

# **DBP - Designer Manual**

#14.1 — Last update: Oct 27, 2023

Interfacing Technologies Corporation

# Table of Contents

|   |           |
|---|-----------|
| <b>1. A Comprehensive Guide to the Digital Business Platform Designer .....</b> | <b>15</b> |
| <b>2. Technical Support .....</b>   | <b>16</b> |
| <b>3. Legal Notices .....</b>   | <b>17</b> |
| <b>4. Release Notes 14.1 .....</b>  | <b>18</b> |
| 4.1. New Features.....  | 19        |
| 4.1.1. General – Keyboard Shortcuts .....                                       | 20        |
| 4.1.2. General Application Management – New Features .....                      | 21        |
| 4.1.2.1. Applications – New Fields for Add, Export, & Import .....              | 22        |
| 4.1.2.2. Settings – Web UI Tab.....   | 26        |
| 4.1.2.3. Application Tree – Option to Increment Version Number .....            | 28        |
| 4.1.3. Task Types – Assign Task Receivers in Web .....                          | 29        |
| 4.1.4. Forms – Save Drafts with Incomplete Required Fields .....                | 31        |
| 4.1.5. Forms – Auto-Save Task Forms.....  | 32        |
| 4.1.6. Forms – Customize Color for Required Fields .....                        | 35        |
| 4.1.7. Forms – Set Disabled Fields as Transparent in Web .....                  | 37        |
| 4.1.8. Forms – Notify and Redirect for Incomplete Required Fields .....         | 39        |
| 4.1.9. Form Behavior Management – New Features .....                            | 40        |
| 4.1.9.1. Form Behavior – Access List of Form Behaviors in Web .....             | 41        |
| 4.1.9.2. Form Behavior – Change Properties in Web .....                         | 42        |
| 4.1.9.3. Form Behavior – Assign Resource to Multiple Forms in Web .....         | 44        |
| 4.1.10. Applications & Form Behaviors – Search Functionality.....               | 45        |
| 4.1.11. Form Category – Change View Order in Web.....                           | 46        |
| 4.1.12. Notifications – Attach Template Output to Email.....                    | 47        |
| 4.1.13. Sequence Service .....  | 49        |
| 4.1.14. EPC Services.....   | 50        |
| 4.2. Feature Improvements .....   | 51        |
| 4.2.1. Performance Improvements .....   | 52        |
| 4.2.2. RBA Import .....   | 53        |
| 4.2.3. RBA Deployment .....   | 54        |
| 4.2.4. SSO Connection – Password Change Not Required .....                      | 55        |
| 4.2.5. Modernized UI for Disabled Fields .....                                  | 56        |
| 4.2.6. Form Validation – Block Duplicate Submissions & Overwrites .....         | 57        |
| 4.2.7. Form Designer – Control Improvements .....                               | 58        |
| 4.2.7.1. Control Governance – Improvements.....                                 | 59        |
| 4.2.7.2. Control Caption – Option to Keep Reserved Space.....                   | 60        |
| 4.2.7.3. Grid Control – Range Validation .....                                  | 61        |
| 4.2.7.4. Tab Control – Reflect Web Appearance in Designer .....                 | 62        |
| 4.2.8. Code Designer – Added Author, Description, & History to Code .....       | 63        |
| 4.2.9. General UI Enhancements .....  | 65        |
| 4.2.9.1. Designer UI Enhancements .....   | 66        |
| 4.2.9.2. Web UI Enhancements .....  | 68        |



|   |            |
|---|------------|
| 4.3. Fixed Issues.....  | 73         |
| 4.3.1. Task Handling – Task Name Used When No Title is Provided.....    | 74         |
| 4.4. Known Limitations .....  | 75         |
| 4.4.1. Sorting in Data Source Dropdown .....                            | 76         |
| 4.4.2. Import – Override Customized Responsibilities .....              | 78         |
| 4.4.3. Grid Control – Quick Forms Feature.....                          | 79         |
| 4.4.4. Pending Tasks Folder – Tasks/Workflows Triggered from Grid ..... | 80         |
| <b>5. BPMS System Requirements .....</b>                                | <b>81</b>  |
| 5.1. BPMS Server Requirements.....                                      | 82         |
| 5.1.1. 2.1.1 Install IIS with Required Modules .....                    | 84         |
| 5.2. BPMS Designer Machine Requirements and Permissions .....           | 85         |
| 5.3. End User Browser Requirements and Permissions.....                 | 86         |
| <b>6. Installing the System.....</b>                                    | <b>87</b>  |
| 6.1. Installing Modules .....   | 89         |
| 6.1.1. Database .....   | 91         |
| 6.1.2. Admin Console.....   | 93         |
| 6.1.3. Web Application .....  | 95         |
| 6.1.4. SMS Service .....  | 99         |
| 6.1.5. Tools.....   | 101        |
| 6.1.6. BPMS Scheduler Service .....                                     | 103        |
| 6.1.6.1. BPMS Scheduler – Developer’s Guide.....                        | 104        |
| 6.1.6.1.1. Getting Started .....  | 105        |
| 6.1.6.1.2. Creating and Deploying Events .....                          | 106        |
| 6.1.6.1.3. Executing Events.....  | 110        |
| 6.1.6.2. BPMS Scheduler – Troubleshooting .....                         | 112        |
| 6.1.6.2.1. Scheduler Initialization (Step 0) .....                      | 113        |
| 6.1.6.2.2. Application Deployment (Step 1) .....                        | 114        |
| 6.1.6.2.3. Raise Event (Step 2) .....                                   | 115        |
| 6.2. Setting Up BPMS Environment.....                                   | 117        |
| 6.2.1. New BPMS Environment – General Setup .....                       | 118        |
| 6.2.2. New BPMS Development Environment – Tools Setup.....              | 121        |
| 6.2.3. New BPMS Development Environment – General Setup .....           | 130        |
| <b>7. Frequently Used Functions.....</b>                                | <b>133</b> |
| 7.1. Data Manipulation .....  | 134        |
| 7.1.1. Filter Options .....   | 138        |
| 7.1.2. Boolean Search .....   | 139        |
| 7.1.2.1. Filter Editor .....  | 141        |
| 7.1.3. Search Field .....   | 142        |
| 7.2. Digital Signature Configuration .....                              | 143        |
| 7.2.1. Step-by-Step Guide for Digital Signatures.....                   | 144        |
| 7.2.2. Digital Signatures in the Web .....                              | 152        |
| <b>8. BPMS Designer General Features .....</b>                          | <b>154</b> |
| 8.1. Title Bar .....  | 156        |

|  |            |
|--|------------|
| 8.2. Menu Bar .....                              | 158        |
| 8.2.1. Home Tab .....                            | 159        |
| 8.2.1.1. Applications .....                      | 160        |
| 8.2.1.2. Current Application .....               | 166        |
| 8.2.1.3. Deploy .....                            | 168        |
| 8.2.1.4. Options .....                           | 169        |
| 8.2.1.4.1. Settings .....                        | 170        |
| 8.2.1.4.1.1. General Tab .....                   | 171        |
| 8.2.1.4.1.2. System Tab .....                    | 174        |
| 8.2.1.4.1.3. Web UI .....                        | 176        |
| 8.2.1.4.1.4. Customization Tab .....             | 177        |
| 8.2.1.4.1.5. Security Tab .....                  | 179        |
| 8.2.1.4.1.6. E-Mail/SMS Tab .....                | 180        |
| 8.2.1.4.1.7. File Storage Tab .....              | 182        |
| 8.2.1.4.1.8. Integration Tab .....               | 184        |
| 8.2.1.4.1.9. Scheduler Tab .....                 | 185        |
| 8.2.1.4.1.10. Logs Tab .....                     | 186        |
| 8.2.1.4.2. Permissions .....                     | 187        |
| 8.2.1.5. Opened Forms .....                      | 190        |
| 8.2.1.6. Help .....                              | 191        |
| 8.2.1.7. Close Application .....                 | 192        |
| 8.2.1.8. Additional Options .....                | 193        |
| 8.2.2. View Tab .....                            | 194        |
| 8.2.2.1. Theme .....                             | 195        |
| 8.2.2.2. Language .....                          | 196        |
| 8.2.2.3. System .....                            | 197        |
| 8.3. Application Tree .....                      | 199        |
| 8.3.1. Application Tree Context Menu .....       | 201        |
| 8.4. Output .....                                | 203        |
| 8.5. System Information .....                    | 204        |
| <b>9. Code Items .....</b>                       | <b>205</b> |
| <b>10. Language Resources .....</b>              | <b>208</b> |
| <b>11. Processes .....</b>                       | <b>211</b> |
| 11.1. Process Design .....                       | 212        |
| 11.1.1. Process Creation .....                   | 213        |
| 11.1.1.1. Process Specification .....            | 216        |
| 11.1.2. Process Designer .....                   | 222        |
| 11.1.2.1. Panel Display Options .....            | 228        |
| 11.1.2.1.1. Resize Design Canvas or Panels ..... | 229        |
| 11.1.2.2. Process Toolbox .....                  | 231        |
| 11.1.2.2.1. Pool and Lane .....                  | 233        |
| 11.1.2.2.2. Start Event .....                    | 236        |
| 11.1.2.2.3. End Event .....                      | 238        |
| 11.1.2.2.4. Human State .....                    | 242        |

|   |            |
|---|------------|
| 11.1.2.2.5. System State .....                              | 246        |
| 11.1.2.2.6. Subprocess .....                                | 250        |
| 11.1.2.2.7. Parallel Gateway .....                          | 254        |
| 11.1.2.2.8. Complex and Exclusive Gateway .....             | 257        |
| 11.1.2.2.9. Common State.....                               | 260        |
| 11.1.2.2.10. Event.....                                     | 262        |
| 11.1.2.2.10.1. Insert and Adjust Events.....                | 266        |
| 11.1.2.2.10.2. Conditional Events: Code Configuration ..... | 268        |
| 11.1.2.3. Menu Bar .....                                    | 269        |
| 11.1.3. Process Alteration.....                             | 272        |
| 11.1.4. Process Context Menus .....                         | 276        |
| 11.2. Human Activities .....                                | 280        |
| 11.2.1. Task Types .....                                    | 281        |
| 11.2.1.1. Translation Management .....                      | 284        |
| 11.2.1.2. Task Type Properties.....                         | 287        |
| 11.2.1.2.1. General Tab .....                               | 288        |
| 11.2.1.2.2. Advanced Tab .....                              | 295        |
| 11.2.1.2.3. Analytic Data Tab .....                         | 298        |
| 11.2.2. Dynamic Task Types.....                             | 300        |
| 11.2.2.1. Dynamic Task Type Properties .....                | 303        |
| 11.2.2.2. Edit Code.....                                    | 304        |
| 11.2.2.3. Designer .....                                    | 306        |
| 11.3. Events.....   | 307        |
| 11.3.1. Event Properties .....                              | 310        |
| 11.3.1.1. Auto and System Event Properties.....             | 311        |
| 11.3.1.1.1. General Tab .....                               | 312        |
| 11.3.1.1.2. Advanced Tab .....                              | 314        |
| 11.3.1.2. Human Event Properties .....                      | 315        |
| 11.3.1.2.1. General Tab .....                               | 316        |
| 11.3.1.2.1.1. Application Files .....                       | 319        |
| 11.3.1.2.2. Advanced Tab .....                              | 321        |
| 11.3.1.2.2.1. Select Color .....                            | 324        |
| 11.3.1.3. Timed Event Properties .....                      | 326        |
| 11.3.1.3.1. General Tab .....                               | 327        |
| 11.3.1.3.1.1. Scheduler .....                               | 329        |
| <b>12. Forms .....</b>                                      | <b>332</b> |
| 12.1. List of Forms .....                                   | 333        |
| 12.1.1. Form Properties .....                               | 335        |
| 12.1.1.1. Form Entity Properties .....                      | 338        |
| 12.1.2. Form Designer .....                                 | 339        |
| 12.1.2.1. Designer Tab .....                                | 342        |
| 12.1.2.1.1. Controls.....                                   | 345        |
| 12.1.2.1.1.1. General Control Properties .....              | 349        |
| 12.1.2.1.1.1.1. Binding .....                               | 350        |
| 12.1.2.1.1.1.1.1. Edit Code.....                            | 354        |

|  |     |
|--|-----|
| 12.1.2.1.1.1.2. Designer .....   | 358 |
| 12.1.2.1.1.1.2. Name .....   | 359 |
| 12.1.2.1.1.1.3. Control Behavior .....                                       | 360 |
| 12.1.2.1.1.1.3.1. Control Governance – Validation .....                      | 363 |
| 12.1.2.1.1.1.3.2. State Matrix – Control-Specific Options.....               | 369 |
| 12.1.2.1.1.1.4. Appearance .....   | 371 |
| 12.1.2.1.1.1.5. Caption Appearance .....                                     | 373 |
| 12.1.2.1.1.2. Control Context Menu.....                                      | 376 |
| 12.1.2.1.1.3. Button .....   | 378 |
| 12.1.2.1.1.4. Checkbox .....   | 381 |
| 12.1.2.1.1.5. Comment .....  | 384 |
| 12.1.2.1.1.5.1. Comment – Binding.....                                       | 387 |
| 12.1.2.1.1.5.1.1. Comment – Attributes .....                                 | 390 |
| 12.1.2.1.1.5.2. Comment – Control Governance .....                           | 403 |
| 12.1.2.1.1.5.3. Comment – Web Display .....                                  | 405 |
| 12.1.2.1.1.6. Date/Time Picker.....  | 407 |
| 12.1.2.1.1.6.1. Date/Time Picker – Web Display .....                         | 409 |
| 12.1.2.1.1.7. File Attachment .....  | 416 |
| 12.1.2.1.1.7.1. File Attachment – Attribute .....                            | 420 |
| 12.1.2.1.1.7.2. File Attachment – Control Governance .....                   | 423 |
| 12.1.2.1.1.7.3. File Attachment – Using Scripts to Manage Data.....          | 425 |
| 12.1.2.1.1.7.4. File Attachment – Web Display .....                          | 426 |
| 12.1.2.1.1.8. Free Draw .....  | 428 |
| 12.1.2.1.1.9. Gantt Chart .....  | 431 |
| 12.1.2.1.1.9.1. Gantt Chart – Binding.....                                   | 434 |
| 12.1.2.1.1.9.1.1. Gantt Chart – Key Components, Entity Conditions, & Binding |     |
| Methods .....  | 436 |
| 12.1.2.1.1.9.1.1.1. Gantt Chart – Attributes .....                           | 440 |
| 12.1.2.1.1.9.1.2. Control Properties – Gantt Chart Tab.....                  | 451 |
| 12.1.2.1.1.9.2. Gantt Chart – Control Governance .....                       | 456 |
| 12.1.2.1.1.10. Gauge .....   | 458 |
| 12.1.2.1.1.10.1. Gauge Edit Form.....  | 462 |
| 12.1.2.1.1.10.1.1. Appearance .....  | 463 |
| 12.1.2.1.1.10.1.2. Label.....  | 468 |
| 12.1.2.1.1.10.1.3. Scale (Base) .....  | 472 |
| 12.1.2.1.1.10.1.3.1. Scale (Minimum, Middle, Maximum) .....                  | 477 |
| 12.1.2.1.1.10.1.4. Needle .....  | 480 |
| 12.1.2.1.1.10.1.5. Cap.....  | 483 |
| 12.1.2.1.1.10.1.6. Data Source.....  | 485 |
| 12.1.2.1.1.10.1.7. Data Binding .....  | 487 |
| 12.1.2.1.1.11. Grid.....   | 490 |
| 12.1.2.1.1.11.1. Grid – Binding .....  | 494 |
| 12.1.2.1.1.11.1.1. Grid – Attributes.....                                    | 496 |
| 12.1.2.1.1.11.1.1.1. Data Grid – Attribute .....                             | 497 |
| 12.1.2.1.1.11.1.1.2. Subform Grid – Attribute .....                          | 505 |

|  |     |
|--|-----|
| 12.1.2.1.1.11.1.2. Grid – Control Properties .....                 | 513 |
| 12.1.2.1.1.11.1.2.1. Grid Column .....                             | 519 |
| 12.1.2.1.1.11.1.2.2. Edit Type .....                               | 522 |
| 12.1.2.1.1.11.1.2.3. Column Behavior Grid.....                     | 523 |
| 12.1.2.1.1.11.1.2.4. Column: Color Styles .....                    | 524 |
| 12.1.2.1.1.11.1.3. Grid – Business Rules for Data Management ..... | 528 |
| 12.1.2.1.1.11.2. Grid – Control Governance .....                   | 531 |
| 12.1.2.1.1.11.3. Grid – Web Display .....                          | 536 |
| 12.1.2.1.1.12. Help .....  | 540 |
| 12.1.2.1.1.13. Image Viewer .....                                  | 544 |
| 12.1.2.1.1.14. Interactive Chart.....                              | 548 |
| 12.1.2.1.1.14.1. Dashboard with Drilldown Functionality .....      | 553 |
| 12.1.2.1.1.14.2. Chart Editor.....                                 | 567 |
| 12.1.2.1.1.14.2.1. General Chart Properties .....                  | 571 |
| 12.1.2.1.1.14.2.1.1. DataSource .....                              | 572 |
| 12.1.2.1.1.14.2.1.2. Design .....                                  | 574 |
| 12.1.2.1.1.14.2.1.3. Detail .....                                  | 576 |
| 12.1.2.1.1.14.2.1.4. Help.....                                     | 577 |
| 12.1.2.1.1.14.2.1.5. Chart Area .....                              | 579 |
| 12.1.2.1.1.14.2.1.6. Schedule .....                                | 580 |
| 12.1.2.1.1.14.2.2. Series Properties .....                         | 581 |
| 12.1.2.1.1.14.2.2.1. Data .....                                    | 582 |
| 12.1.2.1.1.14.2.2.2. Design .....                                  | 583 |
| 12.1.2.1.1.14.2.2.3. Alarm with Color .....                        | 585 |
| 12.1.2.1.1.14.3. Interactive Chart – Web Display .....             | 587 |
| 12.1.2.1.1.15. Label .....   | 590 |
| 12.1.2.1.1.16. Line .....  | 597 |
| 12.1.2.1.1.17. Map.....  | 598 |
| 12.1.2.1.1.17.1. Map Editor Form .....                             | 604 |
| 12.1.2.1.1.17.1.1. Data .....                                      | 605 |
| 12.1.2.1.1.17.1.2. Bubble Size .....                               | 608 |
| 12.1.2.1.1.17.1.3. Coloring .....                                  | 609 |
| 12.1.2.1.1.17.1.4. Tooltip.....                                    | 611 |
| 12.1.2.1.1.18. Multi-Select Dropdown .....                         | 613 |
| 12.1.2.1.1.18.1. Multi-Select Dropdown – Quick Forms.....          | 616 |
| 12.1.2.1.1.18.2. Multi-Select Dropdown – Binding.....              | 618 |
| 12.1.2.1.1.18.2.1. Multi-Select Dropdown – Attributes .....        | 620 |
| 12.1.2.1.1.18.2.1.1. Link to System Entity .....                   | 621 |
| 12.1.2.1.1.18.2.1.2. Link to Custom Entity .....                   | 626 |
| 12.1.2.1.1.18.2.2. Control Properties – Single-Select Tab .....    | 635 |
| 12.1.2.1.1.18.2.3. Control Properties – Multi-Select Tab.....      | 637 |
| 12.1.2.1.1.18.3. Multi-Select Dropdown – Control Governance .....  | 638 |
| 12.1.2.1.1.18.4. Multi-Select Dropdown – Web Display .....         | 640 |
| 12.1.2.1.1.19. Nested Form Link .....                              | 643 |
| 12.1.2.1.1.19.1. Nested Form Link – Binding .....                  | 649 |

|   |     |
|---|-----|
| 12.1.2.1.1.19.1.1. Nested Form Link – Attributes .....            | 651 |
| 12.1.2.1.1.19.2. Nested Form Link – Web Display .....             | 659 |
| 12.1.2.1.1.20. Numeric Textbox .....                              | 661 |
| 12.1.2.1.1.21. Picture.....                                       | 666 |
| 12.1.2.1.1.21.1. User Reference Attribute .....                   | 670 |
| 12.1.2.1.1.22. Radio-Button List.....                             | 673 |
| 12.1.2.1.1.23. Rich Textbox .....                                 | 678 |
| 12.1.2.1.1.23.1. Rich Textbox – Web Display .....                 | 681 |
| 12.1.2.1.1.24. Search Panel.....                                  | 683 |
| 12.1.2.1.1.24.1. Search Panel – Binding .....                     | 686 |
| 12.1.2.1.1.24.2. Search Panel – Build Components .....            | 688 |
| 12.1.2.1.1.25. Single-Select Dropdown .....                       | 691 |
| 12.1.2.1.1.25.1. Single-Select Dropdown – Binding .....           | 694 |
| 12.1.2.1.1.25.2. Single-Select Dropdown – Control Governance..... | 698 |
| 12.1.2.1.1.25.3. Single-Select Dropdown – Web Display .....       | 700 |
| 12.1.2.1.1.26. Static Image .....                                 | 702 |
| 12.1.2.1.1.27. Tab Control .....                                  | 705 |
| 12.1.2.1.1.27.1. Tab Pages Editor .....                           | 710 |
| 12.1.2.1.1.28. Table.....   | 713 |
| 12.1.2.1.1.28.1. Table – Control Properties .....                 | 718 |
| 12.1.2.1.1.29. Textbox (Single-Line, Multi-Line & Password).....  | 722 |
| 12.1.2.1.1.29.1. Textbox Control Properties.....                  | 724 |
| 12.1.2.1.1.30. Time Duration.....                                 | 728 |
| 12.1.2.1.1.30.1. Time Duration Web Display .....                  | 731 |
| 12.1.2.1.1.31. Tree .....   | 732 |
| 12.1.2.1.1.31.1. Tree – Binding.....                              | 735 |
| 12.1.2.1.1.31.1.1. Tree – Key Components & Entity Conditions..... | 737 |
| 12.1.2.1.1.31.1.1.1. Tree – Attributes .....                      | 739 |
| 12.1.2.1.1.31.1.2. Control Properties – Tree Tab.....             | 754 |
| 12.1.2.1.1.31.2. Tree – Control Governance .....                  | 758 |
| 12.1.2.1.1.31.3. Tree – Web Display.....                          | 762 |
| 12.1.2.1.1.32. User Picture .....                                 | 765 |
| 12.1.2.1.1.32.1. User Picture – Binding .....                     | 767 |
| 12.1.2.1.1.33. User Signature .....                               | 769 |
| 12.1.2.1.2. Quick Forms .....                                     | 772 |
| 12.1.2.1.2.1. Quick Forms Feature.....                            | 773 |
| 12.1.2.1.3. Layouts .....   | 777 |
| 12.1.2.1.3.1. Layout Context Menu .....                           | 780 |
| 12.1.2.1.3.2. Layout Properties .....                             | 782 |
| 12.1.2.2. Form Entity Tab .....                                   | 786 |
| 12.1.2.3. Tips and Tricks for Customizing Templates.....          | 792 |
| 12.2. Form Behavior.....  | 794 |
| 12.2.1. Form Behavior Properties .....                            | 796 |
| 12.2.1.1. General Tab.....  | 797 |
| 12.2.1.1.1. Word Template Editor.....                             | 802 |

|   |            |
|---|------------|
| 12.2.1.2. Advanced Tab.....   | 803        |
| 12.2.1.3. Title Tab .....   | 805        |
| 12.3. Form Categories .....   | 807        |
| 12.3.1. Form Category Properties .....  | 809        |
| 12.4. Form Control Actions.....   | 812        |
| 12.5. Form Dynamic Parameters.....  | 814        |
| 12.6. Dropdown Data Sources .....   | 816        |
| 12.6.1. Query Editor.....   | 818        |
| 12.6.1.1. Dynamic Tab .....   | 819        |
| 12.6.1.1.1. Query Executor .....  | 822        |
| 12.6.1.1.2. Sample Queries.....   | 823        |
| 12.6.1.2. Static Tab .....  | 825        |
| <b>13. Business Logics .....</b>  | <b>828</b> |
| 13.1. Rules.....  | 829        |
| 13.1.1. Rule Properties .....   | 831        |
| 13.1.2. Code Designer .....   | 832        |
| 13.1.2.1. Find .....  | 835        |
| 13.1.2.2. Scripts .....   | 837        |
| 13.1.2.2.1. Sample – Dynamically Add Control Validation .....                     | 838        |
| 13.1.2.2.2. Sample – Copy Files from One File Attachment Control to Another ..... | 839        |
| 13.1.2.2.3. Sample – Store & Retrieve Custom Server Files.....                    | 840        |
| 13.1.2.2.4. Sample – Add Template to Email.....                                   | 841        |
| 13.1.2.2.5. Sample – Send Email By Code .....                                     | 842        |
| 13.1.2.2.6. Sample – Extract Activities ID from BPMSProcessInfoString .....       | 843        |
| 13.1.2.2.7. Sample – Terminate Active Tasks for Specific DomainObject .....       | 844        |
| 13.1.2.2.8. BPMS Services – Use Cases and Applications .....                      | 845        |
| 13.1.2.2.8.1. Method – GetFileFromControl (DmsService) .....                      | 852        |
| 13.1.2.2.8.2. Method – CreateDomainObject (DomainObjectService) .....             | 854        |
| 13.1.2.2.8.3. Method – Login (EPService) .....                                    | 855        |
| 13.1.2.2.8.4. Method – GetObjects (EPService) .....                               | 857        |
| 13.1.2.2.8.5. Method – GetUsers (EPService) .....                                 | 858        |
| 13.1.2.2.8.6. Method – SetFilter / ClearFilter (FormService).....                 | 859        |
| 13.1.2.2.8.7. Method – RaiseEvent (ProcessService).....                           | 860        |
| 13.1.2.2.8.8. Method – ExecuteSQLQuery (SQLService) .....                         | 861        |
| 13.1.3. Rule Designer .....   | 862        |
| 13.2. Actions .....   | 867        |
| 13.2.1. Pre-Form Actions .....  | 868        |
| 13.2.1.1. Pre-Form Action Properties.....   | 871        |
| 13.2.2. Post-Form Actions .....   | 872        |
| 13.2.2.1. Post-Form Action Properties .....                                       | 874        |
| 13.2.3. Event Actions .....   | 875        |
| 13.2.3.1. Event Action Properties .....   | 877        |
| 13.3. Guards .....  | 880        |
| 13.3.1. Guard Properties.....   | 882        |
| 13.4. Code Messages .....   | 885        |

|  |            |
|--|------------|
| 13.4.1. Code Message Editor.....                             | 887        |
| <b>14. Entities .....</b>                                    | <b>888</b> |
| 14.1. Entity Designer.....                                   | 889        |
| 14.1.1. Add New Entity .....                                 | 891        |
| 14.1.1.1. Add New Attribute .....                            | 894        |
| 14.1.1.1.1. Data Type Per Form Control .....                 | 900        |
| 14.1.2. Entity-Relationship Diagram .....                    | 901        |
| 14.2. Key Point Indicators (KPIs).....                       | 904        |
| 14.2.1. KPI Properties.....                                  | 906        |
| 14.2.1.1. Naming Tab .....                                   | 907        |
| 14.2.1.2. DataSource Tab.....                                | 909        |
| 14.2.1.3. Execute Parameters Tab .....                       | 911        |
| 14.2.1.4. View and Insert Parameters Tab .....               | 912        |
| 14.2.1.5. Timing Tab.....                                    | 914        |
| 14.3. KPI BAM .....  | 915        |
| <b>15. Notification .....</b>                                | <b>917</b> |
| 15.1. Notification Templates.....                            | 918        |
| 15.1.1. Notification Item Properties .....                   | 923        |
| 15.2. Task Data Items .....                                  | 927        |
| 15.2.1. Data Item Properties .....                           | 932        |
| 15.3. Info Templates .....                                   | 936        |
| 15.3.1. Info Template Properties .....                       | 940        |
| <b>16. Integration.....</b>                                  | <b>942</b> |
| 16.1. Resources.....   | 943        |
| 16.1.1. Add Reference .....                                  | 945        |
| 16.1.2. Add Web Reference .....                              | 946        |
| 16.2. WebService Connectors .....                            | 948        |
| 16.2.1. WebService Connector .....                           | 950        |
| 16.3. Connections .....                                      | 952        |
| 16.3.1. Connect to Database .....                            | 954        |
| 16.3.1.1. MySQL .....  | 955        |
| 16.3.1.2. OLE DB .....                                       | 957        |
| 16.3.1.3. ODBC .....   | 958        |
| 16.3.1.4. Access.....  | 959        |
| 16.4. References.....  | 961        |
| <b>17. Application Programming Interface (API) .....</b>     | <b>963</b> |
| 17.1. Define a Parameter (AddParameter) .....                | 964        |
| 17.2. Calling a Rule (CallRule) .....                        | 965        |
| 17.3. Calling Schedule Service (StartJob) .....              | 966        |
| 17.4. 11.4 Stop Scheduling (StopJob) .....                   | 967        |
| 17.5. 11.5 Sum of two numbers (Arithmetic.Add).....          | 968        |
| 17.6. 11.6 Divide two numbers (Arithmetics.Division) .....   | 969        |
| 17.7. 11.7 Multiply two numbers (Arithmetics.Multiply) ..... | 970        |



|  |      |
|--|------|
| 17.8. 11.8 Subtract two numbers (Arithmetics.Subtract) .....                   | 971  |
| 17.9. 11.9 Concatenate two Strings (Concat) .....                              | 972  |
| 17.10. 11.10 Search Text (Contains) .....                                      | 973  |
| 17.11. 11.11 Add Date (AddDate) .....  | 974  |
| 17.12. 11.12 Insert current date (CurrentDate) of the DateTime type .....      | 975  |
| 17.13. 11.13 Add time to date (CurrentDate.Add) .....                          | 976  |
| 17.14. 11.14 Add Day to Date (CurrentDate.AddDays) .....                       | 977  |
| 17.15. 11.15 Add hours to date (CurrentDate.AddHours) .....                    | 978  |
| 17.16. 11.16 Add milliseconds to date (CurrentDate.AddMilliseconds) .....      | 979  |
| 17.17. 11.17 Add Minute to Date (CurrentDate.AddMinutes) .....                 | 980  |
| 17.18. 11.18 Add months to date (CurrentDate.AddMonths) .....                  | 981  |
| 17.19. 11.19 Add seconds to date (CurrentDate.AddSeconds) .....                | 982  |
| 17.20. 11.20 Add tick to date (CurrentDate.AddTicks) .....                     | 983  |
| 17.21. 11.21 Add year to date (CurrentDate.AddYears) .....                     | 984  |
| 17.22. 11.22 Compare current date with date (CurrentDate.CompareTo) .....      | 985  |
| 17.23. 11.23 Take a Date From (CurrentDate.Date) .....                         | 986  |
| 17.24. 11.24 Getting the day out of the date (CurrentDate.Day) .....           | 987  |
| 17.25. 11.25 Take a day of week (CurrentDate.DayOfWeek) .....                  | 988  |
| 17.26. 11.26 Take a few days from the date (CurrentDate.DayOfYear) .....       | 989  |
| 17.27. 11.27 Getting the time out of the current date (CurrentDate.Hour) ..... | 990  |
| 17.28. 11.28 Insert current date (CurrentMiladiDate) in decimal format .....   | 991  |
| 17.29. 11.29 Insert Current Time (CurrentTime) .....                           | 992  |
| 17.30. 11.30 Difference between two dates (DateDiffGregorian) .....            | 993  |
| 17.31. 11.31 Insert Day of the Week (DayOfWeek) .....                          | 994  |
| 17.32. 11.32 Convert date decimal to datetime (GetDate) .....                  | 995  |
| 17.33. 11.33 Day of the week (GetDayOfWeekByDate) .....                        | 996  |
| 17.34. 11.34 The day of the week (GetDayOfWeekByNumeric) .....                 | 997  |
| 17.35. 11.35 Sum of 2 times (TimeAdd) .....                                    | 998  |
| 17.36. 11.36 Subtraction of 2 time (TimeDiff) .....                            | 999  |
| 17.37. 11.37 Receive Groups (GetAllGroups) .....                               | 1000 |
| 17.38. 11.38 Password Comparison (CheckPassword) .....                         | 1001 |
| 17.39. 11.39 Assignment to User (AssignStaffToUser) .....                      | 1002 |
| 17.40. 11.40 Delete User Assignment (DeleteUserAssignment) .....               | 1003 |
| 17.41. 11.41 Update Unit (UpdateDepartment) .....                              | 1004 |
| 17.42. 11.42 Update Group (UpdateGroup) .....                                  | 1005 |
| 17.43. 11.43 Update Staff (UpdateStaff) .....                                  | 1006 |
| 17.44. 11.44 Update User (UpdateUser) .....                                    | 1007 |
| 17.45. 11.45 Insert member in the Group (AddMemberToGroup) .....               | 1008 |
| 17.46. 11.46 Remove member from the Group (RemoveMemberOfGroup) .....          | 1009 |
| 17.47. 11.47 Remove User from Group (RemoveMemberOfGroup) .....                | 1010 |
| 17.48. 11.48 Create or Update User (SaveUser) .....                            | 1011 |
| 17.49. 11.49 Create or Update an Organizational Unit (saveDepartment) .....    | 1012 |
| 17.50. 11.50 Create or Update Group (saveGroup) .....                          | 1013 |
| 17.51. 11.51 Create or Update a Staff (saveStaff) .....                        | 1014 |
| 17.52. 11.52 Update User (UpdateUserWithoutPassword) .....                     | 1015 |

|  |      |
|--|------|
| 17.53. 11.53 Get current Manager's ID (ManagersIdOfCurrentStaff) .....   | 1016 |
| 17.54. 11.54 Get a list of current Managers (GetManagersOfCurrentStaff).....                                       | 1017 |
| 17.55. 11.55 Get personally identifiable information (GetPrincipal).....   | 1018 |
| 17.56. 11.56 Get admin ID (GetManager).....  | 1019 |
| 17.57. 11.57 Get started User Information (MercuryService.StarterUser) .....                                       | 1020 |
| 17.58. 11.58 Get current User information.....   | 1021 |
| 17.59. 11.59 Get the header side manager information (MercuryService.ManagerOfStarterStaff) .....                  | 1022 |
| 17.60. 11.60 Get Started Information (MercuryService.StarterStaff) .....   | 1023 |
| 17.61. 11.61 Getting Starter Side Manager Information<br>(MercuryService.ManagerOfCurrentStaff).....               | 1024 |
| 17.62. 11.62 Get Current Staff Information (MercuryService.CurrentStaff) .....                                     | 1025 |
| 17.63. 11.63 Get Managers for User Positions (MercuryService.GetUserManagerByUser) .....                           | 1026 |
| 17.64. 11.64 Get Users admin side (MercuryService.GetUserManagerByStaff).....                                      | 1027 |
| 17.65. 11.65 Get Group Members (MercuryService.GetUsersByGroupCode).....   | 1028 |
| 17.66. 11.66 Receive the Organizational Unit (MercuryService.GetDepartmentByStaff) .....                           | 1029 |
| 17.67. 11.67 Obtaining Organizational Units for a single User<br>(MercuryService.GetDepartmentByUser).....         | 1030 |
| 17.68. 11.68 Get the User's positions (MercuryService.GetUserStaffs) .....   | 1031 |
| 17.69. 11.69 Get the position of the top Manager of one side<br>(MercurySide.GetManagerOfBeforeCEODepartment)..... | 1032 |
| 17.70. 11.70 Get the post of a higher-level unit manager<br>(MercuryService.GetManagerOfUpperDepartment) .....     | 1033 |
| 17.71. 11.71 Get current User units (MercuryService.CurrentUserDepartment).....                                    | 1034 |
| 17.72. 11.72 Get current manager User list (MercuryService.CurrentUserManager).....                                | 1035 |
| 17.73. 11.73 Get current User positions (MercuryService.CurrentUserStaffs).....                                    | 1036 |
| 17.74. 11.74 Get Current User (MercuryService.CurrentStaff).....   | 1037 |
| 17.75. 11.75 Receive admin-side admin ID (MercuryService.GetStaffMangerByStaff).....                               | 1038 |
| 17.76. 11.76 Fetch information from the database (ExecuteQuery) .....  | 1039 |
| 17.77. 11.77 Fetch information from the current database (ExecuteHqlQuery) .....                                   | 1040 |
| 17.78. 11.78 Get Objects (GetAll) .....  | 1041 |
| 17.79. 11.79 Get the complete object domain (GetDomainObject) .....  | 1042 |
| 17.80. 11.80 Fetch Database Information (ExecuteSqlQuery) .....  | 1043 |
| 17.81. 11.81 Get all database information (ExecuteSqlQueryFromDB).....   | 1044 |
| 17.82. 11.82 General code tracking system (GenerateTracking) .....   | 1045 |
| 17.83. 11.83 Get Tracking Code (GetTrackingNumber).....  | 1046 |
| 17.84. 11.84 Send Notification to User (SendByNotification) .....  | 1047 |
| 17.85. 11.85 Send notification to (SendByNotification) .....   | 1048 |
| 17.86. 11.86 Send notification to recipient list (SendByNotification) .....  | 1049 |
| 17.87. 11.87 Send notification to recipient email list (SendByNotification) .....                                  | 1050 |
| 17.88. 11.88 Send notification to an email (SendByNotification).....   | 1051 |
| 17.89. 11.89 Create notifications and send to recipients (Send) .....  | 1052 |
| 17.90. 11.90 Create notifications and send to receiver types (Send) .....  | 1053 |
| 17.91. 11.91 Create notifications and send to recipients list (Send).....  | 1054 |
| 17.92. 11.92 Create a notification and send to recipient email list (Send).....                                    | 1055 |
| 17.93. 11.93 Create notification and send to recipient's email (Send).....   | 1056 |

|  |             |
|--|-------------|
| 17.94. 11.94 Send Transaction Type Notice (SendCopy) .....                                   | 1057        |
| 17.95. 11.95 Event call (RaiseEvent) .....   | 1058        |
| 17.96. 11.96 Call event with specified user (RaiseEvent) .....                               | 1059        |
| 17.97. 11.97 Start the process (StartProcess) .....  | 1060        |
| 17.98. 11.98 Start process with the user and the Starter (StartProcess) .....                | 1061        |
| 17.99. 11.99 Start the process automatically (StartProcessOnPersistentObject) .....          | 1062        |
| 17.100. 11.100 Get an object from an entity (GetDomainObject) .....                          | 1063        |
| 17.101. 11.101 Create an object from an entity (CreateDomainObject) .....                    | 1064        |
| 17.102. 11.102 Create an object from an entity with an identifier (CreateDomainObject) ..... | 1065        |
| 17.103. 11.103 Delete an object from an entity (DeleteDomainObject) .....                    | 1066        |
| 17.104. 11.104 Save an object from an entity (SaveDomainObject) .....                        | 1067        |
| 17.105. 11.105 Save an object from an object (SaveObject) .....                              | 1068        |
| 17.106. 11.106 Get objects of an entity (GetAll) .....                                       | 1069        |
| 17.107. 11.107 Get Parents Object (GetParentObject) .....                                    | 1070        |
| 17.108. 11.108 Changing information in the database (CommitTransaction) .....                | 1071        |
| 17.109. 11.109 Get the file from the attachment control (GetFileFromControl) .....           | 1072        |
| 17.110. 11.110 Get the file from the attachment control (GetFile) .....                      | 1073        |
| 17.111. 11.111 Search the folder named (GetFolderDefaultForAttachment) .....                 | 1074        |
| 17.112. 11.112 Search the folder with the name and id (GetFolderByNameAndParentId) .....     | 1075        |
| 17.113. 11.113 Create folder with name and id (CreateFolderSerialize) .....                  | 1076        |
| 17.114. 11.114 Create folder with name and id (CreateFolderSerialize) .....                  | 1077        |
| 17.115. 11.115 Get the children of a node (GetAllTreeNodeFromChildObject) .....              | 1078        |
| 17.116. 11.116 Get the children of root (GetAllTreeNodeFromRootObject) .....                 | 1079        |
| 17.117. 11.117 Get Tree Roots (GetParentsNode) .....   | 1080        |
| 17.118. 11.118 Get root tree with child object (GetRootObject) .....                         | 1081        |
| 17.119. 11.119 Read from excel (ReadOfExcel) .....   | 1082        |
| 17.120. 11.120 Open File or Print (GetWordFormat) .....                                      | 1083        |
| 17.121. 11.121 Call Web Service Connection (InvokeBPMSService) .....                         | 1084        |
| 17.122. 11.122 Calling the Web Service Connection with XMM (InvokeBPMSService) .....         | 1085        |
| 17.123. 11.123 Calling the Web Service with the User and the User (InvokeBPMSService) .....  | 1086        |
| 17.124. 11.124 Calling the Web service with the user and the XML (InvokeBPMSService) .....   | 1087        |
| 17.125. 11.125 X-MEL split into distinct attributes (XmlToProperties) .....                  | 1088        |
| 17.126. 11.126 Convert Dictionary to XMM Format (PropertiesToXml) .....                      | 1089        |
| 17.127. 11.127 Code Environment .....  | 1090        |
| 17.128. 11.128 Design Environment .....  | 1092        |
| <b>18. Business Logic API .....</b>  | <b>1094</b> |
| 18.1. Reservation Word .....   | 1095        |
| 18.2. HQL .....  | 1096        |
| 18.3. SQL .....  | 1097        |
| 18.4. User .....   | 1098        |
| 18.5. Rule .....   | 1100        |
| 18.6. Process .....  | 1101        |
| 18.7. Domain .....   | 1102        |
| 18.8. Task .....   | 1103        |
| 18.9. TimeEvent .....  | 1104        |

|  |             |
|--|-------------|
| 18.10. Notification .....  | 1105        |
| 18.11. Tracking .....  | 1107        |
| 18.12. Data Source .....   | 1108        |
| 18.13. DateTime .....  | 1109        |
| 18.14. Math .....  | 1110        |
| 18.15. Text .....  | 1111        |
| 18.16. Point .....   | 1112        |
| <b>19. Attachments .....</b>                                       | <b>1113</b> |
| 19.1. 13.1 Microsoft IIS 6 SSL Compatibility Components .....      | 1114        |
| 19.2. 13.2 Setting Permission For sharing BPMAApp Folder. ....     | 1118        |
| 19.3. 13.3 MSDTC Services .....                                    | 1127        |
| 19.3.1. 13.3.1 Introduction .....                                  | 1128        |
| 19.3.2. 13.3.2 Steps to Configure MSDTC Services .....             | 1129        |
| 19.4. 13.4 Externalizing the web form .....                        | 1131        |
| <b>20. Troubleshooting .....</b>                                   | <b>1132</b> |
| 20.1. This configuration section cannot be used at this path ..... | 1133        |
| 20.2. Access to the path 'Rayvarz\BPMS\Temp' is denied .....       | 1136        |
| 20.3. Not able to access temp files by visual studio .....         | 1137        |
| 20.4. Error in saving web form when there is an attachment .....   | 1139        |
| 20.5. "Invalid Data" Error on deploy any BPM Application .....     | 1140        |
| 20.6. Application Tree Not Visible .....                           | 1141        |
| 20.7. Application Icons Not Visible in Web .....                   | 1142        |
| 20.8. Form Access Blocked .....                                    | 1143        |
| 20.9. Visual Studio – Error When Building Solution .....           | 1144        |

# 1. A Comprehensive Guide to the Digital Business Platform Designer

---



## 2. Technical Support

---

For help and technical advice, please contact the Interfacing Technical Support department. In North America, the department is open Monday-Friday from 7:00 AM to 5:00PM, Eastern Time.

**Phone:** +1 514 737 7333

**Email:** [support@interfacing.com](mailto:support@interfacing.com)

**Live Chat:** [interfacing.com](https://www.interfacing.com)

Online technical support and worldwide contact information are available at [www.interfacing.com](https://www.interfacing.com)

## 3. Legal Notices

---

### Disclaimer

Copyright © 2022 Interfacing Technologies Corporation. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this document is furnished for informational use only, and due to product development, this information is subject to change without notice, and should not be construed as a commitment by Interfacing Technologies Corporation. Interfacing Technologies Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of Interfacing Technologies Corporation..

The Interfacing Technologies logo and the Enterprise Process Center logo are trademarks of Interfacing Technologies Corporation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Interfacing Technologies has made every effort to supply trademark information about manufacturers and their products mentioned in this publication.

## 4. Release Notes 14.1

---

In our latest release of the **DPB Designer**, we have introduced new features, improved existing ones, revamped the user interface, and fixed bugs.

| Product      | Version | Date       |
|--------------|---------|------------|
| DBP Designer | 14.1    | 2023/04/19 |

In these notes, you will find:

1. [New Features](#)
2. [Feature Improvements](#)
3. [Fixed Issues](#)
4. [Known Limitations](#)



## 4.1. New Features

---



With this latest release, we have introduced the following new features:

1. [General – Keyboard Shortcuts](#)
2. [General Application Management – New Features](#)
3. [Task Types – Assign Task Receivers in Web](#)
4. [Forms – Save Drafts with Incomplete Required Fields](#)
5. [Forms – Auto-Save Workflow Forms](#)
6. [Forms – Customize Color for Required Fields](#)
7. [Forms – Set Disabled Fields as Transparent in Web](#)
8. [Forms – Notify and Redirect for Incomplete Required Fields](#)
9. [Form Behavior Management – New Features](#)
10. [Applications & Form Behaviors – Search Functionality](#)
11. [Form Category – Change View Order in Web](#)
12. [Sequence Service](#)
13. [EPC Services](#)

## 4.1.1. General – Keyboard Shortcuts

---

We have added keyboard shortcuts for several actions, namely **Insert (New)**, **Delete**, **Edit**, and **Refresh**.

- **Insert (New)**: CTRL + I
- **Delete**: CTRL + D
- **Edit**: CTRL + P
- **Refresh**: F5

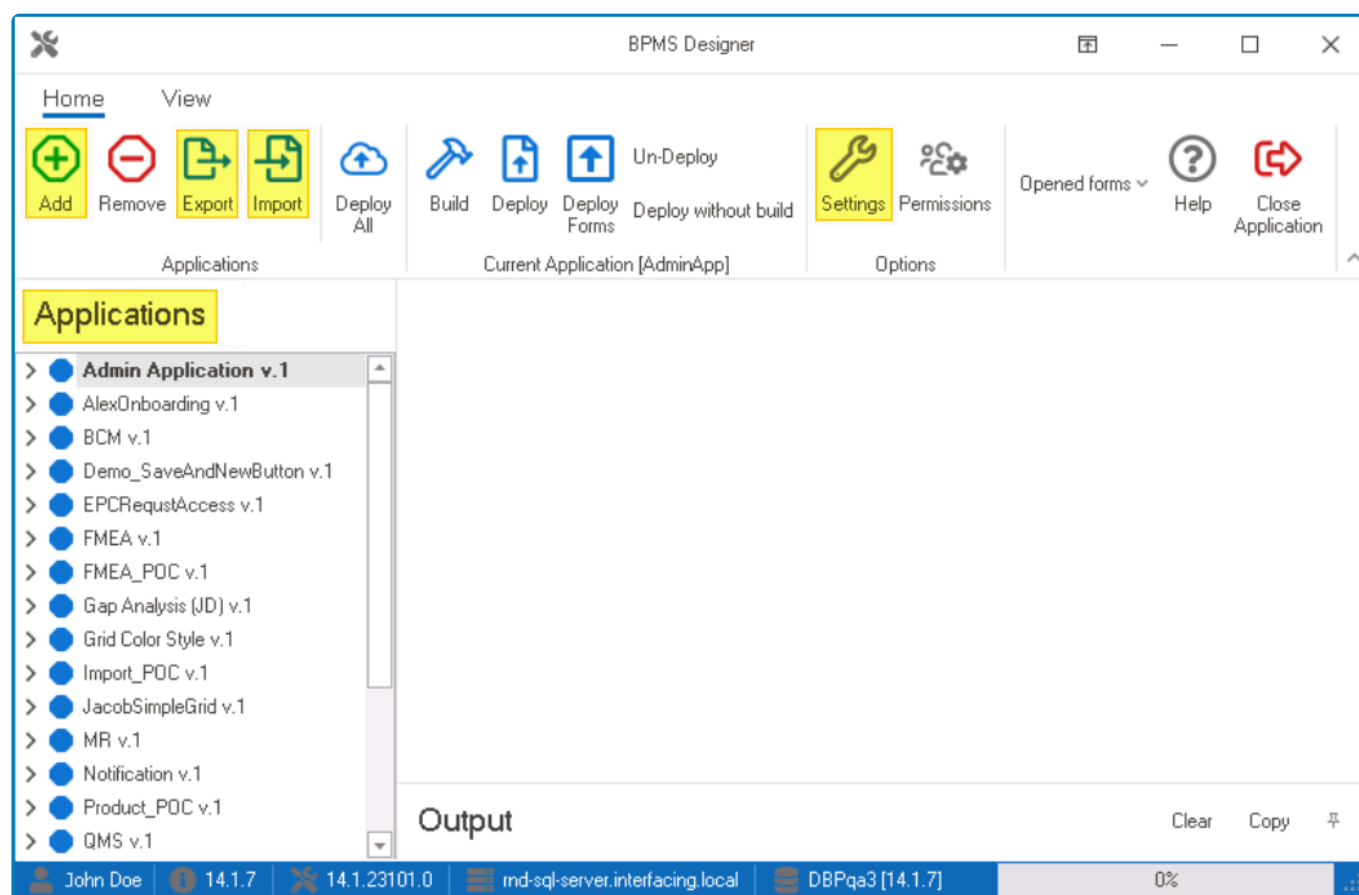
These shortcuts can now be used in all applicable areas of the Designer, particularly in the tabs that consist of a list of items. This change aims to enhance the user experience by providing a more efficient way of performing these actions without relying on the traditional menu buttons.

## 4.1.2. General Application Management – New Features

Our latest updates have introduced new features that improve application management, offering greater functionality and additional options to elevate the user experience.

These new features include:

1. [Applications – New Fields for Add, Export, & Import](#)
2. [Settings – Web UI Tab](#)
3. [Application Tree – Option to Increment Version Number](#)



## 4.1.2.1. Applications – New Fields for Add, Export, & Import

In the **Applications** group of the menu bar, users are provided access to the following set of commands:



In this latest release, we have added new fields to the windows associated with the **Add**, **Export**, and **Import** commands.

### 1. Add – Application Version Number

If the **Add** button is selected, it will display the Application window. In this window, along with the name and title of the application, users can now also set the version number.

- **Version:** This is a numeric updown field that allows users to input the version number of the application.
  - Users can enter this number either manually or by using the increment/decrement arrows.
    - To enable this field, users must first create and save the application. Then, they must right-click on the application from the [Application Tree](#) and select **Change App** from the context menu.

✿ Previously, the program incremented the version number of an application automatically whenever changes were made to the entities. With the addition of this new field, users can now manually set the version number for greater control over the application's versioning.

### 2. Export – Enhanced Settings

If the **Export** button is selected, it will display the Export window. In this window, users can now also choose to exclude sensitive data and automatically increment the application version number after export

**Export**

**Settings**

Application: Gap Analysis (JD)

Export to file:  ...

Password:  Confirm password:

☐ Exclude sensitive data ☐ Auto increment app version after export

**Entities**

☐ Export custom data

|   | Export                              | Name                      | Columns |
|---|-------------------------------------|---------------------------|---------|
| ▶ | <input checked="" type="checkbox"/> | New Gap Analysis          |         |
|   | <input type="checkbox"/>            | Comment Entity            |         |
|   | <input type="checkbox"/>            | Data Analysis Methods Mid |         |
|   | <input type="checkbox"/>            | Actions                   |         |
|   | <input type="checkbox"/>            | Data Analysis Methods     |         |
|   | <input type="checkbox"/>            | NewGapAnalysis            |         |

**Progress**

0%

☒ Proceed ☐ Cancel

- **Exclude Sensitive Data:** This is a checkbox which, if selected, allows users to exclude sensitive data such as connection strings, passwords, and other confidential information from the exported application file.
- **Auto Increment App Version After Export:** This is a checkbox which, if selected, allows users to automatically increment the application version number after export.

✿ If this checkbox is selected, after the export is complete, the version number of the application will be automatically incremented by one.

### 3. Import – Import Timed Event Settings & Override Customized Responsibilities

If the **Import** button is selected, it will display the Import window. In this window, users now have the option to import timed event settings for the application. They can also choose to override customized responsibilities.

The screenshot shows the 'Import' dialog box. In the 'Settings' section, there are five checkboxes. The first checkbox, 'Exported file', is followed by a text input field. The other four checkboxes are 'Get Backup before import', 'Import custom data', 'Import timed event settings', and 'Override Customized Responsibilities (roles and resource assignees)'. The last two checkboxes are highlighted with yellow boxes. Below the settings is a 'Progress' section with a progress bar showing 0%. At the bottom are 'Import' and 'Cancel' buttons.

