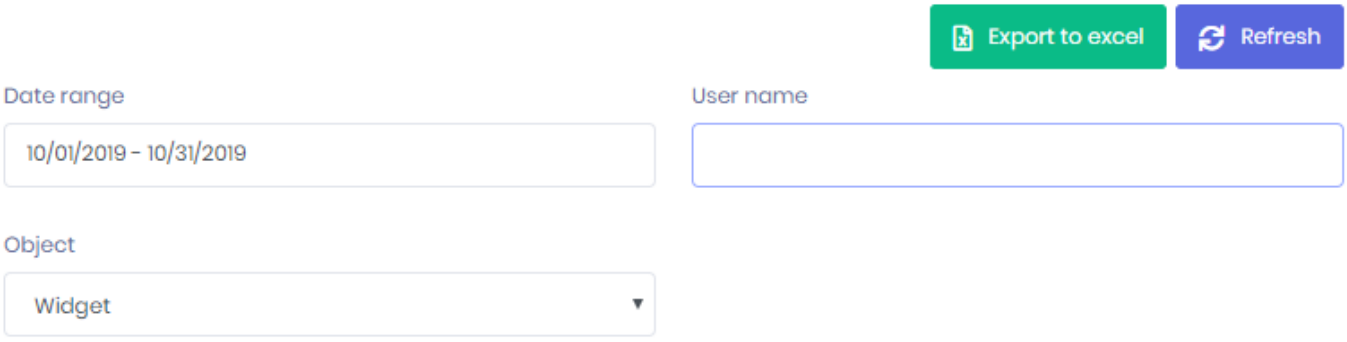
Logs are presented in a list with properties like :

- *Action* : which Action was performed or attempted on the entity
- *Object* : which entity was modified
- *User name* : user authenticated for the operation
- *Time* : when the operation happened



Filtering logs

These properties can be used for filtering the logs : select a *Date range* and/or another property to filter it, then click *Refresh* to refresh the view including the filters.



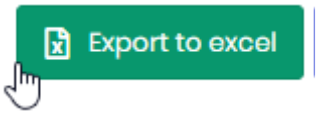
Viewing details

You can click the magnifying glass icon in front of a log entry to view details :

This opens a details window with more details about what changed.

Exporting logs

You can export the filtered logs to Excel by clicking *Export to Excel* :



Last modified: Oct 31, 2019

3.13.4. Website Troubleshooting

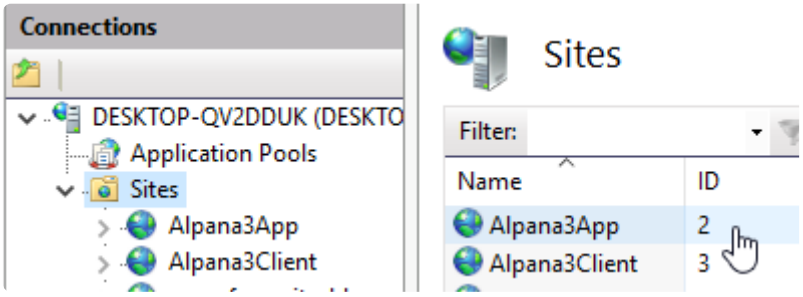
Below are some useful resources and some answers to common issues with reaching or starting the websites.

Useful logs

IIS Requests Logs

If you need to investigate your network configuration to see the content of HTTP requests, you can find IIS log files :

1. Open IIS Manager (run `inetmgr`)
2. Click on the *Sites* list to get the ID of the desired website. For example in this screenshot, the Alpana3App ID is 2 :



3. With Windows file explorer, browse the following folder (replace the **2** with your desired ID from step 2. above) :

```
C:\inetpub\logs\LogFiles\W3SVC2\
```

4. This folder contains logs as space-separated text files ordered by date, containing all requests.

Browser errors

Sometimes, the web browser console gives indications on the nature of the problem.

See your web browser documentation to open the console.
For Google Chrome, press F12 and click the “Console” tab.

Problem : Frontend website loading forever

You are trying to open the Frontend website, but you have a white page with a loading animation that never completes :

Possible explanation : Wrong Frontend address used

Symptom

Check the web browser console for an error like :

```
Access to XMLHttpRequest at 'http://xxxxx.com:123//AbpUserConfiguration/GetAl
l?d=1234567890' from origin 'http://yy.yy.yy.yy' has been blocked by CORS poli
cy: Response to preflight request doesn't pass access control check: No 'Acces
s-Control-Allow-Origin' header is present on the requested resource.
```

Check that you can reach the Backend from your same web browser on the same URL, and see a login page there.

Explanation

If so, a possible explanation is :
Your Backend at `http://xxxxx.com:123` is configured to not respond to queries from `http://yy.yy.yy.yy`.

For example, when you install Alpana Server and configure to use a domain name with `xxxxx.com:80` as Frontend and `xxxxx.com:123` as Backend, you won't be able to access the Frontend using its IP address `yy.yy.yy.yy`.
That's for security reasons : the Frontend will only be allowed to talk to the Backend if it calls from the pre-defined origin (the domain name `xxxxx.com:80`).
That is configured through CORS rule internally.

Solving

You can solve this case by redirecting requests in IIS.
For example, if you wish to be able to reach the Frontend from IP address `yy.yy.yy.yy`, you need to let IIS listen to requests to `http://yy.yy.yy.yy` and redirect them to the accepted domain name `xxxxx.com:80`.

See IIS documentation examples on how to do this.

Last modified: Oct 31, 2019

3.13.5. License troubleshooting

Alpana became “unlicensed”

You may see the following message when logging-in to a Tenant :
“*Alpana is currently unlicensed. Please contact your administrator.*”

There are several possible explanations for this.

License is managed at the Host level for all Tenants, so please login to the Host to get more information and solve the issue.

Computer not recognized

The license system fingerprints the computer where Alpana Server is installed.
When the license system doesn’t recognize the fingerprint, the license system believes that you illegitimately moved the license to a different computer.
The license then becomes invalid, and you get the message :
“*The system could not be identified as the one to which this license was issued.*”

Why this happens

In order to prevent this from happening again, you need to understand why this happens, and fix the relevant I.T. issue.

The license fingerprint is based on several hardware identifiers (CPU ID, BIOS, MAC address, etc).
So when changing these hardware components, the fingerprint changes.

When using virtualization, some of these identifiers are inherited from the virtualization host. For example, when the VM is moved to a different host, the CPU ID changes.
It is also known that some Hyper-V default settings will allocate the MAC address

How to prevent

If you know in advance that such a change will happen (for example, if you are updating the hardware, or moving the VM to a different host), you should Deactivate the license in advance.

If you identify that the MAC address is allocated dynamically, you should change this setting.

If you need help for another case, you may contact Support for help.

How to fix

To fix it, you will need to contact Support.

Once you have identified the cause with Support (see above hints), they may give you a new Activation to let you activate your license again.

In case the new Activation doesn’t work, Support may ask you to remove all license information. When Support asks it, **and only then**, you may use the *Clear license data* button.

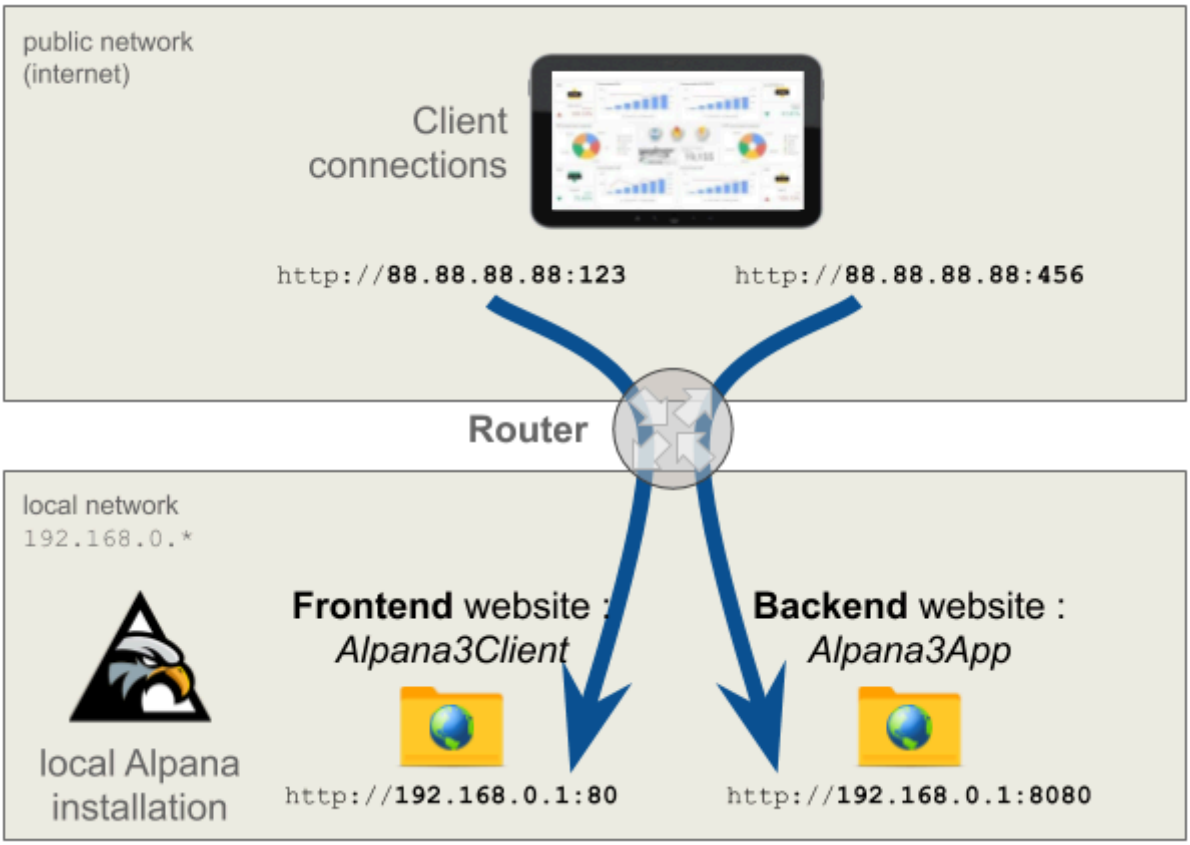
> ! Warning : the *Clear license data* button will completely **remove your license without warning**.

Last modified: Jan 28, 2020

3.13.6. Accessing behind a router

Scenario

In this scenario, Alpana Server is installed in the local network on a machine with IP address `192.168.0.1`, and you wish to reach it from a public network through a router with external IP address `88.88.88.88`.



Configuring

> Note : for this configuration to work, the external IP address `88.88.88.88` must be reachable from the internal network `192.168.0.1`. This is the case with an Internet router for example.

Step 1 : Install locally

First, let's make a simple local installation, in order to define a local address where Alpana Server will respond to the router.

During installation of Alpana Server, configure with the local address `192.168.0.1` and some ports for the Backend and Frontend.

In our example, we configured :

- port `80` for the frontend (`http://192.168.0.1:80`)
- port `8080` for the backend (`http://192.168.0.1:8080`).

Step 2 : Configure router

Define which router ports you want to open for talking to the FrontEnd and BackEnd sites from the outside network.
In our example, we chose:

- port `123` for the FrontEnd. So incoming requests on `http://88.88.88.88:123` are redirected to `http://192.168.0.1:80`
- port `456` for the BackEnd. So incoming requests on `http://88.88.88.88:456` are redirected to `http://192.168.0.1:8080`

Step 3 : Configure authorizations for local access

Let's configure Alpana Server to accept requests from the local addresses that will be requested by the router.

In the CORS whitelist, create exceptions for :

- the local address and port for the FrontEnd : `http://192.168.0.1:80`
- the local address and port for the BackEnd : `http://192.168.0.1:8080`

Step 4 : Re-configure with public address

In order to reconfigure the address where Alpana Server is responding, you will have to uninstall/ reinstall.
This is fast and doesn't delete your configuration and dashboards.

Un-install

Un-install Alpana Server.

Install with public address

Install again Alpana Server.
During installation of Alpana Server, configure with the public address `88.88.88.88` and the correct router ports for the Backend and Frontend.
In our example, we configured :

- frontend : `http://88.88.88.88:123`

- backend : `http://88.88.88.88:456`

Step 5 : Configure to respond locally

Finally, let's configure IIs to accept the queries to local addresses from the router.

Open the IIS Management Console (run `inetmgr`).
For both websites `Alpana3App` and `Alpana3Client`, right-click to *Edit Bindings*, add a new binding and define the local address (`192.168.0.1`) and ports (`80` and `8080`)

> Now Alpana should work on the public address `http://88.88.88.88:123`

Last modified: Jun 10, 2020

4. Alpana Mobile

- Installation
- Usage

Last modified: Apr 12, 2019

4.1. Installation

- How to get Alpana Mobile
- Setup

Last modified: Apr 12, 2019

4.1.1. How to get Alpana Mobile

You can download Alpana mobile through the usual providers:

- Google Play Store for Android devices
- App Store for Apple devices

Alpana Mobile is part of the Alpana software solution and as such is free to download.

Last modified: Jul 01, 2019

4.1.2. Setup

Address

At first launch you will be prompted to enter Alpana Sever address.

The format of your input should exactly match the address set for your server. For instance, if your server has been set to ignore the “www”, you should ignore them in your input.

> It is also important to start with “http://” (“https//” if your server has a security certificate).

Some examples:

- http://www.company.com
- https://company.com
- http://subdomain.company.com

Your server address could also be an simple I.P address “http://123.150.XX.XX”

If your server has been set for Multitenancy with tenancy sub-domains, you can take it into account in your input (“https://tenantX.company.com” , “https://tenantY.company.com”…).
In that case, the application will recognize the subdomain and set it as tenancy name for you.

Port

The port number to input is the one that has been set for your backend Alpana IIS server.

Last modified: Apr 12, 2019

4.2. Usage

- Login
- Tenancy
- Dashboard Page

Last modified: Apr 12, 2019

4.2.1. Login

Login screen

The login screen allows you to enter your login credentials. From there, you can also change the tenancy name and reset the server address.

Login behavior

After your fist connection, the application will securely save your credentials and will automatically log you into the server at application startup.
However, if you haven’t logged in the mobile application or the desktop web application for an extended period of time, your session can expire. When that happen, the mobile application will invite you to log out. You can then log in again and your session and your rights will be renewed.

> Note that you cannot login to the application without a tenancy name set. See Tenancy

You can, through the side menu or the settings menu, log out. Doing so will end your session and navigate you to the login screen. The log out action will clear your saved password but tenancy name and server address will remain saved.

Last modified: Apr 12, 2019

4.2.2. Tenancy

Single tenancy

In the case of single tenancy, the application will try to set “Default” as tenancy name at first launch as is it the default tenant in the Alpana solution.
If your company changed the “Default” name for something else you will be prompted to change it. You can tap the change tenant box on the login screen and input the tenancy name that your company set. Provided that the tenancy name is valid, it will be set as tenant and saved.

Multitenancy

In the case of multitenancy, the application will set your tenancy name according to the sub domain name provided in the server address. You can, however, change the tenancy name at any time via the change tenant box on the login screen.

> ✻ Please note that the mobile application does not allow to connect without tenancy name or “Host” mode.

Last modified: Apr 12, 2019

4.2.3. Dashboard Page

Once logged in, the application navigate you to the dashboard page. Here, you can browse through all the dashboard saved into your server.

This page is divided into five different tabs in appearing order:

- Favorites
- Recent
- Categories
- Public
- All

The dashboard list for each page is fetched once when the page fist appear. You can however refresh them manually by pulling down the list. This features doesn’t apply for subcategories navigation. If you want to refresh a dashboard list seen in a subcategory, you can go back then reopen said subcategory to refresh its content.

Last modified: Apr 12, 2019

5. Alpana Update File Dependency

Alpana dashboards rely on *dependencies* when they use an external file.
Sometimes, you need to update one such dependency. For example : when you need to update the content of an Excel file on a published dashboard without having to re-publish the dashboard.

You have two options :

- write an external application that makes use of Alpana Server API to push the updated file
- use the utility called “Alpana Update File Dependency” which implements this for common cases

Last modified: Aug 12, 2019

5.1. Installation

Installation of Alpana Update File Dependency is light on resource and pretty straightforward.
Proceed to the next chapters to complete the installation.

Last modified: Nov 04, 2019

5.1.1. Pre-Requisistes

This section explains the system requirements to run Alpana Update File Dependency.

Software Requirements

The following minimum software requirements are necessary to run Alpana Update File Dependency :

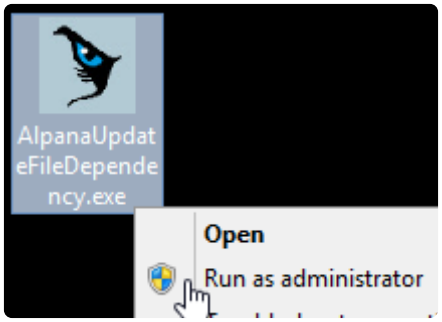
- .NET Framework 4.8 can be found here then download the “Runtime”. Alternately, an offline installer exists on the same page . **Installing this component requires to restart the machine**.
- Alpana Server version 3.3.x or greater is required
- Alpana Advanced license, to support API calls

Last modified: May 18, 2020

5.1.2. Installing

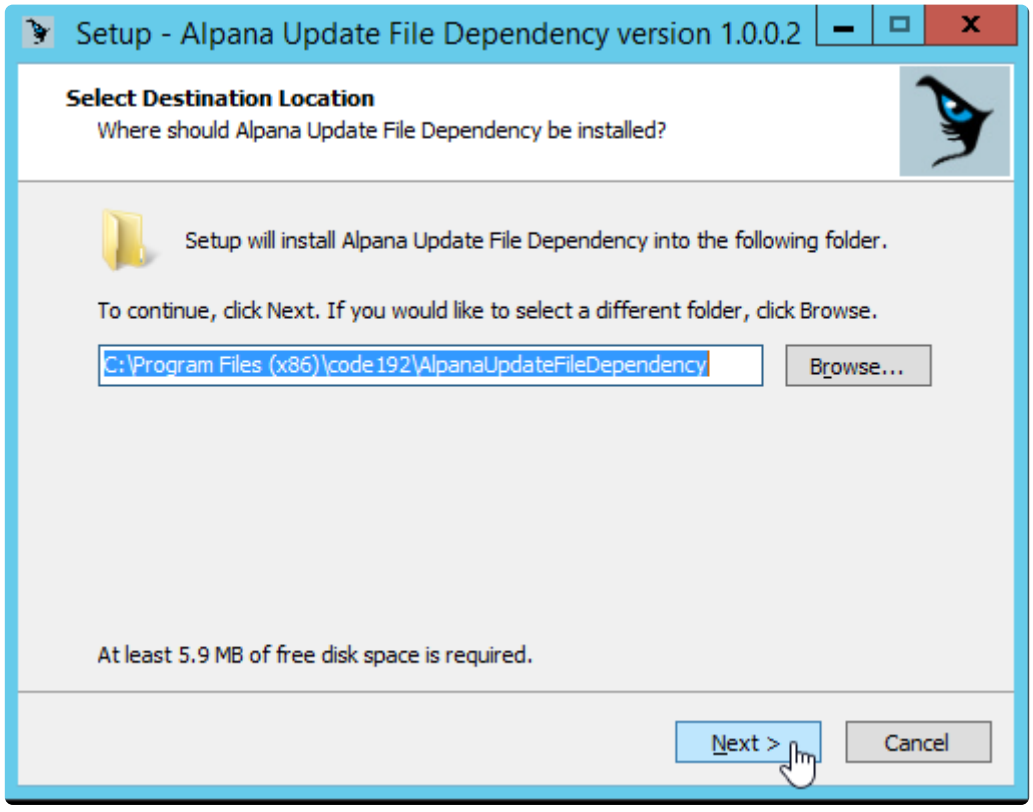
Run installer

Run the installer as Administrator :



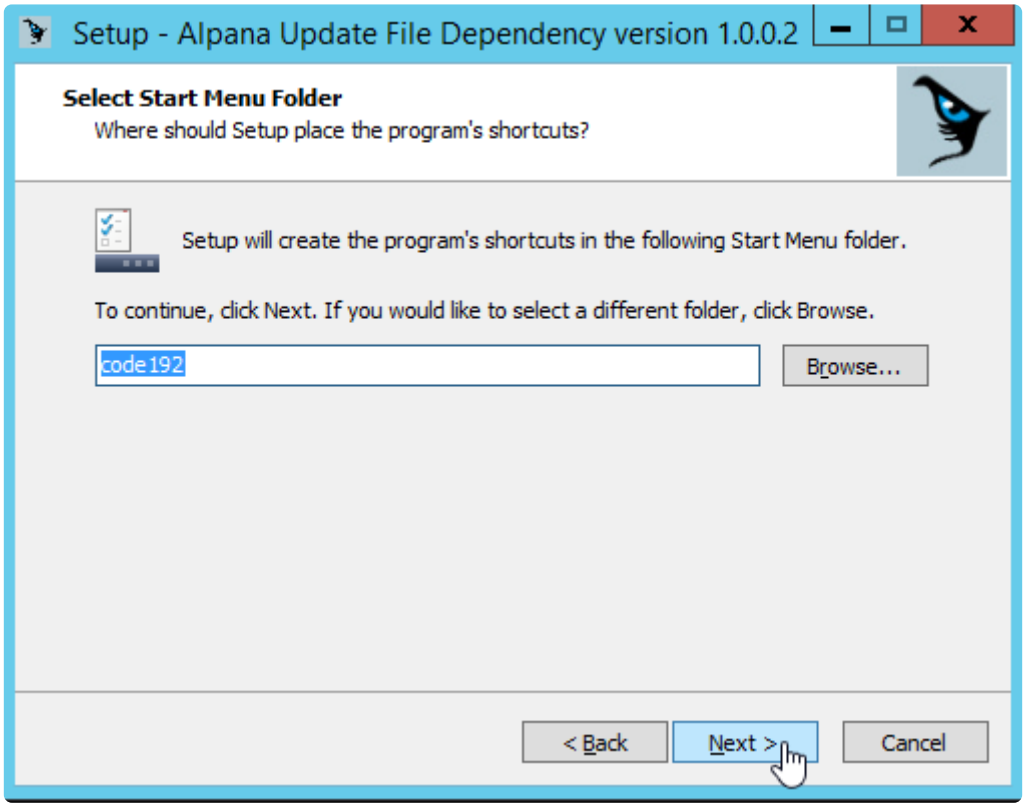
Select folder

Select the installation folder and click *Next* :