- **Import Timed Event Settings:** This checkbox determines whether to include timed event settings during the import process.
  - If this checkbox is selected, importing an RBA file and deploying will **update** the server's timed event settings with the values stored in the RBA file.
  - If this checkbox is left unselected, importing an RBA file and deploying will **preserve** the server's timed event settings.

! It's important to note that when importing an RBA file, any timed event settings that do not already exist on the server will automatically take the values specified in the RBA file, regardless of whether this checkbox was selected or not.

- **Override Customized Responsibilities:** This checkbox determines whether to override customized responsibilities (role and resource assignees) during the import process.
  - If this checkbox is selected, all **existing** and **new** task receiver settings on the server will be taken from the RBA file.
    - Task assignees for tasks or task types that **already exist** on the server will be overwritten with the values in the RBA file.
    - Task assignees for tasks or task types that **do not already exist** on the server will be imported from the RBA file.
  - If this checkbox is left unselected, all **existing** task receiver settings on the server will be preserved. All **new** task receiver settings, however, will take their values from the RBA file.
    - Task assignees for tasks or task types that **already exist** on the server are preserved.
    - Task assignees for tasks or task types that **do not already exist** on the server (i.e., are being imported into the server for the first time) will take the task receiver values from the RBA file.

! When importing an RBA file and selecting this checkbox, it's important to note that only the defined and dynamic receiver settings will be overridden. The specific receivers assigned to individual tasks before

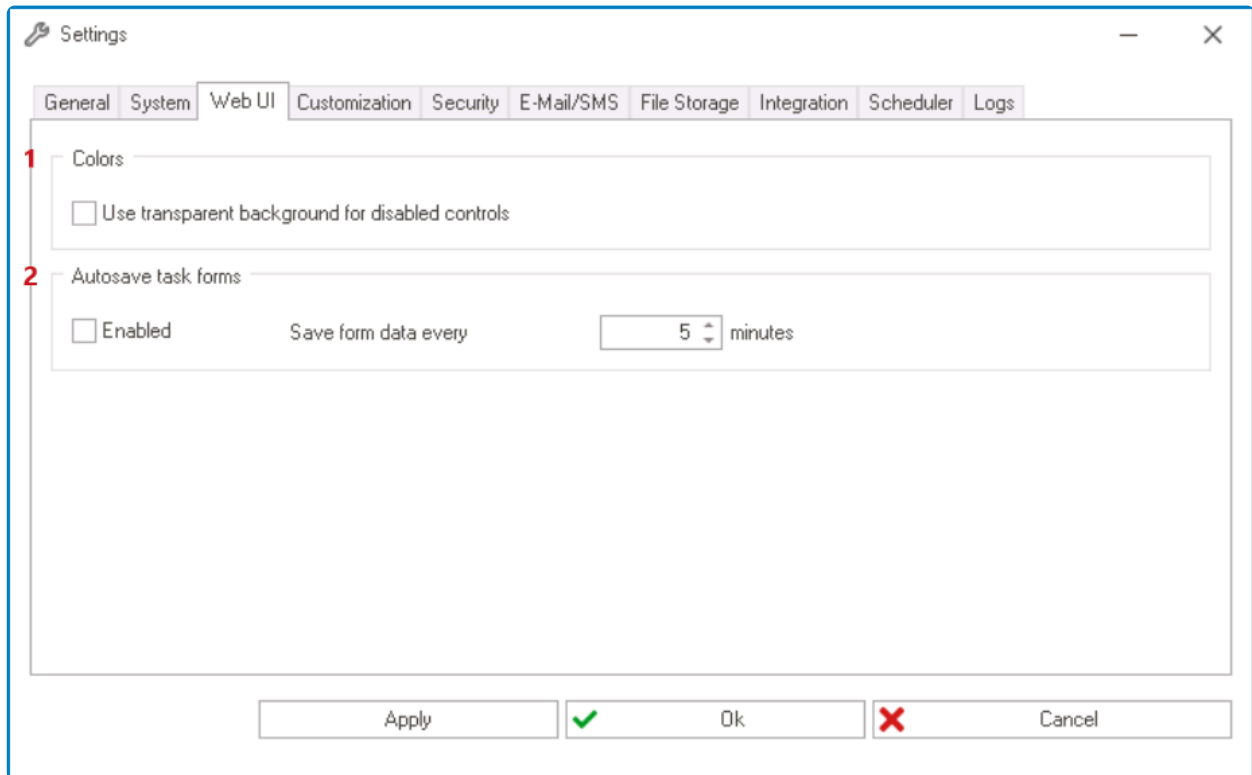
the import will remain unchanged.



For more information on the pre-existing fields in these windows, [click here](#).

## 4.1.2.2. Settings – Web UI Tab

We have added a new tab called Web UI to the **Settings** window. This tab is dedicated to configuring the settings related to the web user interface of the **DBP** application, providing users with a centralized location for managing these options. Additionally, some minor adjustments were made to the layout of the **Settings** window, with certain fields being reorganized to better align with the overall purpose of each tab.



1. **Colors:** In this section, users can modify the color scheme of the interface to suit their preferences.
  - **Use Transparent Background For Disabled Controls:** This is a checkbox which, if selected, allows disabled controls to have a transparent background in the **DBP** application—as opposed to a darker background color.
    - For more details on this feature, [click here](#).

✿ Previously, this checkbox was located in the System tab. It has been moved to to the Web UI tab for better organization of related settings.

2. **Auto-Save Task Forms:** In this section, users can configure the web application to automatically save task forms after a set interval.
  - **Enabled:** This is a checkbox which, if selected, allows users to to enable the auto-save feature.
  - **Save Form Data:** This is a numeric updown field that allows users to set the interval at which the application automatically saves task forms. Users can enter this number either manually or by using the increment/decrement arrows.
    - For more details on this feature, [click here](#).

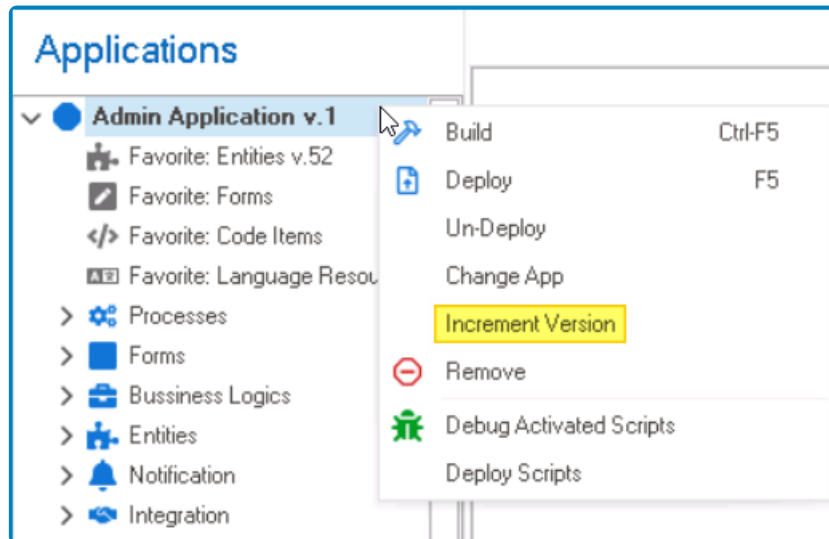




For more information on the pre-existing tabs of the **Settings** window, [click here](#).

## 4.1.2.3. Application Tree – Option to Increment Version Number

We have added a new feature to the Application Tree context menu. When users right-click on an application in the tree, they can now see a new option called **Increment Version**.



- **Increment Version:** This option allows users to manually increase the version number of an application.
  - This can be helpful for keeping track of changes made to the application or for marking significant milestones in the development process.

✿ For more information on the pre-existing options in this context menu, [click here](#).

## 4.1.3. Task Types – Assign Task Receivers in Web

We have introduced a new feature to the **DBP** web application that allows users to assign specific and defined receivers for tasks directly in the web interface. Previously, users could only assign receivers in the [Task Types](#) section of the Designer. Now, with the addition of the new **Form Behavior** application, users can also assign task receivers in the **Process: Task Receiver** form. This feature streamlines the task assignment process and provides greater flexibility to users.

\* The **Form Behavior** application is part of the larger **Admin** application. To access the **Process: Task Receiver** form, users can simply follow the path indicated in the above graphic.

In the **Process: Task Receiver** form, there are two tabs to consider: [Task Types](#) and [Task Type Properties](#).

To manage the assignment of a given task, users must first double-click on it in the [Task Types](#) tab. They will then be automatically redirected to the **Task Type Properties** tab, where they will find the task's properties on display.

### Process: Task Receiver

Task Types

**Task Type Properties**

Application: BCM V11

Title: Review BIA

Receiver's type

☐ Specific ☒ Defined

Specific receivers

Select All

Deselect All

Defined receivers

☐ Current Role

☐ Current Role Department Manager

☐ Starter Role

☐ Starter Role Department Manager

☐ Current User

☒ Starter User

Save



For detailed guidance on the fields within this tab and how to utilize them, [click here](#).



Any changes made to the task receiver in the **DBP** web application will be automatically updated in the Designer, and vice versa.

## 4.1.4. Forms – Save Drafts with Incomplete Required Fields

---

Users can now save forms with incomplete required fields.



Previously, to save a form as a draft, users needed to fill out all of its mandatory fields. With this latest update, users can save partially completed forms and return to them later to complete the required fields before submitting the form.

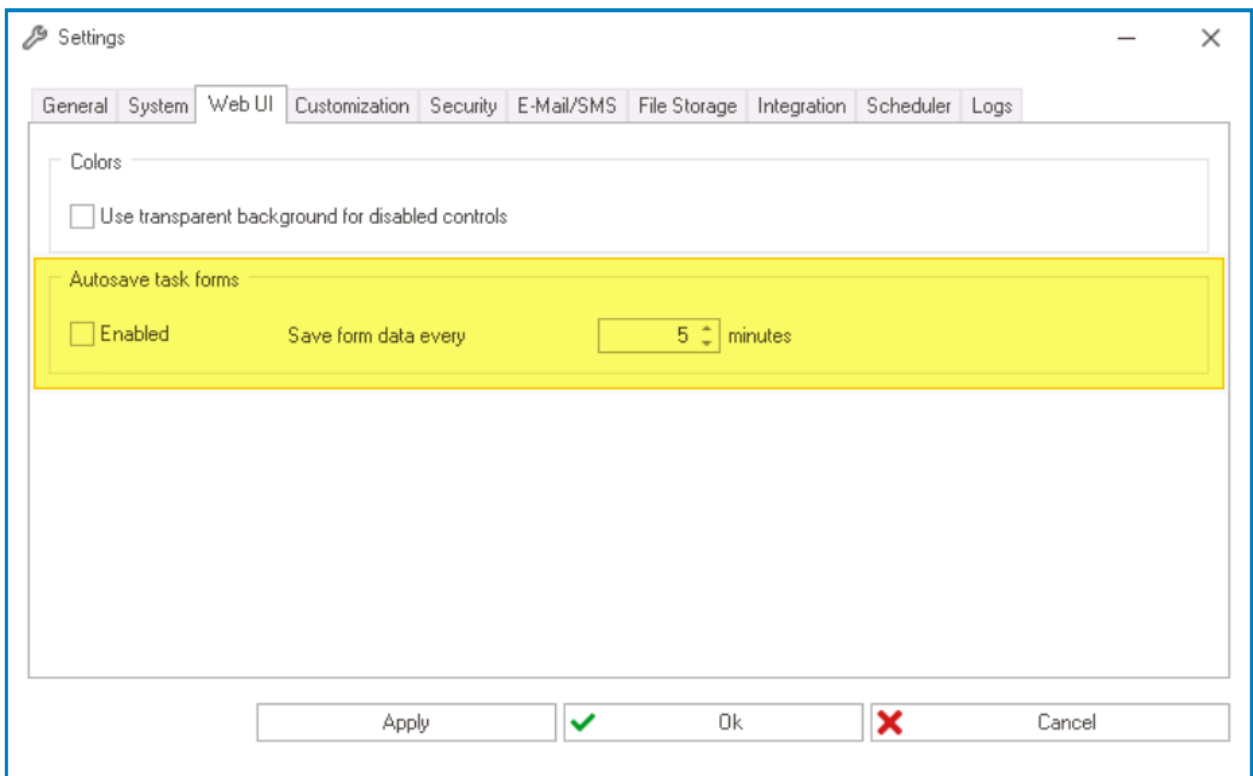
## 4.1.5. Forms – Auto-Save Task Forms

Users can now configure the **DBP** web application to automatically save task forms after a specified interval.

✿ A task form is a form that is linked to a process. It is part of a workflow. If a form is not related to process, it cannot be auto-saved.

Users can enable the auto-save feature for task forms in either the [Settings](#) of the Designer program or the Settings of the **DBP** web application.

- **Designer Settings – Web UI Tab**



The screenshot shows the 'Settings' dialog box with the 'Web UI' tab selected. The 'Autosave task forms' section is highlighted in yellow. It contains a checkbox labeled 'Enabled' and a numeric updown field labeled 'Save form data every' with the value '5' and the unit 'minutes'. The dialog box has tabs for General, System, Web UI, Customization, Security, E-Mail/SMS, File Storage, Integration, Scheduler, and Logs. At the bottom, there are buttons for 'Apply', 'Ok', and 'Cancel'.

- **Enabled:** This is a checkbox which, if selected, allows users to to enable the auto-save feature.
- **Save Form Data:** This is a numeric updown field that allows users to set the interval at which the application automatically saves task forms. Users can enter this number either manually or by using the increment/decrement arrows.

- **DBP Settings – Server-Wide Tab**

## Personalization

Server-wide
User

### Colors

Background color

#0371c3

Background color for required fields

#dbf0ff

Transparent background for readonly fields
☐

Restore default values

### Task Form Auto-Save

Enabled
☐

Interval (min)

5

### Connection to EPC

Base URL

Username

Marie

Password

.....

Test

Default environment

Select...

Save

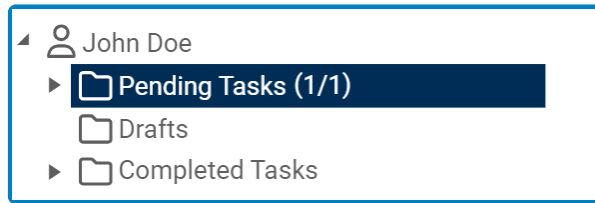
\* Attention: To see the changes, refresh the cache (Ctrl-F5).

- **Enabled:** This is a checkbox which, if selected, allows users to enable the auto-save feature.
- **Interval (Min):** This is a numeric updown field that allows users to set the interval at which the application automatically saves task forms. Users can enter this number either manually or by using the increment/decrement arrows.

✿ By default, the **DBP** web application session times out after 20 minutes of inactivity, so the auto-save interval should be set between 1 and 19 minutes.

An auto-saved form is stored in the same way as any other saved form in the web application.

- If it is the first step in a given workflow, the task can be found in the **Drafts** folder.
- If it is assigned in the middle of a workflow, the task will remain in the **Pending Tasks** folder.



Changes made to the auto-save feature in either the Designer or the **DBP** web application are immediately reflected in the other platform. This real-time synchronization ensures that users can update their auto-save preferences in either platform without having to worry about inconsistencies or delays.



## 4.1.6. Forms – Customize Color for Required Fields

Users can now customize the color of required fields in the **DBP** web application. This new feature enables users to choose their preferred RGB color codes to suit their aesthetic preferences or, if applicable, maintain consistency with their company's branding.

### Personalization

Server-wide

User

#### Colors

Background color  
#0371c3

Background color for required fields  
#dbf0ff

Transparent background for readonly fields

Restore default values

#### Task Form Auto-Save

Enabled

Interval (min)  
5

#### Connection to EPC

Base URL

Username  
Marie

Password  
.....

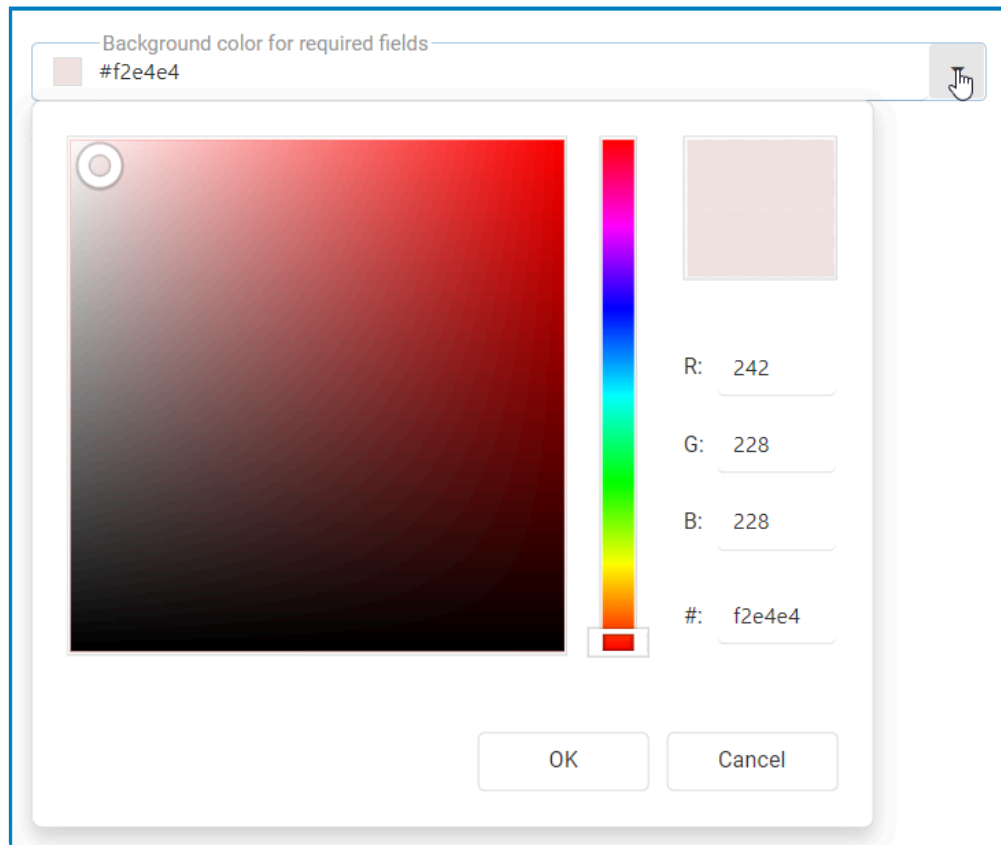
Test

Default environment  
Select...

Save

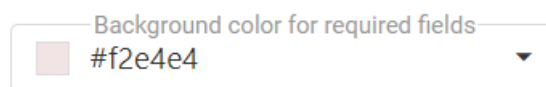
\* Attention: To see the changes, refresh the cache (Ctrl-F5).

- **Background Color for Required Fields:** This is a color picker that allows users to choose the background color of required fields in the **DBP** application.
  - When this field is selected, it displays the following tool:



- The color selection tool offers multiple ways to choose and customize colors.
  - To select a color from a range of options, use the color field and slider. Simply click on the desired color to choose it.
  - To adjust the red, green, and blue components of a color, enter numeric values for each component.
  - To input or modify the hexadecimal code of a specific color, use the # field.

After selecting and saving the desired color, users must refresh the cache (CTRL + F5) to apply the changes in the web application.



## 4.1.7. Forms – Set Disabled Fields as Transparent in Web

Where previously users could only enable this feature in the Designer, they now have the option of setting the background of disabled fields as transparent in the **DBP** web application.

### Personalization

Server-wide

User

#### Colors

Background color

#0371c3

Background color for required fields

#dbf0ff

Transparent background for readonly fields☐

Restore default values

#### Task Form Auto-Save

Enabled☐

Interval (min)

5

#### Connection to EPC

Base URL

Username

Marie

Password

.....

Test

Default environment

Select...

Save

\* Attention: To see the changes, refresh the cache (Ctrl-F5).

- **Transparent Background for Read-Only Fields:** This is a checkbox which, if selected, allows read-only or disabled fields to have a transparent background in the **DBP** application—as opposed to a darker background color.

In this mode, disabled fields nonetheless remain distinguishable from enabled fields in that they cannot be manipulated by the user in any way.



## 4.1.8. Forms – Notify and Redirect for Incomplete Required Fields

---

Users are now notified of incomplete or unfilled required fields when attempting to submit a form in the **DBP** web application. Instead of a simple error message, users are redirected to the specific fields that need to be completed, allowing for a more streamlined and efficient form submission process.



The notification appears at the bottom of the form and lists the names of the fields that require completion. By clicking on the field names, users can easily navigate to the corresponding fields in the form.



This feature applies only to incomplete fields within the currently selected tab. Please note that toggling between tabs is not yet supported.

## 4.1.9. Form Behavior Management – New Features

---

Our latest updates have introduced new features that improve form behavior management, offering greater functionality and additional options to elevate the user experience.

These new features include:

1. [Form Behavior – Access List of Form Behaviors in Web](#)
2. [Form Behavior – Change Properties in Web](#)
3. [Form Behavior – Assign Resource to Multiple Forms in Web](#)

These features have all been made possible by the new **Form Behavior** application in the **DBP**, which enables users to view and manage form behaviors in the **Form Behavior Control** form.



The **Form Behavior** application is part of the larger **Admin** application. To access the **Form Behavior Control** form, users can simply follow the path indicated in the above graphic.



Changes made to form behaviors in either the Designer or the **DBP** web application are immediately reflected in the other platform. This real-time synchronization ensures that users can update their preferences in either platform without having to worry about inconsistencies or delays.

# 4.1.9.1. Form Behavior – Access List of Form Behaviors in Web

We have introduced a new feature to the **DBP** web application that enables users to access the list of form behaviors for all applications directly in the web interface, instead of having to go through the corresponding [Form Behavior](#) section in the Designer program. By streamlining the process of accessing and managing form behaviors, this feature saves users time and effort.

There are three tabs to consider in the **Form Behavior Control** form: [List of Form Behaviors](#), [Form Behavior Properties](#), and [Assign Resource to Multiple Forms](#).

To view a comprehensive list of all available form behaviors across all applications, click on the [List of Form Behaviors](#) tab.

Form Behaviour

List of Form Behavior

Form Behavior Properties

Assign Resource to Multiple Forms

To view the properties for a form behavior, double-click on the respective row in the grid.

|   | Application Name     | Authorized Resources          | Title                    | Type                 | Category                 | Template             |
|---|----------------------|-------------------------------|--------------------------|----------------------|--------------------------|----------------------|
|   | <input type="text"/> | <input type="text"/>          | <input type="text"/>     | <input type="text"/> | <input type="text"/>     | <input type="text"/> |
| 1 | BCM                  | Jane Doe, John Doe, Richar... | Document                 | Starter              | Forms                    | No Tem...            |
| 2 | BCM                  | Jane Doe, John Doe, Richar... | List of Executed Disa... | Starter              | Disaster Recovery        | No Tem...            |
| 3 | BCM                  | Jane Doe, John Doe, Richar... | Business Continuity P... | Starter              | Business Continuity P... | With Te...           |
| 4 | BCM                  | Jane Doe, John Doe, Richar... | Criteria to send to DR   | Starter              | Settings                 | No Tem...            |
| 5 | BCM                  | Jane Doe, John Doe, Richar... | Check List Statuses      | Starter              | Settings                 | No Tem...            |

1

2

3

4

5

...

5

1 - 5 of 112 items

Export

\* When users double-click on a form behavior in this list, they will be redirected to the [Form Behavior Properties](#) tab where they can view and edit the properties of the selected form behavior. [For more information on the features in this tab, click here.](#)

# 4.1.9.2. Form Behavior – Change Properties in Web

We have introduced a new feature to the **DBP** web application that allows users to change the form behavior properties directly in the web interface, instead of having to go through the corresponding [Form Behavior](#) section in the Designer program. By streamlining the process of accessing and managing form behaviors, this feature saves users time and effort.

There are three tabs to consider in the **Form Behavior Control** form: [List of Form Behaviors](#), [Form Behavior Properties](#), and [Assign Resource to Multiple Forms](#).

To change the properties of a given form behavior, users must first double-click on it in [List of Form Behaviors](#) tab. They will then be automatically redirected to the [Form Behavior Properties](#) tab, where they will find the form behavior’s properties on display.

Form Behaviour

List of Form Behavior

Form Behavior Properties

Assign Resource to Multiple Forms

Application: BCM V11

Form Title

Form Behaviour View Order

Category

Category View Order

Business Continuity Plan

1

Business Continuity/I

2

Authorized Resources

Select All

Deselect All

[User] John Doe

[User] Jane Doe

[User] Richard Roe



Select Template Output Type

All file types

Form Title in English - US

Business Continuity Plan

Form Title in French

Plan de continuité des activités

Form Title in German

Geschäftskontinuitätsplan

Form Title in Arabic

خطة استمرارية العمل

Form Title in Swedish

Kontinuitetsplan

Form Title in Hebrew

תכנית המשכיות עסקית

Word Template for English - US ?

No file exists

Word Template for French ?

No file exists

Word Template for German ?

No file exists

Word Template for Arabic ?

No file exists

Word Template for Swedish ?

No file exists

Word Template for Hebrew ?

No file exists

Save the Changes

In this tab, users can view and modify the basic properties of the form behavior, including its title, view order, category, and authorized resources. They can also manage the template settings for the form behavior.



For more detailed instructions on setting and managing these properties, [click here](#).

## 4.1.9.3. Form Behavior – Assign Resource to Multiple Forms in Web

We have introduced a new feature to the **DBP** web application that allows users to assign a resource to multiple forms directly in the web interface.

There are three tabs to consider in the **Form Behavior Control** form: List of Form Behaviors, Form Behavior Properties, and Assign Resources to Multiple Forms.

In the Assign Resources to Multiple Forms tab, a single resource (a department, role, user, or group) can assigned to multiple forms at once.

### Form Behaviour

List of Form Behavior

Form Behavior Properties

**Assign Resource to Multiple Forms**

Assigned Resource

Select...

Assign

Forms

Select All

Deselect All



For detailed guidance on the fields within this tab and how to utilize them, [click here](#).

## 4.1.10. Applications & Form Behaviors – Search Functionality

---

Users can now search for specific applications and starter form behaviors in the **Home Setup** form of the **DBP** web application. By simply typing the name of the desired application or form behavior into the search bar, the list will filter accordingly.



As users enter characters into the search box, the menu items that match the entered text in their name or shown name will become visible while the others are filtered out. To remove the search filter and display the full list, users can simply click on the **x** button.



Regarding form categories, please note that the search function filters results based on the name of the categories, rather than their shown name. In other words, if the shown name of a category does not match its name, the search may either yield inaccurate results or no results at all.

## 4.1.11. Form Category – Change View Order in Web

We have introduced a new feature to the **DBP** web application that enables users to reorganize the view order of form categories directly in the web interface, instead of having to go through the [Form Categories](#) section in the Designer program. This is made possible by the **Home Setup** form in the **DBP** web application.

### Home Settings

To add menus to Home page, select the item and change its settings

Menu items

- QMS
  - Admin Settings
    - Audit App Settings
      - Test Methods
      - Audit type
      - Audit Related To List
      - Audit Method
      - Team list
    - Audit**
      - Test Management
        - Evidence Collection Req...
        - New Test
        - Tests List
      - Requirement Management...
        - Import Requirements fr...
        - New Requirement
        - Requirements List
      - Define Audit
        - New Audit Definition
        - List of Defined Audit
      - Execute Audit
        - Audit Plans-Waiting for ...
        - New Audit Plan
        - List of All Audit Plans
        - List of Audit Plans in Pr...
        - List of Completed Audit ...
        - My Team's Audit Plans ...
        - My Audit Plans (Auditor...
        - Trigger Scheduled Audit...
      - Audit Dashboard
      - Audit Summary Report
    - Manual(s)
      - Audit Manual

Menu item settings

☒ Show tile on Home page
 ☒ Is container menu

☒ Show in first level
 ☐ Hide in top menu

Category Order

Shown name

French name

German name

Arabic name

Swedish name

Hebrew name

Image shape

☒ Circle
 ☐ Square
 ☐ Rectangle

Image width

Image height

Text Location

☐ Top Image
 ☐ Center Image
 ☒ Bottom Image

Text Color

Float Text Co...

Upload image

Delete Image

Please select a file with jpg, png or gif format

Font Awesome icon

Audit

Save

**Attention!**  
To see the changes, refresh the page

- **Category Order:** This field allows users to set the view order of the form category—that is, the order in which the category appears in the web application.



In the **DBP** web application, each application or starter form can be displayed under a specific form category, which acts as a folder. By reorganizing the view order of these form categories, users can affect the order in which the icons of the applications and forms are displayed on the web, as well as the order in which the forms are listed if no icon is associated with them.

## 4.1.12. Notifications – Attach Template Output to Email

Users can now opt to attach the template output to their email notifications. This option is presented as a checkbox in the [Notification Item Properties](#) window, which is located within the **Notification** module.

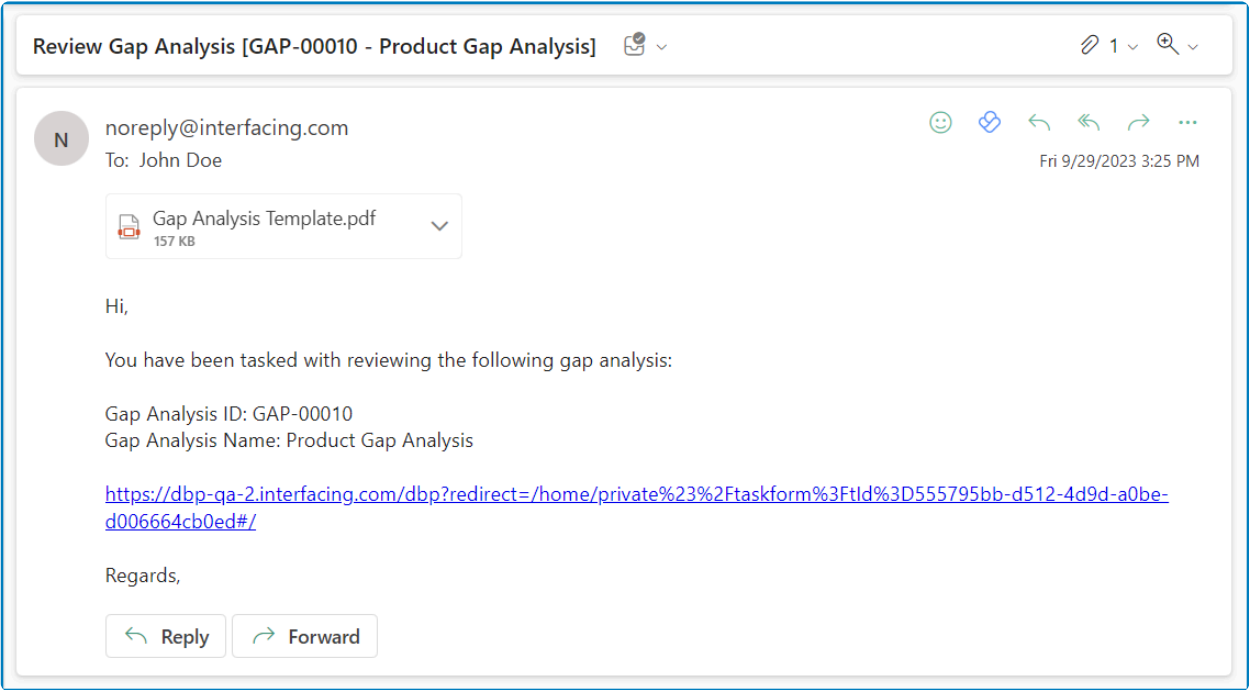
The screenshot shows the 'Notification Item Properties' dialog box. It has a title bar with a bell icon and a close button. Below the title bar are three buttons: 'Apply' (with a green square icon), 'OK' (with a green checkmark icon), and 'Cancel' (with a red X icon). The main area contains several fields: 'Entity' is a dropdown menu showing 'New Gap Analysis (New Gap Analysis)'; 'Name' and 'Title' are empty text boxes; 'Subject Data Item' and 'Body Data Item' are dropdown menus showing '[Empty]', each with '+' (add), 'edit' (pencil), and '-' (remove) icons; 'Notification Agent' is a dropdown menu showing 'Default'; 'Attach exported file' is a checkbox, highlighted with a yellow background; 'Asynchronous Call' is a checked checkbox; 'Recipients' is a section with 'Object Property' (a text box with a table icon and a red minus icon) and 'Task Principals' (a checkbox); and 'Task Type' is a dropdown menu showing '[Empty]' with '+' (add), 'edit' (pencil), and '-' (remove) icons.

- **Attach Exported File:** This is a checkbox which, if selected, allows users to attach the exported template file to their (email) notification.

### Template Attachment – Workflow Implementation

In a practical workflow, each task (e.g., [Task A](#), [Task B](#), [Task C](#)) can be assigned a unique template and notification. Enabling the **Attach Exported File** checkbox for each task ensures that its corresponding template is attached to the email notification.

For example, if [Task A](#) has an assigned template, enabling this option will attach [Task A](#)'s template to the email, and the same applies to [Task B](#) and [Task C](#).



✿ This feature streamlines the workflow of an application, ensuring that all relevant information is readily available in the user’s notifications.

## 4.1.13. Sequence Service

---

We have added a new service that can be used in custom code to automatically increment numeric values by 1.

- To demonstrate how the sequence service can be used, consider the following sample code:

```
var q = bpmAppService.BPMSServices.MiscService.GetSequenceNumber("TestSequence1", 1);  
formObject.Message1 = string.Format("New sequence #: {0:n0}", q);
```

## 4.1.14. EPC Services

---

To simplify the process of calling the EPC API, we have added new EPC services to the code. With these services, users can easily integrate EPC functionality into their applications without the need for complex coding.

The expanded suite of services includes the following:

- **User Authentication**

Users can securely log in to EPC (or connect to it using global settings) directly from the **DBP** web application.

- **Environment & Module Retrieval**

Users can retrieve individual components from EPC, including environments, users, documents, processes, rules, controls, capabilities, risks, and organization units.

- **Action Creation and Deletion**

Users can create and delete actions as needed.

- **Node Description Retrieval**

Users can retrieve descriptions for specific EPC nodes by searching for them using their unique identifiers.



## 4.2. Feature Improvements

---



### What's Been Improved?

With this latest release, we have applied the following feature improvements:

1. [Performance Improvements](#)
2. [RBA Import](#)
3. [RBA Deployment](#)
4. [SSO Connection – Password Change Not Required](#)
5. [Modernized UI for Disabled Fields](#)
6. [Form Validation – Block Duplicate Submissions](#)
7. [Form Designer – Control Improvements](#)
8. [Code Designer – Added Author, Description, & History to Code](#)
9. [General UI Enhancements](#)

## 4.2.1. Performance Improvements

---

This release brings significant improvements to the performance and speed of the **DBP** web application. Several calls have been inspected and rectified, resulting in faster speeds and an overall improvement in loading time. These improvements have been made in the backend and will provide a more efficient and responsive user experience.

## 4.2.2. RBA Import

---

We have made substantial improvements to the speed of importing RBA files into the Designer program. This is especially beneficial for users importing larger RBA files, as they can now complete the process in a matter of seconds.

## 4.2.3. RBA Deployment

Following an optimization of the deployment process, we have significantly improved the speed of deploying RBA files to the **DBP** web application.

✿ This improvement is particularly noticeable when deploying larger RBA files, which can now be completed in a fraction of the time it previously took.

Additionally, users will now be informed of the duration of the build process when deploying an application. This information can be found in the [Output](#) section of the Designer, where a new metric will display the length of time it took to build the application. The metric will read 'Build ended. Took [X] sec.' where '[X]' represents the number of seconds the build process took.

| Output  |  |
|---------|--|
| 8:25 PM | [Gap Analysis (JD)] Processes: Built successfully                                      |
| 8:25 PM | [Gap Analysis (JD)] Start Publish  |
| 8:25 PM | [Gap Analysis (JD)] Process 'New Gap Analysis', version 2 : published                  |
| 8:25 PM | [Gap Analysis (JD)] All process version of all processes of GapAnalysisJD is published |
| 8:25 PM | [Gap Analysis (JD)] Rules: Built successfully  |
| 8:25 PM | [Gap Analysis (JD)] Actions: Built successfully  |
| 8:25 PM | [Gap Analysis (JD)] Processes has been built successfully                              |
| 8:25 PM | [Gap Analysis (JD)] Workflows: Built successfully                                      |
| 8:25 PM | [Gap Analysis (JD)] Task: Built successfully   |
| 8:25 PM | [Gap Analysis (JD)] RAG File: Built successfully                                       |
| 8:25 PM | [Gap Analysis (JD)] Debug activated bussiness logic: 'Action'                          |
| 8:25 PM | [Gap Analysis (JD)] 1 item(s) succeeded  |
| 8:25 PM | [Gap Analysis (JD)] Debug activated bussiness logic: 'DynamicTaskInstance'             |
| 8:25 PM | [Gap Analysis (JD)] 1 item(s) succeeded  |
| 8:25 PM | [Gap Analysis (JD)] Debug activated bussiness logic: 'DynamicParameters'               |
| 8:25 PM | [Gap Analysis (JD)] 3 item(s) succeeded  |
| 8:25 PM | [Gap Analysis (JD)] BPMS Application: Built successfully                               |
| 8:25 PM | [Gap Analysis (JD)] Build ended. Took 32 sec   |
| 8:25 PM | [Gap Analysis (JD)] Generate Project Form Files Succeeded                              |
| 8:25 PM | [Gap Analysis (JD)] Form: Built successfully   |
| 8:25 PM | [Gap Analysis (JD)] TimedEvent deployed successfully                                   |
| 8:25 PM | [Gap Analysis (JD)] Application Resources Published Successfully                       |
| 8:25 PM | [Gap Analysis (JD)] Application Resources Published Successfully                       |
| 8:25 PM | [Gap Analysis (JD)] BPMS Application: Deployed successfully                            |

## 4.2.4. SSO Connection – Password Change Not Required

---

We have applied an improvement to the Single Sign-On (SSO) feature, which removes the requirement for SSO-connected users to change their password when logging in for the first time.



Prior to this improvement, the **DBP** web application prompted all [first-time connections](#) to change their password. This pop-up window has now been removed for users who have enabled and are connecting with SSO.

**Note:** Users can enable SSO with Active Directory (AD) in the [Security](#) tab of the Designer Settings.

## 4.2.5. Modernized UI for Disabled Fields

---

We have improved the user interface (UI) of disabled fields in the **DBP** web application. Users will now experience a cleaner and more modern look without any unnecessary border lines, buttons, or icons on the disabled fields.



Users can set the background of disabled fields as transparent in either the [Settings](#) of the Designer or the [Settings](#) of the **DBP** web application.

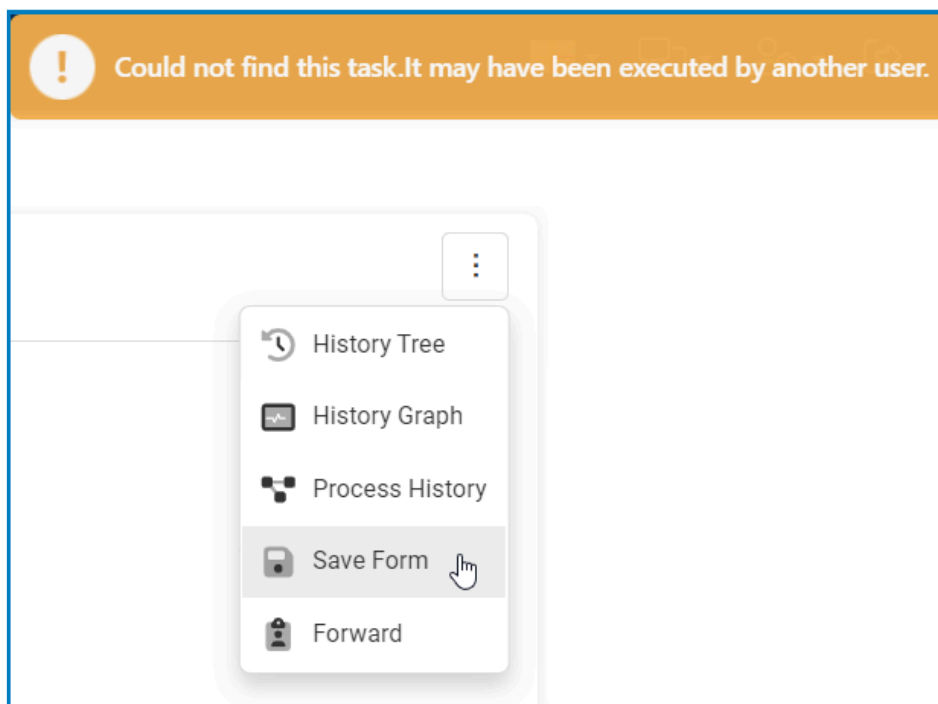
## 4.2.6. Form Validation – Block Duplicate Submissions & Overwrites

We have implemented a feature in the **DBP** web application that warns users when they attempt to save or submit a task that has already been completed by another user.

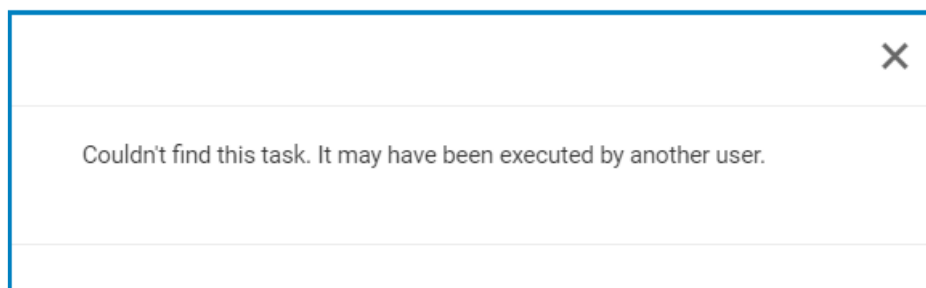
- \* This feature helps to prevent duplicate submissions and overwrites, which is particularly useful in scenarios where multiple users have the same task open.

When the user attempts to save or submit the task, a check is performed to see if the task has already been completed. If so, a clear message is displayed to the user, informing them of the fact and preventing them from saving or submitting the task again.

- **Save – Message**



- **Submit – Message**



## 4.2.7. Form Designer – Control Improvements

---

Our latest updates to the [Form Designer](#) tool in the Designer program bring key improvements to the controls, resulting in increased functionality and usability.

1. [Control Governance – Improvements](#)
2. [Control Caption – Option to Keep Reserved Space](#)
3. [Grid Control – Range Validation](#)
4. [Tab Control – Reflect Web Appearance in Designer](#)



These updates offer new customization options and enhanced capabilities for the controls, allowing users to create more engaging and intuitive interfaces for their web applications.



# 4.2.7.1. Control Governance – Improvements

Our latest update includes an improvement to the process by which control governance properties are configured in the Form Designer tool. Specifically, we have expanded the functionality of the **State Matrix** field, now allowing users to set control properties and validation rules for multiple form behaviors simultaneously.

## Example: State Matrix for Grid Control – Expanded Configuration Options

To illustrate this improvement, consider the window linked to the **State Matrix** field for the grid control.

Control Behaviour

Apply

OK

Cancel

| Form Behavior          | Control State |                          | Add |                          | Inline Add |                          | Edit |                          | Inline Edit |                                     | Delete |                          | View |                                     | Export                              |  | Import |                          | Enable Select |                          | Is Required |  |
|------------------------|---------------|--------------------------|-----|--------------------------|------------|--------------------------|------|--------------------------|-------------|-------------------------------------|--------|--------------------------|------|-------------------------------------|-------------------------------------|--|--------|--------------------------|---------------|--------------------------|-------------|--|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            | <input type="checkbox"/> |      | <input type="checkbox"/> |             | <input checked="" type="checkbox"/> |        | <input type="checkbox"/> |      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |        | <input type="checkbox"/> |               | <input type="checkbox"/> | ...         |  |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            | <input type="checkbox"/> |      | <input type="checkbox"/> |             | <input checked="" type="checkbox"/> |        | <input type="checkbox"/> |      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |        | <input type="checkbox"/> |               | <input type="checkbox"/> | ...         |  |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            | <input type="checkbox"/> |      | <input type="checkbox"/> |             | <input checked="" type="checkbox"/> |        | <input type="checkbox"/> |      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |        | <input type="checkbox"/> |               | <input type="checkbox"/> | ...         |  |

Generally, in this window, users were previously limited to configuring the control state for each form behavior. With our latest update, however, users can now also perform the following actions:

- Enable control properties or functions on a per-form-behavior basis
  - To enable a specific function, select the corresponding checkbox (located either directly under the function or to its left).

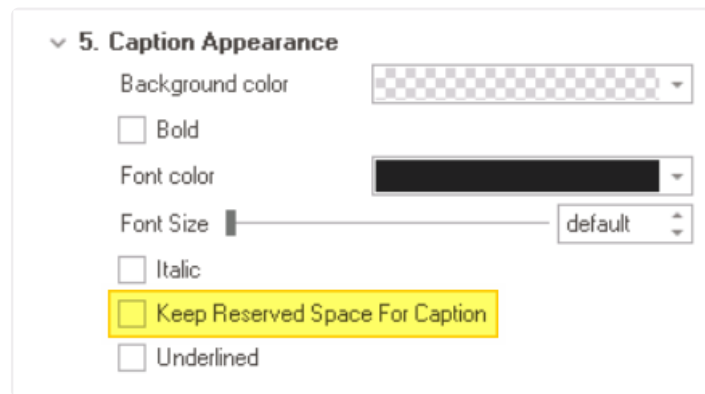
| Form Behavior          | Control State |                          | Add |                          | Inline Add |
|------------------------|---------------|--------------------------|-----|--------------------------|------------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |

- Set the corresponding form behavior for each enabled function (where applicable)
  - Once a function has been enabled for the control, users can select the form behavior associated with it.
- Set validators for the control on a per-form-behavior basis
  - Depending on the control, users can enable required field and/or range validation.

This update streamlines the process of editing control properties and validators across multiple form behaviors, improving overall efficiency and user experience.

## 4.2.7.2. Control Caption – Option to Keep Reserved Space

In the [Form Designer](#), users can now remove the reserved space used for a control's caption when no caption is specified. This new option can be found in the [Caption Appearance](#) category of the Control Properties Panel, and provides greater flexibility for aligning controls on a form.



- **Keep Reserved Space for Caption:** This is a checkbox which, if selected, allows users to reserve space for the control's caption even when there is no caption set.

To illustrate this improvement, consider the following example:

| Category    | Source      |
|-------------|-------------|
| Select... ▼ | Select... ▼ |

- If the caption for the the first single-select dropdown control (currently labelled Category) is removed and the **Keep Reserved Space for Caption** checkbox is selected, it will appear as follows:

|             | Source      |
|-------------|-------------|
| Select... ▼ | Select... ▼ |

- If the caption for the the control is removed but the **Keep Reserved Space for Caption** checkbox is left unselected, it will appear as follows:

|             | Source      |
|-------------|-------------|
| Select... ▼ | Select... ▼ |

## 4.2.7.3. Grid Control – Range Validation

Users can now add range validation to date columns in the grid control of the [Form Designer](#).

The image shows a 'Validator form' dialog box with a close button (X) in the top right corner. It contains two checkboxes: 'Has Required Field Validator' (unchecked) and 'Has Range Validator' (checked). Below these are two tabs: 'Required Field Validator' and 'Range Validator'. The 'Range Validator' tab is active, showing an 'Error message' section with a table. The table has three columns: 'Language', 'Is Default', and 'Text'. It lists 'English', 'French', and 'German'. 'English' is selected as the default with a checked box. Below the table are two date picker fields labeled 'Minimum Date Value' and 'Maximum Date Value'. At the bottom right are 'Save' and 'Cancel' buttons.

| Language | Is Default                          | Text |
|----------|-------------------------------------|------|
| English  | <input checked="" type="checkbox"/> |      |
| French   | <input type="checkbox"/>            |      |
| German   | <input type="checkbox"/>            |      |

Minimum Date Value:

Maximum Date Value:

Save Cancel

By selecting a minimum and maximum date value from the date picker, users can ensure that the date values in the column fall within the specified range.

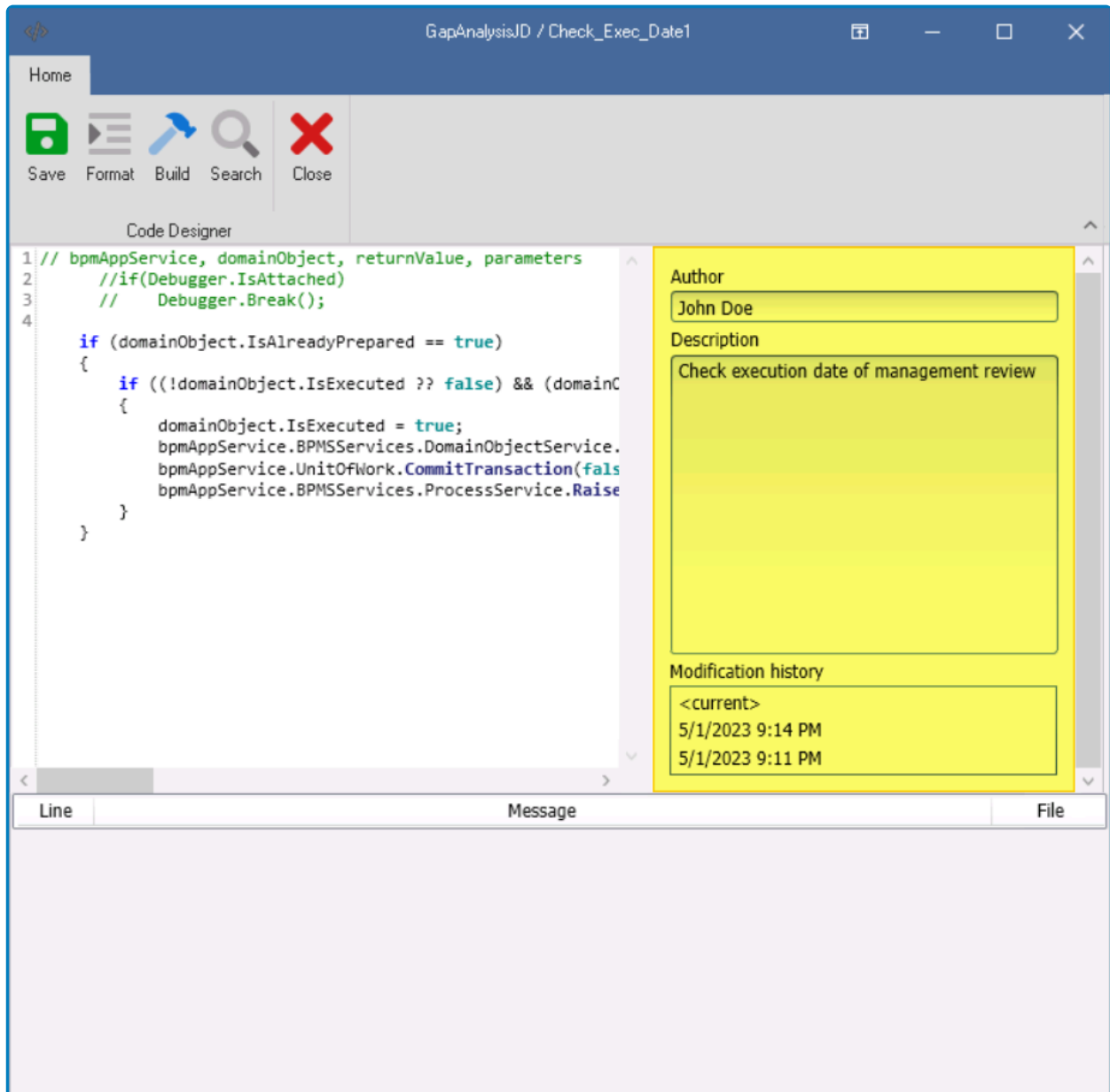
## 4.2.7.4. Tab Control – Reflect Web Appearance in Designer

---

Changes made to the appearance of the tab control in the Form Designer are now immediately reflected in the design environment, providing users with a more accurate representation of how the control will appear in the web application.

## 4.2.8. Code Designer – Added Author, Description, & History to Code

We have added a new panel to the [Code Designer](#) that makes it easier for users to track changes and collaborate on code. The panel, located to the right of the code environment, contains three fields: **Author**, **Description**, and **Modification History**.

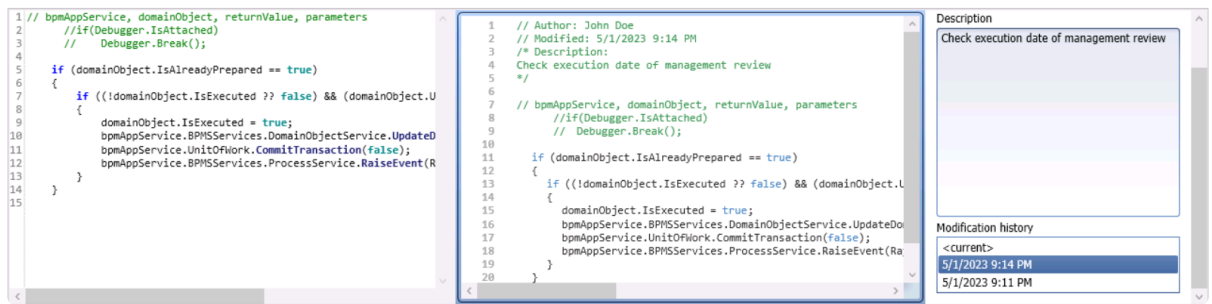


- When users make changes to the code, they can input their name and a brief description of the task they are working on in the **Author** and **Description** fields, respectively.

✿ This information is then saved and can be accessed later by other users who are working on the same code.

- In the **Modification History** field, users can access a log of all changes made to the code, including the date and time of each modification.

- When a specific entry is selected in this field, a small panel appears displaying the code as it was saved by the author at that time. The panel also provides information about the author, including their description, date, and time.



By enabling users to quickly identify the responsible author of each modification, this enhancement streamlines code management and facilitates teamwork.

## 4.2.9. General UI Enhancements

---

We are pleased to introduce a range of refinements that enhance the user experience in both the Designer program and **DBP** web application. These updates bring a new level of efficiency, clarity, and visual polish to the interface, ensuring improved readability and overall usability.

1. [Designer UI Enhancements](#)
2. [Web UI Enhancements](#)

## 4.2.9.1. Designer UI Enhancements

We have implemented key enhancements to the user interface of the Designer program.

Changes in this latest release include:

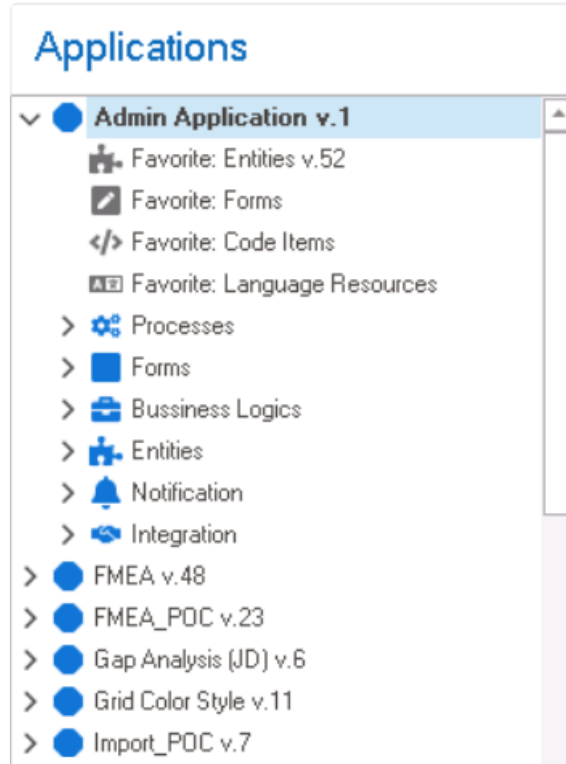
### 1. Menu Bar – Dynamic Labelling

The label of the [Current Application](#) group in the menu bar now dynamically updates to display the name of the currently selected application in the [Application Tree](#). This provides users with a visual cue to ensure that the commands they execute are applied to the intended application.



### 2. Application Tree – Visual Highlighting of Selected Application

In the [Application Tree](#), the selected application is now highlighted and shown in bold text, providing a clear visual indication of the active application. This enhancement improves the user experience by making it easier to quickly locate and work with the desired application.



### 3. Form Designer Toolbox – New Look

In the [Form Designer](#), we have updated the look of the toolbox by refreshing the icons and switching to a monochromatic color palette. This new palette utilizes shades of dark blue and white, which enhance the



visual presentation of the toolbox with a more modern, cohesive design.

## 4.2.9.2. Web UI Enhancements

We have implemented an array of enhancements to the user interface of the **DBP** web application.

Changes in this latest release include:

### 1. Revised Default Color Scheme for Platform Background and Mandatory Fields

We have made adjustments to the default color scheme for the platform's background and mandatory fields.

- The platform's previous dark blue background (#002D56) has been replaced with a brighter and more vibrant blue (#0371C3), creating an atmosphere that is fresh and engaging.
  - **UI Evolution: Before vs. After**
- The color for required fields has been changed from light-yellow (#FFFFE0) to light-blue (#DBF0FF), complementing the new default background color and enhancing visibility by addressing the challenge of the previous color's low contrast.
  - **UI Evolution: Before vs. After**

Analysis Name



Users can customize the platform's background and mandatory field colors through the Settings in the **DBP** web application.

### 2. Timestamp Removal

We have removed the timestamp, which was previously located in the upper right corner above forms, to declutter the interface and optimize space.

### 3. Form Elements – Modernized Border Styles



We have redesigned form elements to provide a more contemporary look and feel.

- **Forms:** Forms now showcase rounded edges, contributing to a softer, more pleasing aesthetic.

- **UI Evolution: Before vs. After**

- **Fields:** Text fields, dropdowns, and other relevant input fields now feature rounded edges and borders on all sides.

- **UI Evolution: Before vs. After**

| Analysis Name        | Analysis Date  | Analysis Type  |
|----------------------|--|--|
| <input type="text"/> | <input type="text"/>  | <input type="text"/>  |

- **Buttons:** Buttons now also sport rounded edges for a cohesive design.





- **UI Evolution: Before vs. After**

|  |  |
|--|--|
| <input type="button" value="Submit form"/> | <input type="button" value="Save Form"/> |
|--|--|

#### 4. **Form Elements – Font Size Adjustments**

To enhance readability and elevate the overall visual appeal, we have implemented a series of font size adjustments across various elements of our forms:

- **Form Titles:** We have increased the font size of form titles from 17px to 25px.
  - **UI Evolution: Before vs. After**
- **Control Caption:** We have increased the default font size of control captions from 13px to 16px.
  - **UI Evolution: Before vs. After**
- **Control Values:** We have increased the font size of values within controls from 13px to 15px.
  - **UI Evolution: Before vs. After**

| Analysis Name  | Analysis Date   | Analysis Type  |
|--|---|--|
| <input type="text" value="Product Gap Ana"/>  | <input type="text" value="15/08/2023"/>   | <input type="text" value="Surveys"/>  |

#### 5. **Style Change for Form-Specific Menu**

We have made an adjustment to the form-specific menu layout. Instead of the previous horizontal arrangement, we have opted for a more compact kebab menu. This change ensures efficient access to contextually relevant actions while maximizing available space within the interface.

#### 6. **Rows – Increased Vertical Padding**

We have increased the vertical padding between rows, allowing for a cleaner, more organized layout.

## Increased Vertical Padding: Designer View

### 7. Control Tooltip – Size & Color Scheme Adjustment

To ensure better visibility, we have enlarged control tooltips and adopted a vibrant blue color scheme, replacing the previous monochromatic grayscale.

**Business Division** ?

### 8. Label Control – Improvements

We have enhanced the visual appeal of the label control by implementing two key changes.

- Firstly, we have added padding to the left and right sides of the control, creating a more balanced and aesthetically pleasing layout.
  - UI Evolution: Before vs. After**



**Identify the scope for the gap analysis**

**Identify the scope for the gap analysis**

- Secondly, we have included vertical padding between the label control and other controls, but only when the **Has Height** checkbox has been selected. This change improves the overall layout and makes it easier for users to distinguish between different elements in the form.
  - UI Evolution: Before vs. After**

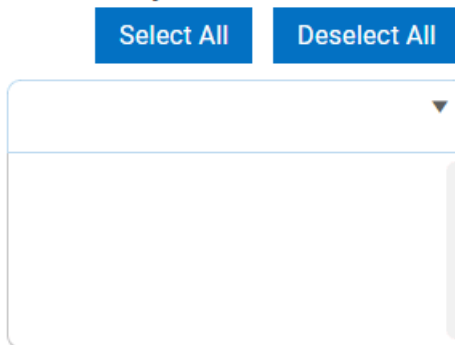
! In addition to the aforementioned improvements, we have removed the **Initial Value** field from the Control Properties panel for the label control after a review showed it to be unnecessary.

## 9. Multi-Select Dropdown Control – Improvements

We have changed the color scheme of the **Select All** and **Deselect All** buttons in the multi-select dropdown control.

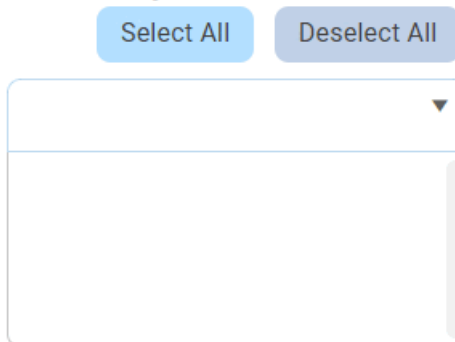
- **Select All** now features a brighter shade of blue that sets it apart from **Deselect All**.
  - **UI Evolution: Before vs. After**

### Data Analysis Methods



- As users hover over these buttons, the color now dynamically adjusts to match the platform's background.
  - **UI Evolution: Before vs. After**

### Data Analysis Methods



## 10. Numeric Textbox Control – Improvements

We have upgraded the numeric textbox control by adding up/down arrows, simplifying the input of numerical values and providing users with a more intuitive interaction.

### Cost



11. Tab Control – Improvements

We have enhanced the visual appeal of the tab control by implementing two key changes.

- Firstly, we now only apply the **bold** effect to the active or currently selected tab within the set, creating a clear visual distinction.
  - UI Evolution: Before vs. After**

|                      |              |                  |                     |               |
|----------------------|--------------|------------------|---------------------|---------------|
| <b>Audit Details</b> | <b>Scope</b> | <b>Resources</b> | <b>Requirements</b> | <b>Agenda</b> |
|----------------------|--------------|------------------|---------------------|---------------|

- Secondly, we have redesigned the tab set with a white background, departing from the previous transparent design. This alteration ensures that, when the floating tab is enabled, it no longer interferes with the text in the form.
  - Tab Control: White Background Preview**

Home Page / Quality Management System / Audit / Requirements Management / Import Requirements from EPC

Import Requirements from EPC

Import Control **Import Document** Import Process Import Capability Import Rule

In this tab, you can select which document(s) to import from EPC.

Folder

Process Docur

Import All

Import Selected

Select All

Deselect All

EPC Documents

Inline Edit

Export to Excel

| Parent                      | Identifier | Name  | Selected For Import |
|-----------------------------|------------|---|---------------------|
|                             |            |   |                     |
| 13.4 Manage Change Reque... |            | 13.4.4 Assess and Evaluate the Change Request | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.5 Authorize the Change Request           | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.3 Review Request                         | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.1 Create and Record the Change Request   | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.2 Manage Emergency Change Requests       | ✓                   |

12. Drafts Folder – Task Context Menu Removal

After reviewing the **Drafts** folder, we have found that the task context menu for saved forms is unnecessary. As a result, we have removed it to simplify the interface.

OLD UI

John Doe

- Pending Tasks (2/4)
- Drafts (1/1)**
- Completed Tasks

Drafts

Open in new window

| Flag | Priority | Subject                   | Created at   | Response... | Done at | Sender   |
|------|----------|---------------------------|--------------|-------------|---------|----------|
|      |          | Saved CAPA [CAPA-00016... | 4/21/2023... |             |         | John Doe |

Label

## 4.3. Fixed Issues

---



### What's Been Fixed?

With this latest release, we have resolved the following issues:

1. [Task Handling – Task Name Used When No Title is Provided](#)

# 4.3.1. Task Handling – Task Name Used When No Title is Provided

We have implemented a fix to address a previous issue in the **DBP** web application. Previously, if a user left the title of a task empty, they were unable to open or act upon it in the web interface. This was due to the creation of an empty subject for the task.

Pending Tasks

Open in new window

Change folder

Search...

| <input type="checkbox"/>            | Flag | Priority | Subject                      | Created at     | Response... | Done at | Sender   |
|-------------------------------------|------|----------|------------------------------|----------------|-------------|---------|----------|
| <input checked="" type="checkbox"/> |      |          |                              | 27/04/2023 ... |             |         | John Doe |
| <input type="checkbox"/>            |      |          |                              | 27/04/2023 ... |             |         | John Doe |
| <input type="checkbox"/>            |      |          | Review CAPA [CAPA-INC-00031] | 27/04/2023 ... |             |         | John Doe |

To resolve this issue, we have made it so that if no title is provided, the task name is used instead in the web interface.


Pending Tasks

Open in new window

Change folder

Search...

| <input type="checkbox"/>            | Flag | Priority | Subject                      | Created at     | Response... | Done at | Sender   |
|-------------------------------------|------|----------|------------------------------|----------------|-------------|---------|----------|
| <input checked="" type="checkbox"/> |      |          | Sys_ReviewGapAnalysis        | 27/04/2023 ... |             |         | John Doe |
| <input type="checkbox"/>            |      |          | Sys_ReviewGapAnalysis        | 27/04/2023 ... |             |         | John Doe |
| <input type="checkbox"/>            |      |          | Review CAPA [CAPA-INC-00031] | 27/04/2023 ... |             |         | John Doe |



In the event that the task name is used, it will be indicated with the prefix “Sys\_”. For example, if the task name is “ABC”, it will appear as “Sys\_ABC”.



## 4.4. Known Limitations

---



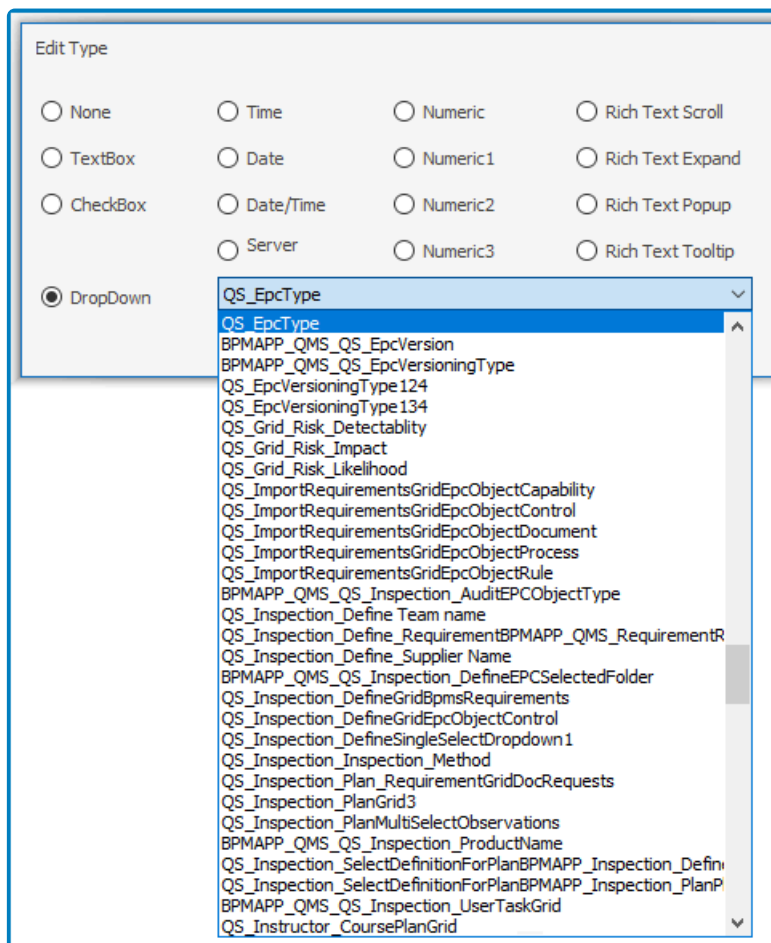
### Known Limitations

In this latest release, we have identified the following limitations:

1. [Sorting in Data Source Dropdown](#)
2. [Import – Override Customized Responsibilities](#)
3. [Grid Control – Quick Forms Feature](#)
4. [Pending Tasks Folder – Tasks/Workflows Triggered from Grid](#)

## 4.4.1. Sorting in Data Source Dropdown

In the current version of the program, there is a known limitation with sorting in the data source dropdown, which can make it challenging to locate the desired query statement.



✿ To locate a specific query statement, please refer to the following organization and sorting system.

### A. Organizational Formats for Data Sources

Data sources can be organized in three formats:

#### 1. Simple Format

These data sources contain a single word or phrase as the title, which follows the naming convention of [Title].

- For example, “QS\_Definitions” or “Categories”.

#### 2. Complex Format

These data sources follow a specific naming convention of BPMAPP\_[ApplicationName]\_[Title].

- For example, “BPMAPP\_QMS\_QS\_Audits”.

### 3. **System Format**

These data sources are generated by the system and are listed last.

- For example, “Users”.

## **B. Sorting of Data Sources in the Dropdown**

Depending on the format of the data source’s name, the sorting system in the dropdown will vary.

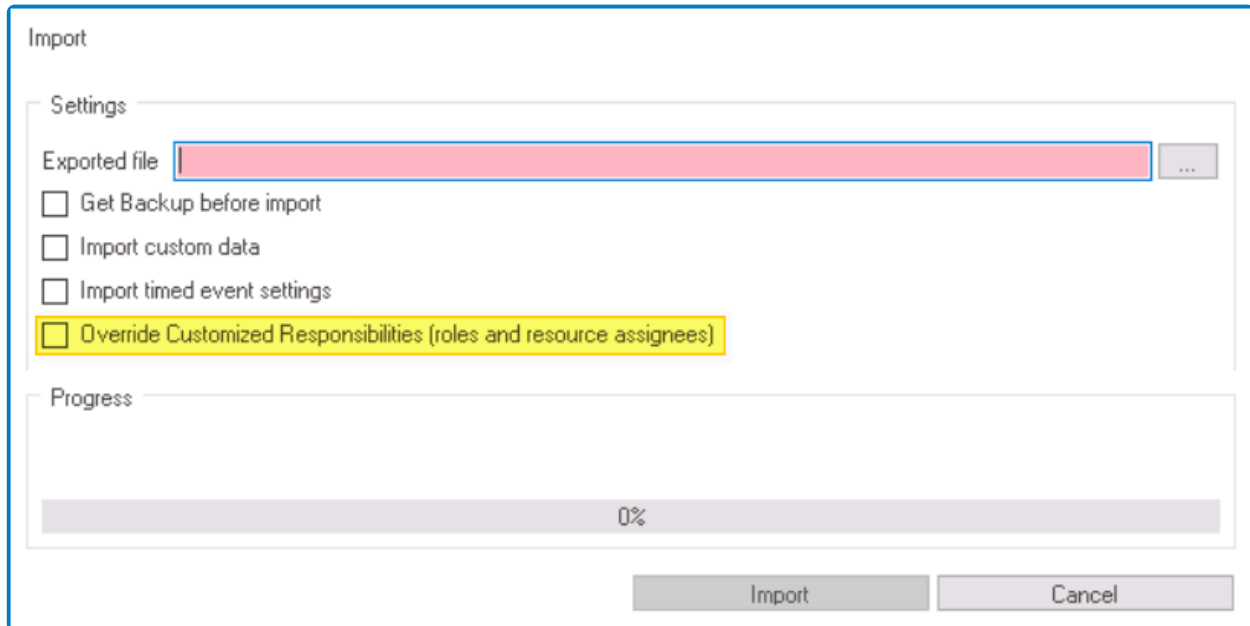
- For data sources in the simple format, they will be sorted alphabetically either by their name or by their name without the prefix “QS\_”.
- For data sources in the complex format, they will be sorted alphabetically by their name without the prefix “BPMAPP\_[ApplicationName]\_”.
- All system-sourced data sources are listed at the end of the dropdown, regardless of the naming convention used.

For example, the aforementioned queries would be sorted in the following order:

- “BPMAPP\_QMS\_QS\_Audits” (real name is “QS\_Audits”)
- “Categories”
- “QS\_Definitions”
- “Users”

## 4.4.2. Import – Override Customized Responsibilities

There is a known limitation to the **Override Customized Responsibilities** option in the [Import](#) window of the program. While this option allows the import process to override the defined and dynamic receiver settings, it will not override the specific receiver settings. As a result, any specific receivers assigned to individual tasks before the import will remain unchanged.



The screenshot shows the 'Import' dialog box. It has a title bar 'Import'. Inside, there is a 'Settings' section with a 'Progress' section below it. The 'Settings' section contains a text field for 'Exported file' with a red highlight, followed by three checkboxes: 'Get Backup before import', 'Import custom data', and 'Import timed event settings'. The checkbox for 'Override Customized Responsibilities (roles and resource assignees)' is highlighted in yellow. The 'Progress' section shows a progress bar at 0%. At the bottom right, there are 'Import' and 'Cancel' buttons.

Import

Settings

Exported file

☐ Get Backup before import

☐ Import custom data

☐ Import timed event settings

☐ **Override Customized Responsibilities (roles and resource assignees)**

Progress

0%

Import Cancel

## 4.4.3. Grid Control – Quick Forms Feature

Currently, the **Quick Forms** feature is limited to creating data grids. If users attempt to apply the feature to subform grids, no foreign key will be created and the control configuration will fail.

✿ A data grid is a type of grid that displays all submitted instances of another form. Conversely, a subform grid is embedded within a form and allows users to directly add and edit supplementary information.

In other words, the **Quick Forms** feature applies exclusively to the creation of grids of the **Base Info** type and does not extend to the **Master-Details** type.

✿ For more information on how the **Quick Forms** feature can be applied to grid controls, [click here](#).

## 4.4.4. Pending Tasks Folder – Tasks/Workflows Triggered from Grid

---

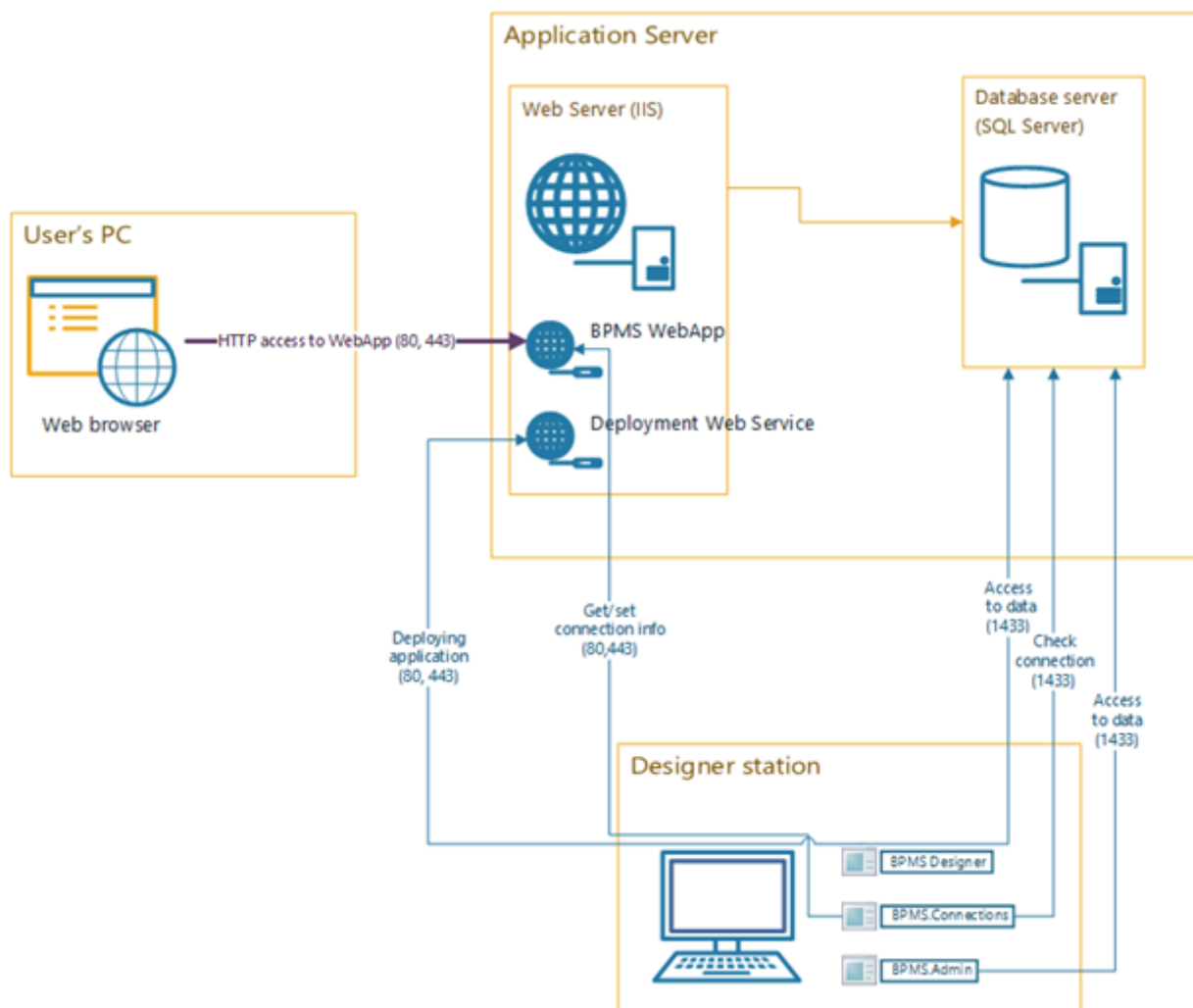
In the **DBP** web application, when a task or workflow is triggered by code from a grid, recipients must manually refresh the page to see an increase in the number of tasks within the **Pending Tasks** folder. This behavior is applicable only when recipients are already logged into the system.

## 5. BPMS System Requirements

The BPMS System is a client-server application.

The BPMS System consists of a web application that is deployed on IIS Server and has a connection to the Database. There are two clients that manage the content and interact with the application by the end user: the BPMS Designer and the Web Client. The BPMS Designer—which is a Windows application—is a thick client used to design, deploy, and modify the web application. A Web Client runs in a browser and is used by the end client to interact with the server. The BPMS System can be integrated with corporate services like the SMTP server to send email messages, Active Directory for centralized user authentication management, and third party systems for data pulling and pushing.

### Generic BPMS System Deployment Diagram



## 5.1. BPMS Server Requirements

---

### Hardware resources for BPMS server:

CPU: 4 Cores

RAM: minimum 8 GB (16 GB recommended)

HDD: 20 GB (of free disk space)

Network connection to SQL Server – minimum 100 Mbps, 1 Gbps recommended

### Hardware resources for SQLserver:

CPU: 4 Cores

RAM: minimum 16 GB (32 GB recommended)

HDD: 100 GB (of free disk space) High IO disk is recommended

Network connection to SQL Server – minimum 100 Mbps, 1 Gbps recommended

### Operating System:

Windows Server 2012 R2 64 bit or more recent

### Web Server:

1. IIS Services
  - a. Note that IIS 6 Management Compatibility is required
  - b. For more information, refer to [Chapter 17.1](#)
2. Net Framework (will be enabled for Windows Roles and Features)
  - a. Admin Console (designer): .net 4.7.2
  - b. Web: .net 4.7.2
  - c. Connection Form: 3.5 service pack 1
3. MSDTC Services
  - a. The Microsoft Distributed Transaction Coordinator (MSDTC) service is a component of modern versions of Microsoft Windows that is responsible for coordinating transactions that span multiple resource managers, such as databases, message queues, and file systems. MSDTC performs the transaction coordination role for components, usually with COM and .NET architectures.
  - b. In case of BPMS application, MSDTC Service is required only on server.
  - c. For more information, refer to [Chapter 17.3](#)

### SQL Server:

Microsoft SQL Server 2019 or more recent

During the deployment we will need to have the following permissions:

- On Windows Server: Administrator
- On SQL Server: dba (sa user)





**Note:** The sa user will only be used during the deployment. It won't be used after the deployment.

## **5.1.1. 2.1.1 Install IIS with Required Modules**

## 5.2. BPMS Designer Machine Requirements and Permissions

---

**The BPMS system consists of multiple parts:**

A Designer, which the user uses to build applications by designing forms. The Designer stores its information in the Database. Applications designed using the Designer get deployed to the Web Server. The Web Server also connects to the Database in order to read and store data. The end user then uses a web browser to access the application (which has been deployed to the Web Server).



**Note:** If the Designer is not supposed to be used by the end client, it can be set up on the same machine as the Web Server.

### Hardware resources for Designer:

CPU: 1 Core

RAM: minimum 8 GB (16 GB recommended)

HDD: 10 GB (of free disk space)

Network connection – minimum 15 Mbps

### Operating System for Designer:

Windows Server 2012 R2 64 bit or Windows 10 64 bit or more recent

Frameworks: .Net 3.5 SP1 and .Net 4.7.2

Office (If users want to see Word or Excel files)

Install Visual Studio (debugger) for debugging codes.

## 5.3. End User Browser Requirements and Permissions

---

To access the web application, end users will need a browser with a network connection.

### Officially Supported Browsers:

1. *Chrome* (Version 65.0.3325 or more recent)
2. *Microsoft Edge* (Version 44.17763 or more recent)
3. *Safari* (Version 12.0 or more recent)
4. *Mozilla Firefox* (Version 63.0 or more recent)
5. *Opera* (Version 53 or more recent)

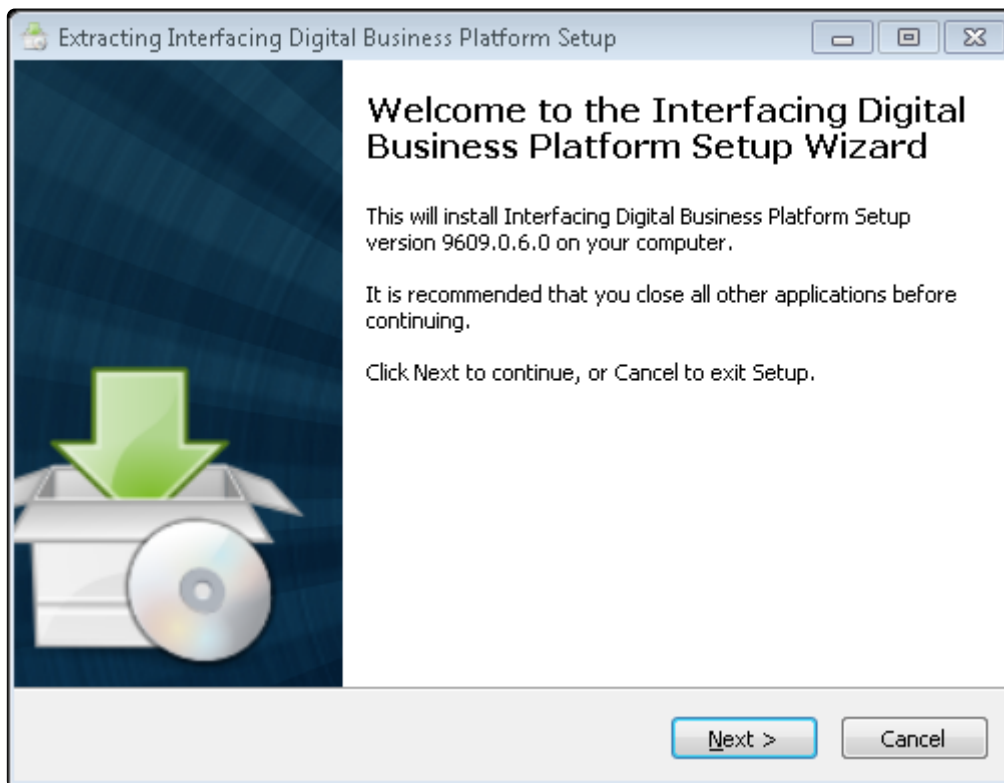


**Note:** The minimum resolution supported is (1090 × 740).

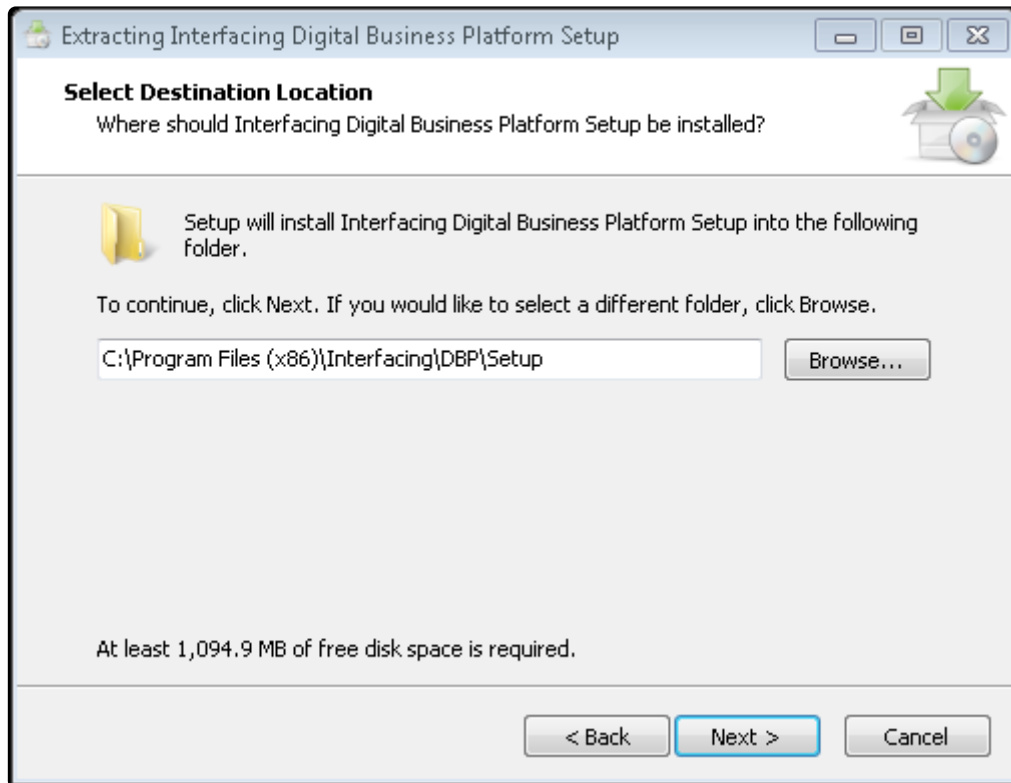
If users would like to export any information or reports, they will need MS Office (Word or Excel).

## 6. Installing the System


To install the Digital Business Platform (DBP) system, extract the installation files to your computer. Upon extracting the installation files, double-click on the Setup program and then select the **Next** button.

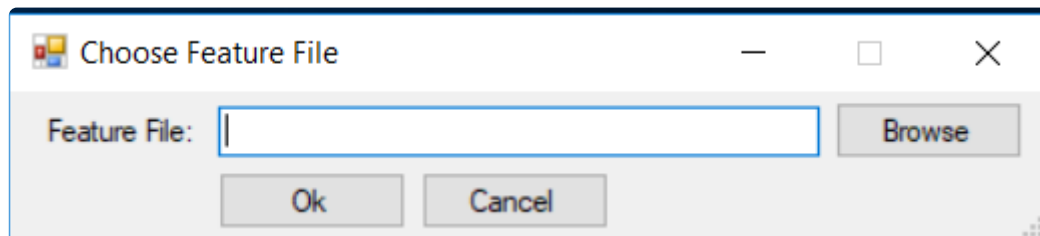


You will be asked to select the directory in which the Digital Business Platform is to be installed. To do so, click on the **Browse** button. Once you have specified a folder, select the **Next** button below.



Once the **Next** button has been selected, the Digital Business Platform will be installed to the specified path. You will be then asked to select a Feature file.

 **Note:** The feature.dat file is a system software lock that assigns access level to the tool module and is provided in accordance with the customer's contract.

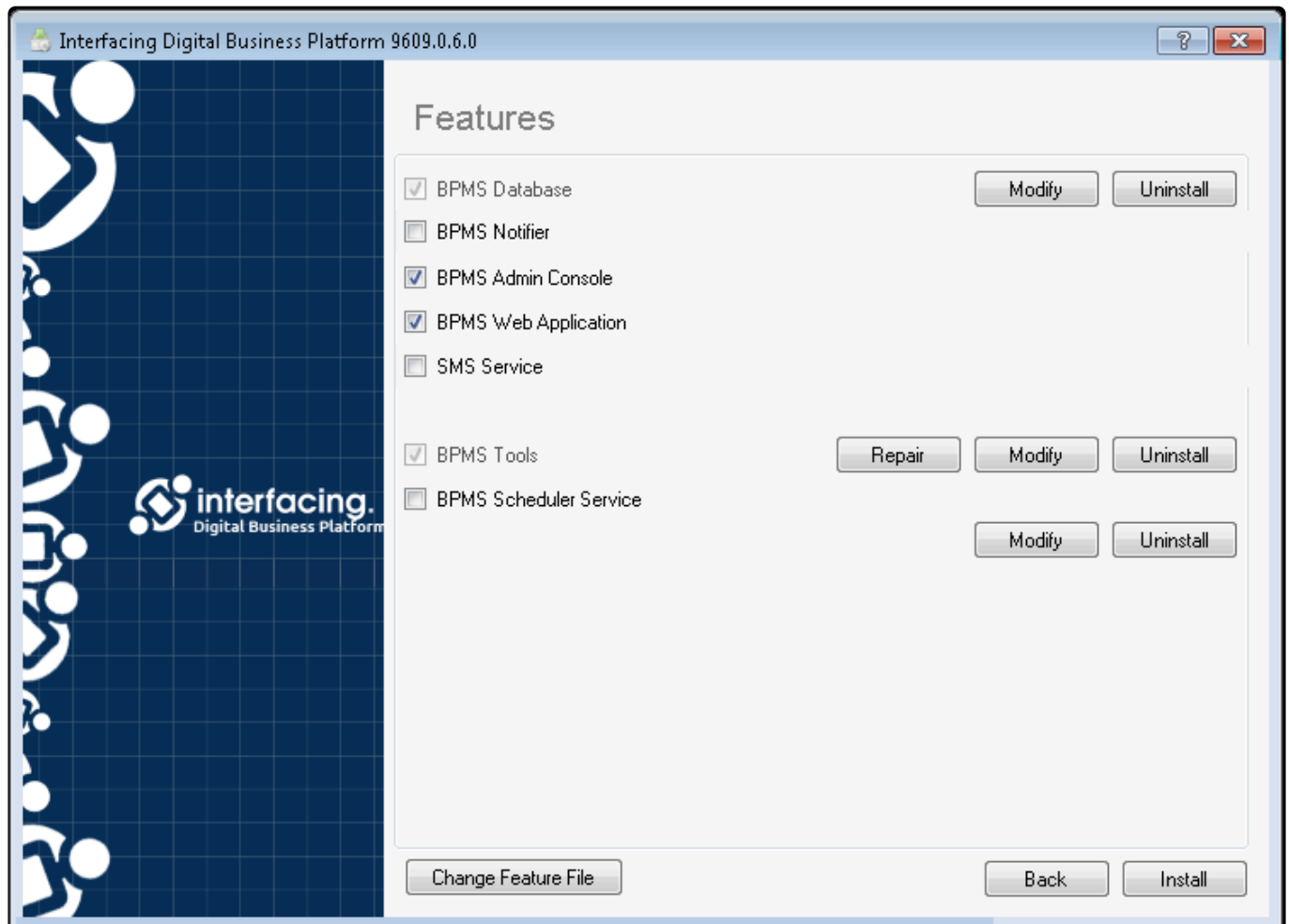


Select the file using the **Browse** button and then click on the **OK** button. If this is the first time that the module is being installed in the system, the [Features](#) form will be displayed. Otherwise, the [Update](#) form will be displayed.

## 6.1. Installing Modules

The Features installer allows users to install, modify, and update modules.

✿ **Note:** The module or demonstration features in this form are displayed based on the feature.dat file selected during installation.



The items and buttons in the Features installer may vary depending on whether or not a module has been installed.

1. **Repair:** This button allows users to repair and reinstall a module.
2. **Modify:** This button allows users to modify and reinstall a module.
3. **Uninstall:** This button allows users to remove an installed module.
4. **Update:** This button allows users to update all installed modules to the latest version.
5. **Change Feature File:** This button allows users to change the software lock or feature.dat file.
6. **Back:** This button allows users to return to the previous form.
7. **Install:** This button allows users to install the module(s) for the system.

To install the module, first select the checkbox next to the desired module, then click on the **Install** button.



**Note:** If the checkbox is selected and disabled next to a particular module, then the module is already installed onto your system.

Depending on the selected module, after clicking the **Install** or **Update** button, the settings associated with each module will be displayed on a single page. The following sections will contain installation configurations associated with each module.



## 6.1.1. Database

By selecting the **BPMS Database** checkbox/module in the [Features](#) installer, the [Database Configuration](#) form will then be displayed. This form is used to install or update the database.

✿ **Note:** Setting up a Digital Business Platform (DBP) requires a database that stores information about the design and build of the system and the operating data of the processes.

! Installing and updating this module is mandatory.

Interfacing Digital Business Platform 9609.0.2.1

### Database Configuration

☐ Connect To Existing Database **1. Connect To Existing Database**

☒ Create New Database **2. Create New Database**

Server:  **3. Server**

Database Name:  **4. Database Name**

DB Username:  **5. Database UserName & Password**

DB Password:  **5. Database UserName & Password**

Server Username:  **6. Server UserName & Password**

Server Password:  **6. Server UserName & Password**

**7. Check Connection**

1. **Connect to Existing Database:** This option allows users to connect to an existing database.

☐ Connect To Existing Database

- If this option is selected, the installation will be performed on the DBP database in the system.

✿ **Note:** This option is the default mode when the modules have been [updated](#).

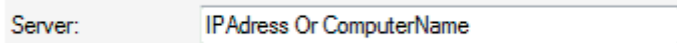
2. **Create New Database:** This option allows users to create a new database.

A button with a blue circular icon containing a white 'C' and the text 'Create New Database'.

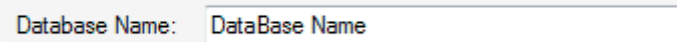
- If this option is selected and the corresponding database profile is then inputted, the DBP database will be created under the chosen name.

 **Note:** This option is the default mode when the modules have been [installed](#).

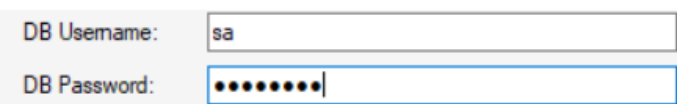
3. **Server:** This is a text field that allows users to input the name or IP of the server where the database is/will be located.

A text field with the label 'Server:' and the placeholder text 'IPAdress Or ComputerName'.

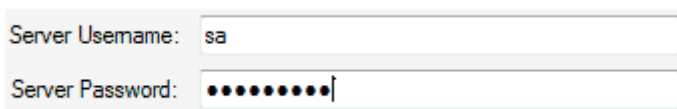
4. **Database Name:** This is a text field that allows users to input the name of the database that they want to update or create.


A text field with the label 'Database Name:' and the placeholder text 'DataBase Name'.

5. **DB Username and Password:** These are text fields wherein the username and password of the database owner can be inputted.

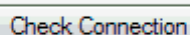
Two text fields. The first is labeled 'DB Username:' and contains the text 'sa'. The second is labeled 'DB Password:' and contains a series of dots.

6. **Server Username and Password:** These are text fields wherein the server username and password for the system administrator can be inputted.

Two text fields. The first is labeled 'Server Username:' and contains the text 'sa'. The second is labeled 'Server Password:' and contains a series of dots.

 **Note:** This section is enabled only if the **Create New Database** option has been selected.

7. **Check Connection:** This button allows users to verify the connection to the database.

A button with the text 'Check Connection'.

## 6.1.2. Admin Console

By selecting the **BPMS Admin Console** checkbox/module in the [Features](#) installer, the [Admin Console Configuration](#) form will be displayed. This form is used to update or install the Admin Console.

! Installing and updating this module is mandatory.

Interfacing Digital Business Platform 9609.0.6.0

### Admin Console Configuration

Installation Folder: C:\Users\mgali\AppData\Roaming\Interfacing\D Browse

User Profile Common User Program Files **1. Installation Folder**

**Scheduler Settings**

If you want to install the scheduler service on another server, change the settings below. **2. Scheduler Server**

Scheduler Server: IP Address or Computer Name

Scheduler Port: 555 Scheduler Service Name: RayBPMSScheduler

Back Next

1. **Installation Folder:** This field displays the Admin Console installation path/folder.
  - To select or change this folder, select the **Browse** button.

Installation Folder: C:\Program Files (x86)\BPMS\BPMSAdr

2. **Scheduler Settings:** If the web application is on a separate platform from the computer (Admin Console), or if the web server's IP address is different from that of the computer, then in this section you can define the link between the Windows server and the server where the timed/scheduler service is installed.

|                   |  |  |
|-------------------|--|--|
| Scheduler Server: | <input type="text" value="IPAddress Or ComputerName"/> |  |
| Scheduler Port:   | <input type="text" value="555"/>                       | Scheduler Service Name: <input type="text" value="BPMSScheduler"/> |

- **Scheduler Server:** This is a text field that allows users to input the name or IP of the server to which the Scheduler is installed.
- **Scheduler Port:** This is a text field that allows users to input the port number used in the scheduler service.
- **Scheduler Service Name:** This is a text field that allows users to input the name of the scheduler service.



**Note:** If the web server and computer addresses are the same, there is no need to complete this section.

## 6.1.3. Web Application

Once users have set the installation details in the [Admin Console Configuration](#) form, they should select the **Next** button. It will direct them to the [Web Application Configuration](#) form.

✿ **Note:** This module contains all of the designer tools (form designer, entity designer, rule designer, etc.) needed to build and/or manage BPM applications.

! Installing and updating this module is mandatory

The screenshot shows the 'Web Application Configuration' window. On the left is a blue sidebar with the 'interfacing. Digital Business Platform' logo. The main area contains the following configuration options:

- 1** ☒ Use Existing Web Application
- ☐ Create New Web Application **2**
- Physical Folder:  **3** [Browse]
- URL:  **4**
- ☐ Install Files **5**
- Local Web Sites:  **6**
- Application Name:  **7**
- Lock Server:  **8**
- 9** Sync
  - SyncServer URL:
  - ☐ Use Sync
- ☐ Enable SSL(Secure login) **10**
- 11** SSO Setting
  - ☐ UseSSO(old version)
  - ☐ UseSSO(new version)
  - ☒ None

At the bottom right are 'Back' and 'Next' buttons.

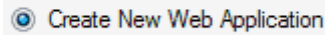
- 1. Use Existing Web Application:** This option allows users to perform the installation on an existing web application in the system.

☒ Use Existing Web Application


- If this option is selected, certain fields in this form will be disabled, as they will contain previously defined information. Where applicable, however, users will be able to either make modifications to the displayed information or input new information.

 **Note:** This option is the default mode when the modules have been updated.

2. **Create New Web Application:** This option allows users to create a new web application.



- If this option is selected and the corresponding web application profile is then defined, the web application will be created under the specified name. All fields will be enabled, allowing users to define new information.

 **Note:** This option is the default mode when the modules have been installed.

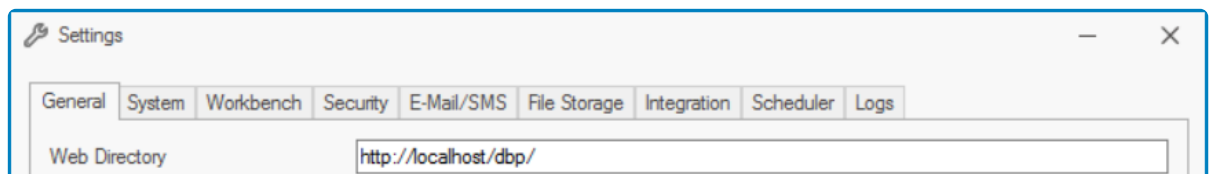
3. **Physical Folder:** This field allows users to select the physical path/folder of the web application.
- To select or change the folder, select the **Browse** button.

Physical Folder:

4. **URL:** This is a text field that allows users to input the URL of the web application.

URL:

- Once the application has been installed, users can change this path by going to the [Settings](#) of the Admin Console.



5. **Install Files:** This is a checkbox which, if selected, allows users to install the files and place them into the IIS physical path/folder.

☒ **Install Files**

6. **Local Websites:** This is a single-select dropdown field that displays the list of local websites. It allows users to select the website that will host the application.

Local WebSites:

7. **Application Name:** This is a text field that allows users to input the name of the web application.

Application Name:

8. **Lock Server:** This is a text field allows users to input the name/IP address of the server where the hardware locks are located.

Lock Server:

9. **Sync:** Here, users can set up and enable the sync server. A sync server allows users to store and synchronize data across platforms.

Sync

SyncServer URL:

☐ Use Sync

- **Sync Server URL:** This is a text field that allows users to input the URL of the sync server.
  - **Use Sync:** This is a checkbox which, if selected, allows users to use the sync server.
10. **Enable SSL:** This is a checkbox which, if selected, allows users to enable SSL. (SSL stands for Secure Sockets Layer. It allows for a secure and encrypted connection.)

☐ Enable SSL

11. **SSO Setting:** Here, users can configure the SSO setting. An SSO is a called service that provides a secure connection across the BPMS system (ARPG & E-Office plugins).

SSO Setting

☐ UseSSO(old version)

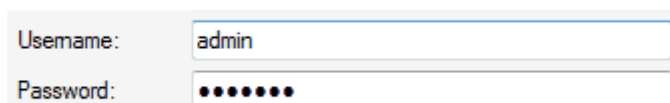
☐ UseSSO(new version)

☒ None

- **Use SSO (Old Version):** This option allows users to connect to the old SSO version.
  - **Use SSO (New Version):** This option allows users to connect to the new SSO version.
  - **None:** This option allows users to not connect to any version of SSO.
12. **Server and Port:** These are text fields wherein the name/IP address of the mail server and its port number, respectively, can be inputted.

Server:  Port:

13. **Username and Password:** These are text fields wherein the username and password of the administrator can be inputted.



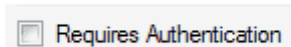
A screenshot of a login form. It contains two labels: 'Username:' and 'Password:'. The 'Username:' label is next to a text input field containing the text 'admin'. The 'Password:' label is next to a text input field containing ten black dots, representing a masked password.

14. **Default Email:** This is a text field that allows users to input the default email.



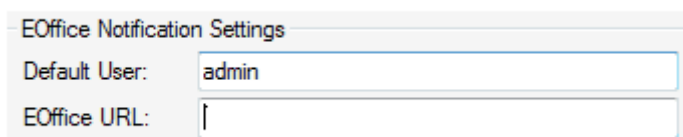
A screenshot of a text input field labeled 'Default EMail:'. The field is empty.

15. **Requires Authentication:** This is a checkbox which, if selected, allows users to mark out authentication as a requirement.



A screenshot of a checkbox labeled 'Requires Authentication'. The checkbox is currently unchecked.

16. **E-Office Notification Settings:** Here, users can configure the E-Office notification settings.



A screenshot of a form titled 'EOffice Notification Settings'. It contains two labels: 'Default User:' and 'EOffice URL:'. The 'Default User:' label is next to a text input field containing the text 'admin'. The 'EOffice URL:' label is next to an empty text input field.

- **Default User:** This is a text field that allows you to input the default user.
- **E-Office URL:** This is a text field that allows you to input the E-Office URL.



## 6.1.4. SMS Service

By selecting the **BPMS SMS Service** checkbox/module in the [Features](#) installer, the “SMS Service Configuration” form will then be displayed. This form is used to update or install the SMS Service.

✿ **Note:** Setting up this service allows users to send and receive SMS for process management purposes.

1. **Use Existing Web Application:** This option allows users to perform the installation on the existing SMS Service available in the system.

☒ Use Existing Web Application

- If this option selected, certain fields in this form will be disabled, as they will contain previously defined information. Where applicable, however, users will be able to either make modifications to the displayed information or input new information.

✿ **Note:** This option is the default mode when the modules have been [updated](#).

2. **Create New Web Application:** This option allows users to create a new SMS Service.

☒ Create New Web Application

- If this option is selected and the corresponding profile is then defined, the SMS Service will be created under the chosen name. All fields will be enabled, allowing users to define new information.

 **Note:** This option is the default mode when the modules have been [installed](#).

3. **Physical Folder:** This field allows users to select the physical path/folder of the SMS service.
  - To select or change the folder, select the **Browse** button.

Physical Folder:

4. **URL:** This is a text field that allows users to input the URL (or web service access path) of the SMS service.

URL:

5. **Install Files:** This is a checkbox which, if selected, allows users to install the files and place them into the IIS physical path/folder.

☒ Install Files

6. **Local Website:** This is a single-select dropdown field that displays the list of local websites. It allows users to select the website that will host the SMS Service.

Local Web Sites:

7. **Application Name:** This is a text field that allows users to input the name of the web application.