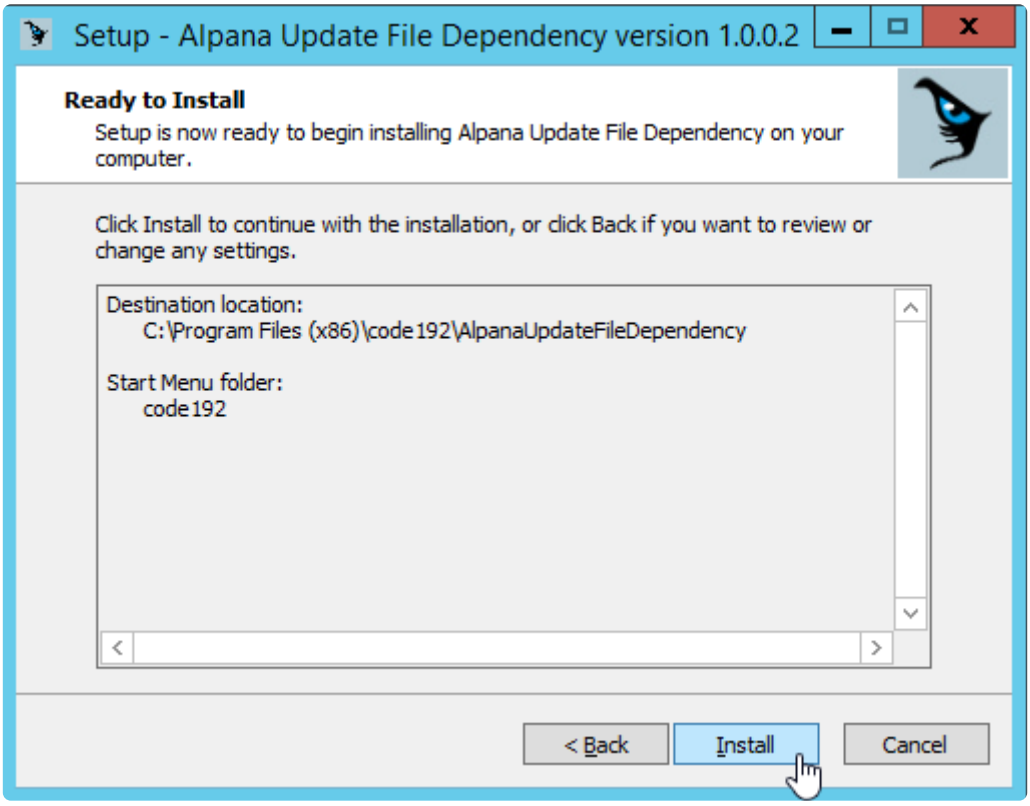
Start Menu folder

Select start menu folder and click *Next* :



Complete installation

Review options, click *Install* :



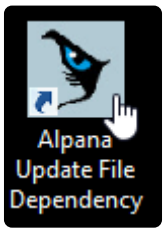
Wait for completion and click *Finish* :

Last modified: Nov 04, 2019

5.2. Running the application

The application runs as a Windows service called “AlpanaDependencyUpdate”.
This allows to run jobs even when no user is logged in.

To start configuring Jobs and start/stop the Service, run the “Alpana Update File Dependency” shortcut :



When finished configuring, you can close the application, or even the Windows session, as the Jobs will be run by the Windows service.

When starting for the first time, the application requires to start the service.

Last modified: Nov 04, 2019

5.3. Settings

Opening Settings

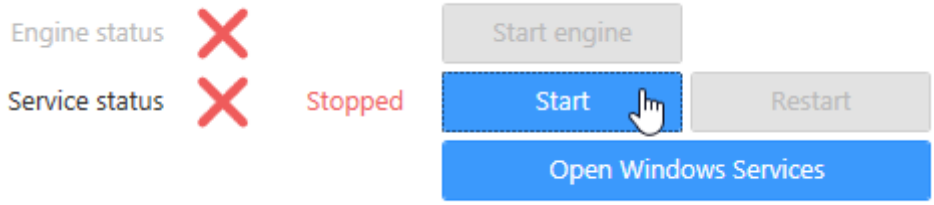
To manage application settings, click the tab *Settings* :

Starting the Service and the Engine

When you start the application for the first time, you need to start the Windows service, and the update engine.
The *Settings* window also allows to see the status of the service and engine.

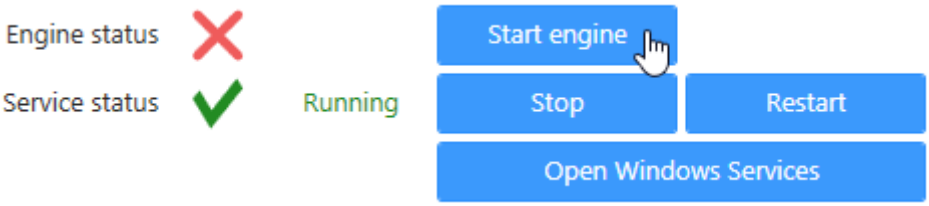
Start Service

Click *Start Service* to start the Windows service :



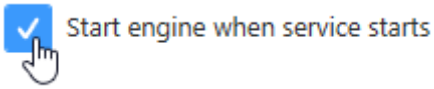
Start Engine

Click *Start Engine* to start the Engine :



Start Automatically

You can also configure the Engine to start automatically when the Service starts



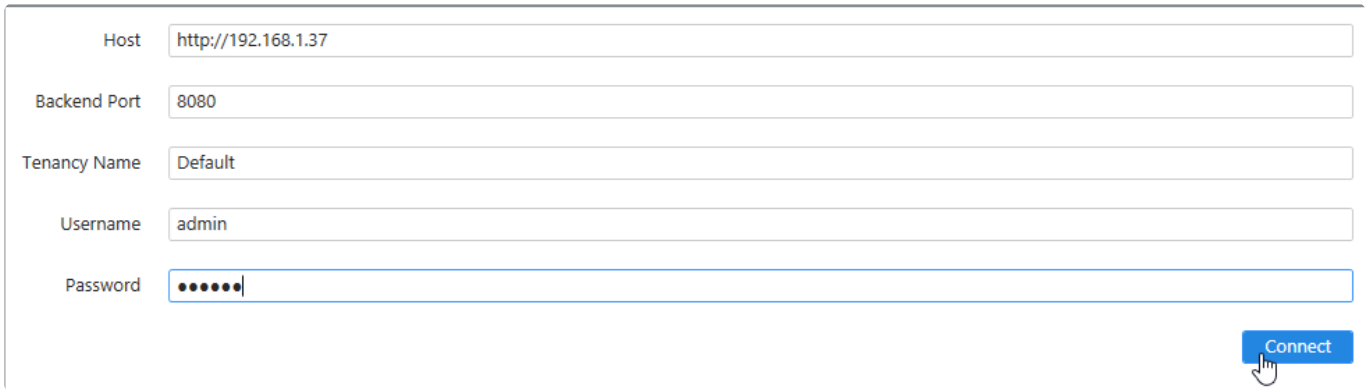
Last modified: Aug 12, 2019

5.4. Connecting To Alpana

To log in or log out to your Alpana Server, click the tab *Configuration* :

Log in

Fill in the login information and click *Connect* :



These credentials will be remembered by the application and will be used to run the Jobs.

Host

URL of the Alpana Server with which you want to communicate.
Include “http” or “https” as relevant.

> A license “Alpana Advanced” must be active on the server, and the Feature “API” must be activated.

Backend Port

Network port to the **Backend** website of the Alpana Server.

Tenancy Name

Name of the Tenant where your dashboards are published.

> The Feature “API” must be activated on the Tenant.

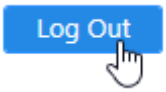
Username and Password

Login information for a User of the Alpana Server with the necessary permissions.

The User must have API permissions, and permissions to the dashboards that will be targeted by the application.

Log out

To log out, simply click *Log Out* on the top right :



Last modified: Nov 04, 2019

5.5. Jobs

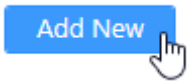
The application will run Jobs that update the file dependencies of Dashboards, based on conditions.

To access the Jobs configuration, click the tab *Configuration* and log in if necessary :

Last modified: Nov 04, 2019

5.5.1. Creating a Job

To create a Job, click the *Add New* button :



This creates an un-configured Job :

Last modified: Nov 04, 2019

5.5.2. Configuring a Job

When a new Job is created, it is not configured.
You can also configure an existing job.

To configure a Job, configure each property :

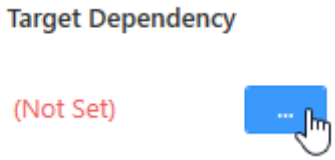
- *Target Dependency*
- *Source File*
- *Update Mode*

Last modified: Nov 04, 2019

5.5.2.1. Target Dependency

Selecting the Target Dependency means selecting which file you want to update from which existing Dashboard.

To configure the Target Dependency, click the "…" button :

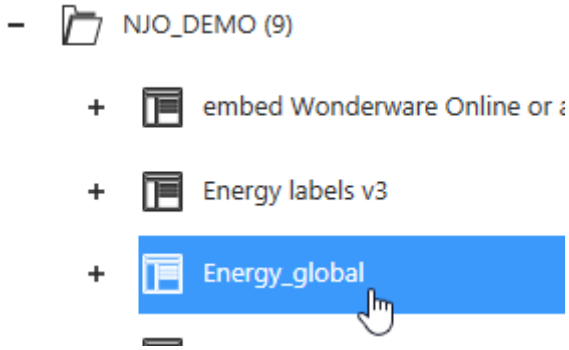


Dashboard or Widget

You can select a published Dashboard or Widget by clicking at the top of the *Select Target Dependency* window :

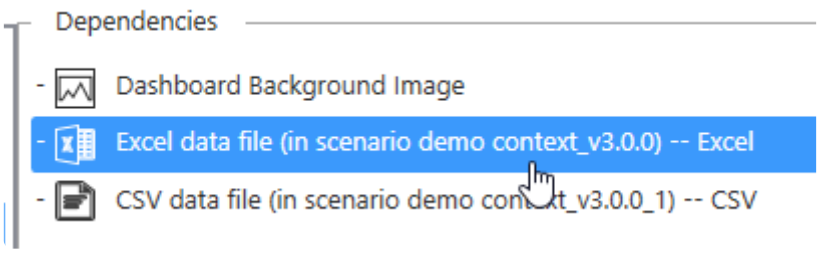
Browse Dashboard

Browse Categories and Dashboards in the left pane :



Browse Dependency

After clicking on a Dashboard on the left pane, the right pane lists all its depedencies :



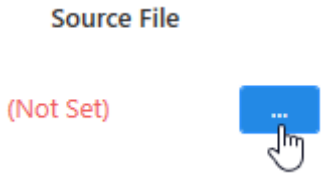
Click *OK* to validate :

Last modified: Nov 04, 2019

5.5.2.2. Source File

Selecting the Source file means selecting the path where you want to fetch the new file that will replace the current dependency.

To configure the Source File, click the “…” button and browse for a desired file :



> ! Important : If you plan to schedule automatic updates, choose well this file path so that it can be accessed by the Windows user running the Service.
> See Troubleshooting for more information.

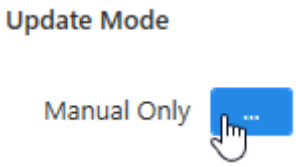
Last modified: Aug 12, 2019

5.5.2.3. Update Mode

Selecting the update Mode means choosing when you want the update to happen.

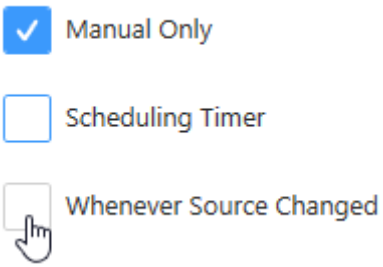
> If you are updating a file data dependency (Excel, CSV), then since Alpana v3.2 you no longer need to configure a Buffer refresh on the Dashboard Connection.
> By default, each update will automatically for the buffer to refresh.

To configure the Update Mode, click the "…" button :



Select mode :

These modes can be combined and selected by checking the corresponding boxes :



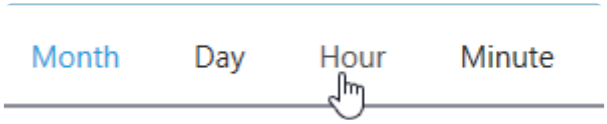
Manual Only

With *Manual Only*, the update will only happen when a user will trigger it manually inside the Alpana Update File Dependency application.
This mode doesn't require additional configuration.

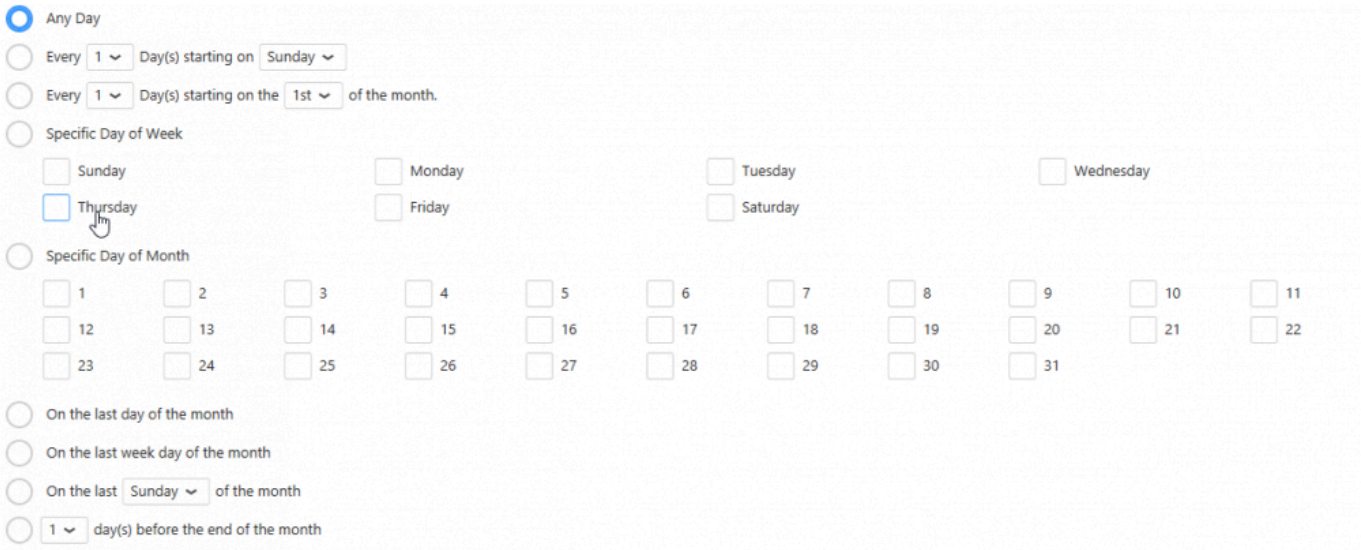
Scheduling Timer

With *Scheduling Timer*, the update will happen at selected time intervals.

Select the field for the time interval at the top of the window :



For the desired field, configure more precisely the desired schedule :



The result is like for a CRON expression that will combine all fields.

Whenever the Source Changed

With *Whenever the Source Changed*, the service will watch the file for any change and will trigger the update.
This mode doesn't require additional configuration.

Last modified: Jun 09, 2020

5.5.3. Running jobs

Run Automatically

Jobs can be run automatically when Enabled.

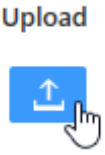
> Jobs are not Enabled by default.

To Enable a Job, check the *Enabled* box :

To Disable a Job, uncheck the *Enabled* box :

Run Manually

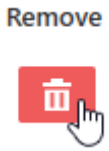
To manually force the update now, click the *Upload* button :



Last modified: Aug 12, 2019

5.5.4. Deleting a Job

To delete a Job, click the *Remove* button in front of it :



> ! This will remove the Job permanently.
> If instead you want to disable it temporarily, just Disable it.

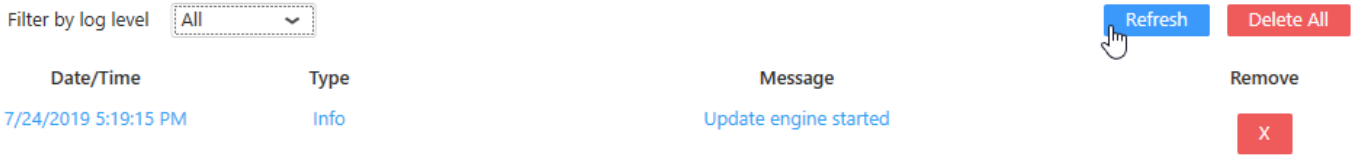
Last modified: Nov 04, 2019

5.6. Troubleshooting

Logs

Application logs are available in the tab *Logs* :

They contain the history of run jobs and useful messages upon failure.



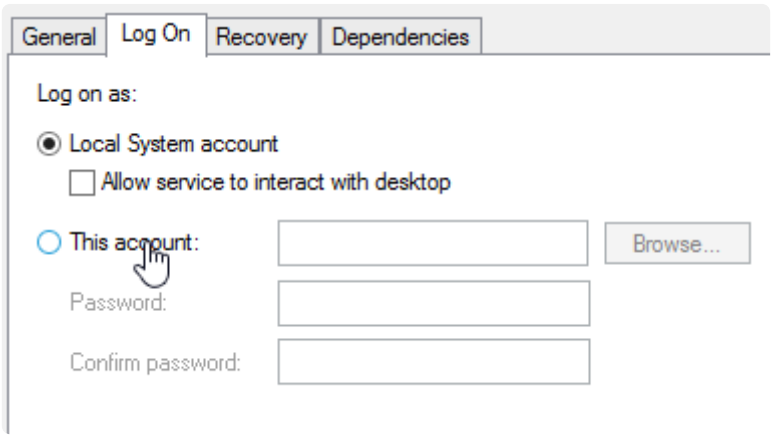
Common Issues

Buffer refresh settings

Since Alpana v3.2, it is no longer necessary to set the Connection with Buffer refresh settings to see th change when a file is updated.

I/O Errors and file permissions

The Windows Service must have access to the source files.
By default, the Service is run by a Local System user.
Check if this user has correct permissions, or change what Windows account is used to run the Service :



Last modified: Oct 15, 2019

6. Alpana Real Time Data

> ! Important : this tool is provided as-is, with no warranty.

The “Alpana Real Time Data” is an external tool used for demonstrating easily the possibilities of the Alpana API for sending data in real time to Dashboards.

The application reads data periodically from a OPC addresses or CSV files and pushes the result into Widgets.

> ✻ To use it, you need to create a Widget that is bound to Real-Time Data instead of a Data Source.
> To configure a Widget for Real-Time Data, see the corresponding chapter.

Last modified: Jun 19, 2020

6.1. Installation

Installation of Alpana Update File Dependency is light on resource and pretty straightforward.
Proceed to the next chapters to complete the installation.

Last modified: Nov 06, 2019

6.1.1. Pre-Requisistes

This section explains the system requirements to run Alpana Real Time Data.