Application Name:

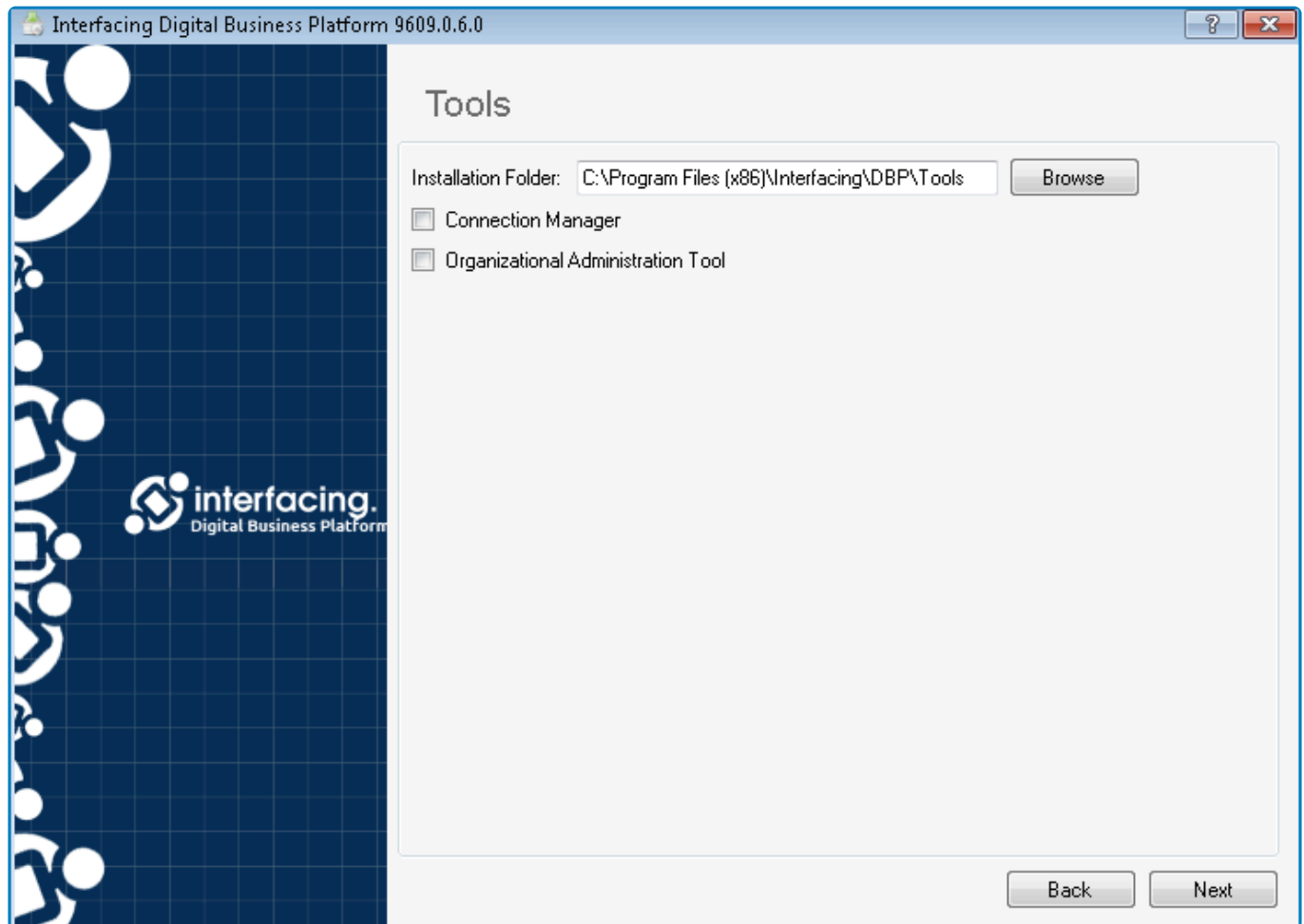
8. **Send Agent:** This is a single-select dropdown field that displays the list of send agents. It allows users to select the server that will be used as the SMS provider.

Send Agent:

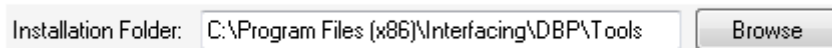
## 6.1.5. Tools

By selecting the **BPMS Tools** checkbox/module in the “[Features](#)” installer, the Tools form will then be displayed. This form is used to update or install tools.

✿ **Note:** By implementing this module, the tools necessary to your work processes’ management system will be installed in the path/folder specified below.



1. **Installation Folder:** This field displays the Tools installation path/folder.
  - To select or change this folder, select the **Browse** button.



2. **Connection Manager:** This is a checkbox which, if selected, allows for the Connection Manager to be installed in the specified path/folder.
  - It is possible to determine the relationship between web applications and Windows web applications within this database.

☐ Connection Manager

3. **Organizational Administration Tool:** This is a checkbox which, if selected, allows for the Admin.exe program to be installed in the specified path/folder.

☐ Organizational Administration Tool

✿ **Note:** Users can access the default database management information system through the E-Office catalog of the Connection.exe program.

4. **Report Runner:** This is a checkbox which, if selected, allows for the Report Runner to be installed in the specified path/folder.

☐ Report Runner

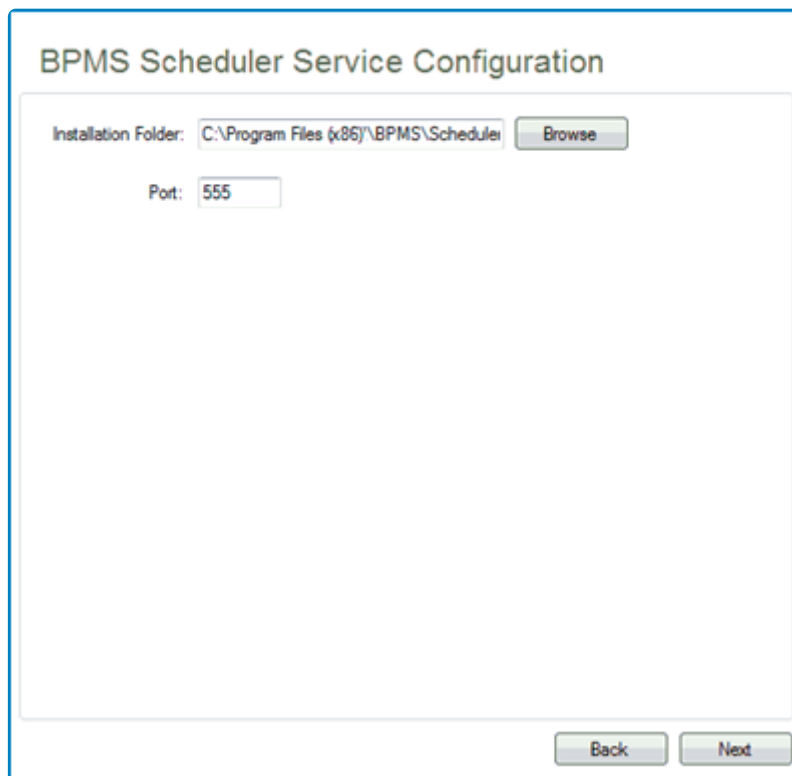
✿ **Note:** Users can access the default database reporting system through the ARPG catalog of the Connection.exe program.

## 6.1.6. BPMS Scheduler Service

By selecting the **BPMS Scheduler Service** checkbox/module in the [Features](#) installer, the BPMS Scheduler Service Configuration form will then be displayed. This form is used update or install the scheduler service.

✿ **Note:** Implementing this module allows the use of scheduled events in the system.

! This service must be installed on the web server.



1. **Installation Folder:** This field displays the installation path/folder of the scheduler service.
  - To select or change this folder, select the **Browse** button.
2. **Port:** This is a text field that allows users to input the port number used by the scheduler service.

## 6.1.6.1. BPMS Scheduler – Developer's Guide

---

This developer's guide for the **BPMS Scheduler** covers the basics of getting started, creating events, and deploying and executing them. The following sections will walk you through each step of the process.

1. [Getting Started](#)
2. [Creating and Deploying Events](#)
3. [Executing Events](#)

## 6.1.6.1.1. Getting Started

The **BPMS Scheduler** is a Windows service designed to execute tasks based on a scheduler. Currently, it only supports one type of task, which is running timed events.

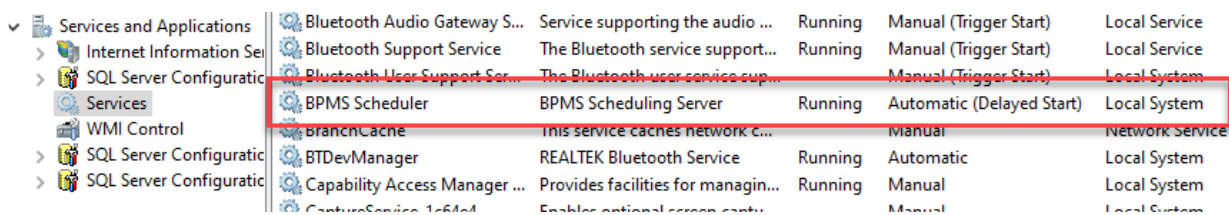
The Scheduler's source code is divided into two projects:

1. **Ray.BPMS.SchedulerWindowsService** – Windows Service
2. **Ray.BPMS.Scheduler.Jobs** – Job Declaration

The source code is compiled and stored in the `<root_project_folder>\RayvarzBPMSSetup\Source\SchedulerWindowsService\` folder.

✿ On the source machine, the Scheduler is placed by default in the `c:\Program Files (x86)\Interfacing\DBP\SchedulerService\` folder.

Once installed, the service can be found in the Computer Management Console.



| Service Name                  | Description                        | Status         | Startup Type                     | Log On As           |
|-------------------------------|------------------------------------|----------------|----------------------------------|---------------------|
| Bluetooth Audio Gateway S...  | Service supporting the audio ...   | Running        | Manual (Trigger Start)           | Local Service       |
| Bluetooth Support Service     | The Bluetooth service support...   | Running        | Manual (Trigger Start)           | Local Service       |
| Bluetooth User Support Ser... | The Bluetooth user service sup...  | Running        | Manual (Trigger Start)           | Local System        |
| <b>BPMS Scheduler</b>         | <b>BPMS Scheduling Server</b>      | <b>Running</b> | <b>Automatic (Delayed Start)</b> | <b>Local System</b> |
| BranchCache                   | This service caches network c...   | Manual         | Manual                           | Network Service     |
| BTDevManager                  | REALTEK Bluetooth Service          | Running        | Automatic                        | Local System        |
| Capability Access Manager ... | Provides facilities for managin... | Running        | Manual                           | Local System        |
| CaptureService 1a5d4d         | Enables optional screen captu...   | Manual         | Manual                           | Local System        |

✿ The Scheduler relies on the [Quartz.NET](#) library to carry out its core actions.

The Scheduler stores information about jobs in the BPMS database, specifically in tables with the naming convention **ray.BPMS\_QRTZ\_**. Examples of such tables include:

- **ray.BPMS\_QRTZ\_TRIGGERS** – stores information about the jobs
- **ray.BPMS\_QRTZ\_SIMPLE\_TRIGGERS** – stores simple schedules for jobs
- **ray.BPMS\_QRTZ\_CRON\_TRIGGERS** – stores cron schedules for jobs

```

1 select * from ray.BPMS_QRTZ_TRIGGERS
2 select * from ray.BPMS_QRTZ_SIMPLE_TRIGGERS
3 select * from ray.BPMS_QRTZ_CRON_TRIGGERS
4

```

| SCHED_NAME | TRIGGER_NAME    | TRIGGER_GROUP                        | JOB_NAME                     | JOB_GROUP     | DESCRIPTION        | NEXT_FIRE_TIME | PREV_FIRE_TIME     | PRIORITY | TRIGGER_STATE | TRIGGER_TYPE | START_TIME  |
|------------|-----------------|--------------------------------------|------------------------------|---------------|--------------------|----------------|--------------------|----------|---------------|--------------|-------------|
| 1          | RayBPMScheduler | 9037ac55-6feb-6e69-338a-ac835bf08aea | TriggerWithoutStartTimeGroup | RaiseEventJob | BpmsEngineJobGroup | NULL           | 637784244000000000 | 5        | WAITING       | SIMPLE       | 6377557320x |
| 2          | RayBPMScheduler | efa103c8-825e-c426-6945-1b72a7fda5ff | TriggerWithoutStartTimeGroup | RaiseEventJob | BpmsEngineJobGroup | NULL           | 637783932000000000 | 5        | WAITING       | SIMPLE       | 6377837880x |

| SCHED_NAME | TRIGGER_NAME    | TRIGGER_GROUP                        | REPEAT_COUNT | REPEAT_INTERVAL | TIMES_TRIGGERED |
|------------|-----------------|--------------------------------------|--------------|-----------------|-----------------|
| 1          | RayBPMScheduler | 9037ac55-6feb-6e69-338a-ac835bf08aea | -1           | 86400000        | 33              |
| 2          | RayBPMScheduler | efa103c8-825e-c426-6945-1b72a7fda5ff | -1           | 600000          | 24              |

| SCHED_NAME | TRIGGER_NAME | TRIGGER_GROUP | CRON_EXPRESSION | TIME_ZONE_ID |
|------------|--------------|---------------|-----------------|--------------|
|------------|--------------|---------------|-----------------|--------------|

## 6.1.6.1.2. Creating and Deploying Events

As an example, let's create a simple task that sends an email every 10 minutes.

### Steps:

1. Navigate to **Processes > Events**. Click on the **New** button in the menu bar to display the [Event Properties](#) window where you can create a timed event.

**Event Properties**

Apply OK Cancel

General Advanced

Event Type: Timed

Name: TestEmail

English Title: Test Email

French Title:

Swedish Title:

Hebrew Title:

Event Icon: [ + ] [ - ]

☐ Can run background ☐ Has TimeSheet

Scheduling

☐ In Process ☒ Run automatically(without API)

Add/Edit Delete

2. Define a schedule for the event.
  - In this case, run the event every 10 minutes indefinitely.

**Scheduler**

Start Time

☐ Run Time

☐ After Run Time

☒ Start Time

0 Second

2022-01-21 11:20 AM

End Time

☒ None

☐ End Time

☒ Simple

Simple

Repeat counter: 0 ☒ Forever

Repeat interval: 10 Minute

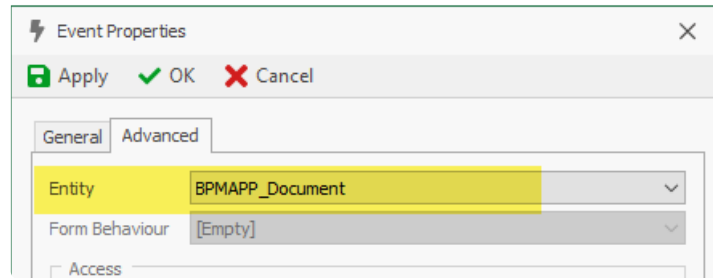
☐ According to work calander

☐ Cron



✿ To access the [Scheduler](#) window, click on the **Add/Edit** button.

3. In the Advanced tab of [Event Properties](#) window, select an entity type.



The screenshot shows the 'Event Properties' dialog box with the 'Advanced' tab selected. The 'Entity' dropdown menu is highlighted in yellow and set to 'BPMAPP\_Document'. The 'Form Behaviour' dropdown menu is set to '[Empty]'. There is an 'Access' checkbox at the bottom which is currently unchecked. The dialog has 'Apply', 'OK', and 'Cancel' buttons at the top.

✿ If the event is not part of a process (as in our case), a new instance of this entity will be created each time the event is triggered and passed to the custom script as `domainObject`.

4. Create a corresponding event action.
  - Navigate to **Business Logics > Actions > Event Actions** and click on the **New** button in the menu bar to display the [Event Action Properties](#) window, where you can create a new event action.

Action Properties: TestEmailAction

Apply OK Cancel

Type: **Action**

Name: TestEmailAction

Title: TestEmailAction

☒ Active

Entity: Document (Document)

Scope

Drag a column header here to group by that column

| Title      | Name      | Access Name                  |
|------------|-----------|------------------------------|
| Test Email | TestEmail | Ray.BPMApp.Demo20191018.Serv |

Record 1 of 1

Event name: Ray.BPMApp.Demo20191018.Services.IBPMAPP\_Document.TestEmail

Process: [Empty]

Process Version:

State Name:

- From the **Entity** field, select the entity associated with the event action (same entity as the one chosen for the timed event).
- From the **Scope** field, select the timed event by double-clicking on it.
  - Its associated access name will then populate the **Event Name** field.

5. Attach code to the newly created event action.

- In this case, write the following code:

```
bpmAppService.BPMSServices.NotificationService.Send("Scheduled email", "This email was sent at " + DateTime.Now.ToString("G"), NotificationAgent.Email, true, "John.Doe@interfacing.com");
```



To write code specifying the behavior of the event action, click on the **Edit Code** button in the menu bar.

6. Deploy an application.

- The **BPMS Scheduler** exposes itself through a TCP connection port 555 to accept job changes.
  - The Designer program connects to the Scheduler to create, modify, and/or delete jobs.
    - Connection settings are:
      1. for the Scheduler in **BPMSSchedulerWindowsService.exe.config** file

```

10 <appSettings>
11 <add key="instanceName" value="RayBPMScheduler" />
12 <add key="threadCount" value="10" />
13 <add key="exporter_type" value="Quartz.Simpl.RemotingSchedulerExporter, Quartz" />
14 <add key="exporter_port" value="555" />
15 <add key="exporter_bindName" value="RayBPMScheduler" />
16 <add key="exporter_channelType" value="tcp" />
17 <add key="exporter_channelName" value="httpQuartz" />
18 <add key="jobStore_type" value="Quartz.Simpl.AdJobStore, Quartz" />
19 <add key="jobStore_instanceName" value="RayBPMScheduler" />

```

## 2. for the Designer in **BPMS.Designer.exe.config** file

```

61 <!-- ***** BPMS Scheduler Config ***** -->
62 <add key="SchedulerServer" value="localhost" />
63 <add key="Port" value="555" />
64 <add key="SchedulerWindowsServiceName" value="RayBPMScheduler" />
65 <add key="DesignerBPMS" value="d45e245e45e45e45e45e45e45e45e45e" />

```

- During deployment, the Designer attaches to the Scheduler:
  - Initiated from:

```
BPMS.Designer.Infrastructure/Services/BPMSAppService.cs/DeployTimeEvents()
```

- Attach to the Scheduler:

```
Ray.BPMS.Core/Scheduler/SchedulerService.cs/AttachToWindowsServiceScheduler()
```

- Sync Events:

```
Ray.BPMS.Core/Scheduler/SchedulerService.cs/SyncStarterProcessTimeEvents()
```

- The Scheduler receives the events and creates the corresponding jobs (tables **ray.BPMS\_QRTZ\_TRIGGERS** and **ray.BPMS\_QRTZ\_SIMPLE\_TRIGGERS**)

## 6.1.6.1.3. Executing Events

When an event must be executed according to a schedule (simple or cron), the **BPMS Scheduler** runs the Execute method of the IJob object.

✿ In this case, it will be: `Ray.BPMS.Scheduler.Jobs/Job/RaiseEventJob.cs/Execute()`

The Execute method extracts the **ServerURL** parameter from the **BPMSchedulerWindowsService.exe.config** file to get the base BPMS address (default: <https://localhost/dbp/>).

```
21 <add key="jobStore_dataSource" value="myDS" />
22 <add key="dataSource_myDS_provider" value="SqlServer" />
23 <add key="ServerUrl" value="https://localhost/dbp/" />
24 </appSettings>
```

Then, the Execute() method sends an HTTP request to the `<baseURL>/api/Scheduler/RaiseEvent` WebAPI and passes parameters such as eventID, objectID (empty), and user (the user who deployed an application in the Designer).

✿ The value of these parameters is fixed during deployment by the Designer.

The BPMS WebApp exposes `api/Scheduler/RaiseEvent` API to receive a command to run the event.

/BPMSFromVS/swagger/ui/index#/Scheduler/Scheduler\_RaiseEvent

**RuleManagement** Show/Hide List Operations Expand Operations

**Scheduler** Show/Hide List Operations Expand Operations

**GET** /api/Scheduler/RaiseEvent/{eventId}/{objectId}/{userId}/{isFirstExecute}

Response Class (Status 200)  
OK

Model Example Value

```
{}
```

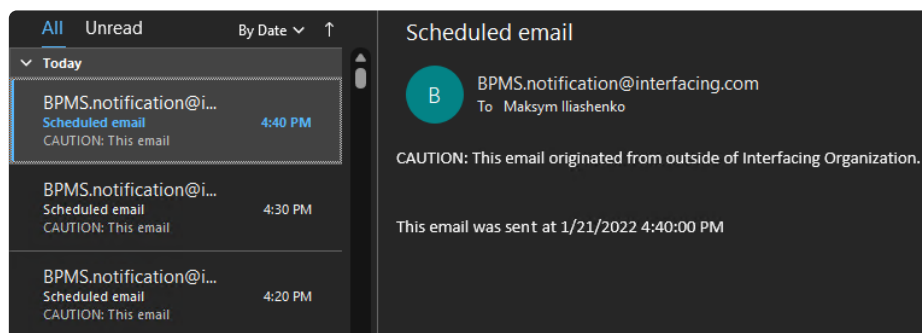
Response Content Type

**Parameters**

| Parameter      | Value  | Description | Parameter Type | Data Type |
|----------------|--|-------------|----------------|-----------|
| eventId        | (required)                                       |             | path           | string    |
| objectId       | (required)                                       |             | path           | string    |
| userId         | (required)                                       |             | path           | string    |
| isFirstExecute | <input checked="" type="checkbox" value="true"/> |             | path           | boolean   |

Try it out!

The BPMS core executes the defined event action and sends an email.

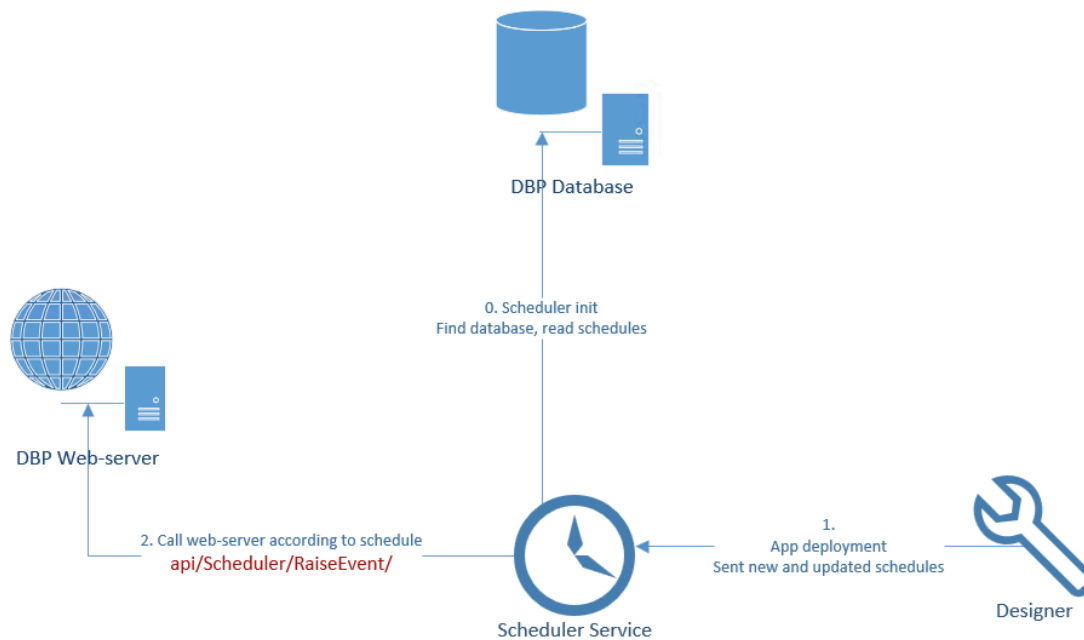


The Scheduler notifies all executions in the log file **BpmsSchedulerWindowsServiceLog.txt**

```
01/21/2022 16:20 RaiseEventJob Executed in: 1/21/2022
01/21/2022 16:20 Raise event. EventId [69451782-58fe-4253-a213-ae2400b882d3], objectId [00000000-0000-0000-0000-000000000000], userId [5bf6a7ef-62d0-4962-8031-88f42d21ec4a]
01/21/2022 16:20 Rest call to /api/Scheduler/RaiseEvent/69451782-58fe-4253-a213-ae2400b882d3/00000000-0000-0000-0000-000000000000/5bf6a7ef-62d0-4962-8031-88f42d21ec4a/False
01/21/2022 16:20 Success!
01/21/2022 16:30 -----
01/21/2022 16:30 RaiseEventJob Executed in: 1/21/2022
01/21/2022 16:30 Raise event. EventId [69451782-58fe-4253-a213-ae2400b882d3], objectId [00000000-0000-0000-0000-000000000000], userId [5bf6a7ef-62d0-4962-8031-88f42d21ec4a]
01/21/2022 16:30 Rest call to /api/Scheduler/RaiseEvent/69451782-58fe-4253-a213-ae2400b882d3/00000000-0000-0000-0000-000000000000/5bf6a7ef-62d0-4962-8031-88f42d21ec4a/False
01/21/2022 16:30 Success!
01/21/2022 16:40 -----
01/21/2022 16:40 RaiseEventJob Executed in: 1/21/2022
01/21/2022 16:40 Raise event. EventId [69451782-58fe-4253-a213-ae2400b882d3], objectId [00000000-0000-0000-0000-000000000000], userId [5bf6a7ef-62d0-4962-8031-88f42d21ec4a]
01/21/2022 16:40 Rest call to /api/Scheduler/RaiseEvent/69451782-58fe-4253-a213-ae2400b882d3/00000000-0000-0000-0000-000000000000/5bf6a7ef-62d0-4962-8031-88f42d21ec4a/False
01/21/2022 16:40 Success!
```

## 6.1.6.2. BPMS Scheduler – Troubleshooting

If you're experiencing issues with the **BPMS Scheduler**, it can be helpful to narrow down the problem by focusing on specific steps in the process.

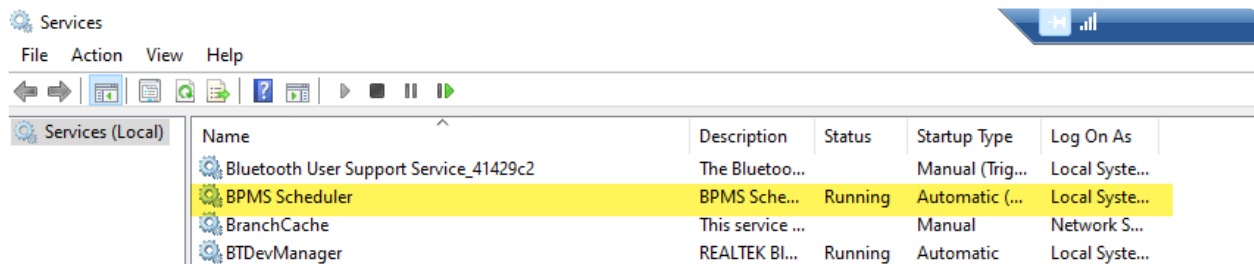


Here are some common troubleshooting points to consider:

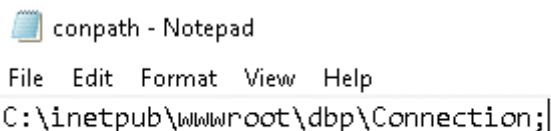
1. [Scheduler Initialization](#)
2. [Application Deployment](#)
3. [Raise Event](#)

## 6.1.6.2.1. Scheduler Initialization (Step 0)

The **BPMS Scheduler** is a Windows service designed to execute tasks based on a scheduler. To start and stop this service, you can use the standard **Services** application.



When the Scheduler starts, it requires a **ConPath.txt** file in its folder to determine how to connect to the database (where all information about the schedule, past, future, and missed events is stored). This file contains a path to a folder where the **Connection.xml** file is located. By default the content is `C:\inetpub\wwwroot\dbp\Connection;`



- If the file is not found in the log file **BpmsSchedulerWindowsServiceLog.txt**, you will see the following error:

```
01/24/2023 14:56 -----
01/24/2023 14:56 Start Configuration
01/24/2023 14:56 Can't find config file ConPath.txt
01/24/2023 14:56 Finish Configuration
01/24/2023 14:56 -----
```

- If the file is found and its content points to the correct location in the log file, you will see:

```
01/23/2023 15:28 -----
01/23/2023 15:28 Start Configuration
01/23/2023 15:28 Connection string leads to Server=rnd-sql-server.interfacing.local;Database=DBPqa2
01/23/2023 15:28 Connectto to sql in successful. Instance name of sql is 'MSSQLSERVER'
01/23/2023 15:28 Finish Configuration
01/23/2023 15:28 -----
```



You can delete the **ConPath.txt** file after a new DBP build installation, but please remember to check this file after setup.

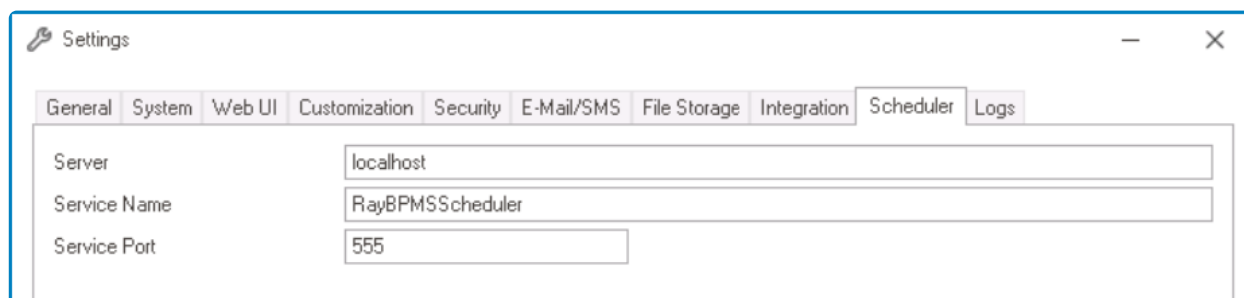


The **BpmsSchedulerWindowsServiceLog.txt** file contains a detailed log of all the Scheduler's activity.

## 6.1.6.2.2. Application Deployment (Step 1)

When a user deploys an application from the Designer, it typically connects to the Scheduler (which must be running) on the local machine.

**Note:** Users have the ability to define how the Designer can locate the Scheduler in the [Designer Settings](#).



The screenshot shows a 'Settings' window with a tabbed interface. The 'Scheduler' tab is selected. It contains three input fields: 'Server' with the value 'localhost', 'Service Name' with the value 'RayBPMSScheduler', and 'Service Port' with the value '555'.

| Tab       | Field        | Value            |
|-----------|--------------|------------------|
| Scheduler | Server       | localhost        |
|           | Service Name | RayBPMSScheduler |
|           | Service Port | 555              |

The Designer sends new and updated schedules directly to the Scheduler, which stores all the information in the database.



## 6.1.6.2.3. Raise Event (Step 2)

When a scheduled event is triggered, the Scheduler calls the **DBP** web server to execute the event (or task), and a new record is added to the **BpmsSchedulerWindowsServiceLog.txt** file.

```
01/24/2023 19:55 -----
01/24/2023 19:55 RaiseEventJob excuted in: 1/24/2023 7:55:00 PM
01/24/2023 19:55 Raise event. EventId [6c28e52d-79bf-492e-ab0e-ae9701415be7], objectId [00000000-0000-0000-0000-000000000000], userId [2926ce8f-7d40-44c9-b0d0-18ea118d421a]
01/24/2023 19:55 Success!
```

The **DBP** web server also logs information about the event in its own [Application Log](#).

### BPMS Log

Drag a column header here to group by that column

| ID      | Date             | Severity | Module          | Message  |
|---------|------------------|----------|-----------------|--|
| 2087051 | 2023-01-24 19:56 | Info     | MISNotification | Application: QMS, required object Ray.BPMApp.QMS.Model.BPMAPP_Audit_CollectEvidences_UserTask  |
| 2087050 | 2023-01-24 19:56 | Info     | MISNotification | Scheduled Event Raised. EventId [6c28e52d-79bf-492e-ab0e-ae9701415be7], objectId [00000000-0000-0000-0000-000000000000], userId [2926ce8f-7d40-44c9-b0d0-18ea118d421a] |
| 2087049 | 2023-01-24 19:55 | Info     | MISNotification | Scheduled Event Ends. Execution time: 130 ms   |
| 2087042 | 2023-01-24 19:55 | Info     | MISNotification | Get object of Ray.BPMApp.QMS.Model.BPMAPP_Audit_CollectEvidences_UserTask  |
| 2087041 | 2023-01-24 19:55 | Info     | MISNotification | Application: QMS, required object Ray.BPMApp.QMS.Model.BPMAPP_Audit_CollectEvidences_UserTask  |
| 2087040 | 2023-01-24 19:55 | Info     | MISNotification | Scheduled Event Raised. EventId [6c28e52d-79bf-492e-ab0e-ae9701415be7], objectId [00000000-0000-0000-0000-000000000000], userId [2926ce8f-7d40-44c9-b0d0-18ea118d421a] |

If the configuration is correct, you should see synchronized records in both the Scheduler's and **DBP** web server's logs.

If an event is present in the **BpmsSchedulerWindowsServiceLog.txt** but not in the Application Log, it means that the Scheduler can't find the **DBP** web server. The path to the web server is configured in the **BPMSSchedulerWindowsService.exe.config** file.

```
BPMSSchedulerWindowsService.exe.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<appSettings>
  <add key="DatabaseSourceName" provider="MySqlProvider" value="mysqlserver" />
  <add key="ServerUrl" value="https://localhost/dbp/" />
</appSettings>
```

If the **DBP** web server encounters an error during the event execution, you will see details about the error in the Application Log.

| ID      | Date             | Severity | Module          | Message  |
|---------|------------------|----------|-----------------|--|
| 2080584 | 2023-01-24 16:36 | Error    | MISNotification | Rollback because of error: Not loaded assembly of Risk application, type: Ray.BPMApp.Risk.Model.BPMAPP_Risk_Assessment, kind:Model |
| 2080033 | 2023-01-24 16:31 | Error    | MISNotification | Rollback because of error: Not loaded assembly of Risk application, type: Ray.BPMApp.Risk.Model.BPMAPP_Risk_Assessment, kind:Model |
| 2080020 | 2023-01-24 16:30 | Error    | MISNotification | Rollback because of error: Error in BPMS.Action.FetchEpcInformationForTraining.Methode1: Value cannot be null. Parameter name: s   |
| 2079134 | 2023-01-24 16:26 | Error    | MISNotification | Rollback because of error: Not loaded assembly of Risk application, type: Ray.BPMApp.Risk.Model.BPMAPP_Risk_Assessment, kind:Model |

- For example, in the above image, errors 1, 2, and 4 indicate that the **Risk** application was removed, but the Scheduler is not aware of it and continues to call events related to the

application. Error 3 indicates an error in the custom script **FetchEpcInformationForTraining**.

## 6.2. Setting Up BPMS Environment

---

Setting up a BPMS environment involves configuring the necessary software and hardware components to automate and manage business processes effectively.

1. [New BPMS Environment – General Setup](#)
2. [New BPMS Development Environment – Tools Setup](#)
3. [New BPMS Development Environment – General Setup](#)



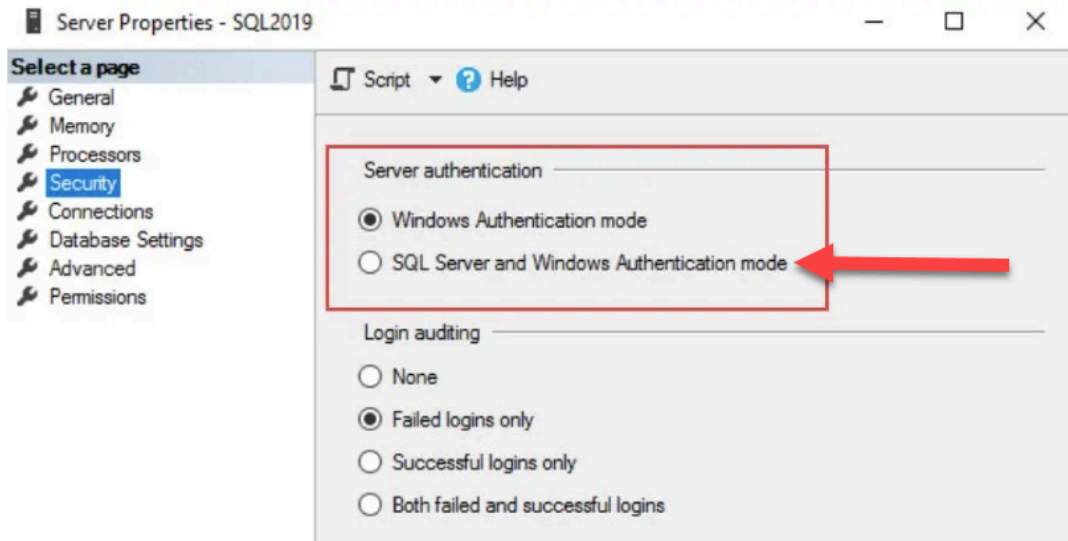
For helpful recordings to guide you through the onboarding process, including beginner trainings for the BPMS platform, please refer to the [Useful Recordings](#) folder in EPC.

## 6.2.1. New BPMS Environment – General Setup

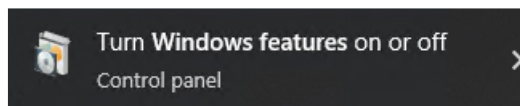
### Step 1 – Set Up Prerequisites

Before setting up the BPMS environment or platform, users must first install the prerequisites.

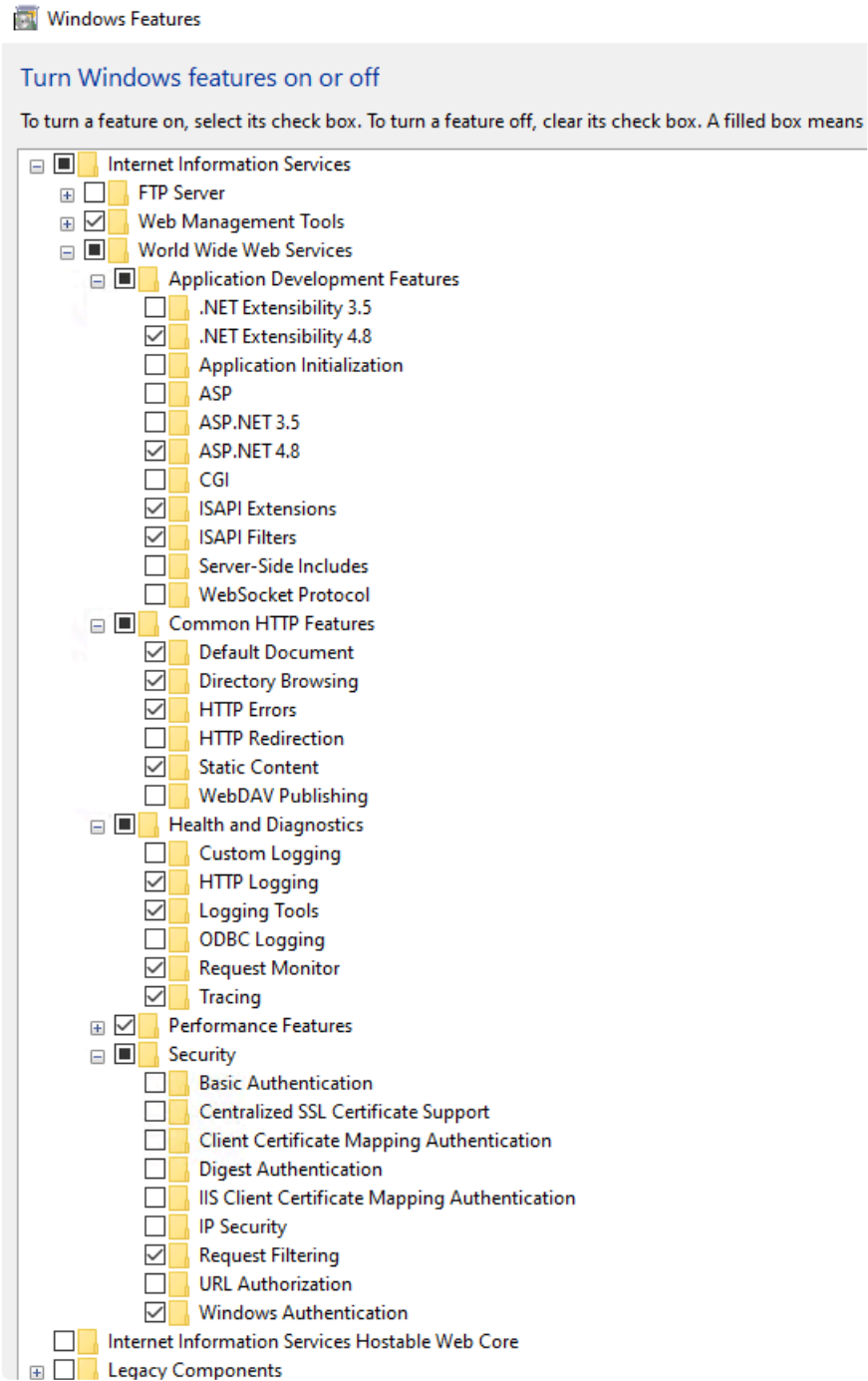
1. Download and install [Visual Studio 2022 Community](#).
2. Download and install [SQL Server 2019 Developer Edition](#).
  - When installing, select Mixed (SQL Server and Windows) Authentication mode.



3. Download and install [SQL Server Management Studio](#).
4. Enable Internet Information Services (IIS).
  - Open the **Windows Start** menu and search for “Turn Windows features on or off”. Click on the corresponding option to open the [Windows Features](#) window.



- Expand the **Internet Information Services** node.
  - Select the following options:



## Step 2 – Set Up BPMS Platform

Once all prerequisites have been installed, users can set up the BPMS platform.

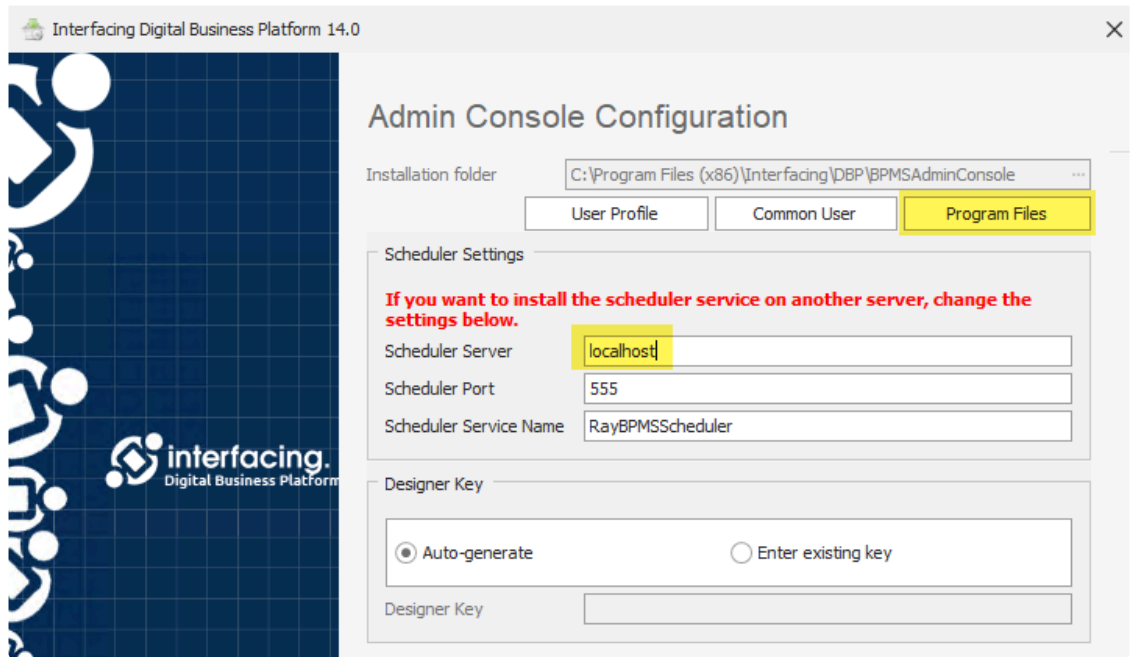
### 1. Run **BPMS Setup**.

- Install all modules (select all options in [Features](#) installer).
  - For the [Database](#) module: create a new database, input the DB username (ray) and password.



Remember to save the DB password somewhere secure and easily accessible.

- For the [Admin Console](#) (Designer) module: if possible, install into the Program Files folder.



Interfacing Digital Business Platform 14.0

### Admin Console Configuration

Installation folder: C:\Program Files (x86)\Interfacing\BPMSAdminConsole

User Profile: ☐ User Profile ☐ Common User ☒ Program Files

Scheduler Settings

**If you want to install the scheduler service on another server, change the settings below.**

Scheduler Server: localhost

Scheduler Port: 555

Scheduler Service Name: RayBPMScheduler

Designer Key

☒ Auto-generate ☐ Enter existing key

Designer Key:

- For the [Web Application](#) module: create a new web application, enable SSL, do not enable SSO (Single Sign-On).

\* To access the latest setup files (including the feature file), navigate to [Research & Development – DBP – Platform – Setup Install – All Documents](#)

## 2. Run **BPMS**.

- To validate if your IIS website is working, open a web browser and enter the following:
  - <https://localhost/dbp/>
  - Default Credentials: admin/admin

! To resolve the issue of the Form Designer not working, it may be necessary to download and install both the [X86](#) and [X64](#) versions of the Microsoft Visual C++ Redistributable package.

## 6.2.2. New BPMS Development Environment – Tools Setup

Tested on Windows 8.1, Windows Server 2019, Windows 10, Windows 11.

### Step 1 – Install Prerequisites

#### 1. Prerequisite Files

Users must download and install the following prerequisites:

- DBPSetupDev.exe \*
- Features.datx \*

\* To access the latest setup files, navigate to [Research & Development – DBP – Platform – Setup Install – All Documents](#).

- AdminApp.rba \*\*

\*\* To access the latest setup files, navigate to [Research & Development – DBP – Platform – RBA Storage – Common Apps to Install on All Servers](#).

#### 2. Optional Tools

Users have the option of downloading and installing the following tools:

- [Windows Terminal](#) (Latest Release)

! Please note that this particular tool requires Windows 10 2004 (build 19041) or later and does not work in Windows 8.1 or Windows Server 2019.

- [PowerShell](#) (Latest Release – 7.3)
  - In the PowerShell Console, run: `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned`

### Step 2 – Install BPMS Platform

#### 1. Install Local SQL Server

Users must install and set up a local SQL Server.

✿ It is recommended for users to install the 2019 Developer edition of the SQL Server.

- [SQL Server Installation](#)

- During the installation process, select:

```
Installation Type = Custom
Installation \ New SQL Server Stand-Alone \ Edition = Developer
Instance Features = Database Engine Services, Client Tools Connect
ivity, Client Tools SDK
Authentication Mode = Mixed, Set sa-password, Add current user as
admin
```

## 2. **Install SQL Server Management Studio (SSMS)**

Users must install the latest release (19.0.2) of the SQL Server Management Studio (SSMS).

- [SSMS Installation](#)

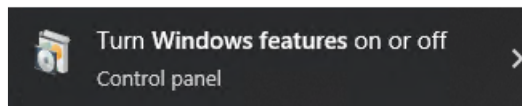
### **CHECKPOINT:**

Run SSMS and connect to local SQL Server as SA user (server=.)

## 3. **Install Internet Information Services (IIS)**

Users must install Internet Information Services (IIS).

- To install **IIS**, follow these steps:
  1. Open the **Windows Start** menu and search for “Turn Windows features on or off”. Click on the corresponding option to open the Windows Features window.



2. In the **Internet Information Services (IIS)** branch, select the following options:

```
Web Management Tools \ (select all except Management Service)
World Wide Web Services \ Application Development Features = .NET
Extensibility 4.7 (4.8), ASP.NET 4.7, ISAPI Extensions, ISAPI Fil
ters
World Wide Web Services \ Common HTTP features \ (select all)
```

### **CHECKPOINT:**

After the installation is complete, open a browser and navigate to <http://localhost/>. If the installation was successful, you should see the IIS welcome page.

## 4. **Add SSL**



Users must add a Secure Sockets Layer (SSL) to their IIS setup.

- Run **IIS Manager**:

```
Default Web Site \ context menu \ Edit Bindings \ Add \ Type = https, S
SL certificate = IIS Express Development Certificate
Default Web Site \ context menu \ Manage Website \ Restart
```

### CHECKPOINT:

Open a browser and navigate to <https://localhost/>. If the installation was successful, you should see the IIS welcome page. Note that the site certificate will initially be invalid. To fix this, import the certificate to the “Trusted root certification authorities” store and then restart the browser. Once you have done this, the certificate will become valid.

## 5. **Install Latest BPMS Platform**

Users must install the latest release of the BPMS platform.

- Install all modules (select all options in [Features](#) installer).
  - For the [Database](#) module: create a new database, server = localhost, DB username = ray, DB password = 12345, admin user = sa, admin password =
  - For the [Admin Console](#) (Designer) module: if possible, install into the Program Files folder.

Interfacing Digital Business Platform 14.0

### Admin Console Configuration

Installation folder:

**Scheduler Settings**

If you want to install the scheduler service on another server, change the settings below.

Scheduler Server:

Scheduler Port:

Scheduler Service Name:

**Designer Key**

☒ Auto-generate
 ☐ Enter existing key

Designer Key:

- For the [Web Application](#) module: create a new web application, local website = default web site, enable SSL, do not enable SSO (Single Sign-On).
- For the [Tools](#) module: select all options

✿ To access the latest setup files (including the feature file), navigate to [Research &](#)

## Development – DBP – Platform – Setup Install – All Documents

### CHECKPOINT:

- Open a browser and navigate to <https://localhost/dbp>. You should see a page without errors in the DevTools console. You should also see a top menu with a **Sign-in** button.
  - Sign-in \ Username = admin, Password = admin
- Once you have logged in, navigate to <https://localhost/dbp/DeployWebService/DeployServiceWS.aspx>. You should see a page without errors in the DevTools console, XML Web service UI.

### 6. **Configure BPMS Designer**

Users must configure the **BPMS Designer**.

- Run **BPMS Designer**:

```
Username = admin, Password = admin
Invalid Web Address = Ok
Settings \ General \ Web Directory = https://localhost/dbp
```

### 7. **Install AdminApp.rba**

Users must install the latest release of the **Admin** application.

✿ To access the latest setup files, navigate to [Research & Development – DBP – Platform – RBA Storage – Common Apps to Install on All Servers](#).

- In **BPMS Designer**:

```
Import AdminApp.rba
Application Tree \ Admin Application \ Forms \ Form Behavior \ select a
ll \ context menu \ Show in Web for All Users
Deploy
```

- Open a browser and navigate to <https://localhost/dbp>.

```
Sign-in \ Username = admin, Password = admin
```

```
Organization Management (Admin) \ Users \ double-click on admin user \  
Select All Permissions, Save
```

### Step 3 – Building BPMS from Sources

#### 1. Install Visual Studio (VS) 2022 Preview Enterprise Edition

Users must install the 2022 Preview Enterprise edition of Visual Studio (VS).

✿ You can also change an existing VS version to Preview: in VS Installer \ More \ Update Setting \ Channel = Preview, then Update VS

- [VS Preview Enterprise Edition Installation](#)
  - During the installation process, select:

```
Workload = ASP.NET, .Net desktop
```

- Sign in to VS with the company account to activate the Enterprise license.

#### 2. Install DevExpress Components

Users must install the required DevExpress components.

- [DevExpress Installation](#)

```
Sign-in \ Email = John.Doe@thecompany.com, Password = DBXPFORE  
My Downloads \ DXperience Subscription \ select version needed to build BPMS \ Unified Component Installer (.exe)
```

- Download and run the program (DevExpressComponentsBundleSetup-21.2.8.exe):

```
Registered install \ Email=John.Doe@thecompany.com, Password = DBXPFORE
```

#### 3. Checkout Sources from GitHub

Users can checkout sources from GitHub.

- Run **VS**:

```
Clone a repository \ GitHub \ https://github.com/the_company/BPMS
```

#### 4. **Switch Branch to the companyDev**

Users must switch their branch in VS to the companyDev.

- In VS:

```
View \ Git Changes \ Branch \ Remotes \ the_companyDev
```

#### 5. **Install NodeJS LTS 16 and Latest Yarn**

Users must install NodeJS LTS 16 and the latest release of Yarn.

- [NodeJS Installation](#)
  - During the installation process, select:

```
Automatically Install Tools (C++, Python, VS Build Tools, Chocolatey) = Off
```

- Run **CMD** (Command Prompt or PowerShell):

```
npm install -global npm  
npm install -global node  
npm install -global yarn
```

#### 6. **Install Dependencies**

Users must install the required dependencies.

- In **CMD**:

```
cd /d %USERPROFILE% \ source \ repos \ the_company \ BPMS \ Source \ Rayvarz.Bpms.Web.App  
yarn install
```

#### 7. **Build JS and CSS bundles**

Users must build the JS and CSS bundles.