Software Requirements

The following minimum software requirements are necessary to run Alpana Real Time Data :

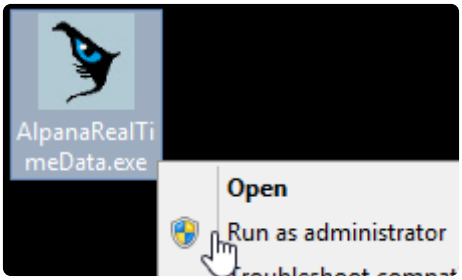
- .NET Framework 4.8 can be found here then download the “Runtime”. Alternately, an offline installer exists on the same page . **Installing this component requires to restart the machine**.
- Alpana Server version 3.3.x or greater is required
- Alpana Advanced license, to support API calls

Last modified: May 18, 2020

6.1.2. Installing

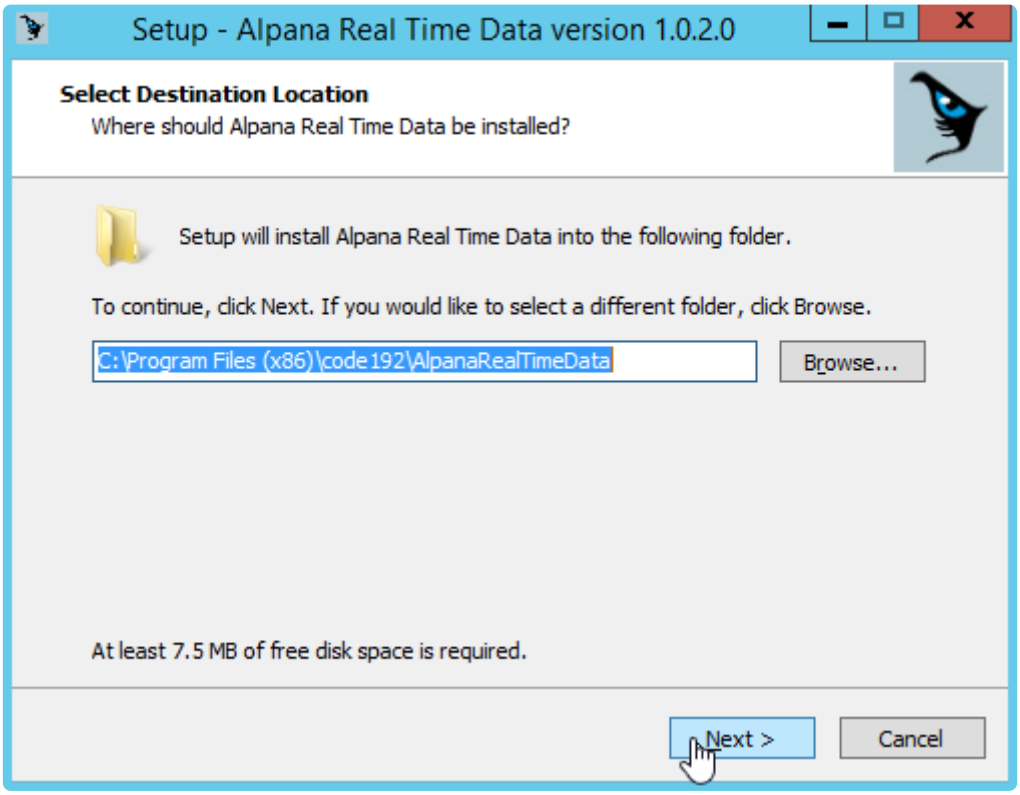
Run installer

Run the installer as Administrator :



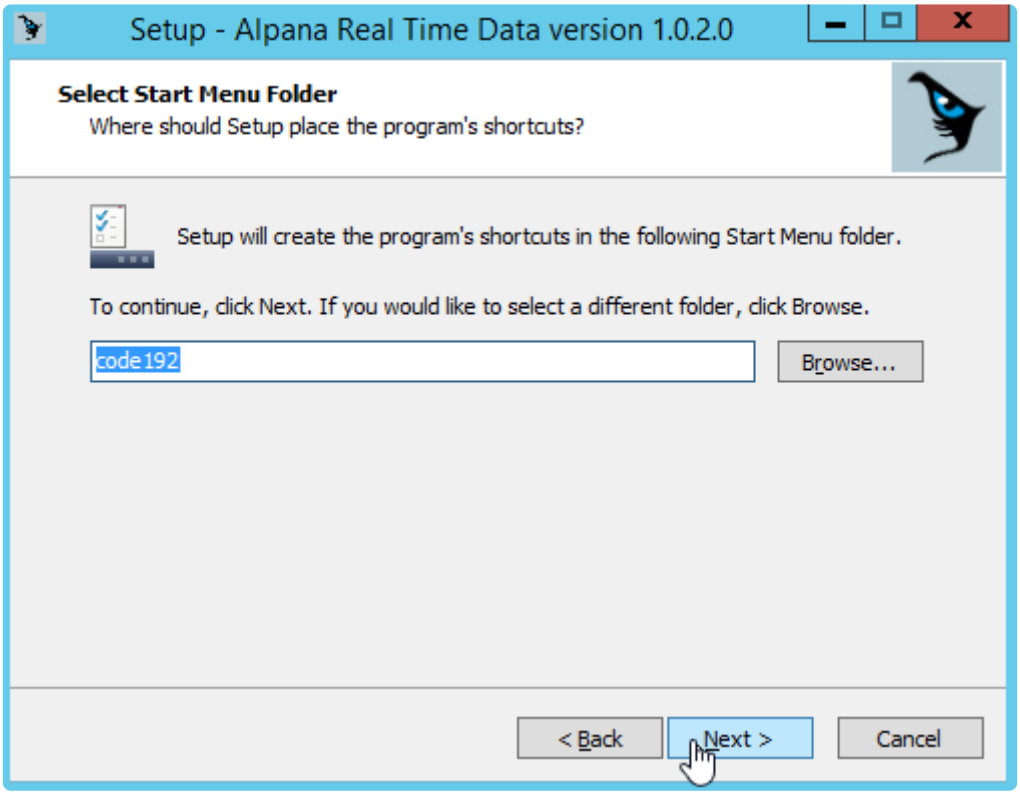
Select folder

Select the installation folder and click *Next* :



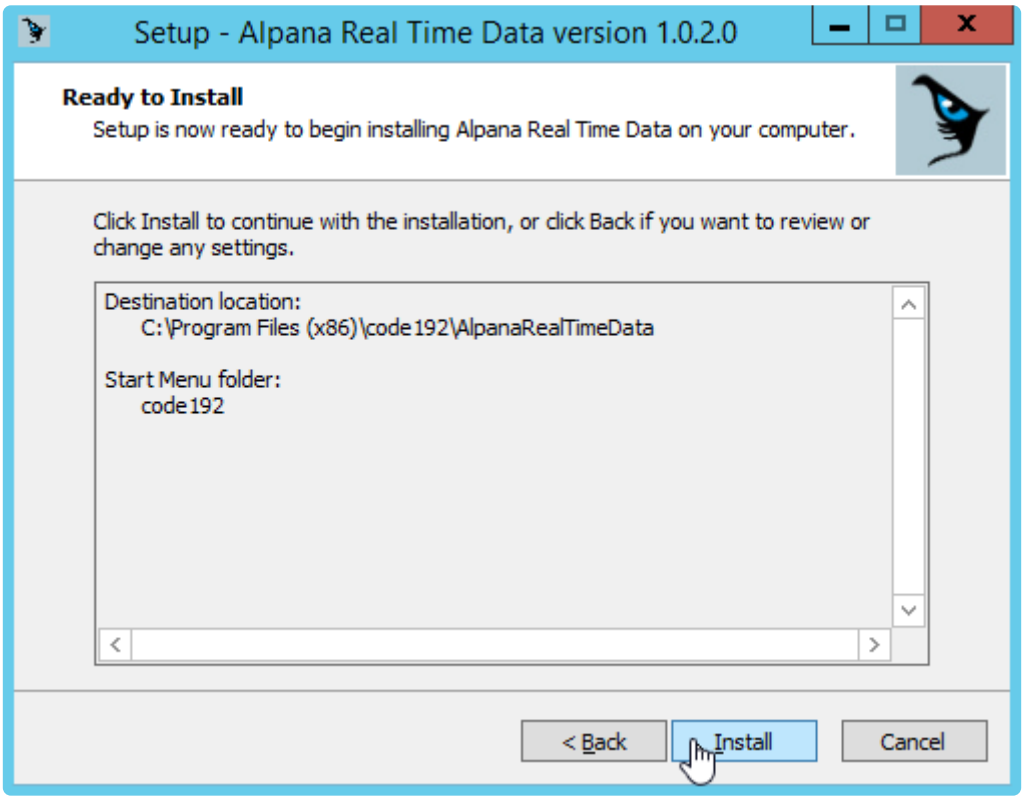
Start Menu folder

Select start menu folder and click *Next* :



Complete installation

Review options, click *Install* :



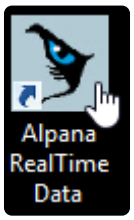
Wait for completion and click *Finish* :

Last modified: Nov 06, 2019

6.2. Running the application

The application runs as a Windows service called “AlpanaRealTimeData”.
This allows to run jobs even when no user is logged in.

To start configuring Jobs and start/stop the Service, run the “Alpana RealTime Data” shortcut :



When finished configuring, you can close the application, or even the Windows session, as the Jobs will be run by the Windows service.

When starting for the first time, the application requires to start the service.

Last modified: Nov 06, 2019

6.3. Settings

Opening Settings

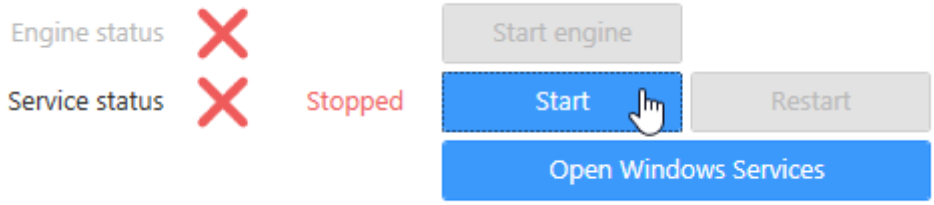
To manage application settings, click the tab *Settings* :

Starting the Service and the Engine

When you start the application for the first time, you need to start the Windows service, and the update engine.
The *Settings* window also allows to see the status of the service and engine.

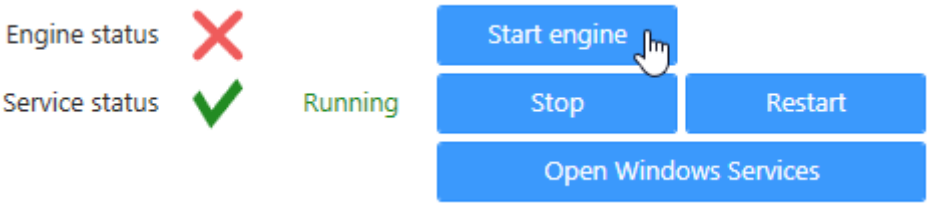
Start Service

Click *Start Service* to start the Windows service :



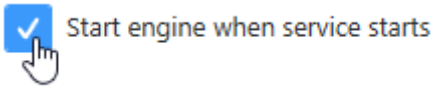
Start Engine

Click *Start Engine* to start the Engine :



Start Automatically

You can also configure the Engine to start automatically when the Service starts



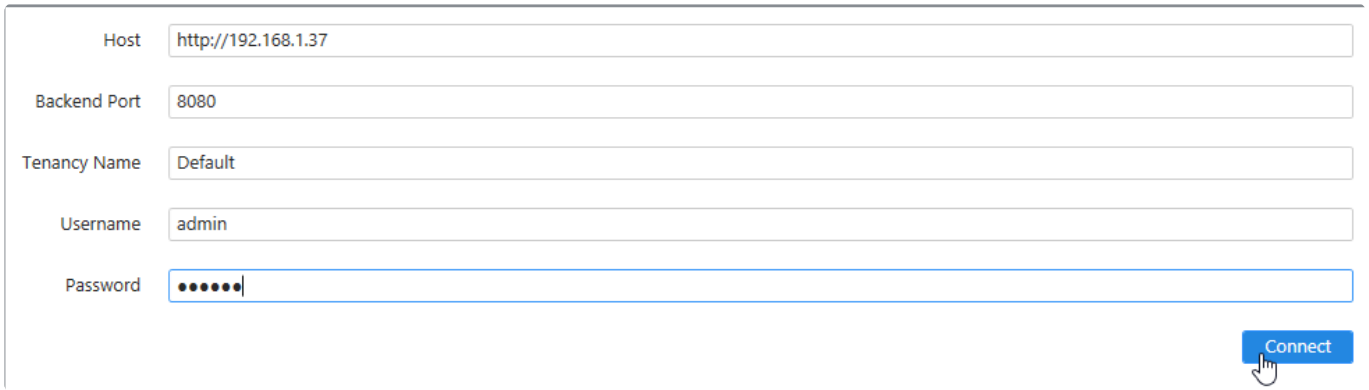
Last modified: Nov 06, 2019

6.4. Connecting To Alpana

To log in or log out to your Alpana Server, click the tab *Configuration* :

Log in

Fill in the login information and click *Connect* :



These credentials will be remembered by the application and will be used to run the Jobs.

Host

URL of the Alpana Server with which you want to communicate.
Include “http” or “https” as relevant.

> A license “Alpana Advanced” must be active on the server, and the Feature “API” must be activated.

Backend Port

Network port to the **Backend** website of the Alpana Server.

Tenancy Name

Name of the Tenant where your dashboards are published.

> The Feature “API” must be activated on the Tenant.

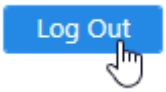
Username and Password

Login information for a User of the Alpana Server with the necessary permissions.

The User must have API permissions, and permissions to the dashboards that will be targeted by the application.

Log out

To log out, simply click *Log Out* on the top right :



Last modified: Nov 06, 2019

6.5. Jobs

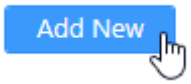
The application will run Jobs that push data to Widgets in real time.

To access the Jobs configuration, click the tab *Configuration* and log in if necessary :

Last modified: Nov 06, 2019

6.5.1. Creating a Job

To create a Job, click the *Add New* button :



This creates an un-configured Job :

Last modified: Nov 06, 2019

6.5.2. Configuring a Job

When a new Job is created, it is not configured.
You can also configure an existing job.

To configure a Job, configure each property :

- *Target Widget* : the Widget that will receive the live data
- *Data Source* : the origin of the data that will be pushed to the Widget

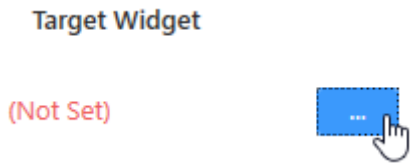
Last modified: Nov 06, 2019

6.5.2.1. Target Widget

Selecting the Target Widget means selecting which Widget will receive the data and display it live.

> ! The selected Widget must be configured for Real-Time Data, see the corresponding chapter.

To configure the Target Widget, click the "…" button :

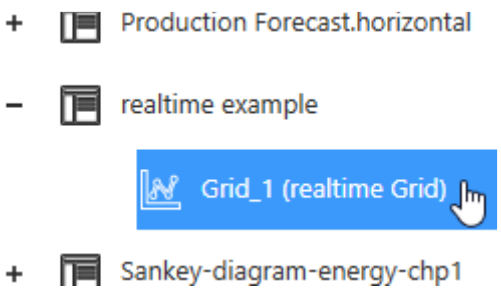


Dashboard or Widget

You can select a published Dashboard or Widget by clicking at the top of the *Select Target Widget* window :

Browse Dashboard

Browse Categories and Dashboards. Unfold the Dashboard node to select the desired Widget :



Click *OK* to validate :

Last modified: Nov 06, 2019

6.5.2.2. Data Source

Selecting the Data Source file means selecting the origin of the data that will be fetched and then pushed to the Target Widget.

To configure the Data Source, click the "…" button :

Select type

Data can come from two types of sources :

- *CSV* : a text file that will be read periodically and sequentially
- *OPC* : a OPC server that will be subscribed

CSV configuration

The tool is configured to fetch data periodically.
The CSV file will be opened each time the tool needs new data.
Each time the file is opened, the next row is fetched. If the last row has been reached, the first row is fetched again, and so on.

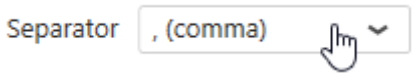
> ! Important : Choose well this file path so that it can be accessed by the Windows user running the Service.

Source file

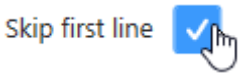
Click "…" to browse for the CSV file :

File format

Select the desired *Separator* :



If the file contains headers on the first row, you can skip them by checking *Skip first line* :



Update size

Each data update can receive 1 or more values.
Select the number of rows to read from the file during each update, by writing a value of *Send X row(s)* :

If the CSV file contains more rows than this size, then it will be read by batches of the desired size.
If the CSV files contains fewer rows than this size, then the output will be padded to the desired size by repeating content.

Update period

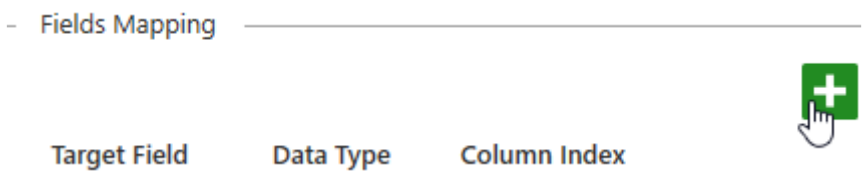
The file is opened and data updated _every X second(s) of your choice :

Fields mapping

Each field of the Target Widget can receive data from a different column of the CSV file.
To configure this relationship, you must define Mappings.

Adding a Mapping

To map a new Widget field with the desired column, click the + button to add a mapping :



Configuring a Mapping

To configure the Mapping :

- Select a *Target Field* from the Target Widget. This is the field that will receive the data.
- Select a *Column Index* from the CSV file, starting at 1. This is the 1-based index of the column of the CSV file that will contain the data.

Removing a Mapping

If you added a Mapping that you want to remove, click the Remove icon next to it

OPC configuration

The tool subscribes to a OPC server.

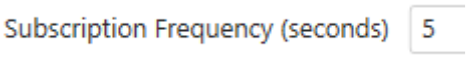
OPC Server connection

Fill-in the desired *Machine Host* to find the OPC Server.
The *Server* list updates. Browse it to select the desired server :

You can click *Test Connection* to confirm that the connection works.

Subscription Frequency

Select how many seconds you want as *Subscription Frequency* :



Fields mapping

Each field of the Target Widget can receive data from a different address on the OPC server.
To configure this relationship, you must define Mappings.

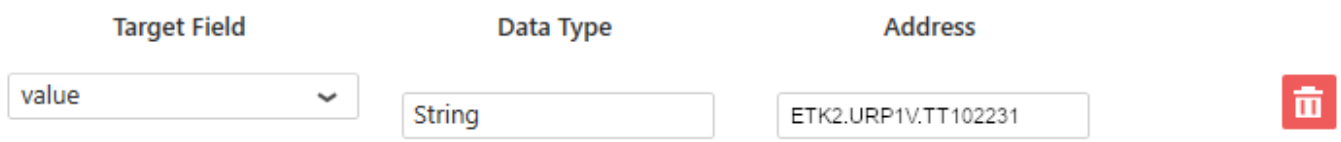
Adding a Mapping

To map a new Widget field with the desired OPC address, click the + button to add a mapping :

Configuring a Mapping

To configure the Mapping :

- Select a *Target Field* from the Target Widget. This is the field that will receive the data.
- Select an *Address* from the OPC server. This is the address that will contain the data.



Removing a Mapping

If you added a Mapping that you want to remove, click the Remove icon next to it

Last modified: Nov 06, 2019

6.5.3. Running jobs

Run Automatically

Jobs are run automatically when Enabled.

Jobs are not Enabled by default.

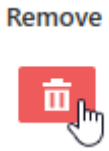
To Enable a Job, check the *Enabled* box :

To Disable a Job, uncheck the *Enabled* box :

Last modified: Nov 06, 2019

6.5.4. Deleting a Job

To delete a Job, click the *Remove* button in front of it :



> ! This will remove the Job permanently.
> If instead you want to disable it temporarily, just Disable it.

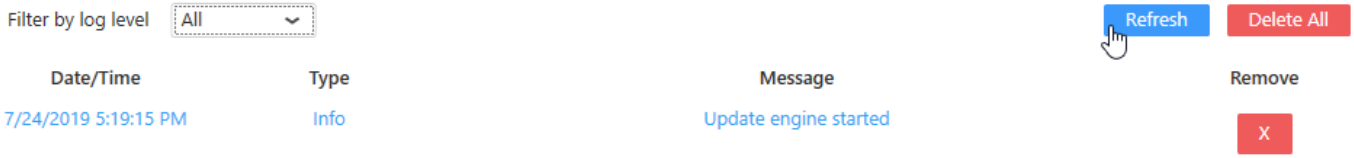
Last modified: Nov 06, 2019

6.6. Troubleshooting

Logs

Application logs are available in the tab *Logs* :

They contain the history of run jobs and useful messages upon failure.



Last modified: Nov 06, 2019

7. Alpana OMI App

The Alpana OMI App allows to display any visuals published on Alpana Server

Usage is very simple :

1. Import the App
2. Add the App to a Layout
3. Configure the App

Last modified: Dec 17, 2019

7.1. Pre-Requisites

The following software pre-requisites are needed :

- Wonderware System Platform 2020, or Wonderware System Platform 2017 Update 3
- Alpana Advanced license
- Dashboards or Widgets published on Alpana Server v3.2.1.5 or later

Note : During the OMI Preview, the Alpana OMI app doesn't display content. That's a known issue.

Last modified: Aug 10, 2020

7.2. Importing

To import the Alpana app for OMi, the procedure is different depending on your WSP version :

- Importing in WSP 2017 U3
- Importing in WSP 2020

Last modified: Jul 06, 2020

7.2.1. Importing in WSP 2017 U3

This page details importing the Alpana app for InTouch OMI 2017 Update 3.

Upgrading

> ! Important Notes – Please read carefully!

If you already have the Alpana App installed, please follow the below instructions to upgrade.

Backup the App

Back up your current version of the Alpana app. To back up the Alpana app:

1. Go the Graphic Toolbox.
2. Select the AlpanaApp object (by default, located under `Default Content > 4. Apps > InTouch OMI Apps`)
3. Right-click and select *Export* > *Object(s)*
4. Select a location to save the file, and click Save.

Remove the App

1. Delete the AlpanaApp object in the galaxy
2. Close the ArchestrA IDE, then delete the folder (folder path listed below).

Add the New App

1. Add the new App as described below.
2. To see changes related to this new Alpana app in run time, redeploy any running ViewApps that use the Alpana app.