- In **CMD**:

```
cd /d %USERPROFILE% \ source \ repos \ the company \ BPMS \ Source \ Rayvarz.Bpms.Web.App
yarn run webpack-dev
```

## 8. **Open Source Folder as a VS Solution**

Users must open the source folder as a VS solution.

- In **VS**:

```
View \ Solution Explorer \ Source \ double-click at Ray.BPMS.sln
```

## 9. **Build WebApp**

Users must build the web application.

- In **VS**:

```
View \ Solution Explorer \ Solution Ray.BPMS\ context menu\ Restore NuGet Packages
Build\ Clean Solution
Build\ Rebuild Solution
```

## 10. **Add WebApp to IIS**

Users must add the web application to IIS.

- In **IIS Manager**:

```
Default Web Site \ context menu \ Add Application \ Alias = dbp-dev, AppPool = BPMS, Path = %USERPROFILE% \ source \ repos \ the company \ BPMS \ Source \ Rayvarz.Bpms.Web.App \
Default Web Site \ Folder "dbp-dev\DeployWebService" \ context menu \ Convert to Application
```

- In **Windows File Explorer**:

```
%USERPROFILE% \ source \ repos \ the company \ BPMS \ Source \ Rayvarz.Bpms.Web.App \ context menu \ Properties \ Security \ Grant Modify permissions
```

```
missions to user NetworkService
```

- Copy:
  - "C:\inetpub\wwwroot\dbp\Connection\Connections.xml" to "%USERPROFILE%\source\repos\the company\BPMS\Source\Rayvarz.Bpms.Web.App\Connection\"
  - "C:\inetpub\wwwroot\dbp\Bin\Features.datx" to "%USERPROFILE%\source\repos\the company\BPMS\Source\Rayvarz.Bpms.Web.App\bin\"
  - "C:\inetpub\wwwroot\dbp\Bin\Features.datx" to "%USERPROFILE%\source\repos\the company\BPMS\Bin\"
  - "%USERPROFILE%\source\repos\the company\BPMS\Source\Rayvarz.Bpms.Web.App\bin\Roslyn" to "%USERPROFILE%\source\repos\the company\BPMS\Source\Rayvarz.Bpms.Web.App\DeployWebService\bin\"

#### CHECKPOINT:

- Open a browser and navigate to <https://localhost/dbp-dev>. You should see a page without errors in the DevTools console. You should also see a top menu with a **Sign-in** button.
  - Sign-in \ Username = admin, Password = admin
- Once you have logged in, navigate to <https://localhost/dbp/DeployWebService/DeployServiceWS.aspx>. You should see a page without errors in the DevTools console, XML Web service UI.

## 11. Build Designer

Users must build the Designer.

- In **VS**:

```
View \ Solution Explorer \ Designer \ BPMS.Designer \ context menu \ Set as Startup Project
Debug \ Run Without Debugging
```

- If the versions of the Designer and the database do not match, you will need to manually apply migration scripts:
  - Connect SSMS to DBP database
  - Choose appropriate scripts from %USERPROFILE%\source\repos\the company\BPMS\Source\Rayvarz.BPMS.Setup\Source\Scripts\ and run it as queries in ascending version order.

- Reopen Designer

## 12. **Configure Designer**

Users must configure the Designer.

✱ Since the database is the same, the AdminApp.rba has already been imported. Therefore, the focus now is just to ensure a smooth deployment process.

- In **BPMS Designer**:

```
Settings \ General \ Web Directory = https://localhost/dbp-dev  
Application Tree \ OrgManagement  
Deploy
```

- Open a browser and navigate to <https://localhost/dbp>.

```
Sign-in \ Username = admin, Password = admin  
Organization Management (Admin) \ Users \ double-click on admin user \  
Select All Permissions, Save
```

## 13. **Troubleshooting**

To effectively troubleshoot issues, various troubleshooting steps can be taken in different components of the development environment

- In **VS**:

```
Close and reopen VS, Clean Solution, Rebuild Solution
```

- In **IIS Manager**:

```
Default Web Site \ context menu \ Manage Website \ Restart
```

- In **BPMS Designer**:

```
Settings \ System \ Debug Mode = On
```

## 6.2.3. New BPMS Development Environment – General Setup

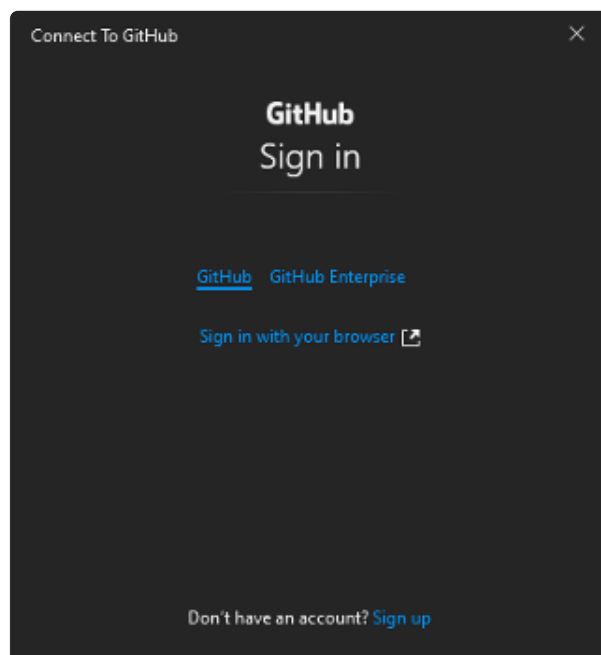
### Step 1 – Install Dev Tools

1. Download and install [Visual Studio 2022 Community](#).
2. Download and install [Node.JS](#).
3. Download and install [SQL Server 2022 Developer Edition](#).
4. Download and install [SQL Server Management Studio](#).
5. Download and install [Git](#).
6. Download and install [DevExpress](#).
  - Installation Path: O:\Product\Product - BPMS\For Developers\DevExpress
7. Open Command Line
  - Install Yarn by running command: `npm install -global yarn`

### Step 2 – Clone BPMS Repository

1. Create an account on [GitHub](#) if you don't already have one.
2. Request access to the BPMS repository from your manager.
3. Open Visual Studio (VS) and clone the repository.
  - Repository Location: [https://github.com/the\\_company/BPMS.git](https://github.com/the_company/BPMS.git) company/BPMS.git.

\* You will be prompted to enter your credentials through the web browser. If the browser doesn't open automatically, try changing the default browser, such as switching from Google Chrome to Microsoft Edge.



4. Open the solution Ray.BPMS.sln.
5. Build the solution.
6. Ask your manager for the feature file and paste it to the folder: C:\Users\YourUserName\source\repos\BPMS\Bin.



- Optionally, you can create a desktop shortcut pointing to: `C:\Users\YourUserName\source\repos\BPMS\Bin\BPMS.Designer.exe`.

### Step 3 – Run BPMS on Your IIS

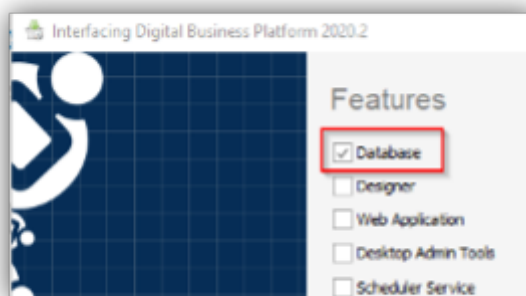
- Enable IIS
  - Installation Instructions: <https://www.howtogeek.com/112455/how-to-install-iis-8-on-windows-8/>
- Open IIS.
- Right-click on Default Web Site and select the **Add Application** option from the context menu.
  - Alias: BPMS
  - Physical Path: `cloned repository path\Source\Rayvarz.Bpms.Web.App`.
    - Like: `C:\Users\YourUserName\source\repos\BPMS\Source\Rayvarz.Bpms.Web.App`
- Expand the application.
- Right-click on DeployWebService and select the **Convert to Application** option from the context menu.

### Step 4 – Install Database

- Open the folder: `O:\Product\Product - BPMS\Install\--Latest--`
- Then, open the most recent folder that follows the format: `YYYY.MM MonthName`
- Copy the latest setup application from that folder and paste it onto your machine.
- Run the setup application.
  - Note:** To run the setup, make sure you have the corresponding feature file.

✿ To access the latest setup files, navigate to [Research & Development – DBP – Platform – Setup Install – All Documents](#).

- From the Features installer, select only the **Database** module.



### Step 5 – BPMS Admin

- Run the Admin.exe file as an administrator, from: `C:\Program Files (x86)\the company\DBP\Setup\Source\Tools\Admin`.
- Click on **Statute**.
- Select **Users** and choose **admin**.
- Check everything under Workflow.

5. Save and close

## Step 6 – Running BPMS

1. To validate if your IIS web site is working, open a browser and navigate to <https://localhost/BPMS>.
2. Run the **BPMS Designer** from: `C:\Users\YourUserName\source\repos\BPMS\Bin\BPMS.Designer.exe`

## 7. Frequently Used Functions

---

Below are some of the most frequently used features of the software that users should be familiar with:

1. [Data Manipulation](#)
2. [Digital Signature Configuration](#)

## 7.1. Data Manipulation

---

To manipulate the data presented in the table of a given tab or window, apply any of the following techniques:

### 1. **Filtering**

There are three ways to filter data in the table: by selecting filter options from the dropdown menu in each column header, by using a Boolean search to filter the table by specific criteria, or by entering a search query in the search field.

For more information on these methods, please refer to the corresponding section listed below.

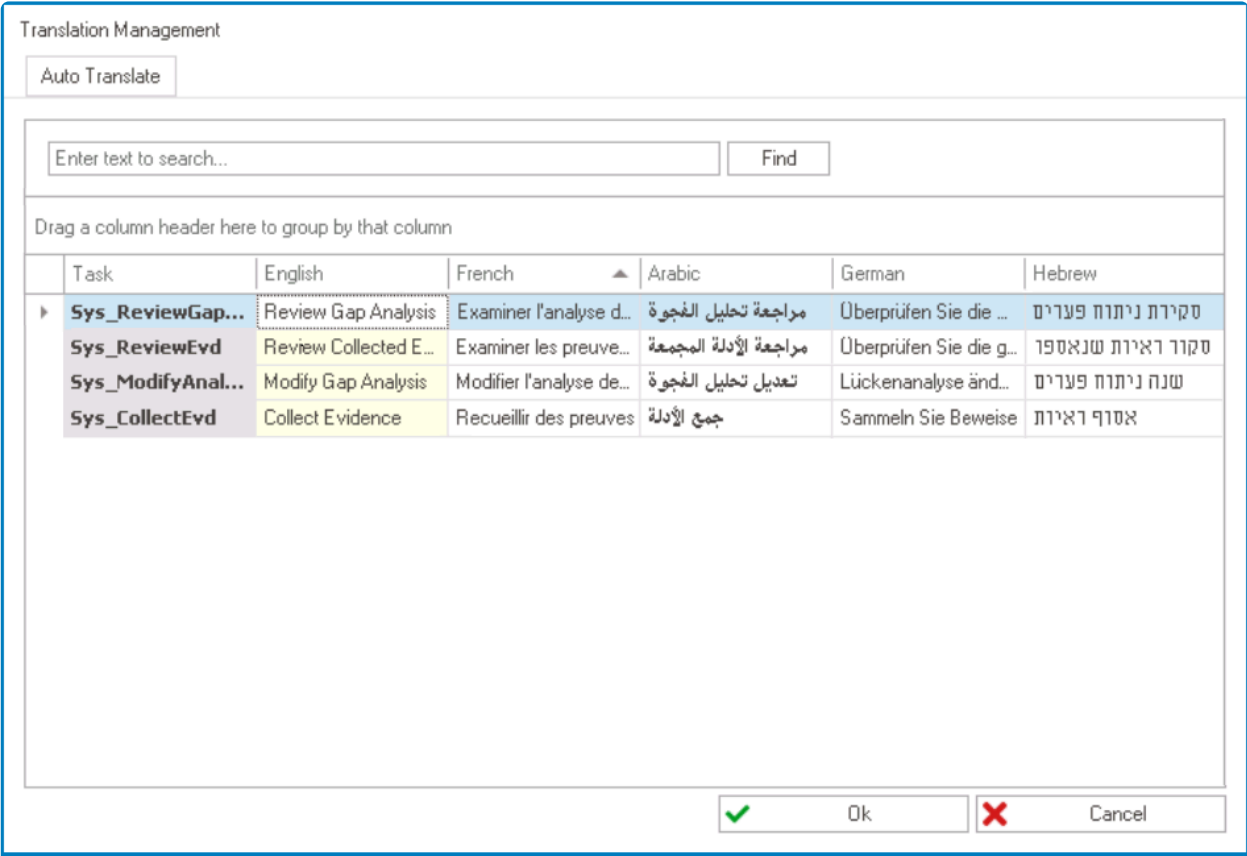
- [Filter Options](#)
- [Boolean Search](#)
- [Search Field](#)



Please note that Boolean search may not be available in all windows or tabs of the program.

### 2. **Sorting**

Users can sort items in the table by clicking on the column headers. For example, to sort by French, click on the 'French' column header.



Items can be sorted by either ascending or descending order.

- Ascending order is indicated by an upward arrow.

French ▲

- Descending order is indicated by a downward arrow.

French ▼

3. **Grouping**

Users can also group items in the table by dragging a column header and dropping it onto the **Drag & Drop** field.

Translation Management

Auto Translate

Enter text to search...

Find

English

|   | Task                                 | French | Arabic | German | Hebrew |
|---|--------------------------------------|--------|--------|--------|--------|
| ▶ | > English: Collect Evidence          |        |        |        |        |
|   | > English: Modify Gap Analysis       |        |        |        |        |
|   | > English: Review Collected Evidence |        |        |        |        |
|   | > English: Review Gap Analysis       |        |        |        |        |

✓

Ok

✗

Cancel

\* The column header remains toggleable in this scenario, allowing users to sort items by ascending or descending order.

Multiple column headers can be used at a time for grouping purposes.

Translation Management

Auto Translate

Enter text to search...

Find

English

French

| Task                                 | Arabic | German | Hebrew |
|--------------------------------------|--------|--------|--------|
| English: Collect Evidence            |        |        |        |
| > French: Recueillir des preuves     |        |        |        |
| > English: Modify Gap Analysis       |        |        |        |
| > English: Review Collected Evidence |        |        |        |
| > English: Review Gap Analysis       |        |        |        |

✓

Ok

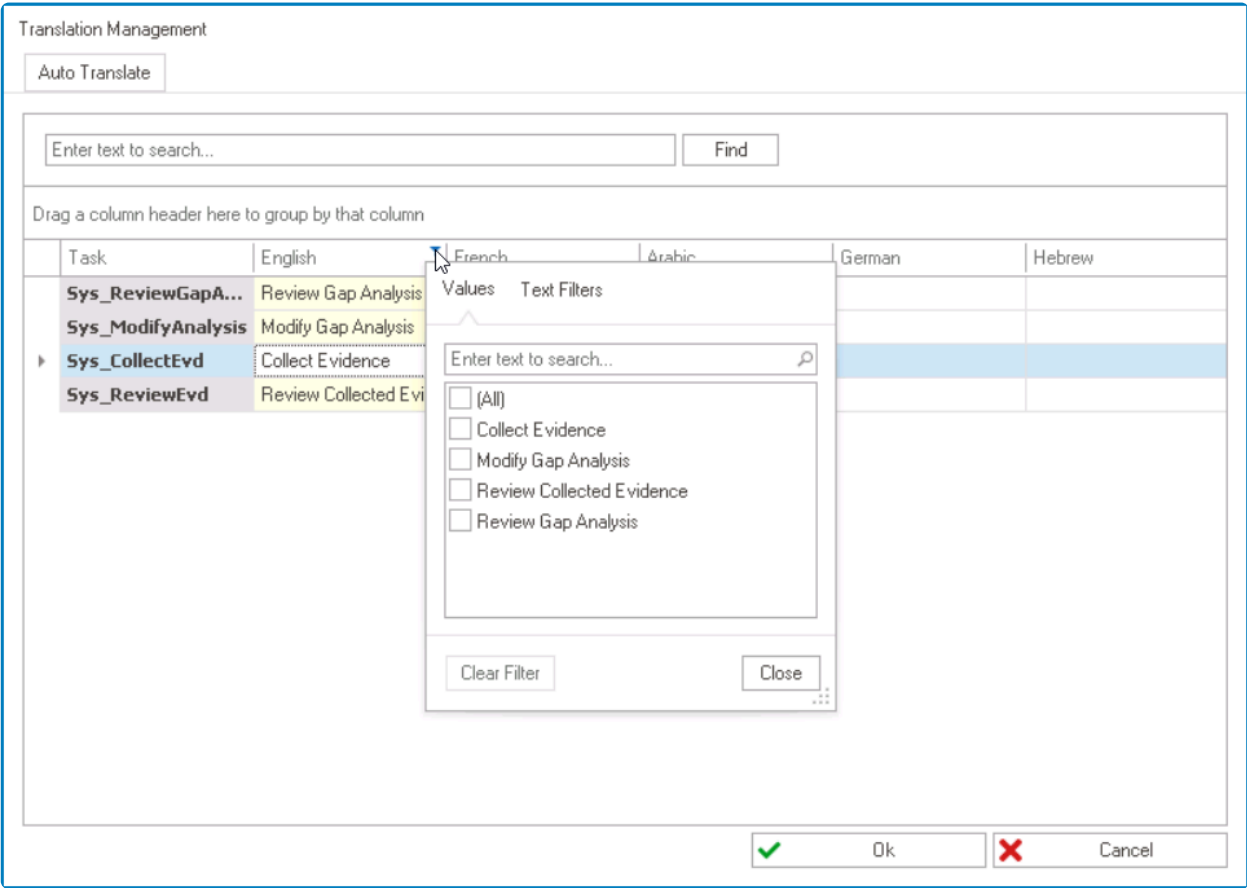
✗

Cancel

✿ To ungroup items, simply drag and drop the column header back onto the table.

# 7.1.1. Filter Options

Users can apply a filter by selecting the filter icon on a column header. It will display the list of values in that column. Users can then select one or more values to filter the data and display only the rows that meet the specified criteria.



✿ To access the filter icon, simply hover the cursor over the column header.

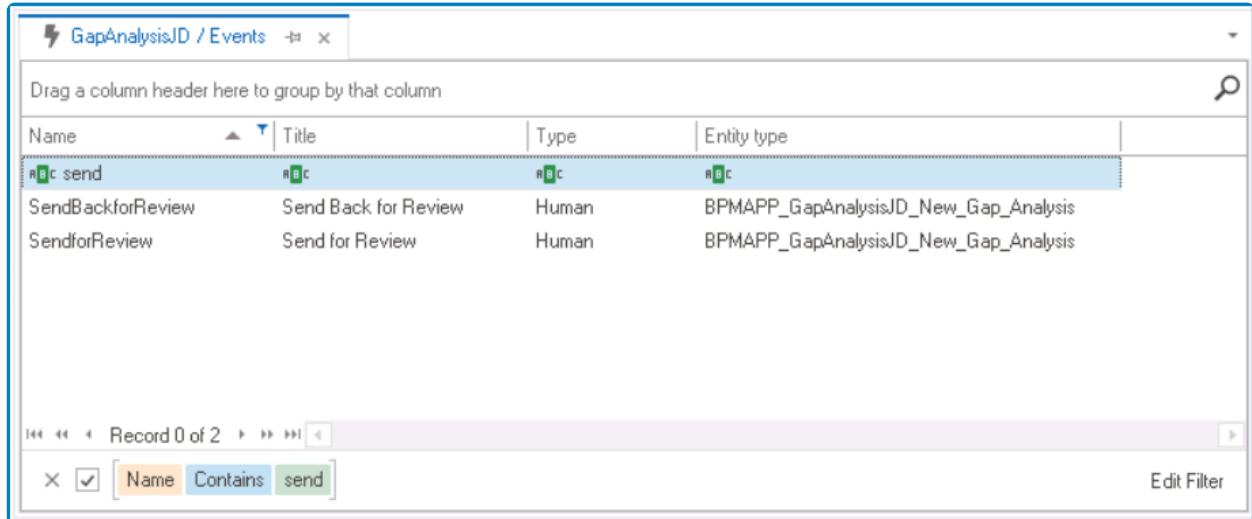


## 7.1.2. Boolean Search

Users can use a Boolean search to refine their search results by combining search terms using logical operators such as AND, OR, and NOT.

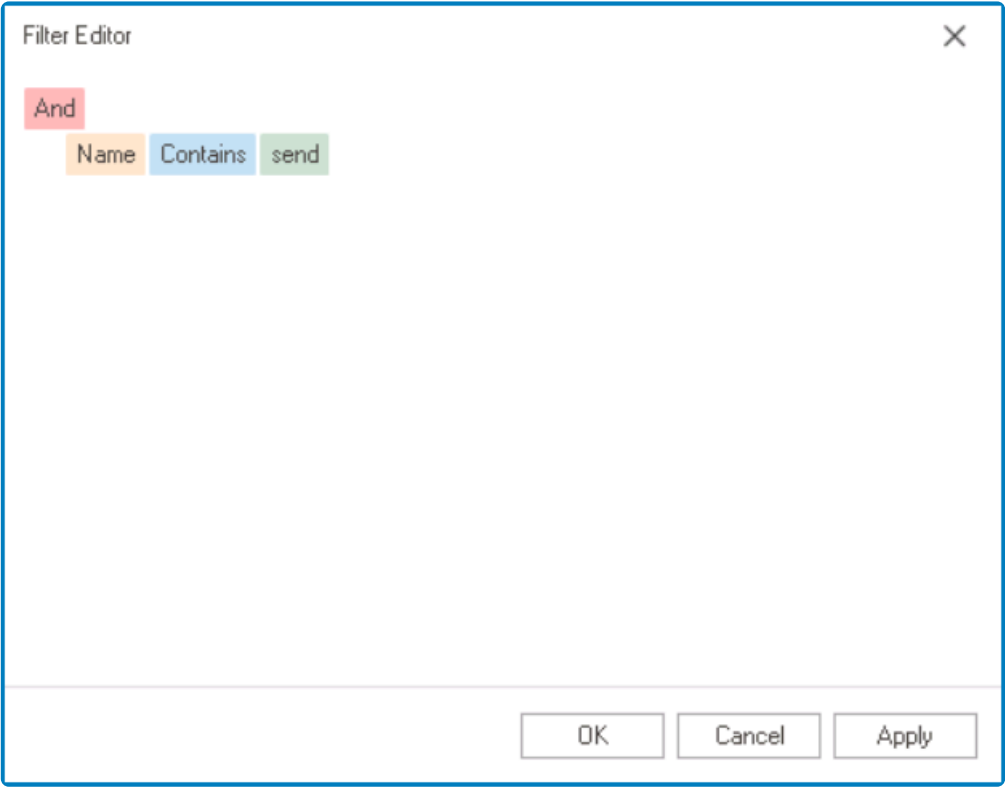
To perform a Boolean search, apply the following steps:

1. Enter a search term in the search bar located directly below the appropriate column header. It will both filter the results in the tab and display the search filter panel.



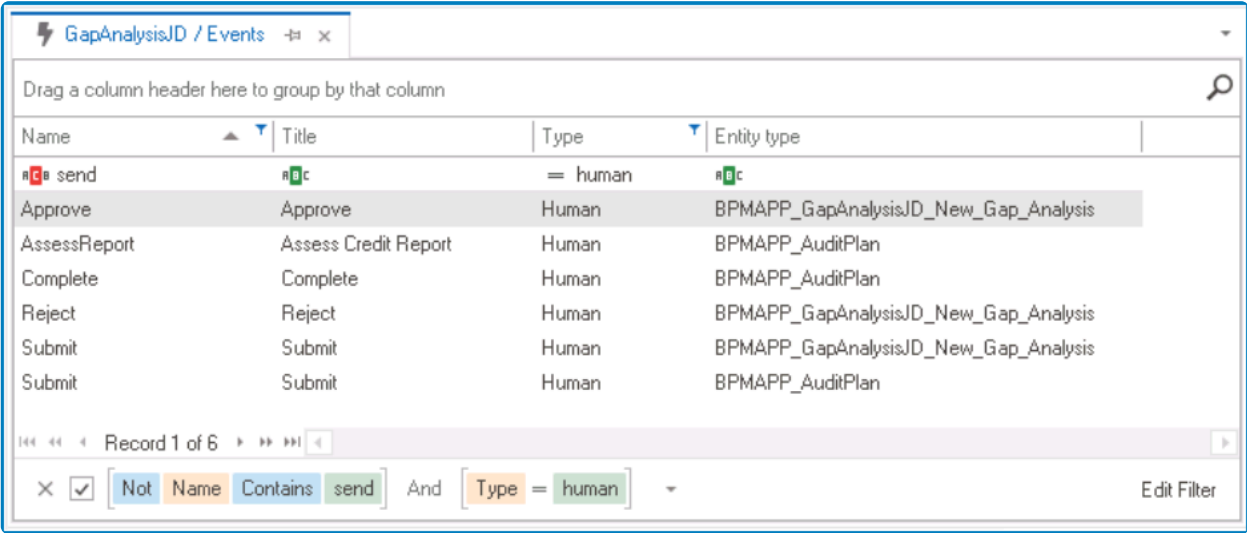
✿ Each search bar in this row allows users to search for specific information within that column. For example, to search for a name, enter your search term in the search bar located under the 'Name' column header.

2. In the search filter panel, click on the **Edit Filter** button. It will display the Filter Editor window, in which you can modify the search filter as needed.



✿ For more information on the features of the [Filter Editor](#) window, [click here](#).

3. After adjusting the search filter as needed, click on the **OK** button to apply your changes and exit the window. The table will now display items filtered according to your search criteria.

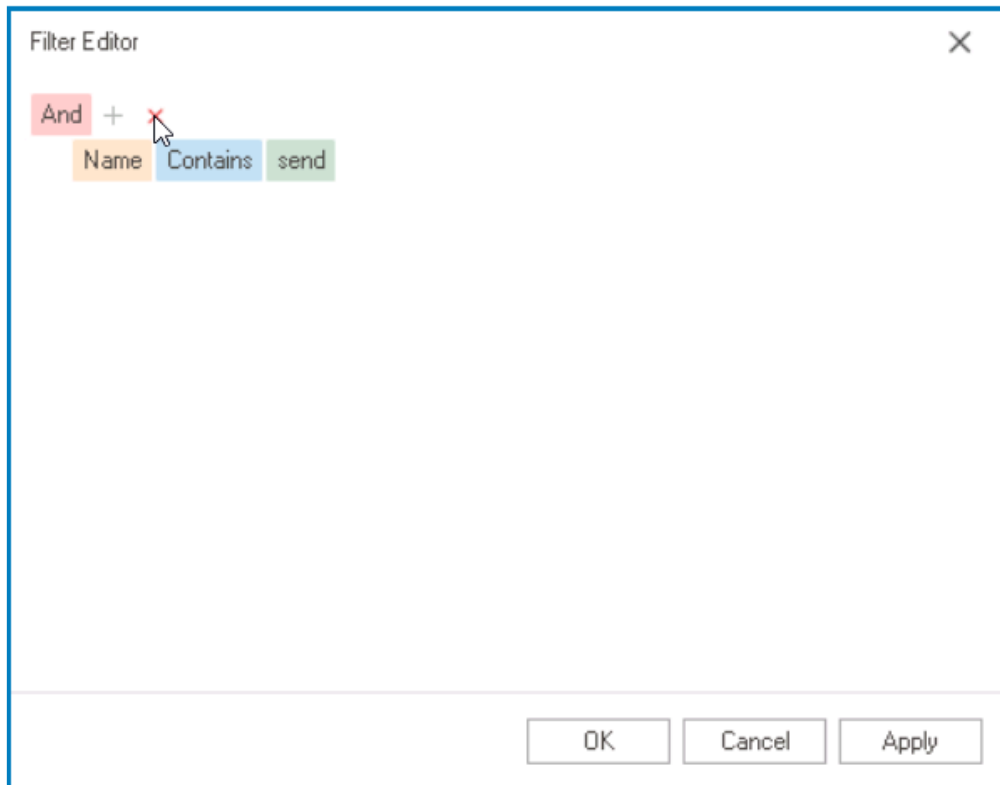


✿ To cancel the search, click on the **x** button. To undo the search term and stop filtering items, unselect the checkbox. Note that the search remains active, and you can activate it again by selecting the checkbox.

## 7.1.2.1. Filter Editor

In the Filter Editor window, users can add new conditions and groups, as well as delete existing ones to refine their search results.

- To add new conditions and/or groups, click on the **+** button or press the **Insert** or **Add** key.
- To remove conditions and/or groups, click on the **x** button or press the **Delete** or **Subtract** key.



Users can also adjust the search filter by editing the logical operator, column value, search parameter, and search term.



By default, the search parameter is set to 'Contains', meaning that it will search the table for items that contain the inputted search term. Users can modify this search parameter as needed.

# 7.1.3. Search Field

Users can filter the results displayed in the table by entering a query in the search field and clicking either the **Enter** key or the **Find** button.

Translation Management

Auto Translate

modify

Find

Drag a column header here to group by that column


|   | Task              | English             | French | Arabic | German | Hebrew |
|---|-------------------|---------------------|--------|--------|--------|--------|
| ▶ | Sys_ModifyAnal... | Modify Gap Analysis |        |        |        |        |

✓

Ok

✗

Cancel

\* To display the search field in a given tab, users will need to click on the  icon in the far right corner.

## 7.2. Digital Signature Configuration

Digital signatures provide a secure way to confirm approvals and validate actions within a process or workflow. In the upcoming sections, we will explore the intricacies of configuring and implementing digital signatures for specific [events](#) within a workflow. We will also discuss their practical application in the web environment.



An event is essentially a type of action that occurs during the execution of a process. By acting as point of communication between flow objects, an event enables the process to move from one stage to another. It is represented in the process diagram by the transition arrow (→) between the flow objects.

Before diving into the step-by-step instructions, please make sure that you have the necessary prerequisites in place.

### Prerequisites

- **Application:** Ensure that you have created an application.
  - [For information on how to create an application in the Designer program, click here.](#)
- **Starter Form:** Ensure that you have created and designed the initial form for your workflow.
  - [For information on how to create and design a form in the Designer program, click here.](#)
- **Application Workflow:** Ensure that you have created and designed a process or workflow for your application. Your workflow should include a start object, at least one human state, an end object, and events linking these components.
  - [For information on how to create and design a process in the Designer program, click here.](#)



While the start event is automatically generated for the transition between the start object and the first human state, you will need to create additional events in the process, linking from the human state onward. [For information on how to create events in the Designer program, click here.](#)

- **Form Behaviors:** Ensure that you have configured form behaviors for each human task in your workflow.
  - [For information on how to create form behaviors in the Designer program, click here.](#)

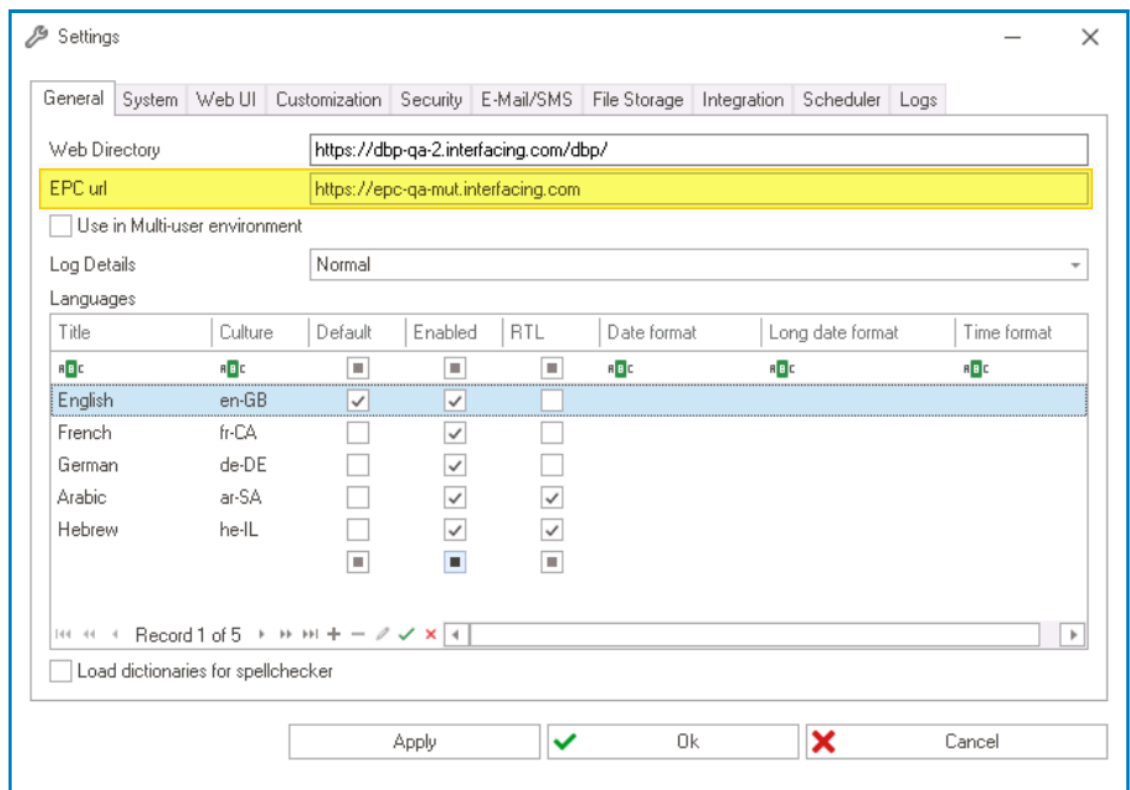
1. [Step-by-Step Guide for Digital Signatures](#)
2. [Digital Signatures in the Web](#)

## 7.2.1. Step-by-Step Guide for Digital Signatures

In this section, we will provide comprehensive step-by-step instructions to guide users through the process of configuring and implementing digital signatures for specific events within their workflow.

### Step 1 – Configure Designer Settings

1. Launch the Designer program and log in.
2. In the menu bar, click on the **Settings** button. This will open the [Settings](#) window.
3. In the [General](#) tab of the [Settings](#) window:
  - Enter the URL of the EPC server associated with the **DBP** web application.
    - For example: **<https://epc-qa-mut.interfacing.com>**



4. In the [Integration](#) tab of the [Settings](#) window:
  - Select the **Enabled** checkbox to activate the PDF E-Signature functionality.
  - Enter the EPC server URL with '/signature' appended to it to call the service.
    - For example: **<https://epc-qa-mut.interfacing.com/signature>**

The screenshot shows a 'Settings' window with a key icon in the title bar. It has several tabs: General, System, Web UI, Customization, Security, E-Mail/SMS, File Storage, Integration (selected), Scheduler, and Logs. Under the 'Integration' tab, there are two sections. The first is 'Exchange Web Services' with fields for 'Server URL', 'Proxy User name', and 'Password'. The second section, 'PDF E-Signature', is highlighted with a yellow background and contains a checked 'Enabled' checkbox and a 'Service URL' field with the value 'https://epc-qa-mut.interfacing.com/signature'. At the bottom of the window are buttons for 'Apply', 'Ok' (with a green checkmark icon), and 'Cancel' (with a red X icon).

5. Finally, click the **OK** button to save your changes and close the window.

## Step 2 – Configure DBP Settings

1. Open the **DBP** web application and log in.
2. In the menu bar, click on **Facilities**, then select **Settings** from the dropdown. This will open the Personalization form.
3. In the Personalization form, under the **Connection to EPC** category, set up a connection to the specified EPC server.
  - Provide the base EPC server URL (the same URL used in Designer).
  - Enter the username and password of an EPC administrator.
  - Click the **Test** button to verify the provided credentials.
  - Select an environment from which data will be pulled.

### Personalization

**Server-wide**   User

#### Colors

Background color: 

#0371c3

Background color for required fields: 

#dbf0ff

Transparent background for readonly fields: ☒

Restore default values

#### Task Form Auto-Save

Enabled: ☐

Interval (min): 

5

#### Connection to EPC

Base URL: 

https://epc-qa-mut.interfacing.cor

Username: 

epcadmin

Password: 

.....

Test

Default environment: 

Aerospace and Defense

Save

\* Attention: To see the changes, refresh the cache (Ctrl-F5).

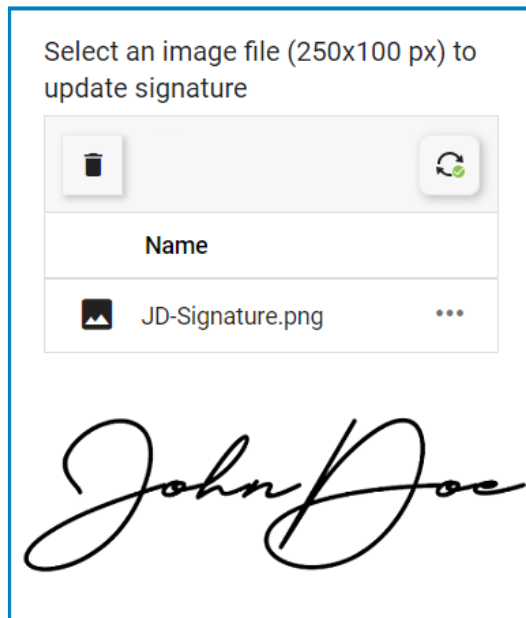
4. Finally, click the **Save** button to save your settings and establish the EPC connection.

! To ensure that changes are reflected in the web application, remember to refresh the cache by pressing **CTRL + F5** after saving.

### Step 3 – Upload User Signatures

1. From the **DBP** web application's Home Page, navigate to the [Admin](#) application.
2. Access the [Organization Management](#) module.
3. In the [Users](#) tab, ensure that for the users requiring a signature, an image file (PNG format) of their signature has been uploaded.






- These uploaded images will be used to display the users' signatures in the generated PDF files.

#### Step 4 – Configure Human Event Signature Requirement

1. Return to the Designer program, and once there:
  - Navigate to **Processes > Process Design** (double-click to open). This will open the Process Edit tab.
  - In the Process Edit tab, double-click on the latest version of your process. This will open the Process Designer in a new tab.
2. In the Process Designer, select the human event (transition arrow) for which you want to require a signature.
  - The associated properties will be displayed in the **Properties** panel.

! You cannot require a signature for the first event in a workflow—that is, the event linked to a workflow's start object, referred to as the '**start human event**'. This feature is only available from the second event onward.

3. Next to the **Event** field in the panel, click on the  button. This will open the Event Properties window.
4. In the General tab of the Event Properties window, select the **Signature Required** checkbox.

**Event Properties**

Apply OK Cancel

General Advanced

Event Type: Human

Name: Approve

English Title: Approve

French Title:

German Title:

Arabic Title:

Hebrew Title:

Event Icon: [Icon] [Add] [Remove]

☐ Can run background ☐ Has TimeSheet ☒ Signature Required


Scheduling

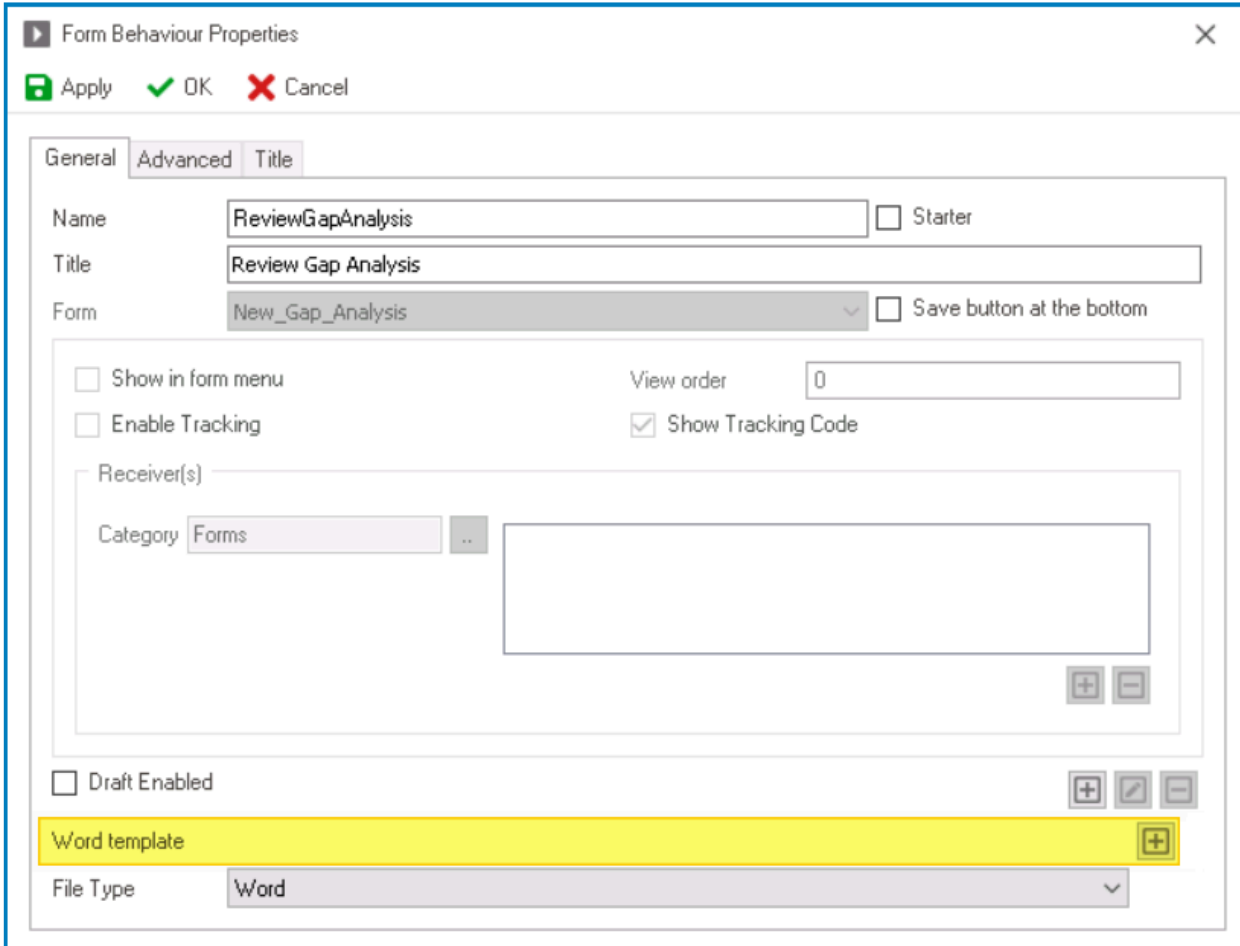
☐ In Process ☐ Run automatically(without API)

Add/Edit Delete

5. Click the **OK** button to save your changes and close the window.
  - A message will remind you to assign a template to the appropriate form behavior.
6. Click the **Save** button to save the process.
7. Finally, click the **Publish** button to publish the process.

### Step 5 – Assign Template to Form Behavior

1. While still in the Designer program:
  - Navigate to **Forms > Form Behavior** (double-click to open). This will display the Form Behavior tab.
2. Inside the Form Behavior tab, double-click on the form behavior for the first task in the workflow with a signature requirement. This will open the [Form Behavior Properties](#) window.
3. In the [General](#) tab of the [Form Behavior Properties](#) window, click on the  button next to the **Word Template** field. This will open the [Word Template Editor](#) window.

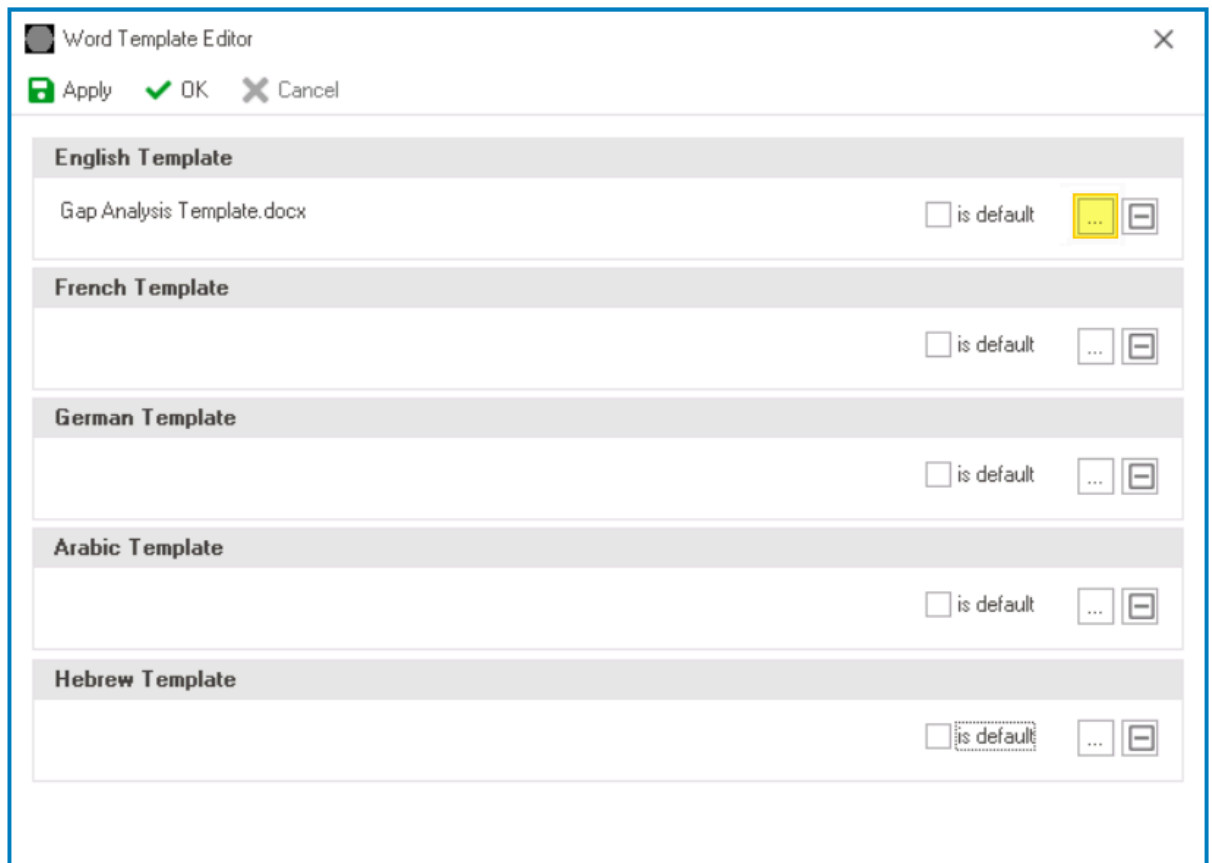


The image shows a 'Form Behaviour Properties' dialog box with three tabs: 'General', 'Advanced', and 'Title'. The 'General' tab is active. It contains the following fields and controls:

- Name:** Text box containing 'ReviewGapAnalysis'. A checkbox labeled 'Starter' is to its right.
- Title:** Text box containing 'Review Gap Analysis'.
- Form:** Dropdown menu showing 'New\_Gap\_Analysis'. A checkbox labeled 'Save button at the bottom' is to its right.
- Options:** Two checkboxes on the left: 'Show in form menu' and 'Enable Tracking', both unchecked. On the right, a 'View order' text box contains '0', and a checked checkbox labeled 'Show Tracking Code'.
- Receiver(s):** A section containing a 'Category' dropdown menu with 'Forms' selected, followed by a large empty text box and two small square buttons with '+' and '-' symbols.
- Draft Enabled:** An unchecked checkbox.
- Word template:** A yellow-highlighted text box containing '\word template' and a '+' button.
- File Type:** A dropdown menu showing 'Word'.

4. In the Word Template Editor window:

- Upload the template file you intend to use.
- Click on the **OK** button to save your changes and close the window.



5. In the General tab of the Form Behavior Properties window:

- Select PDF from the **File Type** dropdown.
- Click on the **OK** button to save your changes and close the window.

**Form Behaviour Properties**

Apply OK Cancel

General Advanced Title

Name  ☐ Starter

Title

Form  ☐ Save button at the bottom

☐ Show in form menu View order

☐ Enable Tracking ☒ Show Tracking Code

Receiver(s)

Category  ..

☐ Draft Enabled

Word template **en: Gap Analysis Template.docx**

File Type

\* Tasks in the same process do not necessarily require individual templates. Assigning one template will apply it to all other tasks within the same process that have a signature requirement.

! A current limitation of this feature is that signatures cannot be applied to subprocesses within a workflow. The template and task to which the signature is related must belong to the same main workflow.

## Step 6 – Deploy Application to Web

1. In the menu bar of the Designer program, click on the **Deploy** button to deploy the application. This action ensures that the latest changes, including the configured digital signature(s), are accessible on the web.

## 7.2.2. Digital Signatures in the Web

With your digital signatures configured and implemented, it's essential to understand how they function within the **DBP** web application.

When an event requiring a signature is triggered in the web application, a pop-up window will appear prompting users to authenticate their signature with a unique code.


### Approval Authentication

We sent you a secret code to your email. Please copy and paste it here.

Code:

*By Approving with your e-signature, you are signing the document electronically. You agree that your electronic signature has the same legal validity and effect as your handwritten signature on the document, and that it has the same meaning as your handwritten signature.*

Users will receive this code via email.

Validation code 

N

noreply@interfacing.com

To: John Doe

Tue 9/12/2023 9:07 AM

Your validation code is: 342992  
This code expires on: 12/09/2023 07:43:01

Once the code is entered, the event will be successfully signed off, and the task will be marked as completed. Users can then access their signature by viewing a PDF template of the form.



✿ Multi-signature is supported on PDF files.

## 8. BPMS Designer General Features

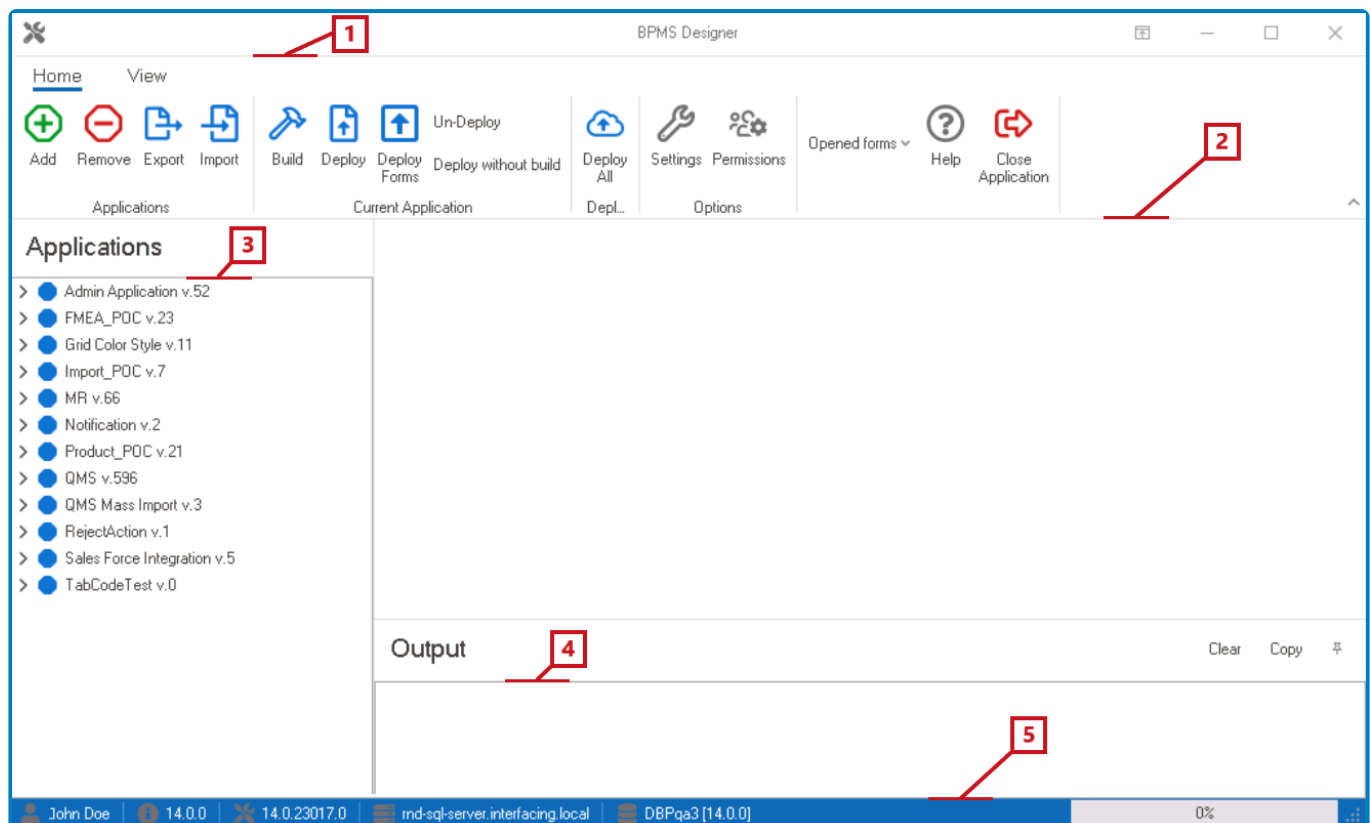
Once users have installed/updated the [Admin Console](#) module, they can access the Digital Business Platform (DBP) system by running a program called “AdminConsole.exe” on their computer.

Users will be asked for their credentials upon running this program.



✿ Depending on the access level specified in Admin.exe or during installation, user authentication can be either enabled or on Windows.

Once users have logged in, they will be directed to the main window of the program.



This window consists of the following elements:




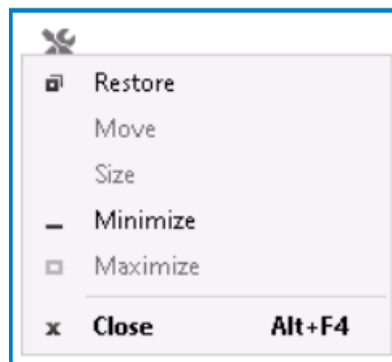
1. [Title Bar](#)
2. [Menu Bar](#)
3. [Application Tree](#)
4. [Output](#)
5. [System Information](#)

## 8.1. Title Bar


The title bar is located at the top of the window. It displays the name of the program and provides users access to a range of functions.



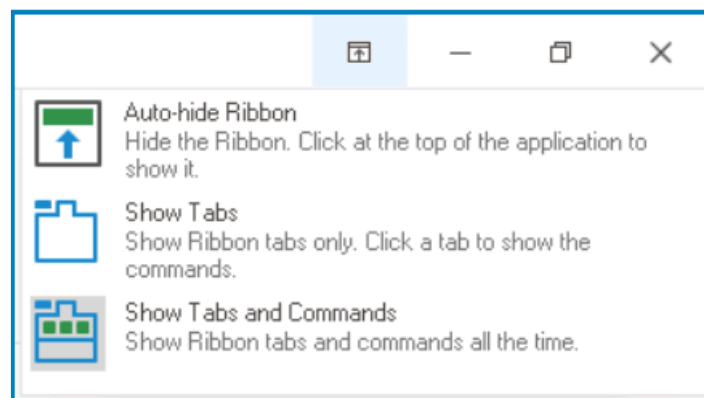
1. **System Menu:** This is a dropdown menu that displays the list of window operation options.
  - Users can access this menu by either clicking on the  icon or right-clicking on the title bar.



- Depending on the current state of the window, certain operations might be disabled in the system menu. To access these operations, users will need to alter the state of the window by either maximizing or restoring it.

 To enable the Move option in the system menu, for example, users will need to restore the window to its default size. Users can also, however, move a maximized window by dragging the title bar.

2. **Ribbon Display Options:** This button allows users to access/modify the ribbon display options. The ribbon, here, refers to the [menu bar](#).
  - If this button is selected, it will reveal the following menu:



- **Auto-Hide Ribbon:** This option allows for both the ribbon and title bar to be hidden. To show them again, users will need to click on the blue bar now at the top of the

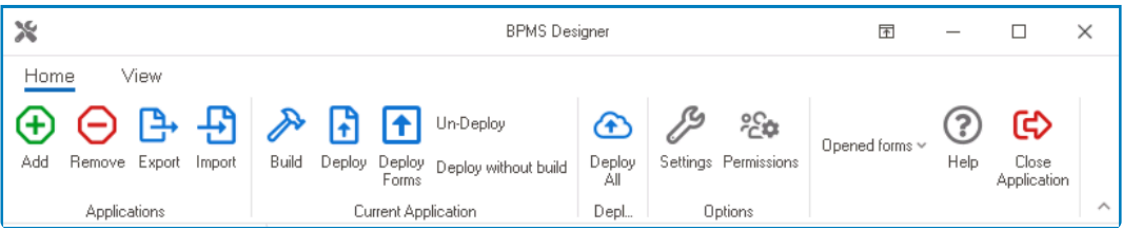
application.



- **Show Tabs:** This option allows for only the tabs of the ribbon to be shown. To access its commands, users will need to click on the relevant tab.




- **Show Tabs and Commands:** This option allows for the ribbon's tabs and commands to be shown at all times. This is the default setting.



- 3. **Minimize:** This button allows users to minimize the window.
- 4. **Maximize:** This button allows users to maximize the window.
  - Once a window has been maximized, the **Maximize** button disappears and is replaced with the **Restore Down** button.



- **Restore Down:** This button allows users to restore the window back down to its default size.

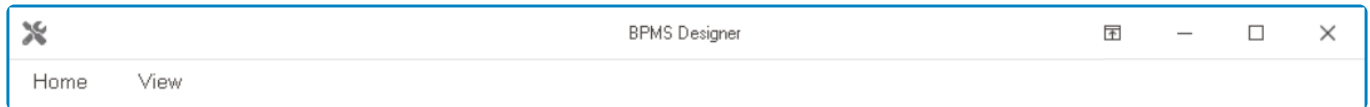


To resize a window, users must first restore it to its default size. A maximized window cannot be resized.

- 5. **Close:** This button allows users to close the window.


## 8.2. Menu Bar

The menu bar, also known as the ribbon, is located just below the title bar.



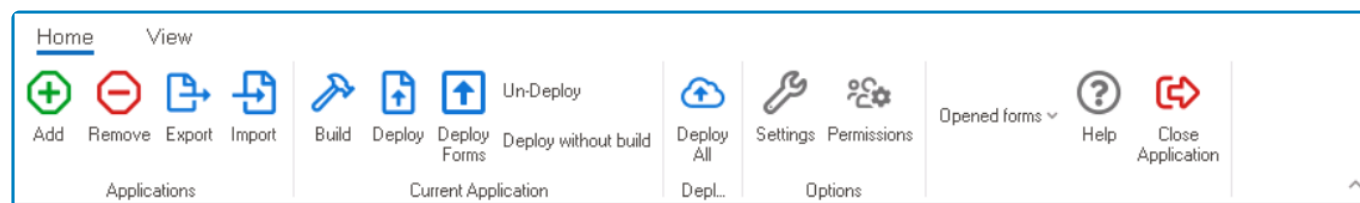
It is comprised of two tabs:

1. [Home](#)
2. [View](#)

✿ By default, the ribbon is set to classic view. To switch between the classic and simplified ribbon, toggle the  icon in the lower right corner.

## 8.2.1. Home Tab

The Home tab is the default tab of the menu bar, displayed every time that the Designer is launched. It is divided into groups of related commands.



These groups/commands include:

1. [Applications](#)
2. [Current Application](#)
3. [Deploy](#)
4. [Options](#)
5. [Opened Forms](#)
6. [Help](#)
7. [Close Application](#)

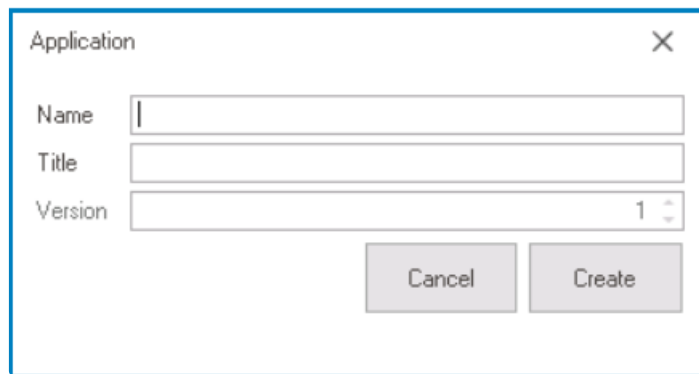
✿ Depending on which module of the [Application Tree](#) is in use, the Home tab may expand to include additional options. For more details on this matter, [click here](#).

## 8.2.1.1. Applications

In the **Applications** group of the Home tab, users are provided access to the following set of commands:



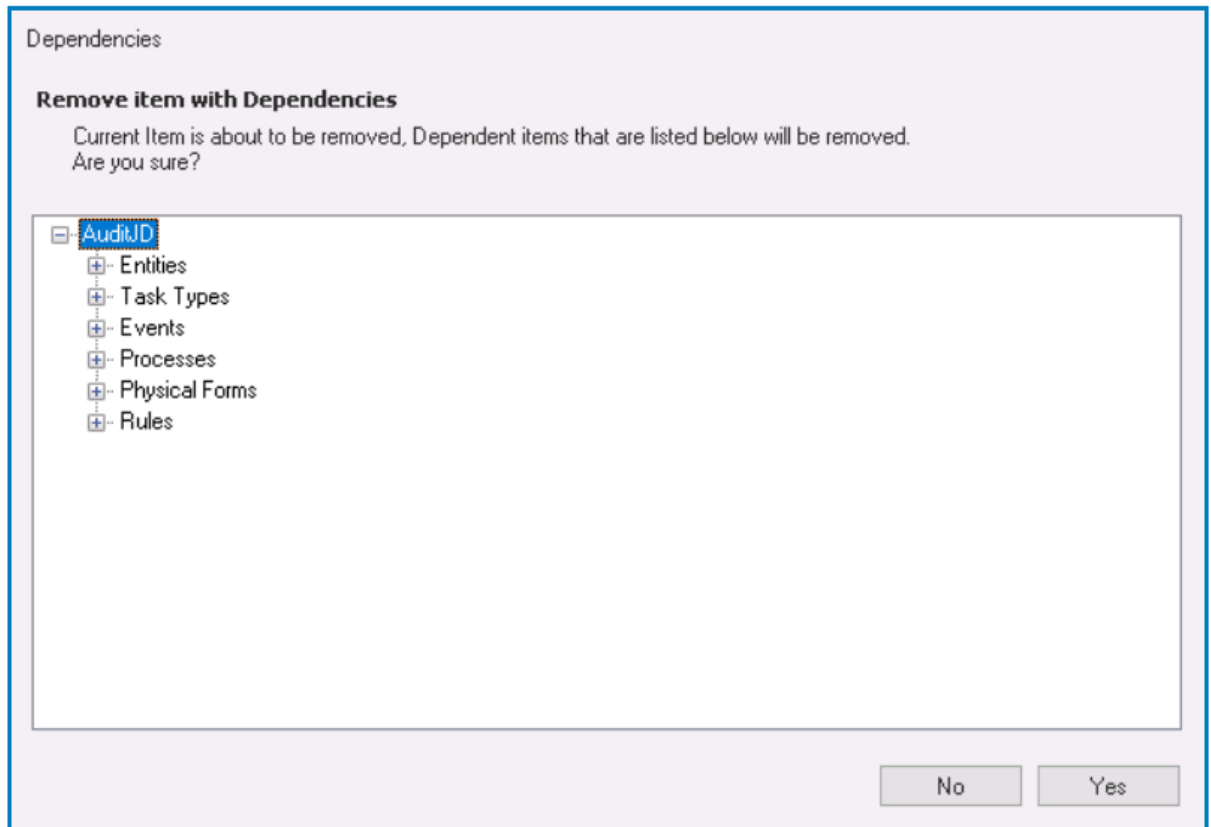
1. **Add:** This button allows users to add a new application.
  - If selected, it will display the Application window. In this window, users can set the name and title of the application.



- **Name:** This is a text field that allows users to input the name of the application.
  - This is the name that the application will be registered under in the database. This name will also be used later on as part of the established naming convention.
  - It is recommended that the name of an application be kept short and simple. It **cannot** contain spaces or special characters.
- **Title:** This is a text field that allows users to input the title of the application.
  - This is the title that the application will be registered under in the [Application Tree](#) of the Designer.
  - Unlike the name, the title of an application can be longer and more descriptive. It can include spaces and special characters.
- **Version:** This is a numeric updown field that allows users to input the version number of the application.
  - Users can enter this number either manually or by using the increment/decrement arrows.
    - To enable this field, users must first create and save the application. Then, they must right-click on the application from the [Application Tree](#) and select **Change App** from the context menu.
- **Create or Cancel:** To create and save the application, click on the **Create** button. To cancel this process, click on the **Cancel** button.
  - Once an application has been created, the results are recorded in the [Output](#) section of the Designer.

- ✿ To avoid database corruption, if multiple users are working on the same server and using the same naming convention, it is recommended that you include your initials as part of the application name and title.

2. **Remove:** This button allows users to remove an application.
  - To begin the removal process, select the application from the [Application Tree](#) and then click on the **Remove** button. It will display the [Dependencies](#) window. In this window, users can access an overview of the selected application, including its dependent objects.



- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
    - Once an application has been removed, the results are recorded in the [Output](#) section of the Designer.
3. **Export:** This button allows users to export an application.
    - If selected, it will display the [Export](#) window. Depending on the settings configured in this window, a backup file will be created for the selected application in the .rba format.

**Export**

**Settings**

1 Application: Audit (JD)

2 Export to file: [Red field] ...

3 Password: [Empty] Confirm password: [Empty] 4

5 ☐ Exclude sensitive data 6 ☐ Auto increment app version after export

**Entities**

7

8 ☐ Export custom data

|   | Export                   | Name                        | Columns |
|---|--------------------------|-----------------------------|---------|
| ▶ | <input type="checkbox"/> | Audit (JD) - Audit Plan     | [Empty] |
|   | <input type="checkbox"/> | Audit (JD) - New Audit Test | [Empty] |
|   | <input type="checkbox"/> | Audit (JD) - New Audit      | [Empty] |

**Progress**

9

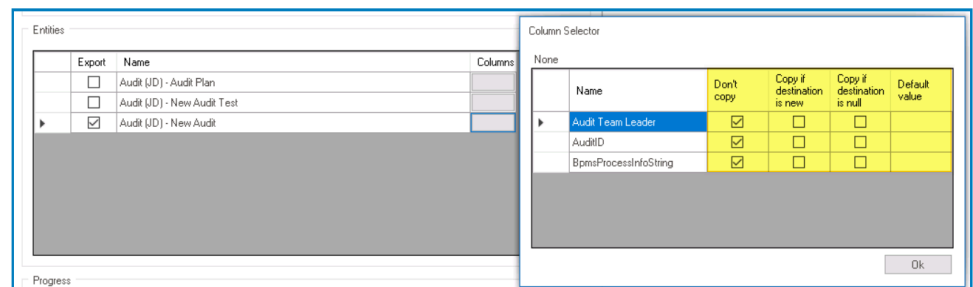
10 ☒ Proceed ☒ Cancel

- **Application:** This is a single-select dropdown field that retrieves the list of applications (from the [Application Tree](#)). It allows users to select the application that they would like to export.
  - Users can also determine the application in advance by selecting it from the [Application Tree](#) before clicking on the **Export** button.
- **Export to File:** This field allows users to specify the physical path/folder to which the file will be exported. It is a **mandatory** field.
  - To select this location, click on the  button. Users will also need to specify the name under which the file will be saved.
- **Password:** This fields allow users to set the password to protect the exported application file.
  - Once a password has been set, the output file will be locked and thenceforth can only be accessed using the password.
    - To import the application, users will be required to input the password.
- **Confirm Password:** This field requires users to re-enter the password they set to confirm it and ensure it was typed correctly.
- **Exclude Sensitive Data:** This is a checkbox which, if selected, allows users to exclude sensitive data such as connection strings, passwords, and other confidential information from the exported application file.
- **Auto Increment App Version After Export:** This is a checkbox which, if selected,




allows users to automatically increment the application version number after export.

- After the export is complete, the version number of the application will be automatically incremented by one.
- **Entities:** In this section, users can access the list of the entities belonging to the application.
  - To select an entity for export, enable the checkbox adjacent to it.
    - Once the user has determined that they would like to proceed with the export, all selected entities/field data will be transferred to the specified database.
  - Selecting an entity automatically selects all attributes of that entity. To alter this selection, use the **Columns** field adjacent to the entity.
    - If the **Columns** field is selected, it will display the Column Selector window.




- In this window, users can specify their preferences as to which (and in what situation) attributes are copied
  - **Export Custom Data:** This is a checkbox which, if selected, allows users to export custom data associated with the application.
    - All entities belonging to the application will, in turn, be selected for export.
  - **Progress:** This bar is used to visualize the progress of the data transfer.
  - **Proceed or Cancel:** To continue with the export, click on the **Proceed** button. To cancel, click on the **Cancel** button.
    - To enable the **Proceed** button, users must first set up the file in its intended folder.
4. **Import:** This button allows users to import an application into the Designer.
- If selected, it will display the Import window.

- **Exported File:** This field allows users to select the file/application that they would like to import. It is a mandatory field.
  - To select the .rba file, click on the  button.
- **Get Backup Before Import:** This is a checkbox which, if selected, allows users to backup existing data before the import.

 It is advisable to backup existing data before importing data.

- **Import Custom Data:** This is a checkbox which, if selected, allows users to import custom data associated with the application.
- **Import Timed Event Settings:** This checkbox determines whether to include timed event settings during the import process.
  - If this checkbox is selected, importing an RBA file and deploying will **update** the server's timed event settings with the values stored in the RBA file.
  - If this checkbox is left unselected, importing an RBA file and deploying will **preserve** the server's timed event settings.

 It's important to note that when importing an RBA file, any timed event settings that do not already exist on the server will automatically take the values specified in the RBA file, regardless of whether this checkbox was selected or not.

- **Override Customized Responsibilities:** This checkbox determines whether to override customized responsibilities (role and resource assignees) during the import process.
  - If this checkbox is selected, all **existing** and **new** task receiver settings on the server will be taken from the RBA file.
    - Task assignees for tasks or task types that **already exist** on the server will be overwritten with the values in the RBA file.

- Task assignees for tasks or task types that **do not already exist** on the server will be imported from the RBA file.
- If this checkbox is left unselected, all **existing** task receiver settings on the server will be preserved. All **new** task receiver settings, however, will take their values from the RBA file.
  - Task assignees for tasks or task types that **already exist** on the server are preserved.
  - Task assignees for tasks or task types that **do not already exist** on the server (i.e., are being imported into the server for the first time) will take the task receiver values from the RBA file.



When importing an RBA file and selecting this checkbox, it's important to note that only the defined and dynamic receiver settings will be overridden. The specific receivers assigned to individual tasks before the import will remain unchanged.

- **Progress:** This bar is used to visualize the progress of the data transfer.
- **Import or Cancel:** To proceed with the import, click on the **Import** button. To cancel, click on the **Cancel** button.
  - To enable the **Import** button, users must first select the intended file/application.

## 8.2.1.2. Current Application

In the **Current Application** group of the Home tab, users are provided access to the following set of commands:

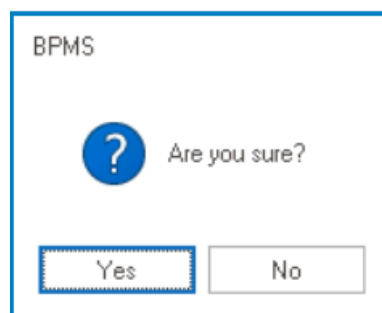


\* The label of the **Current Application** group in the menu bar dynamically updates to display the name of the currently selected application in the [Application Tree](#). This provides users with a visual cue to ensure that the commands they execute are applied to the intended application.

1. **Build:** This button allows users to build the selected application.
  - When an application is built, its components are compiled and sent to the web server.
2. **Deploy:** This button allows users to deploy the selected application.
  - When an application is deployed, its components are built and executed in the web server.
    - Any changes made to the application are published in the web as well.

\* Deployment, in this context, refers to the process of creating application-specific files and copying them into the program's web directory path.

3. **Deploy Forms:** This button allows users to deploy forms that have been deployed previously but have since had their layout modified.
4. **Un-Deploy:** This button allows users to undo the deployment of the selected application, removing it from the web.
  - If this button is selected, users will be alerted of the action with the following confirmation message:



- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
5. **Deploy Without Build:** This button allows users to deploy the selected application without building it.

- When an application is deployed without build, its changes are updated only in the web server.



Users must deploy an application in order to be able to access it in the web server.



When any of the commands in this group are selected, the results of the operation are recorded in the Output section of the Designer.

## 8.2.1.3. Deploy

In the **Deploy** group of the Home tab, users are provided access to the following command:



1. **Deploy All:** This button allows users to deploy all applications listed in the [Application Tree](#).
  - The results of the deployment are recorded in the [Output](#) section of the Designer.

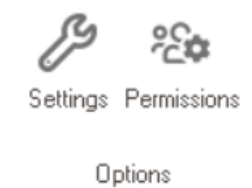
! Users must deploy an application in order to be able to access it in the web server.

\* Deployment, in this context, refers to the process of creating application-specific files and copying them into the program's web directory path.

## 8.2.1.4. Options

---

In the **Options** group of the Home tab, users are provided access to the following commands:



1. [Settings](#)
2. [Permissions](#)

## 8.2.1.4.1. Settings

---

Selecting the **Settings** button will display the Settings window.

There are nine tabs to consider in this window:

1. [General](#)
2. [System](#)
3. [Customization](#)
4. [Security](#)
5. [E-Mail/SMS](#)
6. [File Storage](#)
7. [Integration](#)
8. [Scheduler](#)
9. [Logs](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Settings window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.



## 8.2.1.4.1.1. General Tab

In the General tab, users can configure the general settings.

Settings

General System Web UI Customization Security E-Mail/SMS File Storage Integration Scheduler Logs

1 Web Directory

2 EPC url

3 ☐ Use in Multi-user environment

4 Log Details

5 Languages

| Title   | Culture | Default                             | Enabled                             | RTL                                 | Date format | Long date format | Time format |
|---------|---------|-------------------------------------|-------------------------------------|-------------------------------------|-------------|------------------|-------------|
| English | en-GB   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |             |                  |             |
| French  | fr-CA   | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |             |                  |             |
| Arabic  | ar-SA   | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |                  |             |
| German  | de-DE   | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |             |                  |             |
| Hebrew  | he-IL   | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |                  |             |

Record 1 of 6

6 ☐ Load dictionaries for spellchecker

Apply Ok Cancel

- Web Directory:** This is a text field that allows users to specify the web address of the BPM application.
  - To view the web application, the web directory must be entered correctly.
- EPC URL:** This is a text field that allows users to specify the URL of the EPC server that will be linked to the application.
- Use in Multi-User Environment:** This is a checkbox which, if selected, allows users to enable application use in a multi-user environment—that is, an environment where multiple users can work on the same application, but not the same forms.
  - Selecting this checkbox will display the **Multi-User Settings** section.

☒ Use in Multi-user environment

Multi-user settings

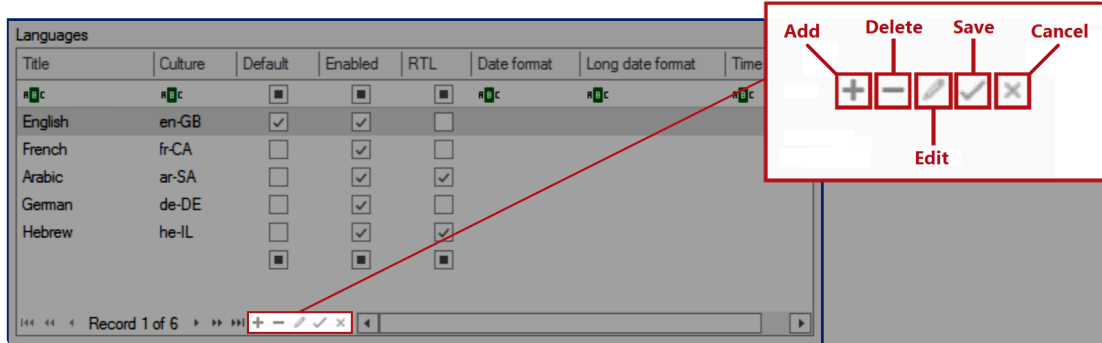
BPM Applications Directory  ...

The specified folders might not be usable for other databases. ☐ Rebuild an application for the first use

- BPM Applications Directory:** This field allows users to specify the shared physical path/folder of the BPM applications (in the .rba format).
  - To select this location, click on the  button.
- Rebuild an Application For the First Use:** This is a checkbox which, if selected, allows users to automatically rebuild an application for the first use.
  - When an application is imported for the first time, all its forms will be

automatically rebuilt.

4. **Log Details:** This is a single-select dropdown field that displays the list of log detail options. It allows users to select the method by which details are logged in the system.
  - Log Details: Critical Only, Normal, or Verbose.
5. **Languages:** In this section, users can configure the language (and date) settings.



- **Title:** This is a text field that allows users to input the title of the language.
- **Culture:** This is a text field that allows users to input the cultural variant of the language.
- **Default:** This option allows users to set a language as the default.

! Only one language can be set as the default at a time.

- **Enabled:** This is a checkbox which, if selected, allows users to enable a language for use in the web application.

! Multiple languages can be enabled at a time.

- **RTL:** This is a checkbox which, if selected, allows users to mark out the languages that use right-to-left script.
- **Date Format:** This is a text field that allows users to specify the date format for the language.
- **Long Date Format:** This is a text field that allows users to specify the long date format for the language.
- **Time Format:** This is a text field that allows users to specify the time format for the language.

\* Please note that for the date format, the month is represented in uppercase as "MM," while the day and year are in lowercase ("dd" and "yyyy" respectively). A lowercase "m" signifies minutes in the time format.

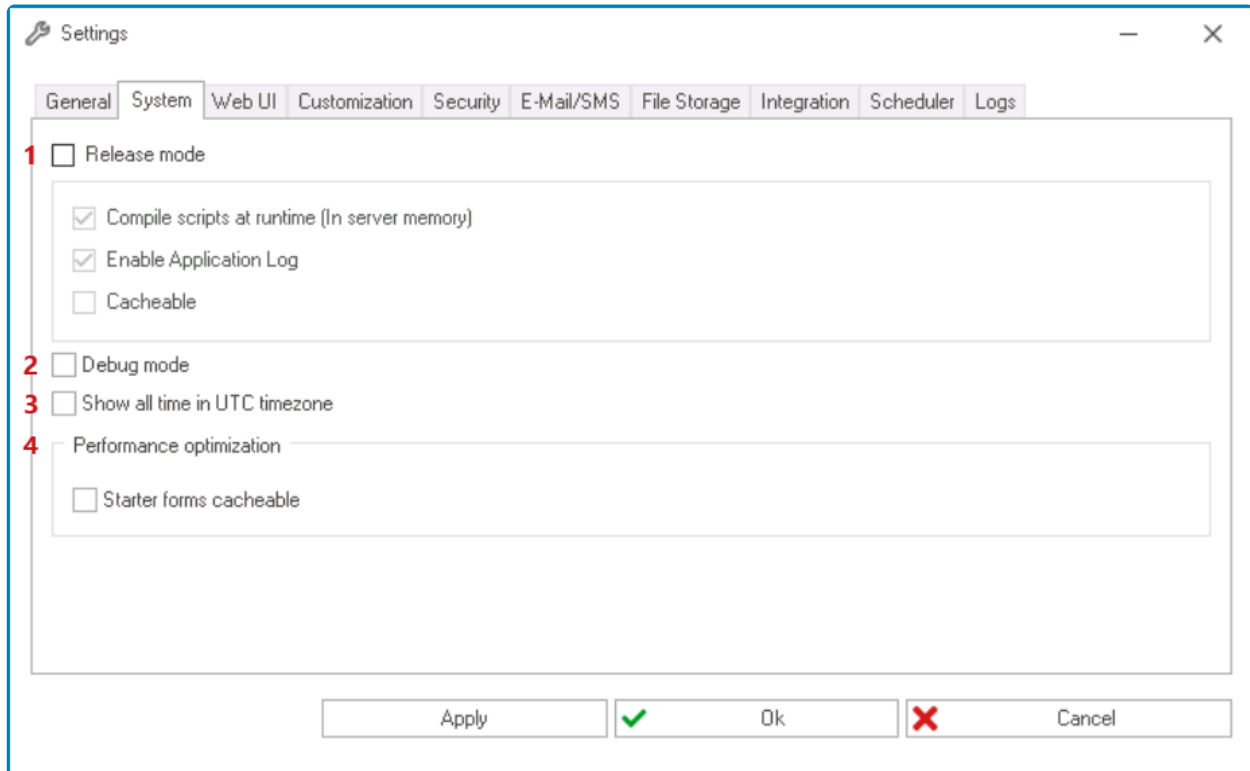
! Currently, only five languages are supported on the web server: English, French, German, Arabic (right-to-left), and Hebrew (right-to-left). In the General tab, users can customize which of these languages are made available in the web UI.

6. **Load Dictionaries for Spellchecker:** This is a checkbox which, if selected, allows users to load

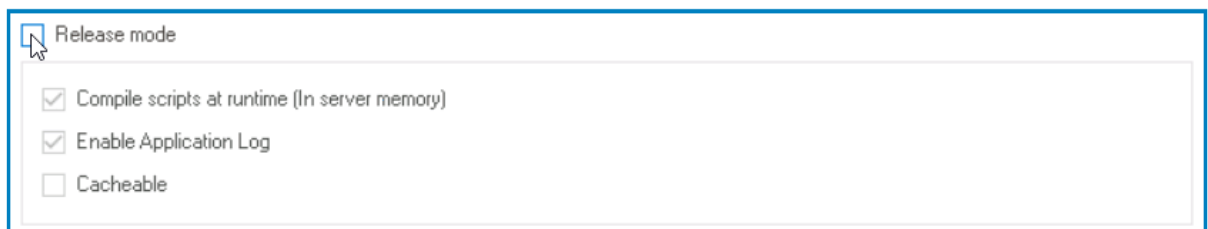
dictionaries for the spellchecker function.

## 8.2.1.4.1.2. System Tab

In the System tab, users can configure details relating to the running of the system.



1. **Release Mode:** This is a checkbox which, if selected, allows for the program to run in release mode. In this mode, the program is fully optimized for performance—stripped of symbolic debug information.
  - Along with final release distribution, this mode is typically used in the final stages of testing and production readiness.
    - When operating in release mode, it is no longer necessary to deploy the application every time modifications are made.
  - Depending on whether this checkbox is selected, the options below it may be selected or unselected.



1. **Compile Scripts at Runtime:** This is a checkbox which, if selected, allows users to compile scripts at runtime. This means that changing the script and saving it does not require for the application to be deployed.
2. **Enable Application Log:** This is a checkbox which, if selected, allows for relevant activities performed in the system to be recorded in the Application Log.
  - For more details on the Application Log of the DBP application, [click here](#).
3. **Cacheable:** This is a checkbox which, if selected, activates data retrieval from the server memory.

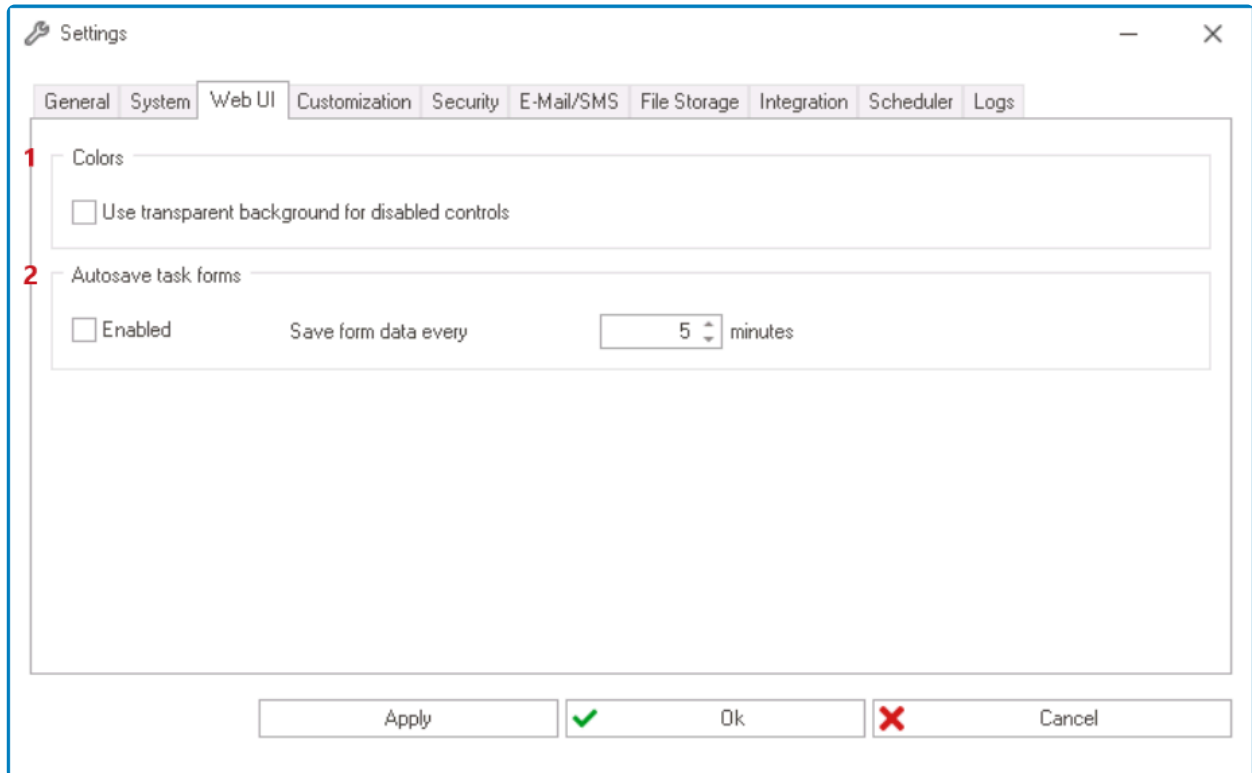


As demonstrated above, these three options remain disabled at all times. To select or unselect them, users will need to manipulate the **Release Mode** checkbox.

2. **Debug Mode:** This is a checkbox which, if selected, allows for the program to run in debug mode. In this mode, the program is not optimized, compiling with full symbolic debug information.
  - Typically, this mode is used in the initial stages of development and testing.
3. **Show All Time in UTC Time Zone:** This is a checkbox which, if selected, converts all time to the Coordinated Universal Time (UTC) time zone.
4. **Performance Optimization:** In this section, users can adjust details relating to performance optimization.
  - **Starter Forms Cacheable:** This is a checkbox which, if selected, activates data retrieval for starter forms.

## 8.2.1.4.1.3. Web UI

In the Web UI tab, users can customize various aspects of the **DBP** application's user interface.



1. **Colors:** In this section, users can modify the color scheme of the interface to suit their preferences.
  - **Use Transparent Background For Disabled Controls:** This is a checkbox which, if selected, allows disabled controls to have a transparent background in the **DBP** application—as opposed to a darker background color.
    - For more details on this feature, [click here](#).
2. **Auto-Save Task Forms:** In this section, users can configure the web application to automatically save task forms after a set interval.
  - **Enabled:** This is a checkbox which, if selected, allows users to to enable the auto-save feature.
  - **Save Form Data:** This is a numeric updown field that allows users to set the interval at which the application automatically saves task forms. Users can enter this number either manually or by using the increment/decrement arrows.

## 8.2.1.4.1.4. Customization Tab

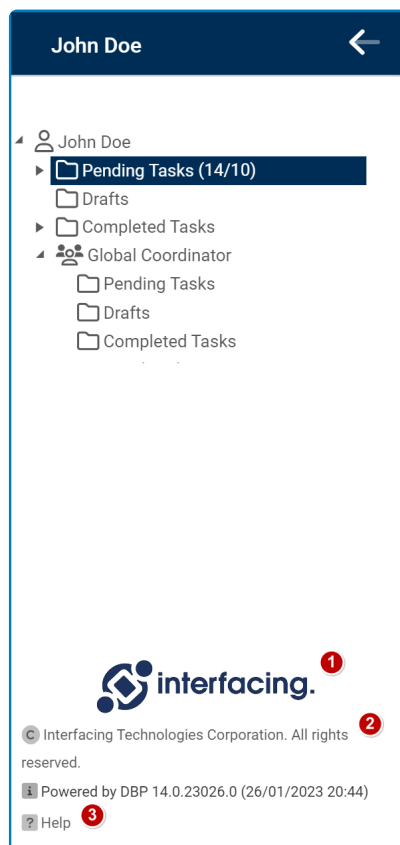
In the Customization tab, users can customize how their company is presented in the **DBP** web application. Along with their company logo and website, users can set up the copyright notice for the application as well as link to its user guide.

The screenshot shows the 'Settings' window with the 'Customization' tab selected. The 'Company Logo' section (labeled 1) includes a text input field with the 'interfacing.' logo, a 'Delete' button, and a button with three dots to select a new logo. The 'Company title' section (labeled 2) includes three text input fields: 'Company Website' with 'https://www.interfacing.com/', 'Copyright text' with 'Interfacing Technologies Corporation. All rights reserved.', and 'Help url' with 'https://www.manula.com/manuals/interfacing-technologies/bpms-user-manual/2/en/topic/1-0-the'. The bottom of the window has 'Apply', 'Ok', and 'Cancel' buttons.

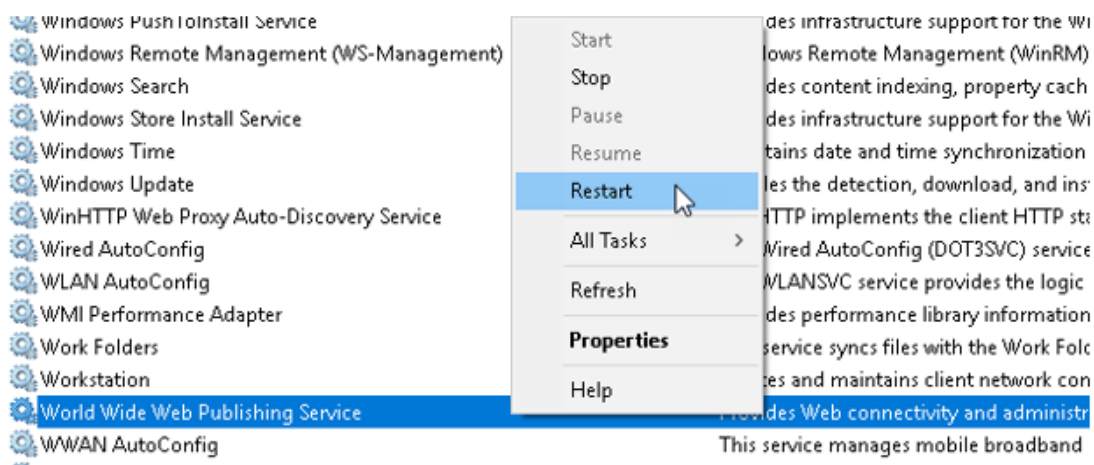
1. **Company Logo:** In this section, users can set up their company logo.
  - To select the logo, click on the button.
    - Once selected, the logo can be removed using the **Delete** button.
  - In the **DBP** application, this logo will be displayed in the lower half of the [Task Navigation Pane](#).
2. **Company Title:** In this section, users can set up the title of the company website. They can also set up other details pertaining to the web application.
  - **Company Website:** This is a text field that allows users to input the website of the company.
    - When users click on the company logo in the **DBP** application, they will be directed to the website indicated in this field.
  - **Copyright Text:** This is a text field that allows users to input the copyright notice for the web application.
    - In the **DBP** application, this notice will be displayed in the lower half of the Task Navigation Pane.
  - **Help URL:** This is a text field that allows users to input the URL for the application's user guide.

- When users click on the **Help** button in the **DBP** application, they will be directed to the web address indicated in this field.

✿ In the **DBP** application, users can access the company logo, copyright notice, and user guide in the lower half of the Task Navigation Pane.



✿ To update the web application, users will need to restart the “World Wide Web Publishing Service” in Windows Services. This is done to clear the cache.





## 8.2.1.4.1.5. Security Tab

In the Security tab, users can configure the security settings.

The screenshot shows a 'Settings' window with a tabbed interface. The 'Security' tab is selected. The tabs are: General, System, Web UI, Customization, Security, E-Mail/SMS, File Storage, Integration, Scheduler, and Logs. The Security tab contains the following settings:

- 1 ☐ Use Single Sign-On
- 2 Identity Provider URL
- 3 WebApp URL (optional)
- 4 Entity ID (optional)
- 5 User login claim name (optional)

At the bottom of the window are four buttons: 'Apply', 'Ok' (with a green checkmark icon), 'Cancel' (with a red X icon), and 'Cancel'.

1. **Use Single Sign-On:** This is a checkbox which, if selected, allows users to enable the use of single sign-on (SSO).
  - SSO allows users to authenticate with multiple applications using a single set of credentials.
2. **Identity Provider URL:** This is a text field that allows users to input the URL of the identity provider.
  - An identity provider is a system used to create, store, and manage users' digital identities. It provides user authentication as a service.
3. **WebApp URL:** This is a text field that allows users to input the URL of the web application. This is an optional field.
4. **Entity ID:** This is a text field that allows users to input the entity ID. This is an optional field.
5. **User Login Claim Name:** This is a text field that allows users to input the user login claim name. This is an optional field.

## 8.2.1.4.1.6. E-Mail/SMS Tab

In the E-Mail/SMS tab, users can configure the e-mail and SMS settings.

**Settings**

General System Web UI Customization Security **E-Mail/SMS** File Storage Integration Scheduler Logs

**1 Mail Server**

Host server: 10.10.100.7

Port number: 25 ☐ TLS Encrypted ☐ Enable SSL

☐ Requires Authentication

User Name: Password: xxx

Mails from: rnd-noreply@interfacing.com

Web Site Url: https://dbp-qa-3.interfacing.com/dbp

**2 AWS settings for SMS**

Region: CA Central1

Access Key: AKIA5RUXGZ62J20NMTVG

Secret Key: G+rYSX5em9i3x1aREnzqebPPr1TGy/9IP3AF1KDY

Apply ☒ Ok ☒ Cancel

1. **Mail Server:** In this section, users can set up the mail server.

- **Host Server:** This is a text field that allows users to input the IP address of the host server.
- **Port Number:** This is a numeric updown field that allows users to input the port number. Users can enter this number either manually or by clicking the increment/decrement arrows.
- **TLS Encrypted:** This is a checkbox which, if selected, allows users to enable Transport Layer Security (TLS) encryption.
- **Enable SSL:** This is a checkbox which, if selected, allows users to enable Secure Sockets Layer (SSL) encryption.
- **Requires Authentication:** This is a checkbox which, if selected, allows users to mark out authentication as a requirement.
- **User Name:** This is a text field that allows users to input the username.
- **Password:** This is a text field that allows users to input the password.
- **Mails From:** This is a text field that allows users to input the address from which the emails will be sent.
- **Website URL:** This is a text field that allows users to input the URL of the website to which the mail server will be connected.
  - Users should specifically input the web address of the server hosting the **DBP** application.



TLS and SSL are both cryptographic protocols used to secure data transmission on the

Internet. They encrypt and protect communications between servers, applications, and users. TLS, however, is the successor protocol to SSL.

2. **AWS Settings for SMS:** In this section, users can configure the AWS settings for SMS.
  - **Region:** This is a single-select dropdown field that displays the list of AWS regions.
  - **Access Key:** This is a text field that allows users to input the access key.
  - **Secret Key:** This is a text field that allows users to input the secret key.

## 8.2.1.4.1.7. File Storage Tab

In the File Storage tab, users can configure the file storage settings.

1. **File Storage Type:** This is a single-select dropdown field that displays the list of file storage types. Depending on the type selected from this dropdown, different fields will be enabled/disabled in the tab.
  - File Storage Types: DB, Local Folder, Remote Folder, or AWS S3.
2. **Local Folder:** This is a text field that allows users to input the local folder.
3. **Remote Folder:** This is a text field that allows users to input the remote folder.
4. **AWS Settings:** In this section, users can configure the AWS settings.
  - **Region:** This is a single-select dropdown field that displays the list of AWS regions.
  - **Access Key:** This is a text field that allows users to input the access key.
  - **Secret Key:** This is a text field that allows users to input the secret key.
  - **Bucket Name:** This is a text field that allows users to input the bucket name.
  - **Folder in the Bucket:** This is a text field that allows users to specify the folder in the bucket.
5. **Custom File Storage:** In this section, users can configure a custom file storage system.
  - **Storage Type:** This is a single-select dropdown field that displays the list of storage types. Depending on the type selected from this dropdown, different fields will be enabled/disabled in the tab.
    - Storage Types: Local Folder, Remote Folder, or AWS S3.
  - **Local Folder:** This is a text field that allows users to input the local folder.

- ***Remote Folder:*** This is a text field that allows users to input the remote folder.
- ***AWS Folder:*** This is a text field that allows users to input the AWS folder.

## 8.2.1.4.1.8. Integration Tab

In the Integration tab, users can configure third-party integrations.

The screenshot shows the 'Settings' window with the 'Integration' tab selected. The window has a title bar with a key icon and the text 'Settings'. Below the title bar are tabs for 'General', 'System', 'Web UI', 'Customization', 'Security', 'E-Mail/SMS', 'File Storage', 'Integration', 'Scheduler', and 'Logs'. The 'Integration' tab is active and contains two sections:

- 1 Exchange Web Services:** This section contains three text input fields: 'Server URL', 'Proxy User name', and 'Password'.
- 2 PDF E-Signature:** This section contains a checkbox labeled 'Enabled' and a text input field labeled 'Service URL'.

At the bottom of the window are three buttons: 'Apply', 'Ok' (with a green checkmark icon), and 'Cancel' (with a red X icon).

1. **Exchange Web Services:** In this section, users can configure the Exchange Web Services (EWS) settings.
  - **Server URL:** This is a text field that allows users to input the URL of the Exchange server.
  - **Proxy User Name:** This is a text field that allows users to input the proxy username.
  - **Password:** This is a text field that allows users to input the password.
2. **PDF E-Signature:** In this section, users can configure the PDF E-Signature settings.
  - **Enabled:** This is a checkbox which, if selected, allows users to enable digital signatures for PDF files.
  - **Service URL:** This is a text field that allows users to input the URL of the EPC server.

## 8.2.1.4.1.9. Scheduler Tab

In the Scheduler tab, users can configure the scheduler settings.

The screenshot shows a 'Settings' dialog box with a tabbed interface. The 'Scheduler' tab is selected. It contains three labeled text input fields: 'Server' (containing 'localhost'), 'Service Name' (containing 'RayBPMSScheduler'), and 'Service Port' (containing '555'). The fields are numbered 1, 2, and 3 respectively. At the bottom, there are four buttons: 'Apply', 'Ok' (with a green checkmark icon), 'Cancel' (with a red X icon), and another 'Cancel' button.

1. **Server:** This is a text field that allows users to input the server name.
2. **Service Name:** This is a text field that allows users to input the service name.
3. **Service Port:** This is a text field that allows users to input the service port number.

## 8.2.1.4.1.10. Logs Tab

In the Logs tab, users can configure the log settings.

The screenshot shows the 'Settings' window with the 'Logs' tab selected. The 'Application log' section displays 'Total record: 844,413' and 'Purge records older than days' set to 30, with a 'Purge now' button. The 'Error log' section displays 'Total record: 5,397' and 'Purge records older than days' set to 30, with a 'Purge now' button. The bottom of the window has buttons for 'Apply', 'Ok', and 'Cancel'.

1. **Application Log:** In this section, users can set the time limit for the automatic purging of records in the [Application Log](#). They can also manually purge records of application events.
  - **Purge Records Older Than Days:** This is a numeric updown field that allows users to input the number of days after which records will automatically be purged. Users can enter this number either manually or by clicking the increment/decrement arrows.
  - **Purge Now:** This button allows users to manually trigger an immediate purge of the applicable records.
    - To alter the number of records included in this purge, users will need to manipulate the numeric updown field.
2. **Error Log:** In this section, users can set the time limit for the automatic purging of records in the [Error Log](#). They can also manually purge records of error occurrences.
  - **Purge Records Older Than Days:** This is a numeric updown field that allows users to input the number of days after which records will automatically be purged. Users can enter this number either manually or by clicking the increment/decrement arrows.
  - **Purge Now:** This button allows users to manually trigger an immediate purge of the applicable records.
    - To alter the number of records included in this purge, users will need to manipulate the numeric updown field.

✿ In this tab, along with automatically and/or manually purging records, users can access relevant statistics for both the Application and Error Log. Specifically, users can view the total number of records as well as the number of old and new records.




## 8.2.1.4.2. Permissions

Selecting the **Permissions** button will display the Permission Setting Form window. This window is used to define permissions to the Designer's features.

\* When a receiver is selected, the **User Info** label will adjust to display relevant information about them. For example, it may state “This user does not have access to the Designer,” or “This user is admin and has all the permissions.”

To add a user with specific access rights, take the following steps:

1. Select the application from the **Applications** field.
2. Next, select the  button below the **Principals** field. It will open the Receivers window.
  - By default, this window displays the list of users in the system.

Receivers

Select receiver type: **User**

Drag a column header here to group by that column


| Code                | Title                                |
|---------------------|--------------------------------------|
| e9a7a2ac-efb6-48... | EYBABU TWVWHQJJUZKX (qms_user_08931) |
| 3fdb7ea8-8c89-44... | SPAUJM DJZDZGZKOFHQ (qms_user_00052) |
| 30470cca-c5f8-4f... | MKFTGU WMQGMKLPMTFA (qms_user_04497) |
| 1599d338-643a-4...  | TCOYMT CMQKNVFMQJTH (qms_user_05592) |
| f9286e1d-427d-40... | THMCVK SJHIMQGZEHYV (qms_user_09621) |
| 3e8ea492-b8dc-4...  | JIMUIE OPLLCVMKJPEU (qms_user_09653) |
| b079a139-b6bb-4...  | OTZUYX ENBNTAISZGXC (qms_user_06983) |
| f6d76e71-14d5-4b... | KAPINN SILKYDMHEVIY (qms_user_05553) |
| 47bd69f6-9e8b-49... | VAQVUW HTOKHDMQFZNZ (qms_user_06310) |
| 0e322001-9696-4...  | MTHMHD TDLQFVRNOLVS (qms_user_03891) |
| 9ca12f70-e266-4c... | WYFODD LDUVXMCShRET (qms_user_08350) |
| 0d2be284-707b-4...  | MAVTKT RTFRHAXFTDUT (qms_user_04239) |
| c9e65b9b-aad5-4...  | ADZYUQ RIWxEVULMJPP (qms_user_03377) |
| ebd32f46-b6fc-44... | PTXSIO HMBVBIUYAXJO (qms_user_03935) |
| 33a629f0-0ac1-43... | WBHQJL VAMTCKGBVCYM (qms_user_03566) |
| 83d2c5dc-ecb7-4...  | User 767 Perf (PerfUser767)          |
| 7a7ad17d-1561-4...  | XZPAFQ DDSUDEOPSQCV (qms_user_06156) |

Record 1 of 11695

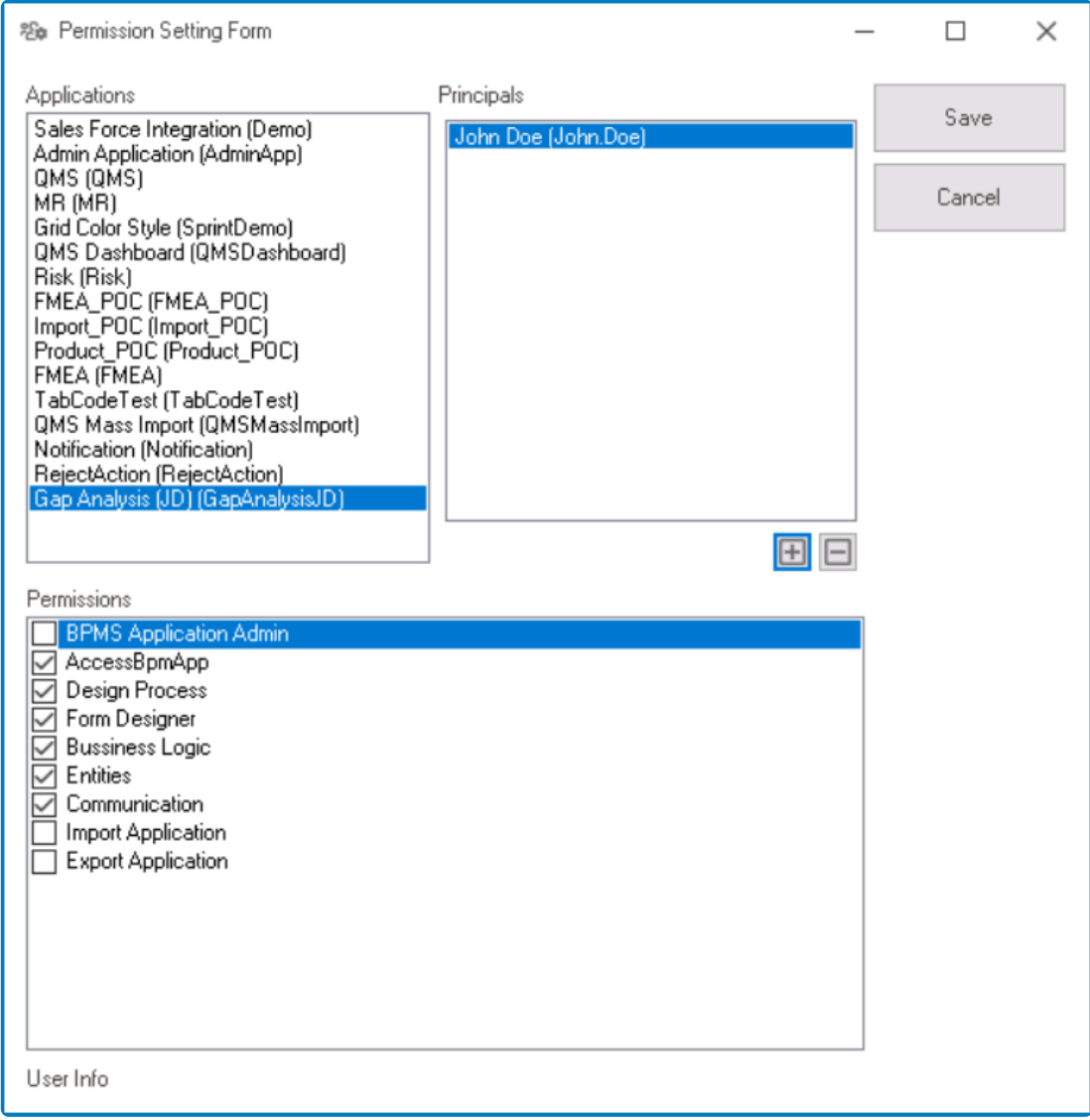
Select Cancel

\* In the Receivers window, you can filter the list by receiver type—that is, by user, role, or group. To instead add a role or group with specific access rights, first select the corresponding receiver type before applying the steps below.