Adding the New App

> ! The dll files included with this package must be copied to the IDE node(s) only. Close the ArchestrA IDE before copying over the dll files.

Create the App folder

1. Close the ArchestrA IDE
2. Create the following folder (on 64-bit operating systems) : `<RootDrive>:\Program Files (x86)\ArchestrA\Framework\Bin\ModernEditors\Controls\AlpanaApp\` (on 32-bit operating systems, the folder location is : `<RootDrive>:\Program Files\ArchestrA\Frame`

`work\Bin\ModernEditors\Controls\AlpanaApp\`)

3. Copy the following files in this folder :
    - `code192.Alpana.API.Client.dll`
    - `code192.Alpana.API.Desktop.dll`
    - `code192.Alpana.OMI.Shared.ArchestrAClient.dll`
    - `code192.Alpana.OMI.Shared.Controls.dll`
    - `code192.Alpana.OMI.Shared.dll`
    - `code192.Alpana.OMI.WidgetViewer.dll`
    - `code192.Alpana.Shared.WPF.dll`
    - `MetroSet UI.dll`
    - `Newtonsoft.Json.dll`
    - `Ninject.dll`

Import the App

1. Start the ArchestrA IDE and connect to the galaxy.
2. Import the `AlpanaApp.aaPKG` file from this package into the galaxy, as follows:
    a. From the *Galaxy* menu, select *Import* > *Object(s)*
    b. Navigate to the location that contains the `AlpanaApp.aaPKG` file and click OK

Then you can add the Alpana App to a Layout using the Layout or ViewApp editor.

> By default, the app is located under `Default Content > 4. Apps > InTouch OMI Apps`

Last modified: Jun 26, 2020

7.2.2. Importing in WSP 2020

This page details importing the Alpana app for InTouch OMI 2020.

For WSP 2020, the Alpana app is provided as an ArchestrA object `.aaPKG`.

Adding the New App

To import the Alpana app in ArchestrA :

1. Start the ArchestrA IDE and connect to the galaxy.
2. Import the `AlpanaApp.aaPKG` file from this package into the galaxy, as follows:
    a. From the *Galaxy* menu, select *Import > Object(s)*
    b. Navigate to the location that contains the `AlpanaApp.aaPKG` file and click OK

Then you can add the Alpana App to a Layout using the Layout or ViewApp editor.

Upgrading

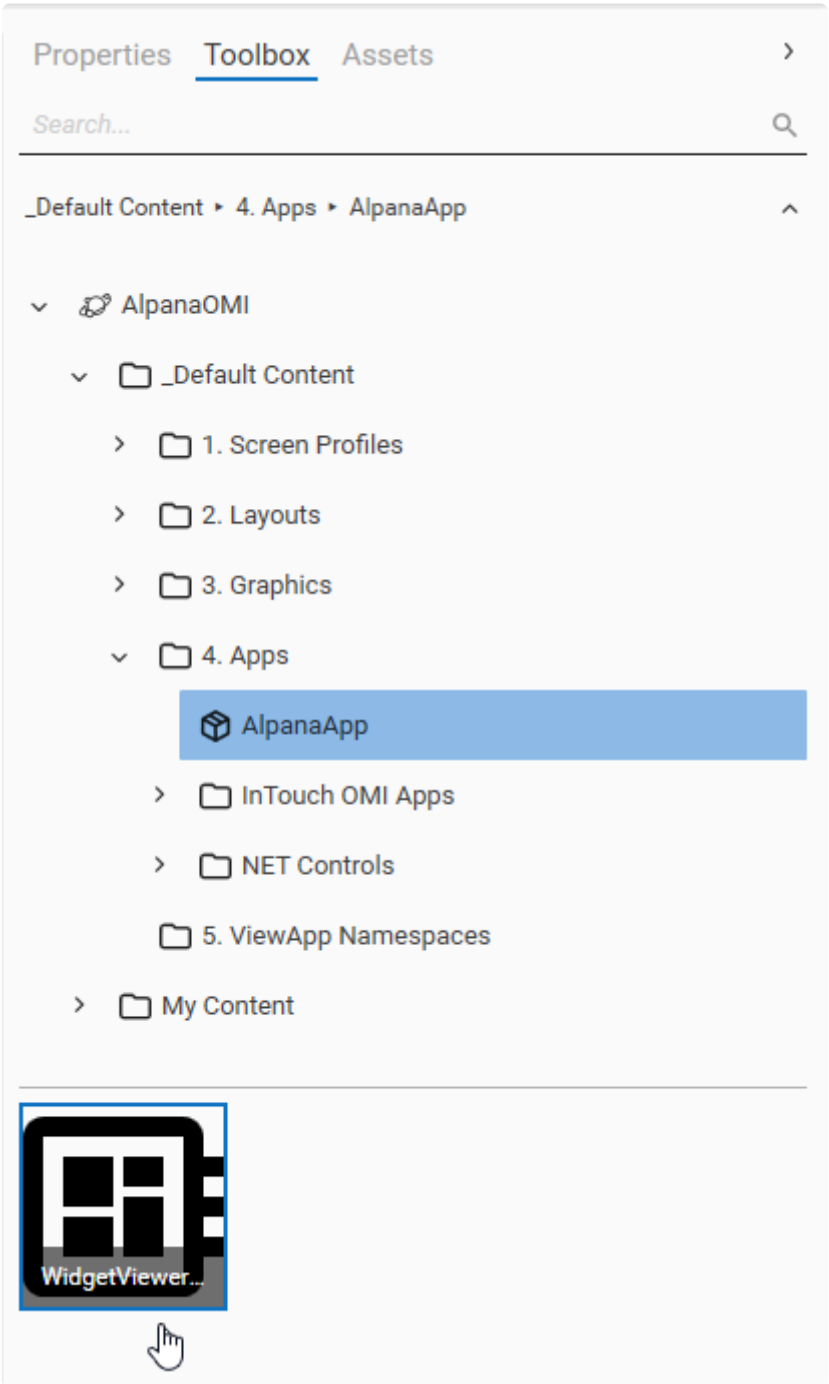
To upgrade :

1. delete the Alpana app object
2. import the new object

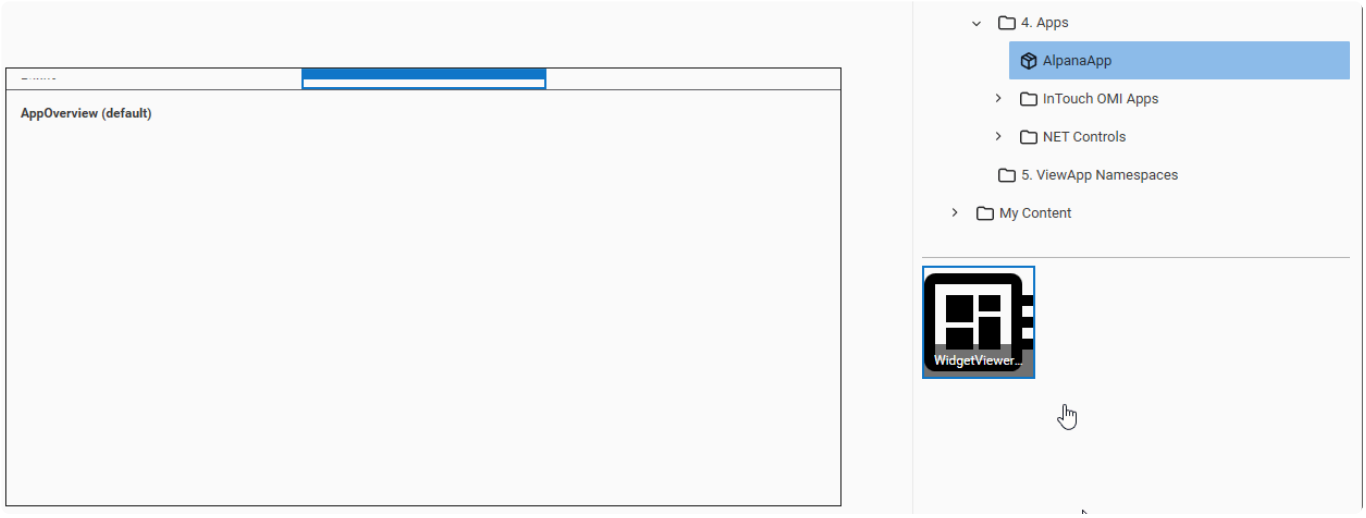
Last modified: Jun 26, 2020

7.3. Adding to a Layout

To add the Alpana app to a layout in OMI, simply browse the Toolbox to find the *AlpanaApp* object :



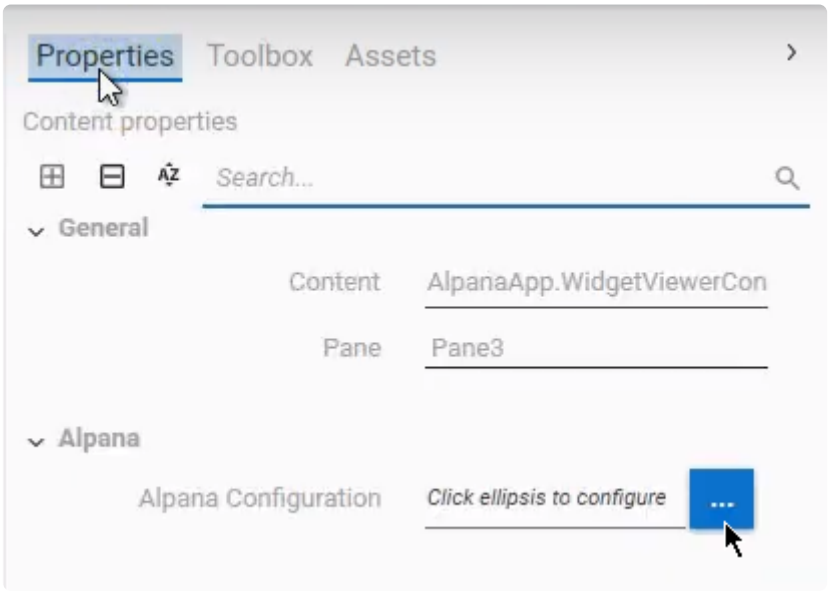
Then drag the *WidgetViewerContainer* and drop on the desired layout :



Last modified: Dec 17, 2019

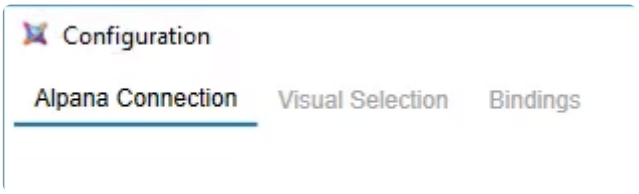
7.4. Configuring

To open the Configuration window for the Alpana app, simply click the ellipsis in the *Properties* of the element :



This opens the *Configuration* window with 3 tabs corresponding to the 3 steps to configure the app :