- Choose the user from the list and click on the **Select** button. (You can also just double-click on the entry from the list.) You will then be redirected to the Permission Setting Form, where you will find the selected user listed in the **Principals** field.

\* To remove a user/role/group from the **Principals** field, select the receiver and click on the  button.

- In the Permission Setting Form, you will then be able to determine the user's access rights by selecting/deselecting permissions in the **Permissions** field.
  - To save your changes, click on the **Save** button. To cancel, click on the **Cancel** button.



The screenshot shows a 'Permission Setting Form' window. It has three main sections: 'Applications', 'Principals', and 'Permissions'. The 'Applications' list on the left includes items like 'Sales Force Integration (Demo)', 'Admin Application (AdminApp)', 'QMS (QMS)', 'MR (MR)', 'Grid Color Style (SprintDemo)', 'QMS Dashboard (QMSDashboard)', 'Risk (Risk)', 'FMEA\_POC (FMEA\_POC)', 'Import\_POC (Import\_POC)', 'Product\_POC (Product\_POC)', 'FMEA (FMEA)', 'TabCodeTest (TabCodeTest)', 'QMS Mass Import (QMSMassImport)', 'Notification (Notification)', 'RejectAction (RejectAction)', and 'Gap Analysis (JD) (GapAnalysisJD)'. The 'Principals' list on the right contains 'John Doe (John.Doe)'. The 'Permissions' section at the bottom has a list of checkboxes: 'BPMS Application Admin' (unchecked), 'AccessBpmApp' (checked), 'Design Process' (checked), 'Form Designer' (checked), 'Business Logic' (checked), 'Entities' (checked), 'Communication' (checked), 'Import Application' (unchecked), and 'Export Application' (unchecked). A 'User Info' label is at the bottom left. 'Save' and 'Cancel' buttons are on the right.

Permission Setting Form

Applications

- Sales Force Integration (Demo)
- Admin Application (AdminApp)
- QMS (QMS)
- MR (MR)
- Grid Color Style (SprintDemo)
- QMS Dashboard (QMSDashboard)
- Risk (Risk)
- FMEA\_POC (FMEA\_POC)
- Import\_POC (Import\_POC)
- Product\_POC (Product\_POC)
- FMEA (FMEA)
- TabCodeTest (TabCodeTest)
- QMS Mass Import (QMSMassImport)
- Notification (Notification)
- RejectAction (RejectAction)
- Gap Analysis (JD) (GapAnalysisJD)

Principals

- John Doe (John.Doe)

Permissions

- ☐ BPMS Application Admin
- ☒ AccessBpmApp
- ☒ Design Process
- ☒ Form Designer
- ☒ Business Logic
- ☒ Entities
- ☒ Communication
- ☐ Import Application
- ☐ Export Application

User Info

Save

Cancel

! If the selected receiver either does not have access to the Designer or is an admin and already has all permissions enabled (as indicated by the **User Info** label), you will not be able to set or modify their access rights.

## 8.2.1.5. Opened Forms

---

To access a list of all tabs currently open in the Designer, users can click on the **Opened Forms** dropdown.

Opened forms ▾

✿ Users can navigate between these different tabs by selecting them from the dropdown.

## 8.2.1.6. Help

---

To access more information about the Designer's features, users can click on the [Help](#) button.



It will direct them to the latest published version of the [DBP Designer User Manual](#).

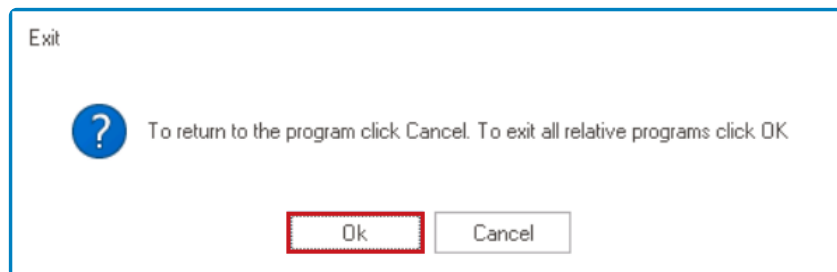
## 8.2.1.7. Close Application

To close the application, users can click on the **Close Application** button.

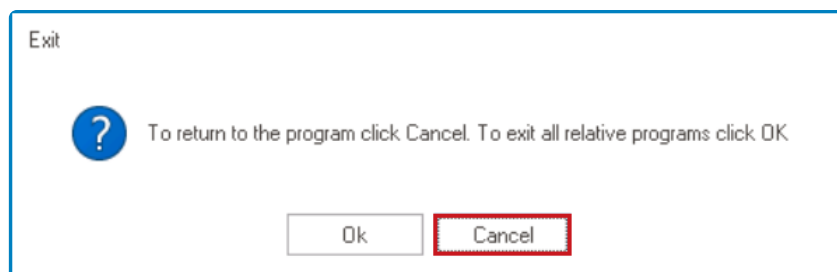


It will open a pop-up window in which users are given the option of either proceeding with or canceling the closing process.

- To exit the application, click on **OK** button.



- To return to the application, click on the **Cancel** button.

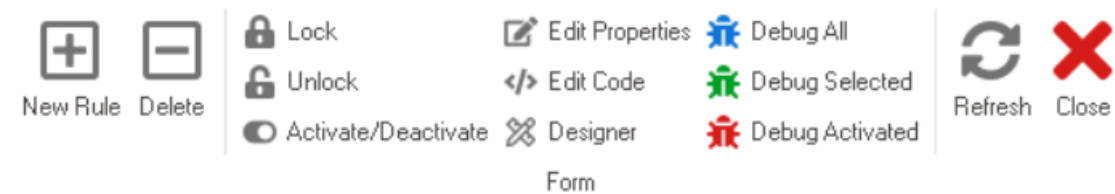


## 8.2.1.8. Additional Options

---

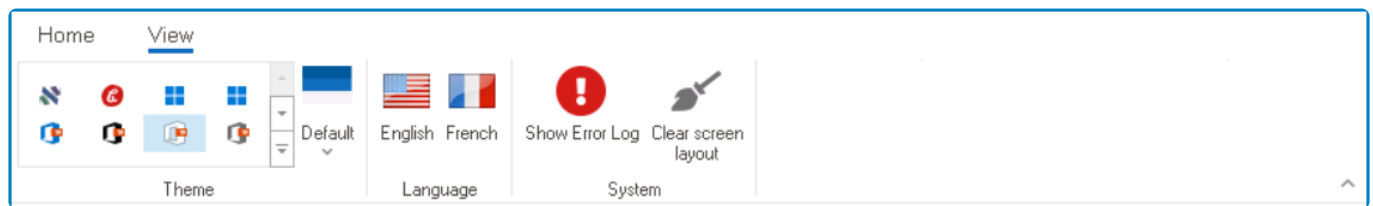
Depending on which module or element of the [Application Tree](#) is in use, the Home tab may expand to include additional options.

- For example, when the **Rules** element is in use, the following options are made visible in the menu bar:



## 8.2.2. View Tab

Like the Home tab, the View tab is divided into groups, each of which contains a set of related commands.



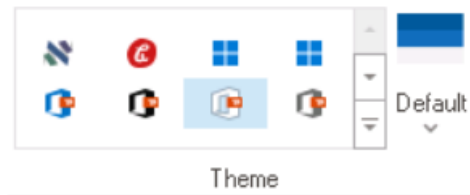
These groups include:


1. [Theme](#)
2. [Language](#)
3. [System](#)

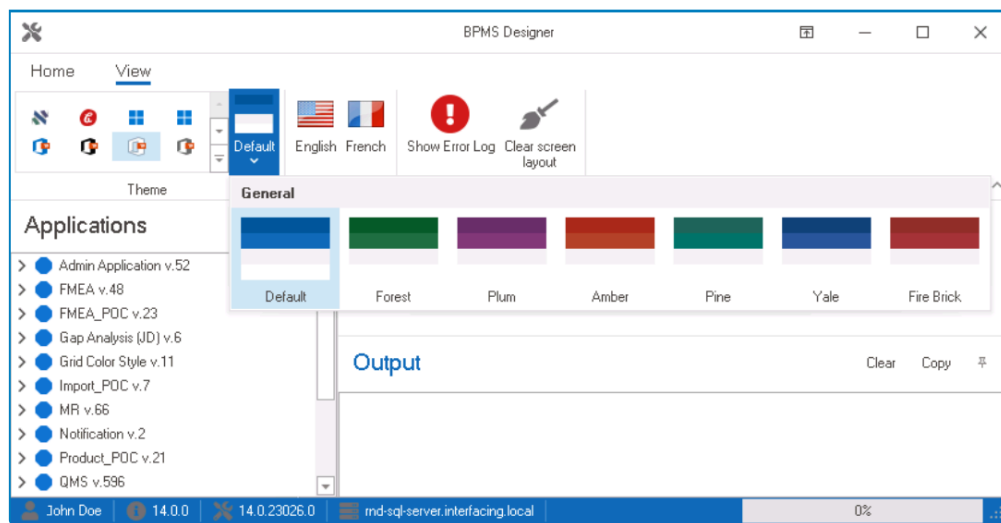


## 8.2.2.1. Theme

In the **Theme** group of the View tab, users are provided access to a variety of themes which they can use to customize the appearance of the Designer.



1. **Theme Gallery:** This field displays the gallery of available themes (also known as skins).
  - To browse through the gallery, click the up and down arrows. To display the full gallery, however, click on the **More**  button.
  - When applied, these themes alter the look and feel of the program.
2. **Color Schemes:** This field displays the gallery of available color schemes.
  - Depending on which theme is selected, the options in this gallery may vary.



- When applied, these schemes act as accent colors, providing a contrast to the primary palette of the program.



When a theme and/or color scheme is selected, it will remain as the default until the user changes it.

## 8.2.2.2. Language

---

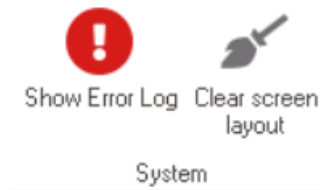
In the **Language** group of the View tab, users can select their language preference for the Designer.



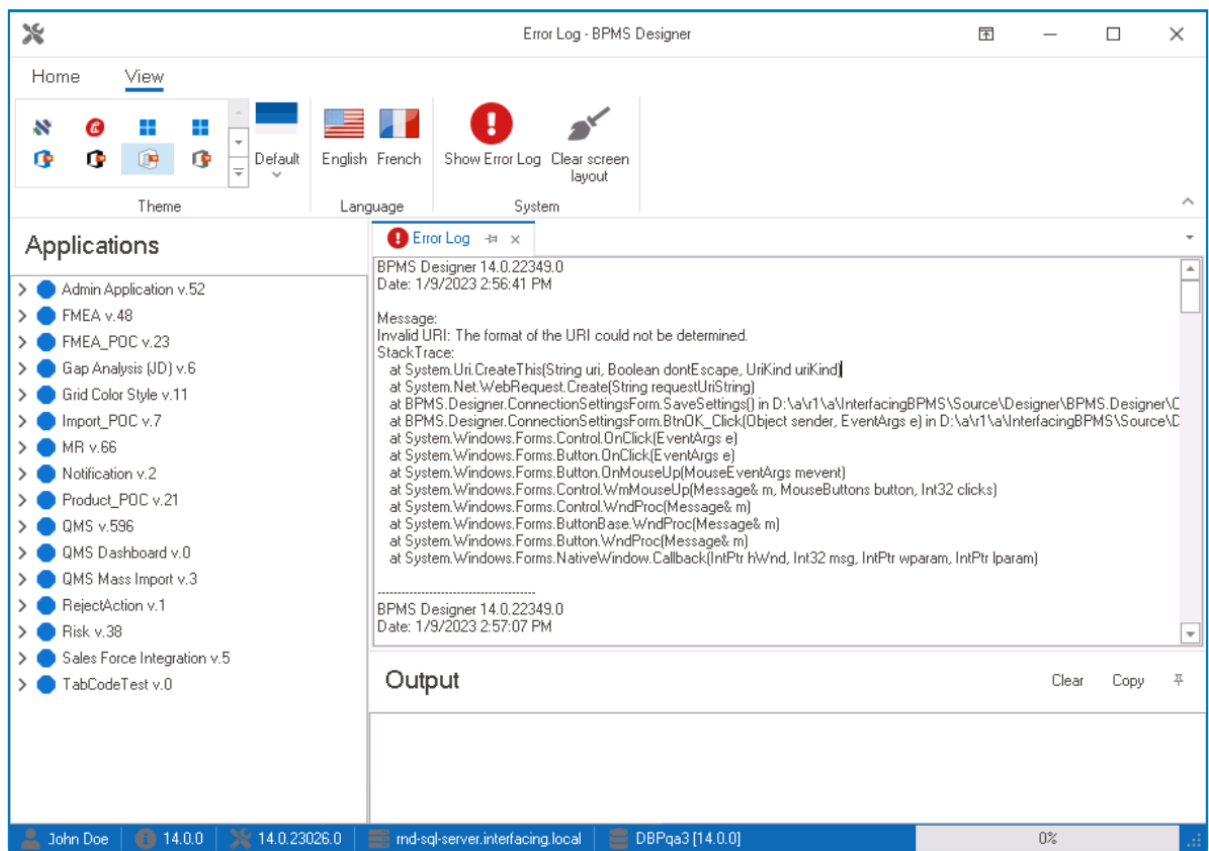
\* Currently, we only support English and French as language options.

## 8.2.2.3. System

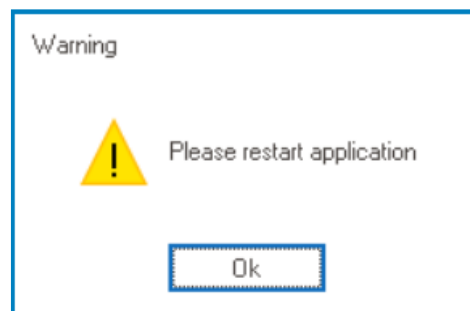
In the **System** group of the View tab, users are provided access to the following commands:



1. **Show Error Log:** This button allows users to access the Error Log for the program.
  - If selected, the Error Log will be displayed in a new tab.



2. **Clear Screen Layout:** This button allows users to clear the screen layout.
  - If selected, users will receive a warning message informing them that they need to restart the application.

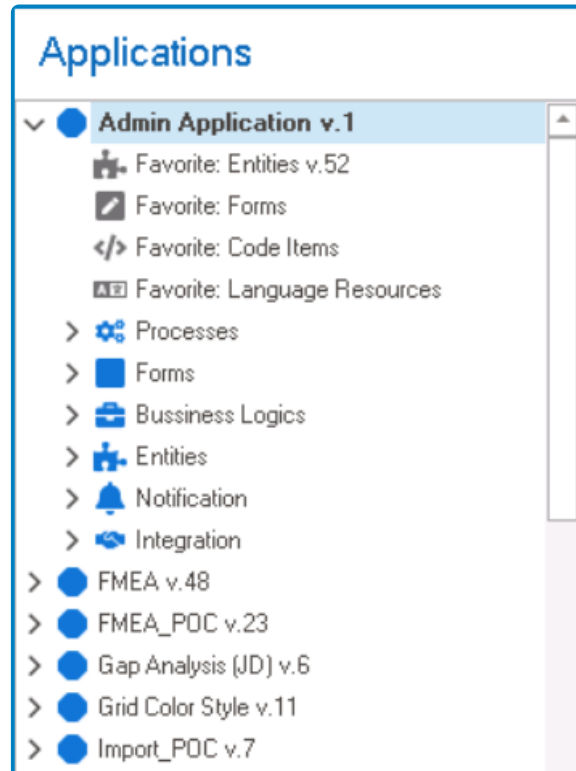


- To close this window, click on the **OK** button.
- Once users have restarted the application, the layout of the screen will be reset to its default

**state.**

## 8.3. Application Tree

In the Application Tree, users can access the list of applications with their corresponding set of modules.

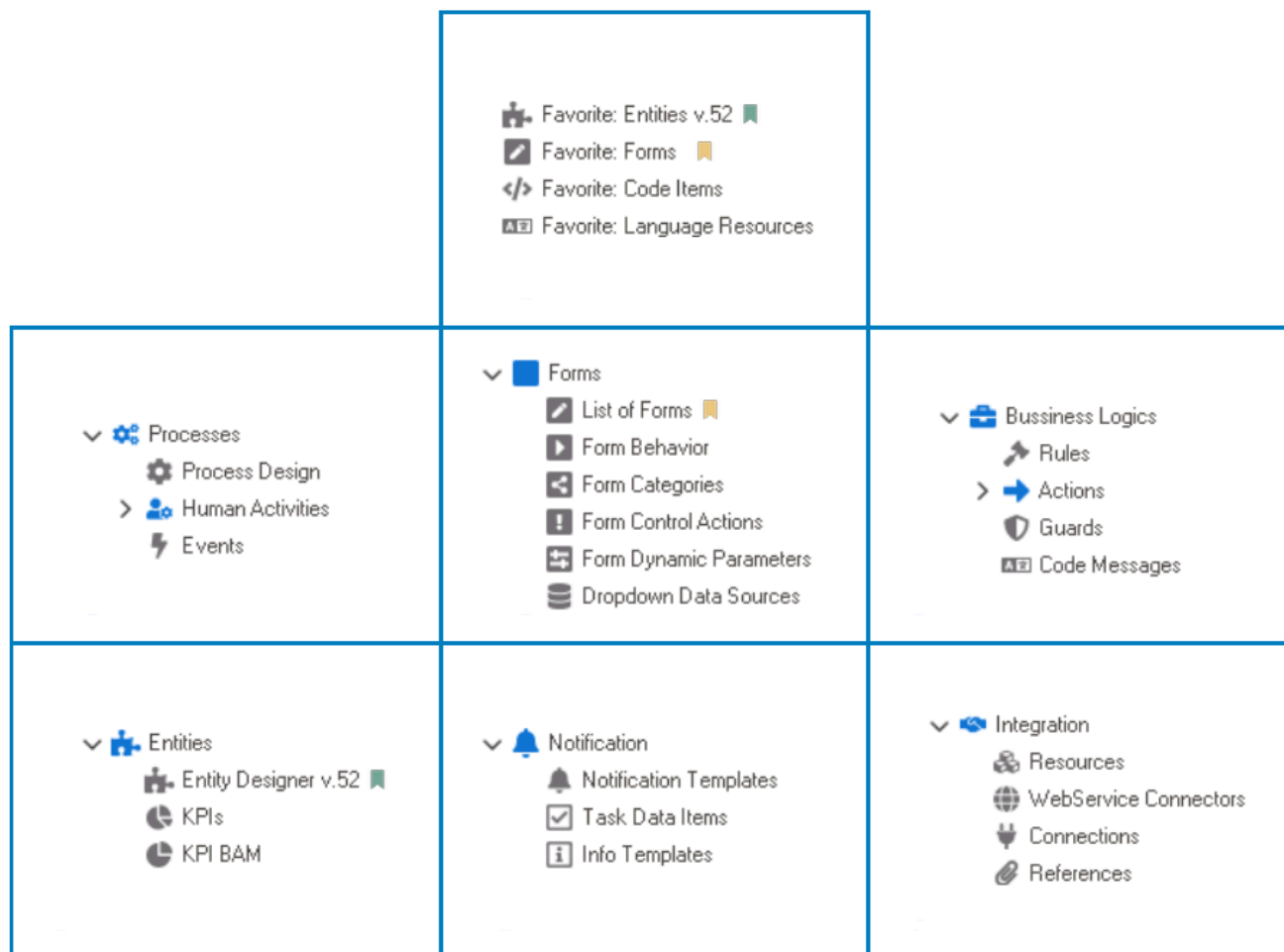


✿ To change the position of the Application Tree in the Designer, simply drag and drop it to one of the alternative positions provided.

For each application in the list, there is the following set of modules:

- [Code Items](#)
- [Language Resources](#)
- [Processes](#)
- [Forms](#)
- [Business Logics](#)
- [Entities](#)
- [Notification](#)
- [Integration](#)

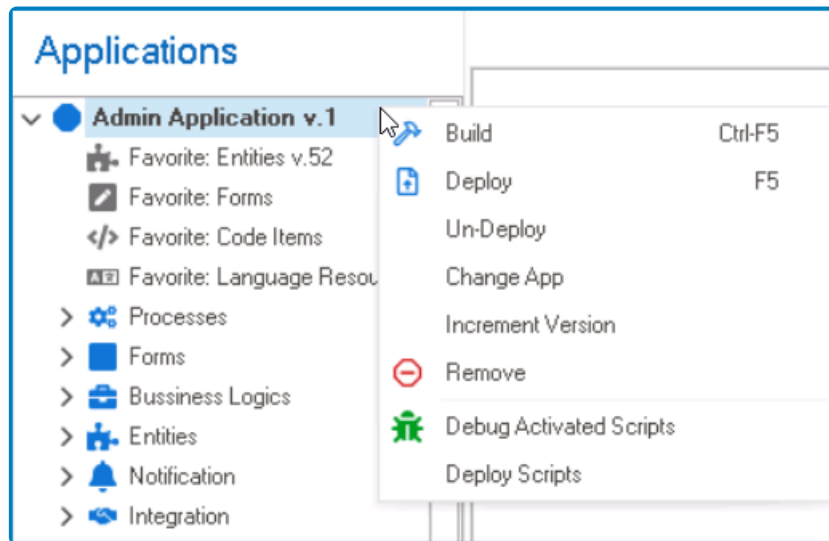
With the exception of the 'favorite' modules, each of these modules has its own subset of elements.



✿ By right-clicking on an application listed in the Application Tree, users can access a context menu containing various options to manage the selected application. [For more information on the options in this menu, click here.](#)

## 8.3.1. Application Tree Context Menu

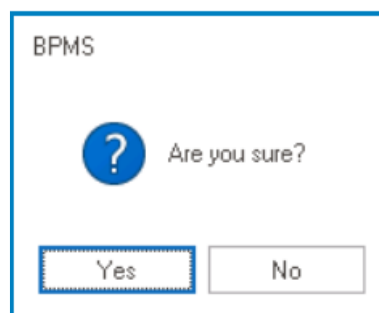
By right-clicking on an application listed in the [Application Tree](#), users can access a context menu containing various options to manage the selected application.



1. **Build:** This option allows users to build the application.
  - When an application is built, its components are compiled and sent to the web server.
2. **Deploy:** This option allows users to deploy the application.
  - When an application is deployed, its components are built and executed in the web server.
  - Any changes made to the application are published in the web as well.

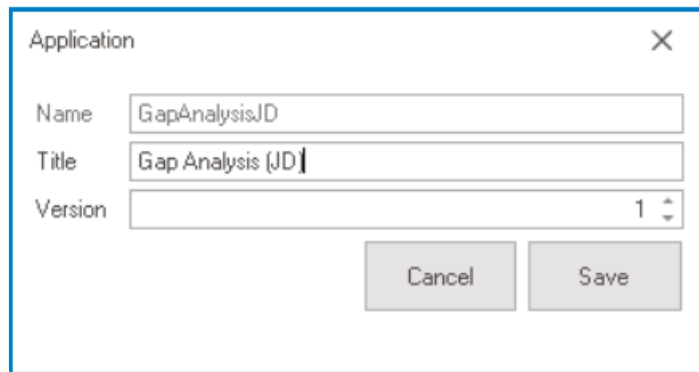
\* Deployment, in this context, refers to the process of creating application-specific files and copying them into the program's web directory path. Users must deploy an application in order to be able to access it in the web server.

3. **Un-Deploy:** This option allows users to undo the deployment of the application, removing it from the web.
  - If this option is selected, users will be alerted of the action with the following confirmation message:



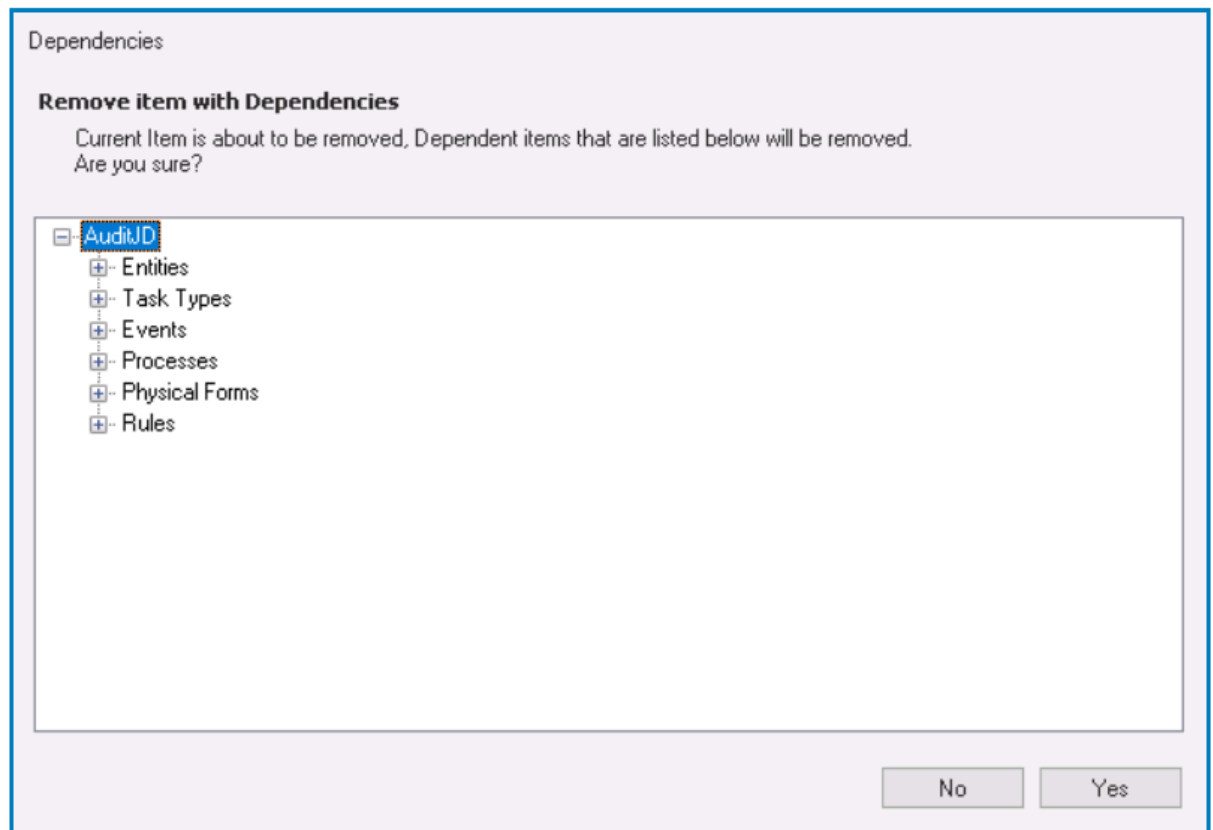
- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
4. **Change App:** This option allows users to change or modify the application properties.
    - If this option is selected, it will display the [Application](#) window. In this window, users can

change the title and version number of the application.



The 'Application' dialog box contains three input fields: 'Name' with the value 'GapAnalysisJD', 'Title' with the value 'Gap Analysis (JD)', and 'Version' with a spinner set to '1'. At the bottom right are 'Cancel' and 'Save' buttons.

5. **Increment Version:** This option allows users to manually increase the version number of an application.
  - This can be helpful for keeping track of changes made to the application or for marking significant milestones in the development process.
6. **Remove:** This option allows users to remove the application.
  - If this option is selected, it will display the Dependencies window. In this window, users can access an overview of the selected application, including its dependent objects.



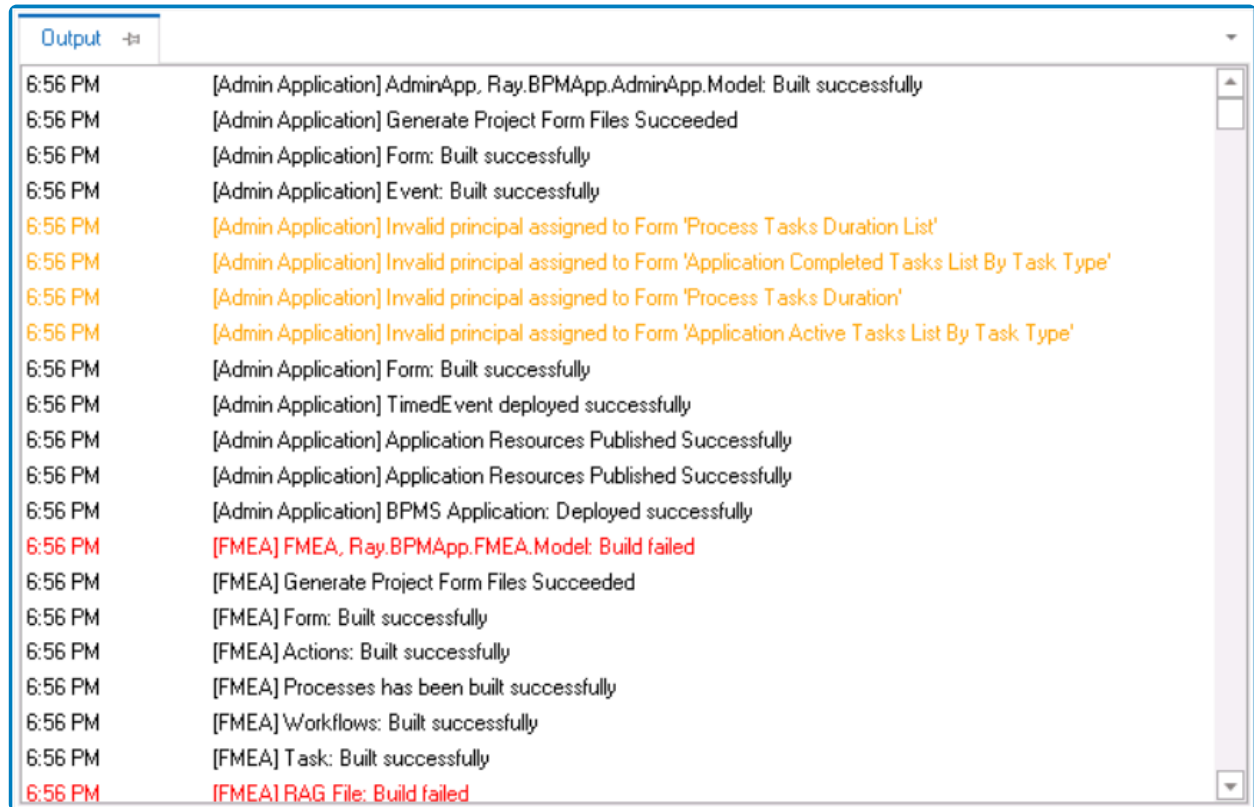
The 'Dependencies' dialog box has a title bar 'Dependencies'. Below it is the section 'Remove item with Dependencies' with the text: 'Current Item is about to be removed, Dependent items that are listed below will be removed. Are you sure?'. A tree view shows a folder 'AuditJD' expanded, containing sub-items: 'Entities', 'Task Types', 'Events', 'Processes', 'Physical Forms', and 'Rules'. At the bottom right are 'No' and 'Yes' buttons.

- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
7. **Debug Activated Scripts:** This option allows users to debug scripts that have been activated in the application.
  8. **Deploy Scripts:** This option allows users to deploy scripts to the application.



## 8.4. Output

Whenever an application is created or undergoes any major change (e.g., building, deployment, or removal), the program automatically performs a validity check. All items in the application are examined and the results are displayed in the Output field.



\* To change the position of the Output field in the Designer, simply drag and drop it to one of the alternative positions provided.

In the Output field, colors are used to indicate operation status.

| Color  | Operation Status   |
|--------|--|
| Black  | Success—All items have been properly created, built, deployed, or removed.   |
| Yellow | Warning—An error occurred but did not interrupt the completion of the operation. It can be repaired, if necessary. |
| Red    | Error—There is an error that prevents building/deployment and must be resolved.                                    |

## 8.5. System Information

---

At the very bottom of the Designer, there is the following status bar:

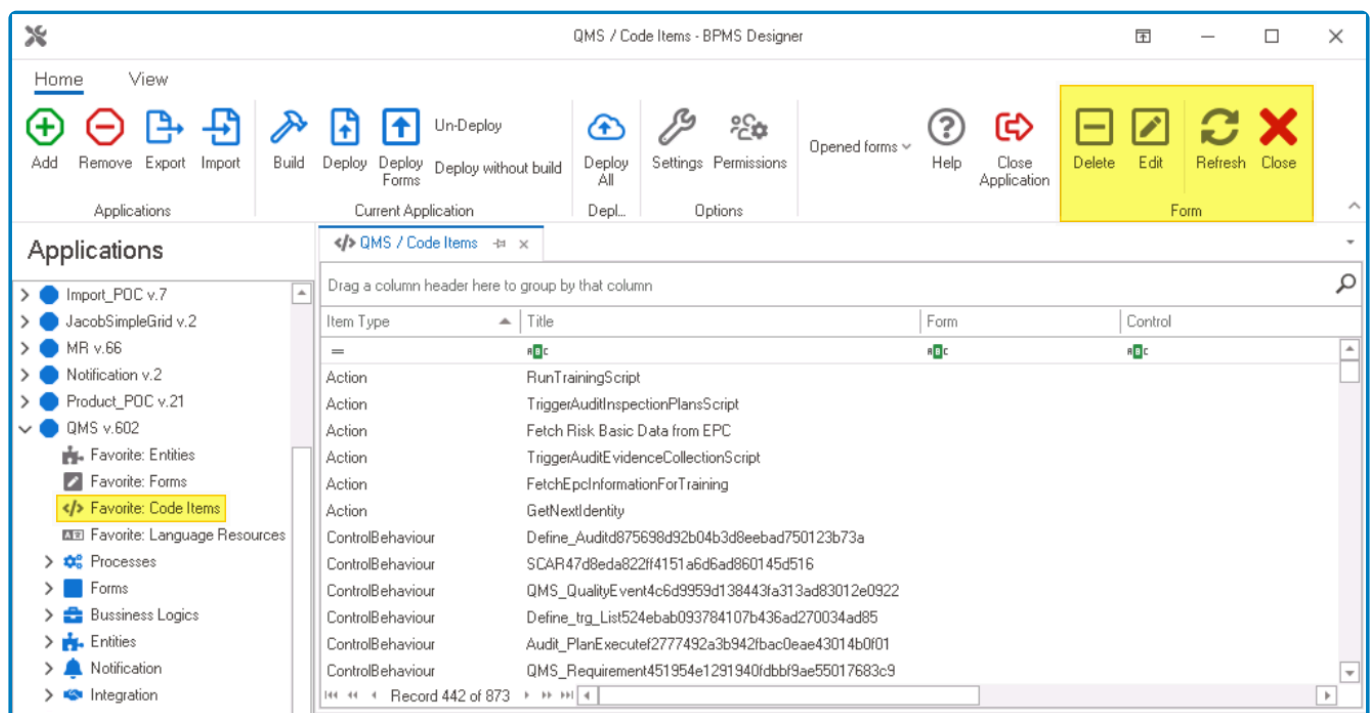


In this bar, users are provided access to system-related information. This includes the current user's name, the BPMS version, the Designer version, the application server name, and the database name.

## 9. Code Items

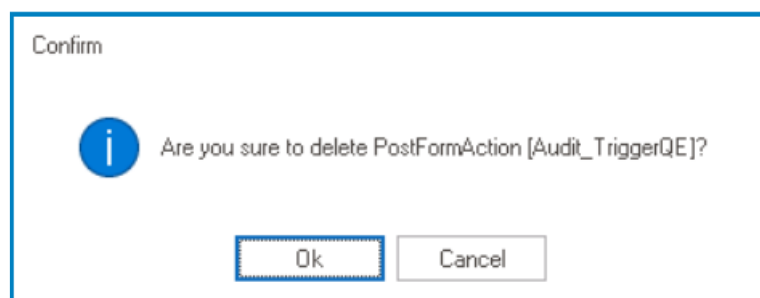
In the **Code Items** module, users can access all defined code items for the application, including actions, control behaviors, dynamic parameters, info templates, rules, and any other elements that have been defined as part of the application's codebase.

✿ By providing a centralized location for viewing and managing these components, this module simplifies the process of maintaining and updating the application's behavior and functionality over time.



1. **Delete:** This button allows users to delete an existing code item.

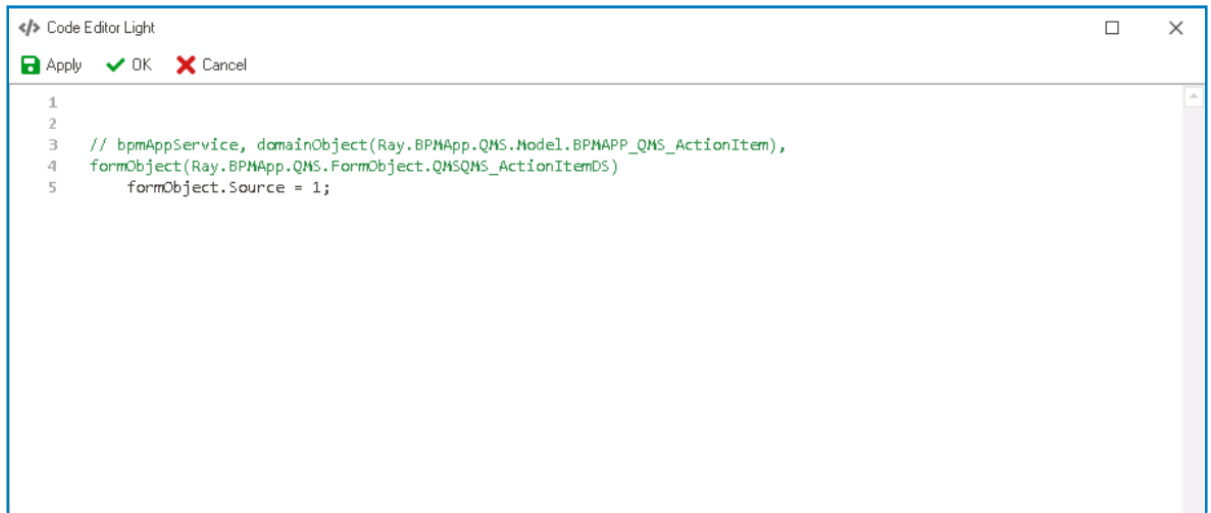
- If this option is selected, users will be alerted of the action with the following confirmation message:



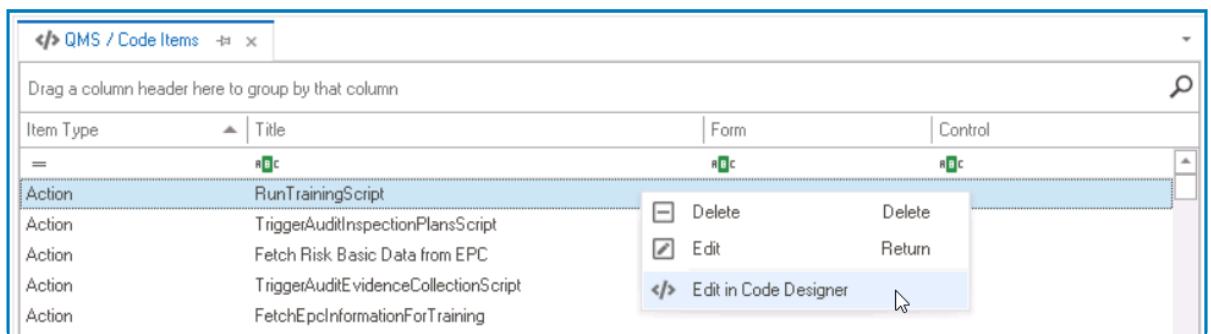
- **OK or Cancel:** To confirm the removal, click on the **OK** button. To cancel, click on the **Cancel** button.

2. **Edit:** This button allows users to edit an existing code item.

- If this option is selected, it will launch the Code Editor Light, in which the existing code will be displayed.



- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- To edit an existing code item in the Code Designer, simply right-click on the code item and select the **Edit in Code Designer** option.



✿ Unlike the Code Designer, the Code Editor Light is a simpler and more lightweight tool that is designed specifically for editing and modifying existing code. It has comparably fewer features and capabilities; however, it is faster to load and easier to use for smaller editing tasks.

3. **Refresh:** This button allows users to refresh the tab.
4. **Close:** This button allows users to close the tab.

**Note:** To view the script for a particular code item, click on the corresponding **Code** column.

QMS / Code Items

Drag a column header here to group by that column

| Title                                   | Form                    | Control | Control Type | Active                              | Code   |
|---|-------------------------|---------|--------------|-------------------------------------|--|
| DQ_QualityEvent_ActionItemGrid          | QualityEvent_ActionItem | Grid    | RayGrid      | <input checked="" type="checkbox"/> |  |
| Set audit as approved                   |                         |         |              |                                     | <div><div></div><div></div></div>                                |
| ExtensionRequest                        |                         |         |              | <input checked="" type="checkbox"/> | //bpmAppService.domainObject.eventObject,                        |
| Conduct Trainings                       |                         |         |              | <input checked="" type="checkbox"/> | Ray.BPMS.RuleModel.GuardResult result;                           |
| User Training                           |                         |         |              | <input checked="" type="checkbox"/> | if (System.Diagnostics.Debugger.IsAttached)                      |
| ManagerTraining                         |                         |         |              | <input checked="" type="checkbox"/> | System.Diagnostics.Debugger.Break();                             |
| UpcomingRecurrence                      |                         |         |              | <input checked="" type="checkbox"/> | domainObject.IsApproved = true;                                  |
| AuditPlanConductInfoTemplate            |                         |         |              | <input checked="" type="checkbox"/> | bpmAppService.BPMS.Services.DomainObjectService.SaveDomainObject |
| TrainingInstanceInfoTemplate            |                         |         |              | <input checked="" type="checkbox"/> | (domainObject);  |
| Training_DeleteNotificationInfoTemplate |                         |         |              | <input checked="" type="checkbox"/> | bpmAppService.UnitOfWork.CommitTransaction(false);               |
| ActionItem_ExtensionRequestInfo         |                         |         |              | <input checked="" type="checkbox"/> | ...  |

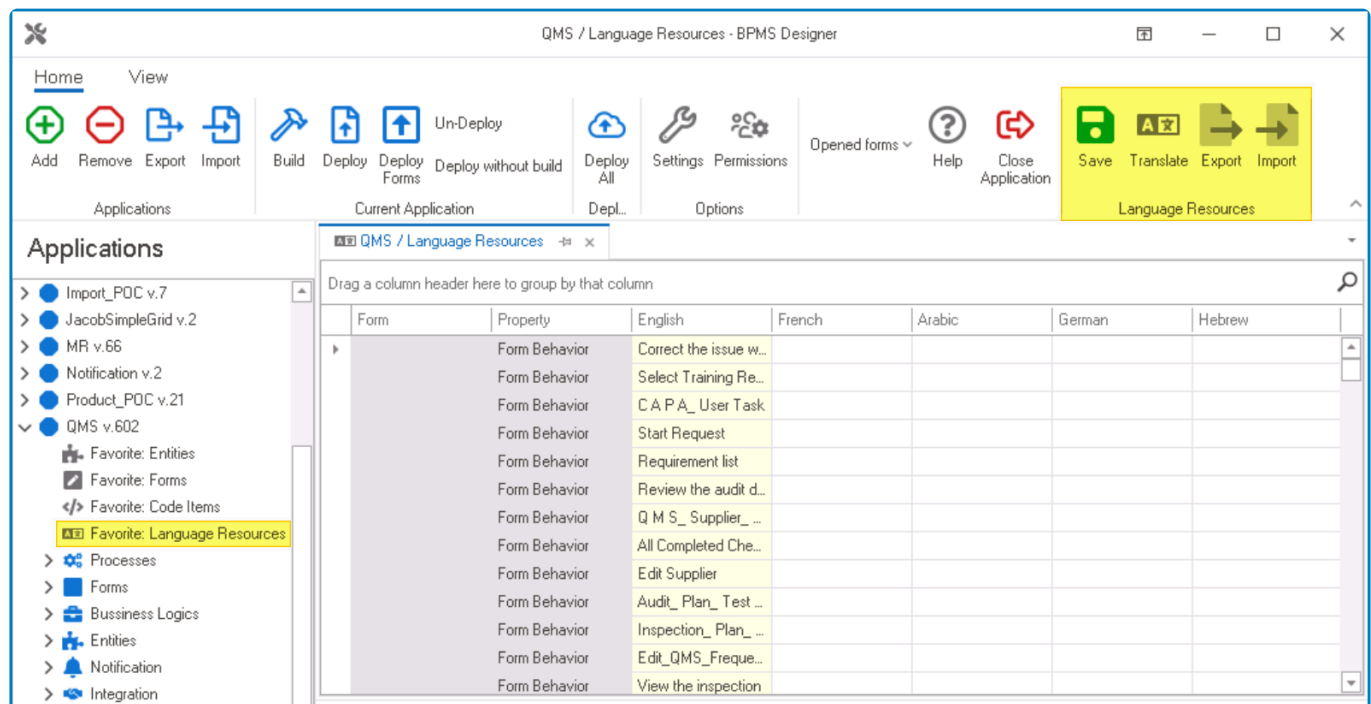
OK

Cancel

# 10. Language Resources

In the **Language Resources** module, users can either automatically or manually translate the user interface of the application into different languages.

- ✿ These UI elements include form behaviors, control captions, column headers, notification items, code messages, events, dropdown options, and any other visible elements in the web application.



- ✿ Users can determine which languages are available for translation in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

1. **Save:** This button allows users to save the translation.
2. **Translate:** This button allows users to automatically translate UI elements into all available languages.
  - If selected, the system will generate the translations in the corresponding language columns of the table.

| QMS / Language Resources                          |      |               |                        |                         |                      |                        |                        |
|---|------|---------------|------------------------|-------------------------|----------------------|------------------------|------------------------|
| Drag a column header here to group by that column |      |               |                        |                         |                      |                        |                        |
|   | Form | Property      | English                | French                  | Arabic               | German                 | Hebrew                 |
|   |      | Form Behavior | Correct the issue w... | Corrigez le problèm...  | تصحيح المشكلة مع...  | Beheben Sie das P...   | תקן את הבעיה ב...      |
|   |      | Form Behavior | Select Training Re...  | Sélectionnez la de...   | اختر طلب تدريب       | Wählen Sie Schul...    | בחר בקשת הדרכה         |
|   |      | Form Behavior | C A P A_ User Task     | CAP A_ Tâche utili...   | مهمة المس...         | CAP A_ Benutzerta...   | CAP A_ משימת מ...      |
|   |      | Form Behavior | Start Request          | Démarrer la deman...    | بدء الطلب            | Anfrage starten        | התחל בקשה              |
|   |      | Form Behavior | Requirement list       | Liste des exigences     | قائمة المتطلبات      | Anforderungsliste      | רשימת דרישות           |
|   |      | Form Behavior | Review the audit d...  | Examiner la définiti... | راجع تعريف التدقيق   | Überprüfen Sie die ... | סקור את הגדרות ה...    |
|   |      | Form Behavior | Q M S_ Supplier_ ...   | Q M S_ Fournisseur...   | Q M S_ المورد ب...   | Q M S_ Lieferant_ ...  | Q M S_ Supplier_ Sc... |
|   |      | Form Behavior | All Completed Che...   | Tous les chèques t...   | جميع الشيكات المك... | Alle abgeschlossen...  | כל המוצאח הושלמה       |
|   |      | Form Behavior | Edit Supplier          | Modifier le fournis...  | تعديل المورد         | Lieferant bearbeiten   | ערוך ספק               |
|   |      | Form Behavior | Audit_Plan_Test ...    | Audit_Plan_Test ...     | الخطة جولة ...       | Audit_Plan_Testb...    | ביקורת תוכנית ...      |
|   |      | Form Behavior | Inspection_Plan_ ...   | Inspection_Plan_ ...    | التفتيش الخطة ...    | Inspektion_Plan_ ...   | בדיקה תוכנית ...       |
|   |      | Form Behavior | Edit_QMS_Freque...     | Edit_QMS_Freque...      | Edit_QMS_Freque...   | Edit_QMS_Freque...     | Edit_QMS_Frequen...    |
|   |      | Form Behavior | View the inspection    | Voir le contrôle        | عرض التفتيش          | Sehen Sie sich die ... | צפה בבדיקה             |

! It is important to note that this feature may not be as accurate or contextually relevant as manual translations, so it is recommended to review the results before finalizing.

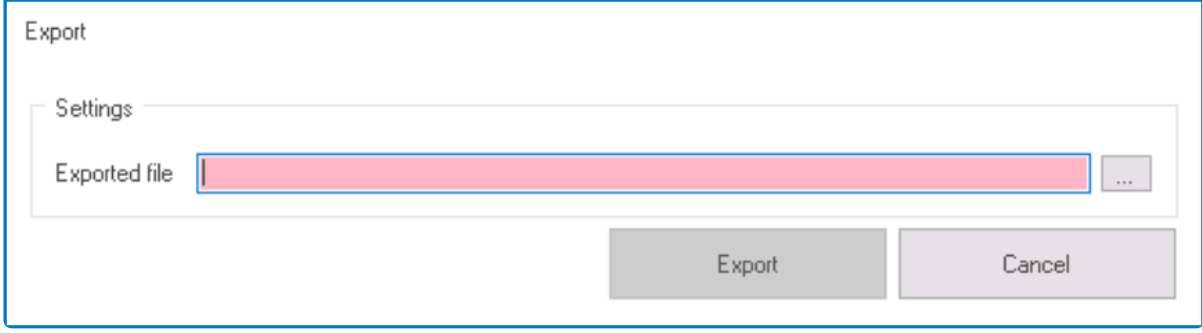
#### Manual Translation


To manually translate UI elements into the available languages, select the corresponding column in the table and enter the translation.

| QMS / Language Resources                          |      |               |                        |        |        |        |        |
|---|------|---------------|------------------------|--------|--------|--------|--------|
| Drag a column header here to group by that column |      |               |                        |        |        |        |        |
|   | Form | Property      | English                | French | Arabic | German | Hebrew |
|   |      | Form Behavior | Correct the issue w... |        |        |        |        |
|   |      | Form Behavior | Select Training Re...  |        |        |        |        |
|   |      | Form Behavior | C A P A_ User Task     |        |        |        |        |
|   |      | Form Behavior | Start Request          |        |        |        |        |
|   |      | Form Behavior | Requirement list       |        |        |        |        |
|   |      | Form Behavior | Review the audit d...  |        |        |        |        |
|   |      | Form Behavior | Q M S_ Supplier_ ...   |        |        |        |        |
|   |      | Form Behavior | All Completed Che...   |        |        |        |        |
|   |      | Form Behavior | Edit Supplier          |        |        |        |        |
|   |      | Form Behavior | Audit_Plan_Test ...    |        |        |        |        |
|   |      | Form Behavior | Inspection_Plan_ ...   |        |        |        |        |
|   |      | Form Behavior | Edit_QMS_Freque...     |        |        |        |        |
|   |      | Form Behavior | View the inspection    |        |        |        |        |

! Please note that for the translation to be effective on forms and their controls, the form must be saved and closed beforehand.

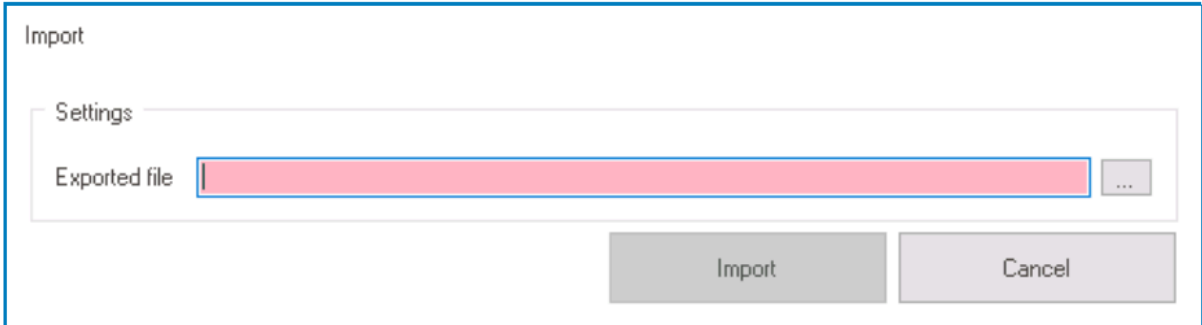
3. **Export:** This button allows users to export the translation table (in the XLSX format).
  - If selected, it will display the Export window.