1. *Alpana Connection* : Connect to Alpana Server
2. *Visual Selection* : Select Alpana visuals
3. *Bindings* : Configure interaction bindings *(optional)*

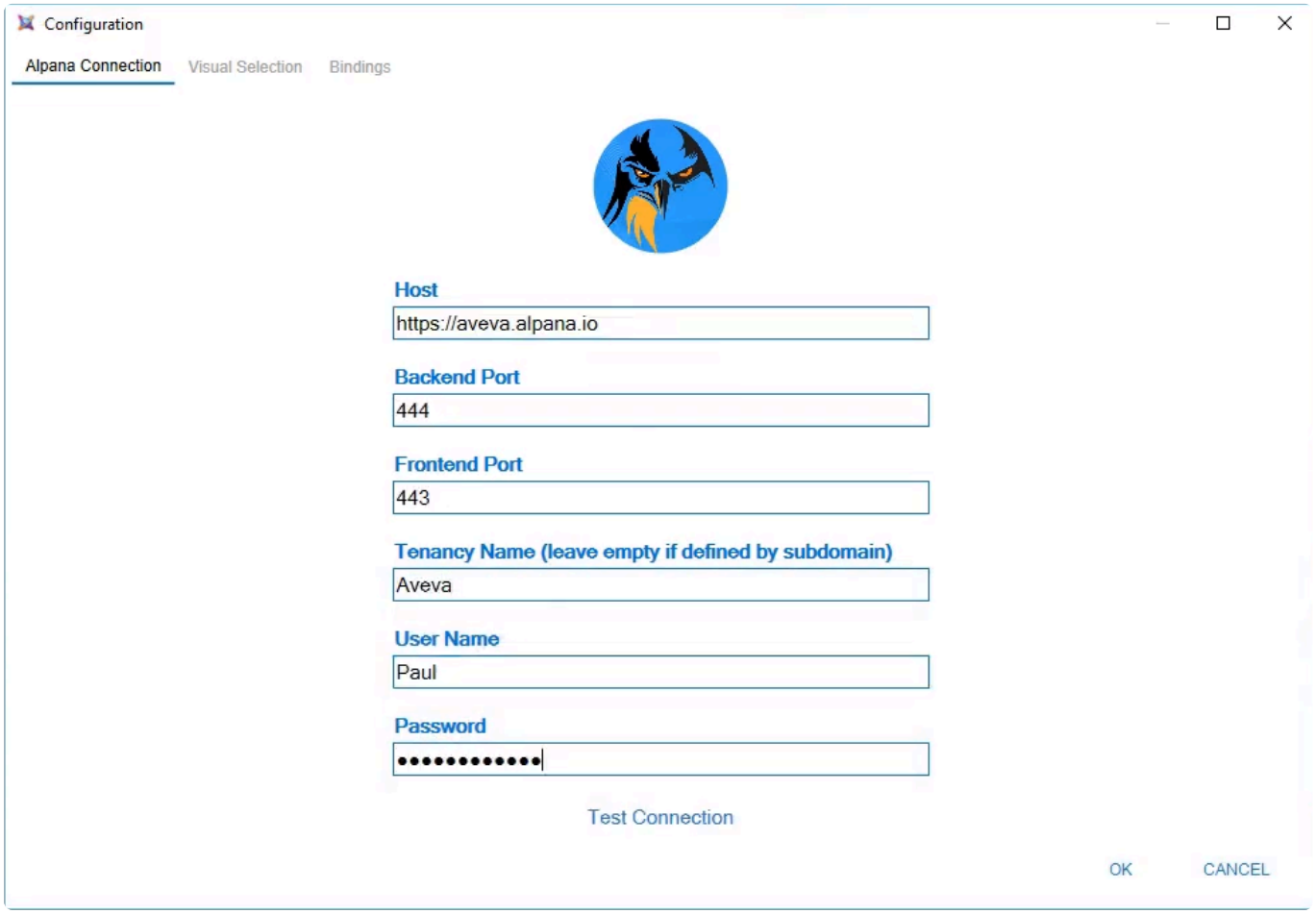


Last modified: Dec 17, 2019

7.4.1. Connecting to Alpana

Login

The first tab *Alpana Connection* allows to connect to an existing Alpana Server.



Connection Parameters

The following fields are required :

Host

The full URL to connect to the Alpana Server installation.
This must include :

- the right protocol (HTTP / HTTPS)
- the full name or IP address. If Alpana Server is configured to have Tenants as sub-domains, you can include the sub-domain here too.

Backend Port

The port number of the Alpana backend website.

Frontend Port

The port number of the Alpana frontend website.

Tenancy Name

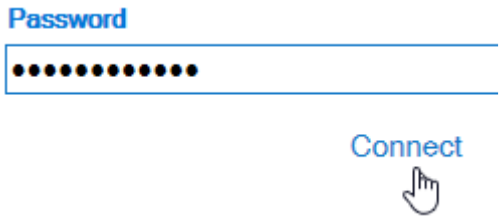
The name of the Tenant you want to connect to.
If Alpana Server is configured to have Tenants as sub-domains, you can leave this field empty.

User Name* and *Password

Use the same credentials as for logging in to Alpana Server from the web interface (frontend).

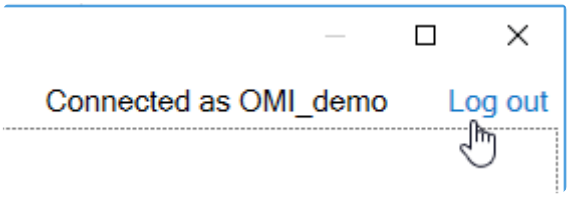
Logging in

To login, click *Connect* at the bottom of the window :



Logout

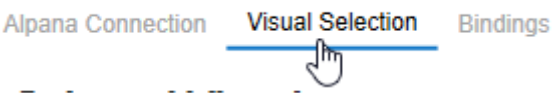
To logout, click *Log out* at the top right of the window :



Last modified: Dec 17, 2019

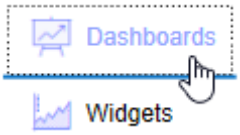
7.4.2. Selecting Visuals

To select Alpana visuals, open the *Visual Selection* tab :

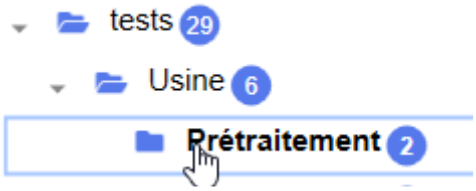


Dashboards

Click on *Dashboards* to select from the list of published Dashboards :

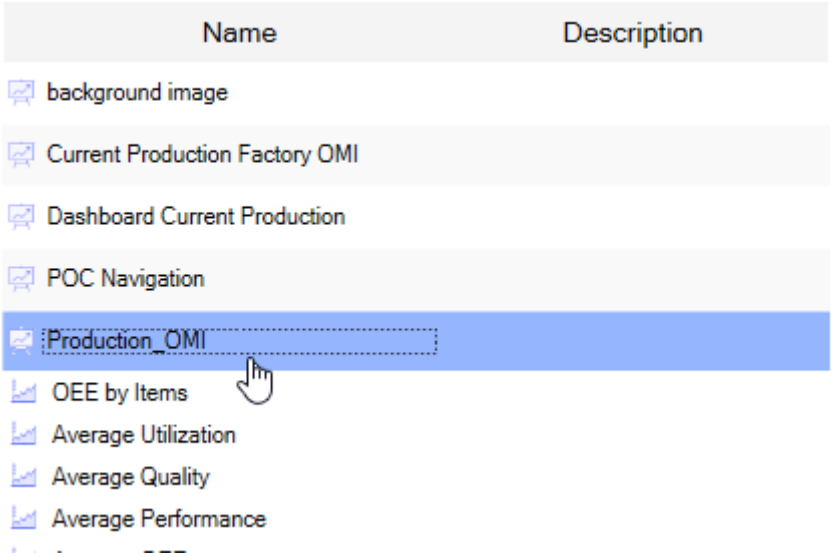


Browse the tree of Categories to find your dashboard :



Full Dashboard

From the list of dashboards on the right, you can click a dashboard to select it :



The *Selected Visual* will say “published dashboard”.

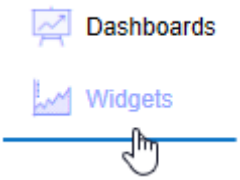
Widget from a Dashboard

When you click on a Dashboard, the list of all Widgets that are contained inside is unfolded.
From there, you may select any Widget that is contained inside a published Dashboard.

The *Selected Visual* will say “widget in dashboard XXX”.

Widgets

Click on *Widgets* to select from the list of published Widgets :



From the list of widgets on the right, you can click a widget to select it :

The *Selected Visual* will say “published widget”.

Last modified: Dec 17, 2019

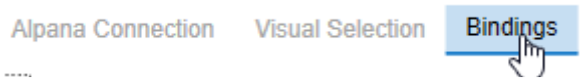
7.4.3. Configuring interactions

What are bindings

In addition to just displaying a Widget or Dashboard, you can also let your InTouch application interact with it.

Bindings allow you to define a relationship between a Alpana Parameter and a ArchestrA Attribute.

To select configure interactions, open the *Bindings* tab :



Managing Bindings

Adding

To add a binding, click the **+** icon :

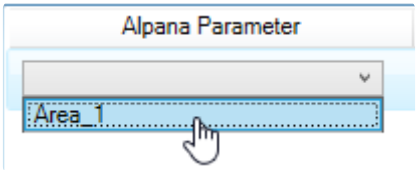
Removing

To remove a binding, click the **–** icon at the right of the corresponding binding :

Binding Properties

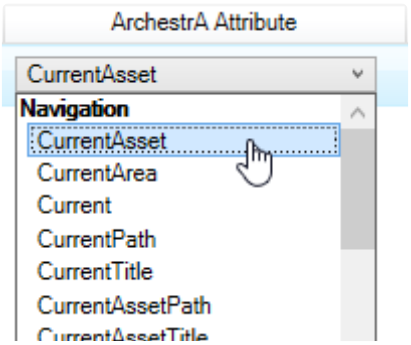
Alpana Parameter

Open the list to select from the Alpana Parameters that exist in the selected dashboard :



ArchestrA Attribute

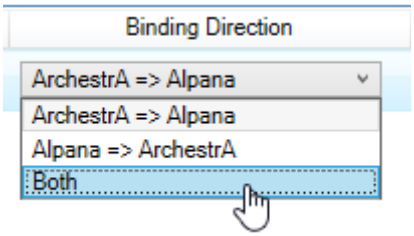
Open the list to select from the ArchestrA attributes that exist in your galaxy :



Binding Direction

Open the list to select a direction of the data flow :

- *ArchestrA => Alpana* : data flows from ArchestrA to Alpana
- *Alpana => ArchestrA* : data flows from Alpana to ArchestrA
- *both* : data flows both ways



Last modified: Dec 17, 2019