The 'Export' dialog box has a title bar 'Export'. Below it is a 'Settings' section containing an 'Exported file' text field with a pink background and a blue border. To the right of the text field is a small grey button with three dots. At the bottom right of the dialog are two buttons: 'Export' and 'Cancel'.

- **Exported File:** This field allows users to specify the physical path/folder to which the translation file will be exported. It is a **mandatory** field.
  - Clicking on the  button will prompt a Save As dialog to appear, enabling users choose a location on their local system to save the file.
- **Export or Cancel:** To proceed with the export, click on the **Export** button. To cancel, click on the **Cancel** button.
  - To enable the **Export** button, users must first set up the file in its intended location.

4. **Import:** This button allows users to import a translation table (in the XLSX format).

- If selected, it will display the Import window.

The 'Import' dialog box has a title bar 'Import'. Below it is a 'Settings' section containing an 'Exported file' text field with a pink background and a blue border. To the right of the text field is a small grey button with three dots. At the bottom right of the dialog are two buttons: 'Import' and 'Cancel'.

- **Exported File:** This field allows users to select the translation file that they would like to import. It is a **mandatory** field.
  - Clicking on the  button will prompt a file dialog to appear, enabling users to browse and choose a file from their local system.
- **Import or Cancel:** To proceed with the import, click on the **Import** button. To cancel, click on the **Cancel** button.
  - To enable the **Import** button, users must first select the intended file.



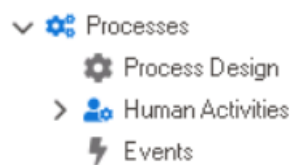
Users can manipulate the data presented in this tab using certain techniques. For more information on these techniques, [click here](#).



# 11. Processes

---

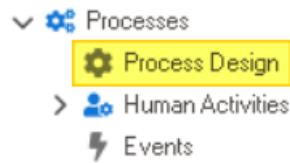
In the **Processes** module, users are provided access to the following elements:



1. [Process Design](#)
2. [Human Activities](#)
3. [Events](#)

# 11.1. Process Design

In the **Process Design** section of this module, users can create and design processes for the application.



✿ A process is a series of interconnected tasks that function together to produce a result known as the workflow. For various reasons—including efficiency, productivity, and accuracy—businesses strive to automate their processes. An essential part of automating processes, however, involves identifying the proper processes and subprocesses and designing them.

The DBP Designer enables users to identify and define the different processes used by organizations.

- When creating a process, the following elements should be considered and documented:
  - Inputs
  - Outputs
  - Roles Involved in the Process
  - Rules
  - Operations
  - Operational Sequence and Workflow
  - Connection to Other Processes
  - Connection with External Information Systems
- For more information on creating processes, [click here](#).

Once they have been identified and defined, processes can then be designed and implemented in the Process Designer.

- For more information on designing processes, [click here](#).

## 11.1.1. Process Creation

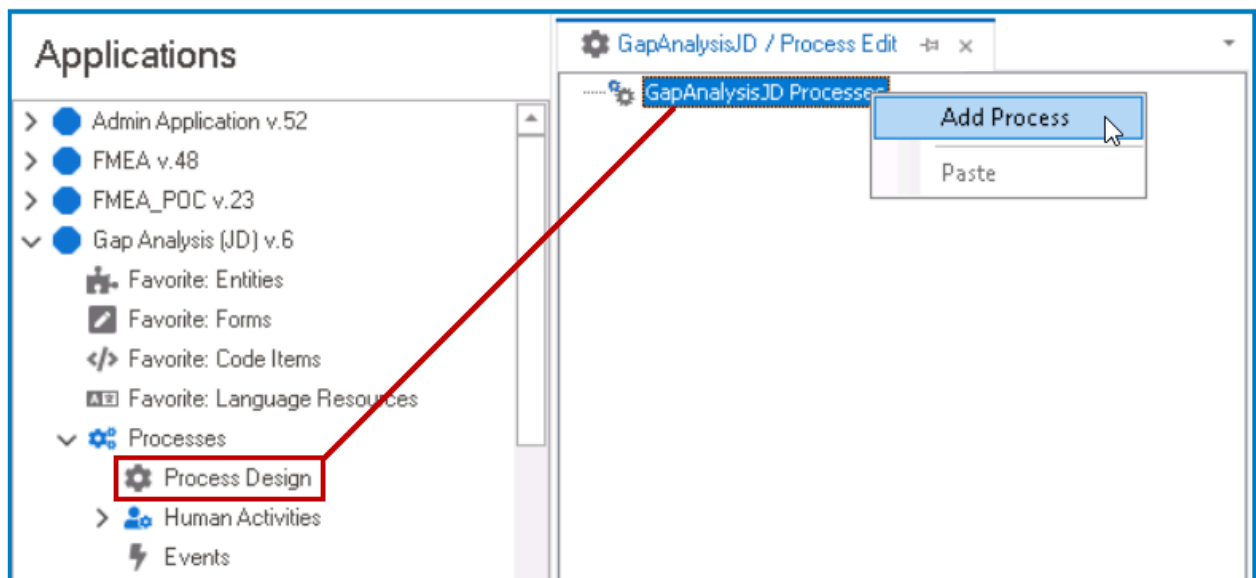
Before a process can be designed, it must first be identified and created.

To create a process for an application, take the following steps:

1. From the **Processes** module (inside the [Application Tree](#)), double-click on **Process Design**. It will open a new tab called Process Edit.

✿ When processes are created in this tab, they are organized into sets of tree-like diagrams. These diagrams are referred to as process trees.

2. In the Process Edit tab, right-click on “[AppName] Processes” and select the **Add Process** option.



3. Selecting **Add Process** will display the Process Edit window. In this window, users can input the process details. Upon doing so, save the process by clicking on the **OK** button.

A screenshot of the 'Process Edit' dialog box. It contains the following fields: 'Type' (a dropdown menu set to 'BPMS Human Process'), 'Name' (a text box with 'New Process'), 'Title' (a text box with 'New Process'), and 'Entity' (a dropdown menu set to 'New Gap Analysis'). There is an unchecked checkbox for 'Start manually' and a blue hyperlink for 'Specification'. At the bottom right are 'OK' and 'Cancel' buttons.

- **Type:** This is a single-select dropdown field that displays the list of process types.
  - All processes in an organization can be categorized into one of three types: human,

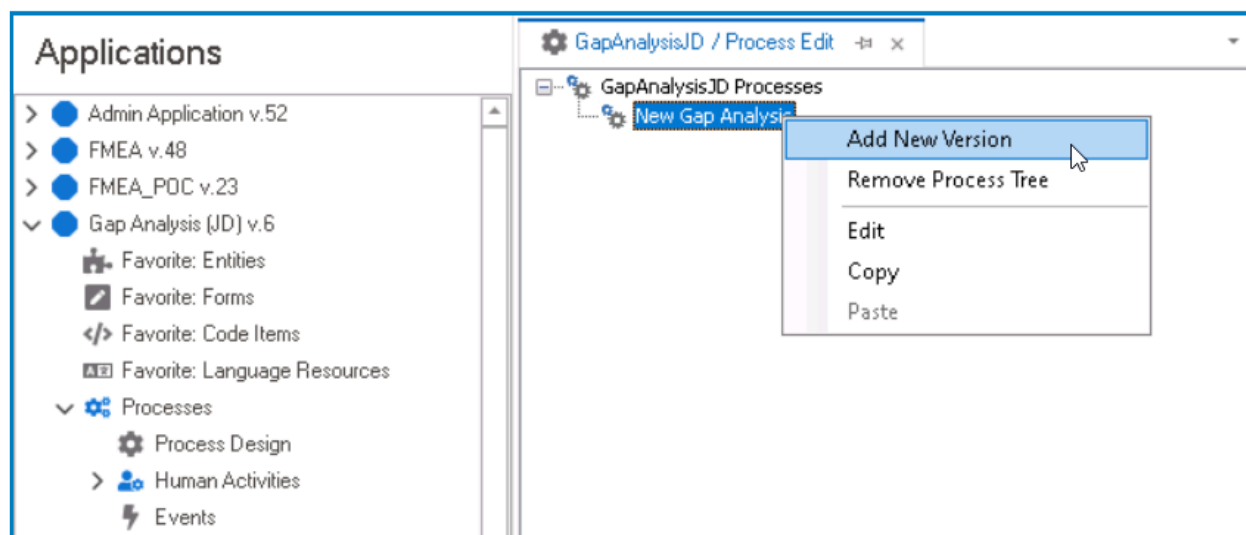
state machine, or simple.

- **Human Process:** a type of process in which the roles essential to its execution are performed by individuals/users.
  - These users can be used as tools in human tasks, system tasks, common tasks, sub-processes, general aggregation, sum, etc.
- **State Machine Process:** a type of process that uses a finite number of states in its execution.
  - Entities have two types of attributes—attributes with a dynamic nature and attributes with a static nature. Dynamic attributes change based on the circumstances and events of the entity. In this system, dynamic attributes can be implemented as “state machine.”
- **Simple Process:** a type of process that requires limited efforts for its execution.
- Depending on the type selected in this dropdown, different features will be displayed in the process toolbox.
  - For more information on this toolbox, [click here](#).
- **Name:** This is a text field that allows users to input the name of the process.
  - A process must have a unique name. It cannot use the same name as another process in the application.
- **Title:** This is a text field that allows users to input the title of the process.
  - Unlike its name, the title of a process does not have to be unique.
- **Entity:** This is a single-select dropdown field that displays the list of entities. It allows users to select the entity to which the process will be linked.
- **Start Manually:** This is a checkbox which, if selected, allows users to start the process manually.
  - If this checkbox is left unselected, by creating an instance of the entity, all the processes defined for that entity will be executed.
  - If this checkbox is selected, however, the process will only be executed by calling the custom code.
    - For more information on the process callback API, [click here](#).
- **Specification:** This is a hyperlink which, if selected, directs users to the [Process Specification Edit](#) window.



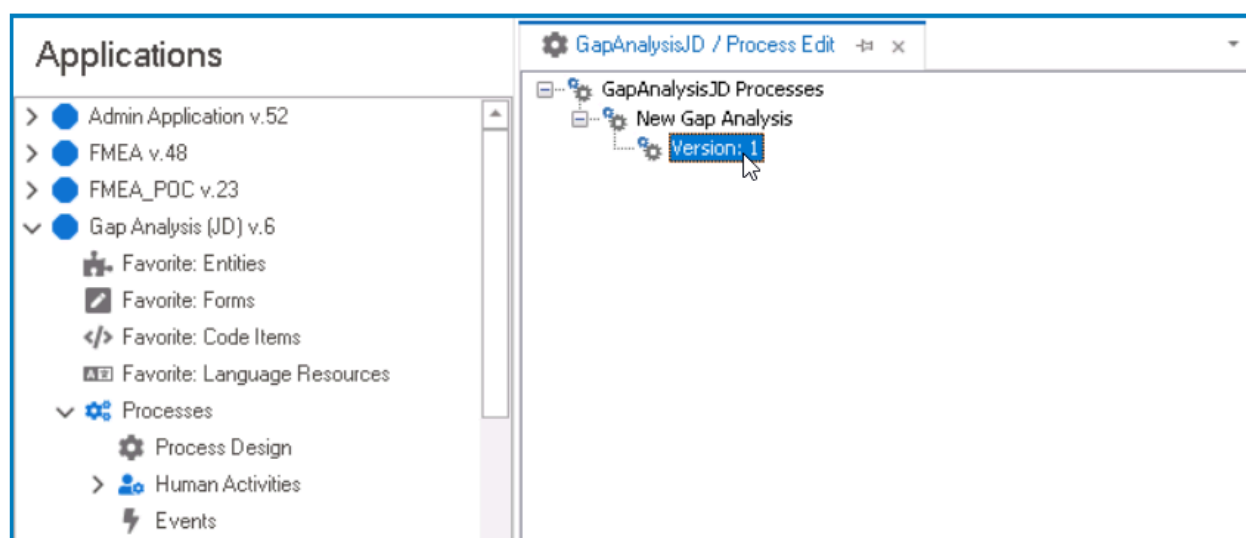
In this window, users can detail the process specifications, including its inputs, outputs, acceptance criteria, and performance indicators.

4. Next, in the Process Edit tab, right-click on the newly defined process and select the **Add New Version** option.



✿ For more details on the other options provided in this context menu, [click here](#).

5. Selecting **Add New Version** will generate a new version below the process. Double-click on the newly generated version. It will launch the [Process Designer](#) in a new tab.



## 11.1.1.1. Process Specification

---

In the Process Specification Edit window, users can detail the process specifications.

There are five tabs to consider in this window: (1) Specifications, (2) Inputs, (3) Outputs, (4) Acceptance Criteria, and (5) Performance Indicators.



To save changes per tab, click on the **Apply** button. To save changes and exit the Process Specification Edit window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

### 1. Specifications

In the Specifications tab, users can provide key information about the process, including (but not limited to) its goals, triggers, requirements, and rules. Users can also identify the roles and responsibilities of the stakeholders in the process—the participants, responsible party, accountable party, etc.

**Process Specification Edit**

Apply OK Cancel

Specifications Inputs Outputs Acceptance Criteria Performance indicators

Process Name: New Gap Analysis

Process Number:

Type: ☒ Main ☐ Support

Goal:

Responsible:

Description:

Trigger:

Participants:

Related Applications:

Requirements:

Geographical Distribution:

Designed by:

Accountable:

Reports:

Forms:

Rules and Regulations:

Related Policies:

- For the most part, this tab is composed of text fields into which users can enter the relevant information.
  - There is, however, the following exception:
    - Type:** This is a radio button that allows users to select only one option from a set of mutually exclusive options. Users can distinguish the process as either a main or support process.

## 2. Inputs

In the Inputs tab, users can set up the inputs of the process.

Process Specification Edit

Apply

OK

Cancel

Specifications

Inputs

Outputs

Acceptance Criteria

Performance indicators

| Output | Form | Time | Receiver |
|--------|------|------|----------|
| nbc    | nbc  | nbc  | nbc      |

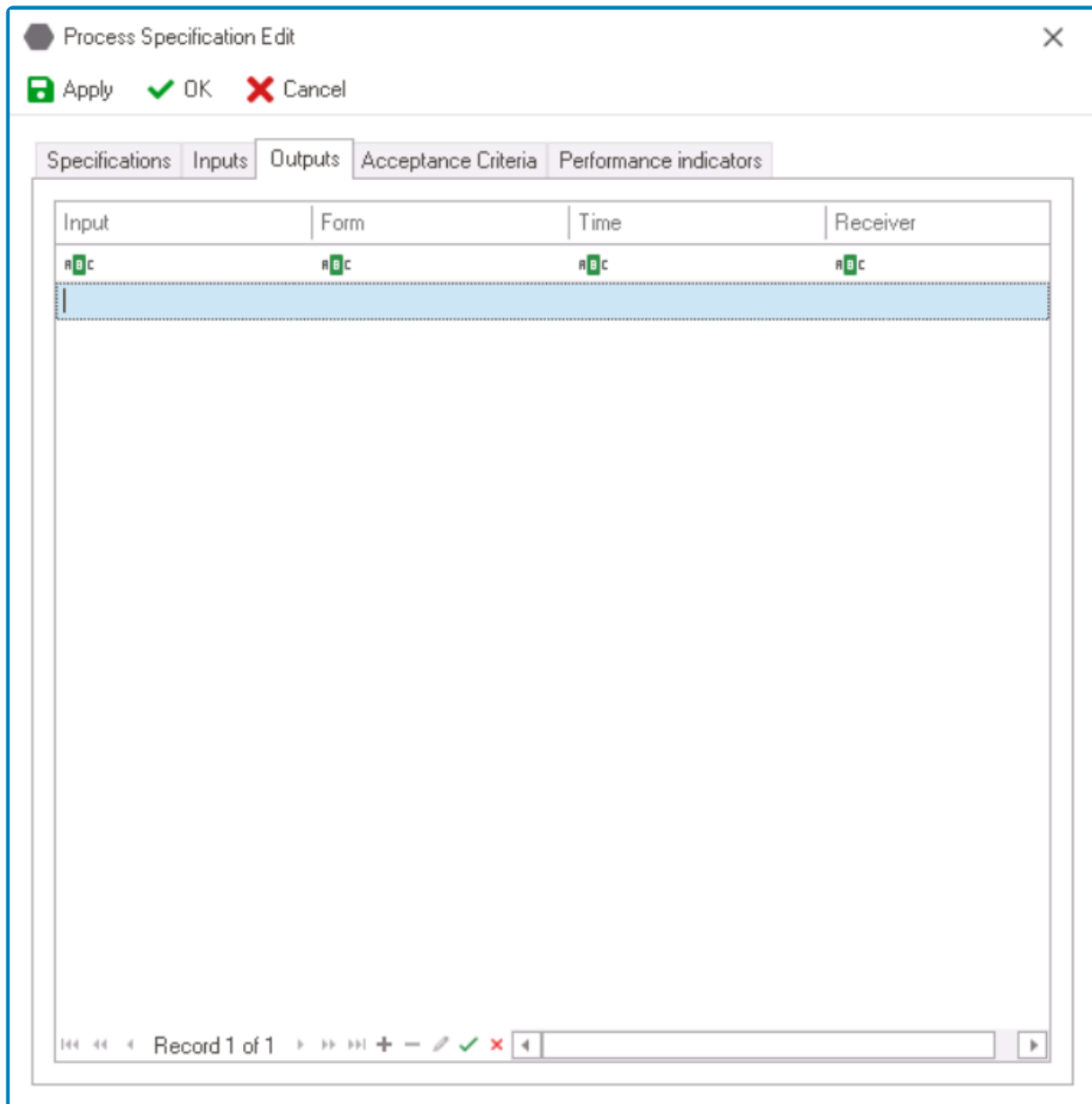
Record 1 of 1

- For each entry in the table, users can specify the output, form, time, and receiver.

3. **Outputs**

In the Outputs tab, users can set up the outputs of the process.





The image shows a 'Process Specification Edit' dialog box. At the top, there is a title bar with a close button (X). Below the title bar, there are three buttons: 'Apply' (with a floppy disk icon), 'OK' (with a green checkmark icon), and 'Cancel' (with a red X icon). The dialog box contains five tabs: 'Specifications', 'Inputs', 'Outputs', 'Acceptance Criteria', and 'Performance indicators'. The 'Inputs' tab is currently selected. Inside the 'Inputs' tab, there is a table with four columns: 'Input', 'Form', 'Time', and 'Receiver'. The first row of the table contains the text 'nbc' in each of the four columns. Below the table, there is a large empty rectangular area. At the bottom of the dialog box, there is a status bar that says 'Record 1 of 1' and includes several navigation icons (back, forward, search, etc.) and a small input field.

| Input | Form | Time | Receiver |
|-------|------|------|----------|
| nbc   | nbc  | nbc  | nbc      |

- For each entry in the table, users can specify the input, form, time, and starter.

#### 4. **Acceptance Criteria**

In the Acceptance Criteria tab, users can set up the acceptance criteria for the process.

Process Specification Edit

Apply OK Cancel

Specifications Inputs Outputs Acceptance Criteria Performance Indicators

| Name | Criterion | Average | Target |
|------|-----------|---------|--------|
| nbc  | nbc       | nbc     | nbc    |

Record 1 of 1

- For each entry in the table, users can specify the name, criterion, average, and target.

## 5. Performance Indicators

In the Performance Indicators tab, users can set up the performance indicators for the process.

Process Specification Edit

Apply OK Cancel

Specifications Inputs Outputs Acceptance Criteria Performance indicators

| Name | Approximate |
|------|-------------|
| nbc  | nbc         |

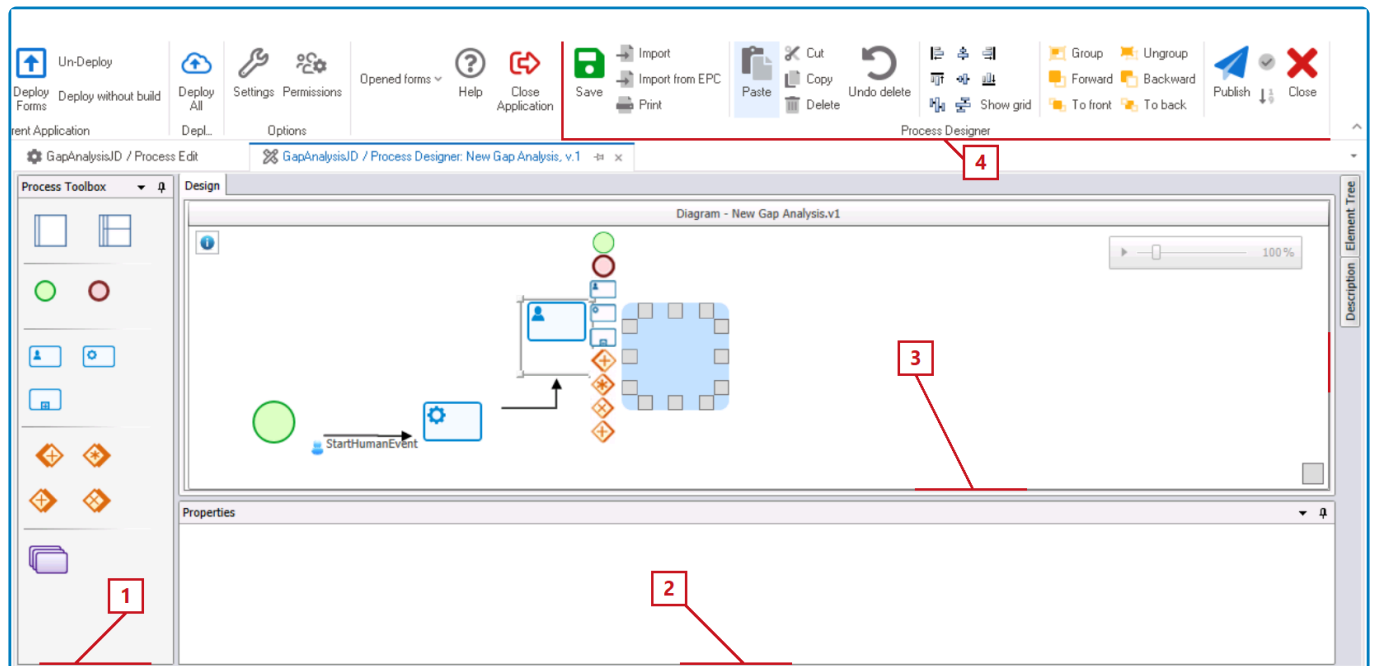
Record 1 of 1

- For each entry in the table, users can specify the name and approximate.

✿ For more information on how to add, edit, and delete records inside tables, [click here](#).

## 11.1.2. Process Designer

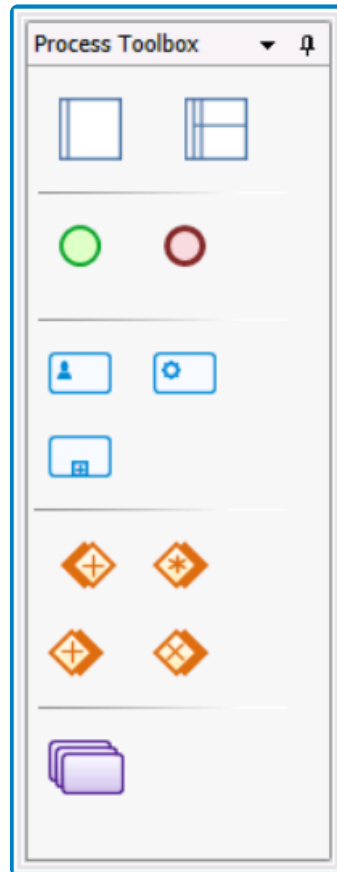
Once users have identified and defined the process, they can launch the Process Designer. The Process Designer is a tool which enables users to design and implement processes.



It consists of the following elements:

1. **Process Toolbox**

In the process toolbox, users can access the elements or flow objects needed to design a process.



To insert flow objects into the diagram, take the following steps:

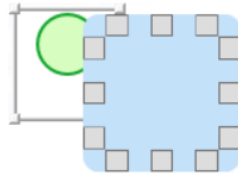
- First, drag and drop an object (start event, state, gateway, etc.) from the toolbox onto the design canvas.
  - If you have opted for a swimlane diagram, you will first need to drag and drop a pool, then a lane, then a flow object before proceeding to the next step.



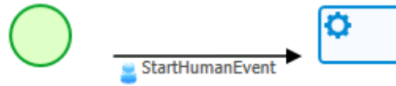
- Next, select the object now situated on the canvas. It will display to its right a list of thumbnails, each representing a different element that the object can be linked to.



- Select the appropriate thumbnail—that is, the one representing the next object needed for your process—and from the resulting blue field, click on the grey box that most closely aligns with your desired location for the second object.



- Once you have determined its location, the second object will appear in the diagram with an arrow linking it to the first object. This arrow is known as an event.



- To modify the position of any object in the diagram, simply select it and move as needed.



- To resize the object, select it and drag its corners.



 For more details on the process toolbox, [click here](#).

## 2. Properties


In the Properties panel, users can access the properties of the flow object selected on the canvas.

Properties

Name  Data Item [Empty]    Task Type [Empty]    ☐ Hide tree history

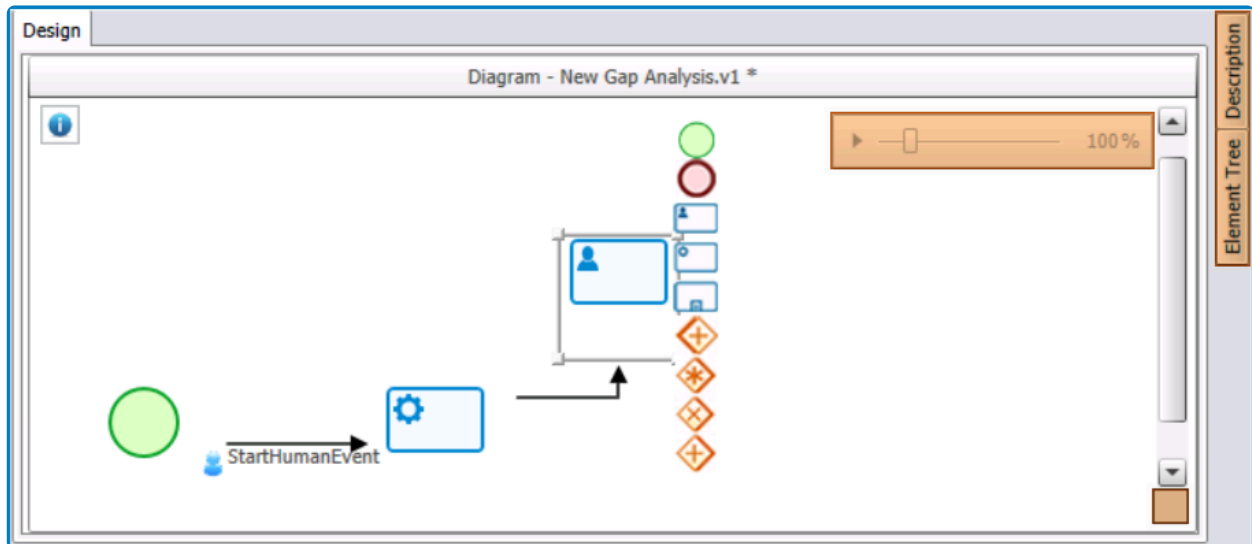
Title  Notifications  ☐ Active ☐ Hide graph history

- For each object, users can either define new information or modify existing information.

 Depending on the flow object selected, the properties displayed in this panel may differ. For more details on this matter, [click here](#).

## 3. Design Canvas:

On the design canvas, users can map out their processes.

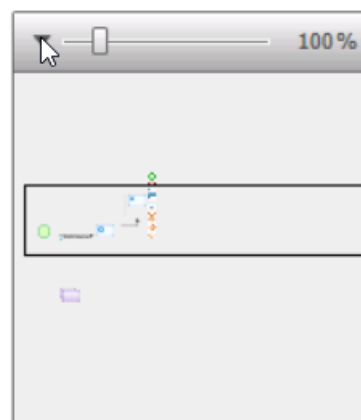


\* Users can resize both the design canvas and the various panels of the Process Designer (including the toolbox). [For more information on this option, click here.](#)

- There are four primary components to this canvas:
  1. **Zoom Slider**
    - Users can zoom in or out of the map using the zoom slider.
      - By default, the magnification level is set to 100%. To modify this setting, simply slide to the zoom percentage that you want.

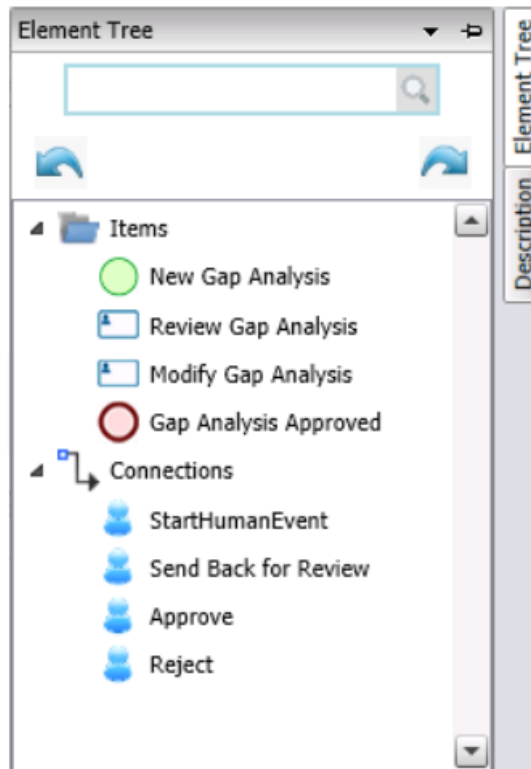


- Minimum Magnification Level: 25%
  - Maximum Magnification Level: 500%
- To expand the slider, click on the ► button. It will display a comprehensive view of the map, including any sections that are not immediately visible in the window.
  - In other words, it displays sections of the map that you would otherwise need to scroll down to see.



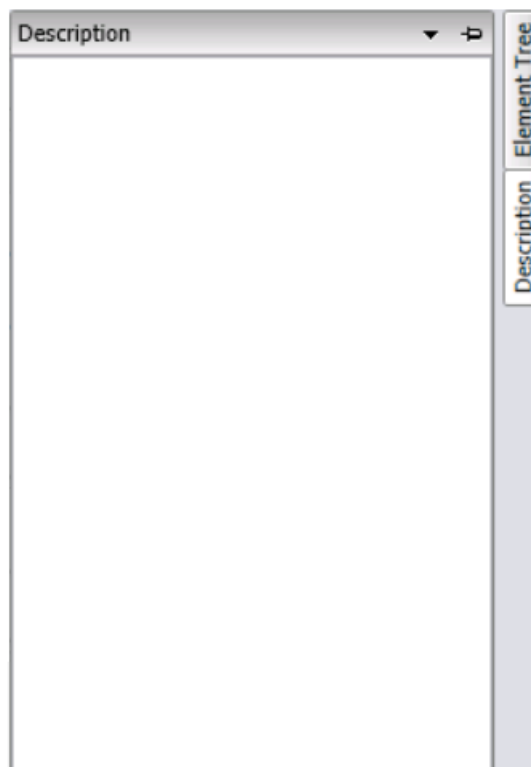
## 2. Element Tree

- To display the Element Tree panel, hover your cursor over the docked element.
  - In the panel, users can access the list of elements currently being used in the Process Designer. These elements are listed as “Items” (Flow Objects) and “Connections” (Data Flows).



## 3. Description

- To display the Description panel, hover your cursor over the docked element.
  - In the panel, users can provide a description of the process.





#### 4. **Validate**

- Users can check the validity of the process design by clicking on the **Validate** button in the canvas.
  - Icon: 
    - If the process design is valid, a message will appear in the lower left corner of the diagram stating “Process Validated Successfully.”
    - If the process design is invalid, a pop-up window will appear identifying the errors for the user.
      - An **X** will also appear adjacent to the **Validate** button.

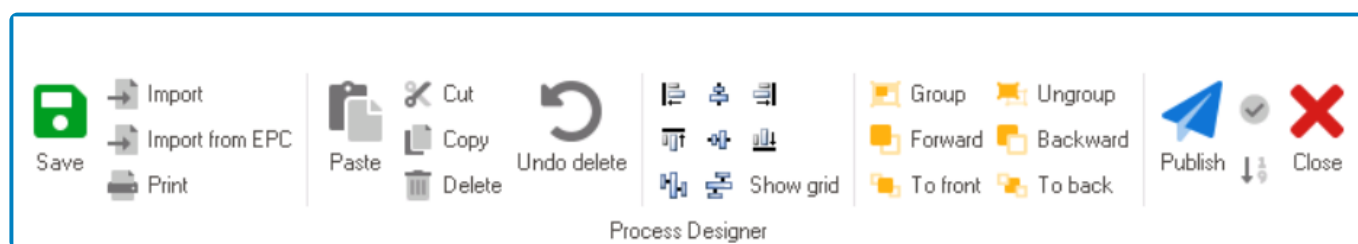


- For more details on this function, see the corresponding button in the [menu bar](#).

\* All panels in the Process Designer (including the toolbox) can be either docked or pinned. For more information on these display options, [click here](#).

#### 4. **Menu Bar:**

In the menu bar, users can access other features related to the process design.



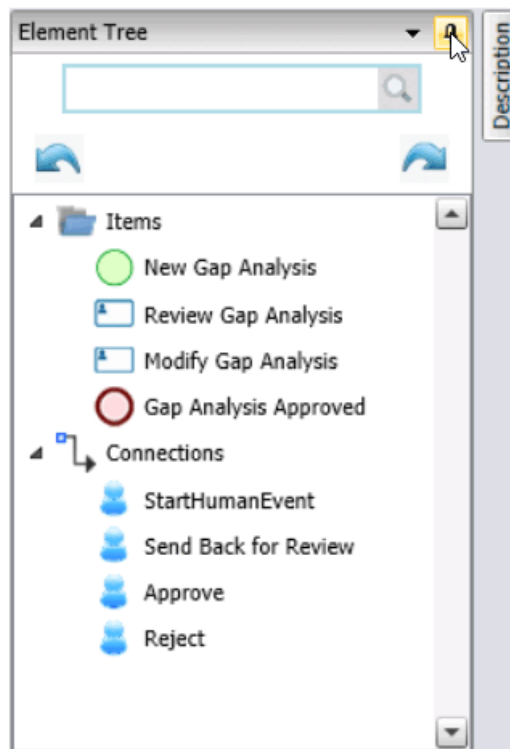
\* For more details on these features, [click here](#).

## 11.1.2.1. Panel Display Options

All panels in the Process Designer (including the toolbox) can be either docked or pinned.

To pin or dock a panel, toggle the  icon.

- A pinned panel remains on screen at all times.
  - For more information on how to resize a pinned panel, [click here](#).
- A docked panel appears as a tab on the side of the screen. When not in use, a docked panel is automatically hidden.



- To view a docked panel, user will need to hover over the tab.

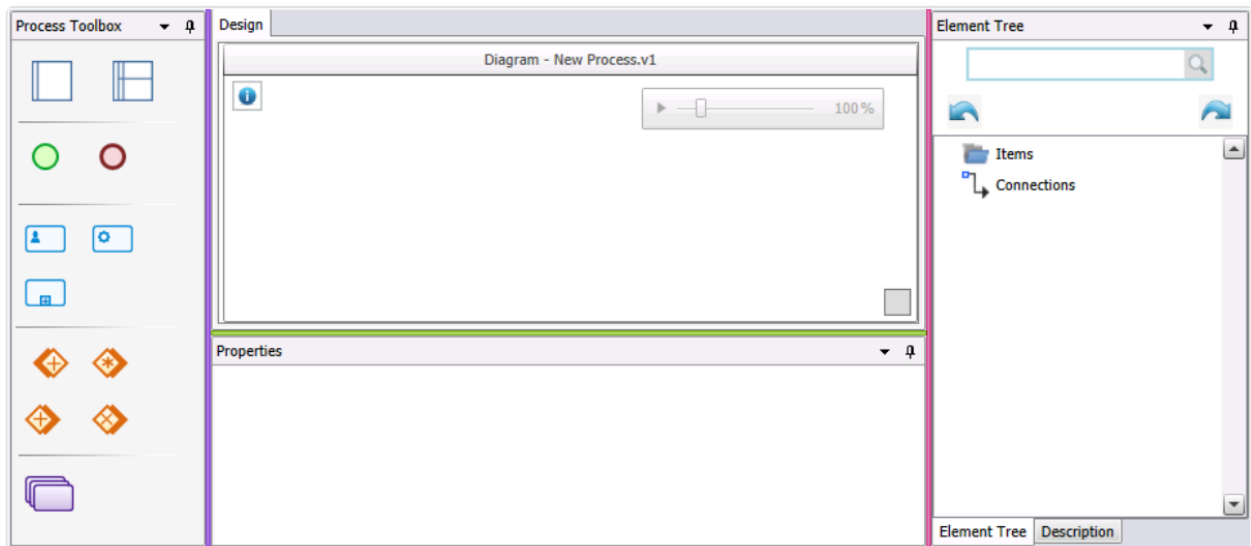
## 11.1.2.1.1. Resize Design Canvas or Panels

In the Process Designer, users can resize the design canvas and/or any pinned panels.

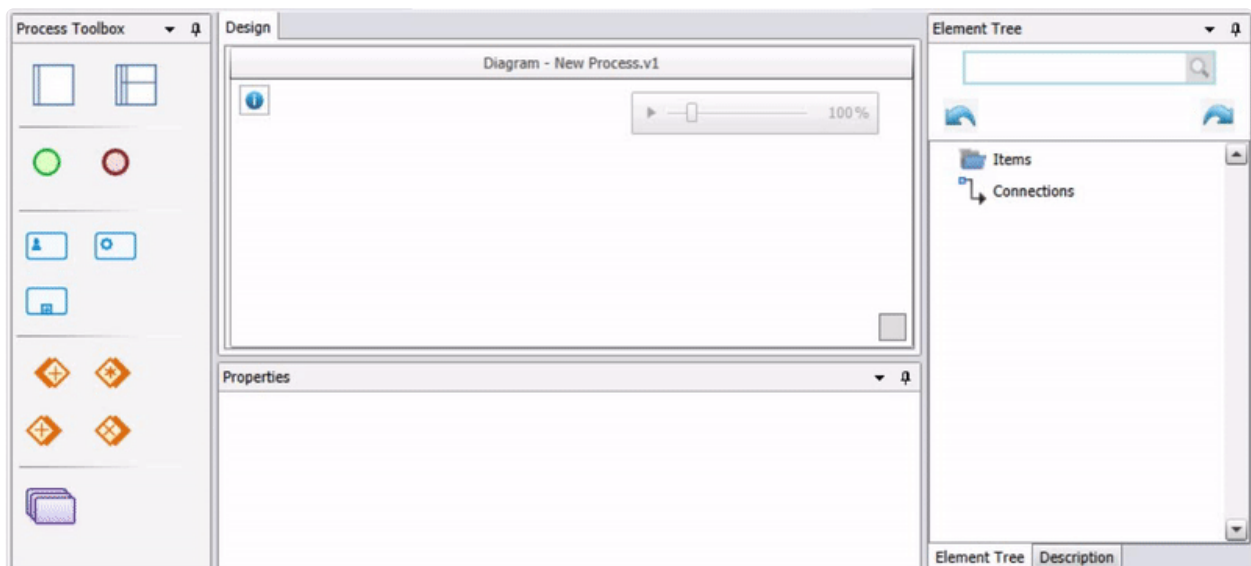
- ✿ For the purposes of this operation, it is helpful to think of the canvas and panels as cells populating the Process Designer. To resize a field, users will need to rely on the padding (or spaces) between these cells. When properly selected, the padding will be highlighted in yellow.

To resize a field, take the following steps:

- Place the cursor at the edge of one of the spaces between the canvas or panel borders (highlighted below).



- Click and drag the cursor until you get the desired size.

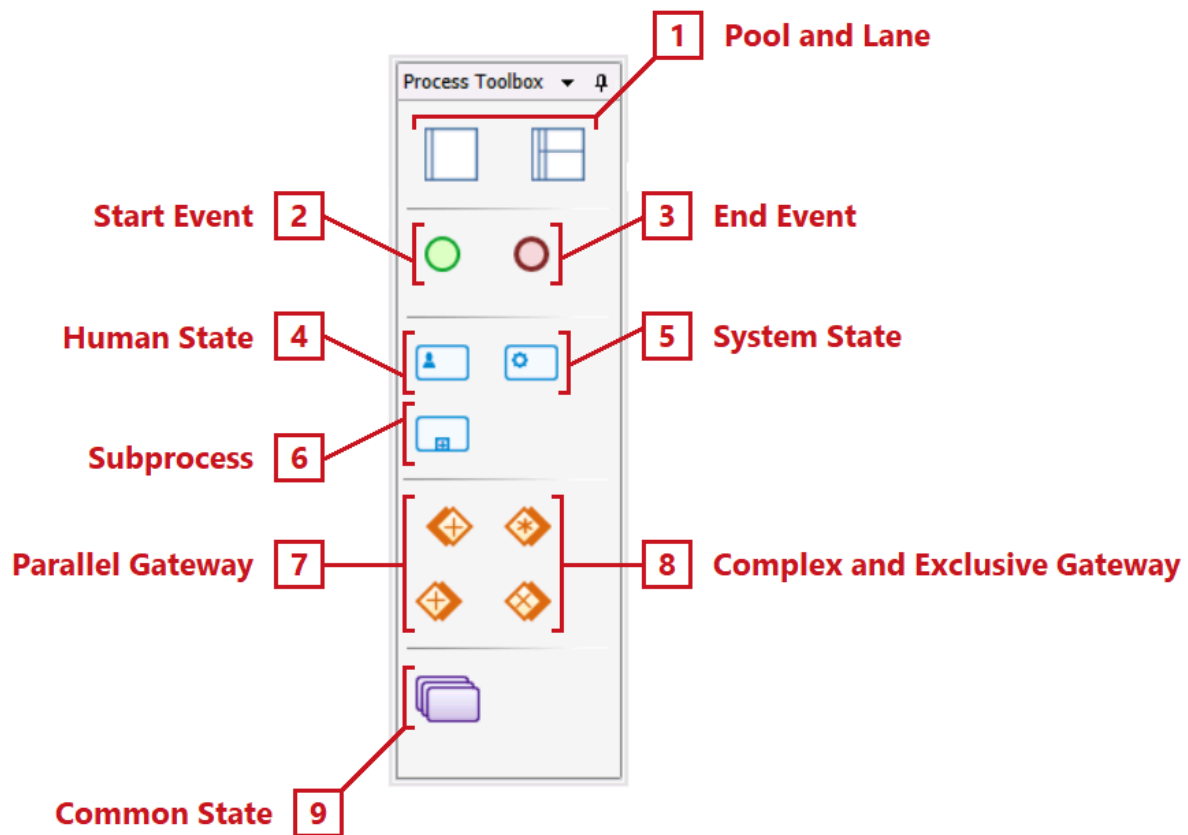




When selecting a space, make sure that the area is first highlighted in yellow before dragging the cursor. Otherwise, the resizing operation will not work correctly.

## 11.1.2.2. Process Toolbox

In the process toolbox, users can access the elements or flow objects needed to design a process.



\* Each element has its own set of properties. To view the properties for a specific element, please refer to the corresponding section listed below.

1. [Pool and Lane](#)
2. [Start Event](#)
3. [End Event](#)
4. [Human State](#)
5. [System State](#)
6. [Subprocess](#)
7. [Parallel Gateway](#)
8. [Complex and Exclusive Gateway](#)
9. [Common State](#)

\* Although not part of the toolbox itself, an essential element of any process map is the transition between flow objects. In the Designer, this transition is known as an event. For more information on events and their properties, [click here](#).

Depending on the selected process type, different features will be displayed in the process toolbox.



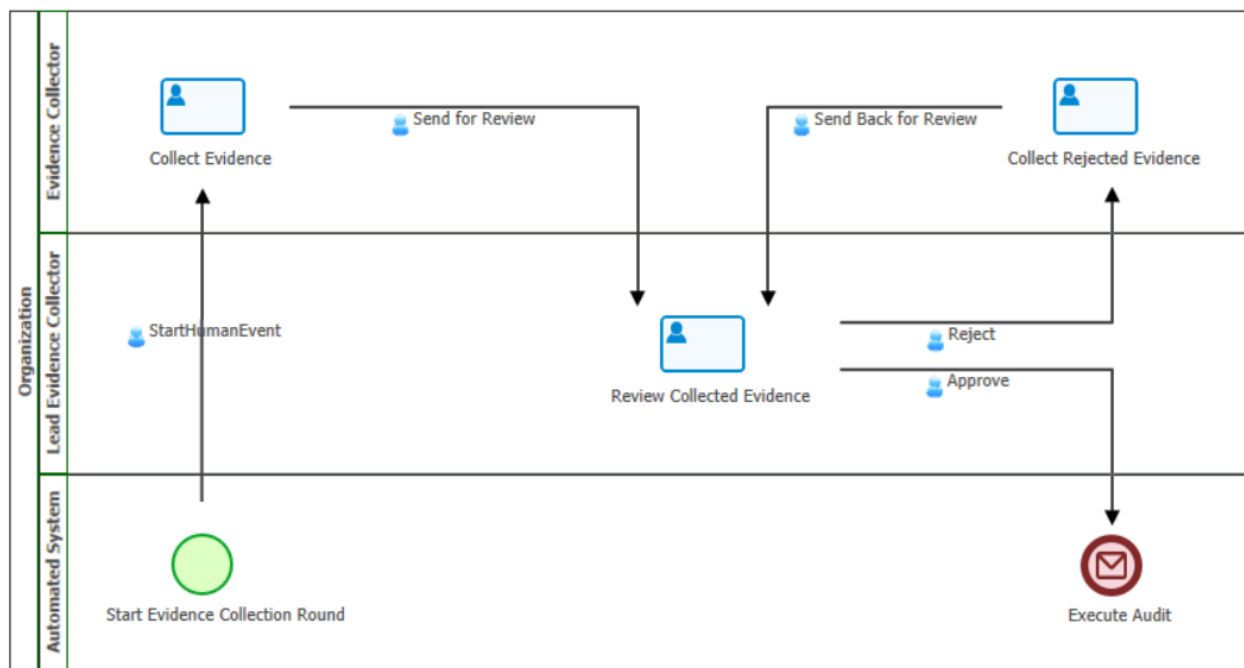
## 11.1.2.2.1. Pool and Lane

Collectively making up what is known as a “swimlane,” the pool and the lane are used to categorize and group the activities of a process. They allow users, essentially, to visually organize the steps and stakeholders of a process.

\* The term *stakeholders*, in this context, refers to the various parties involved in the workflow. Whether they affect it or are affected by it, these parties play a role in the process.

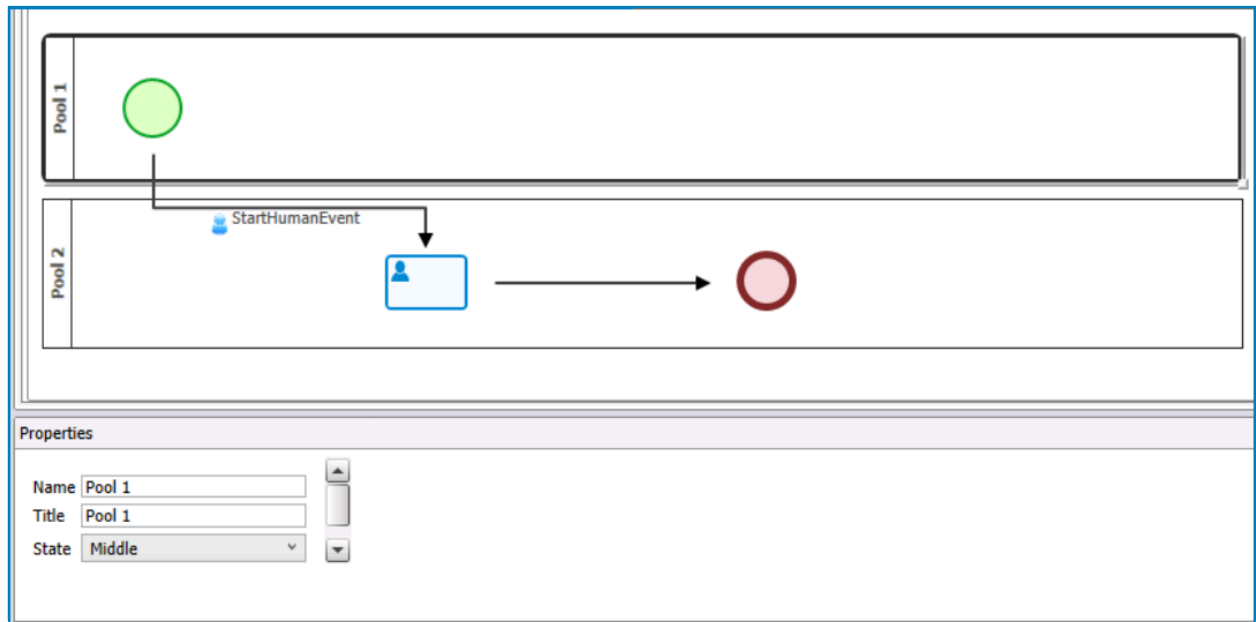
To use these two elements, first drag the **Pool** from the toolbox and drop it onto the design canvas. Next, drag the **Lane** from the toolbox and drop it onto the pool. (Users can add multiple lanes simply by dragging and dropping more onto the pool.)

\* In the Process Designer, the pool and the lane must be used together and only in this order (pool followed by lane). Neither element can be used alone.



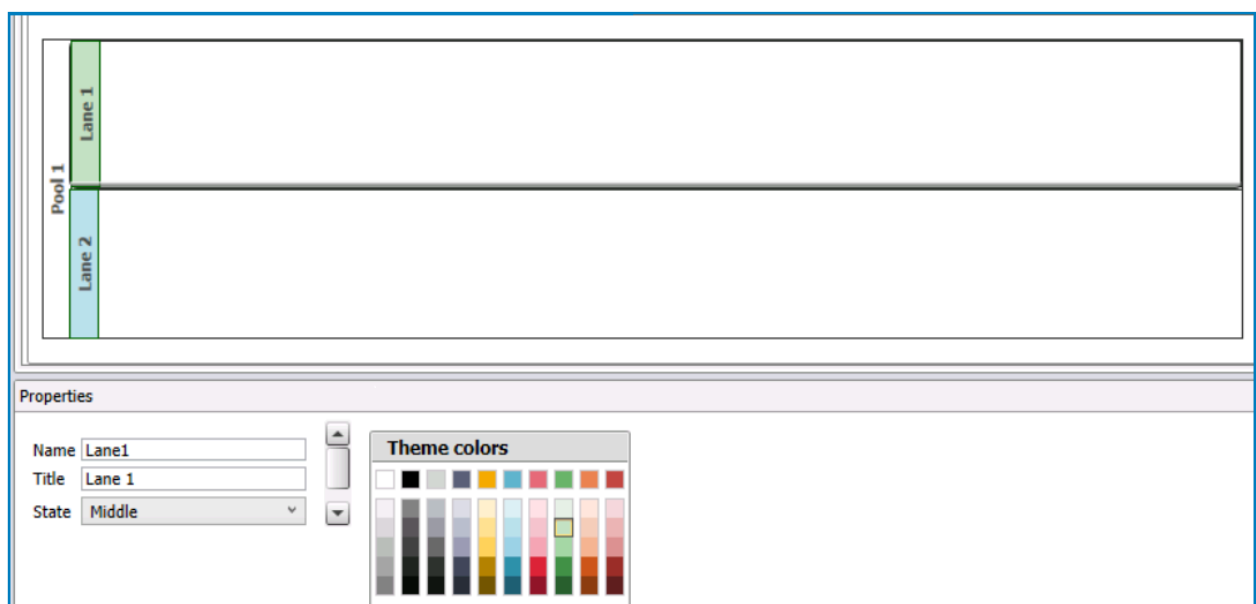
Selecting pools and/or lanes on the design canvas will display their associated properties in the Properties panel.

### 1. Pool



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **State:** This is a single-select dropdown field that displays the list of states. It allows users to select the state of the element.
  - States: Start, Middle, or End.

## 2. Lane





- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **State:** This is a single-select dropdown field that displays the list of states. It allows users to select the state of the element.
  - States: Start, Middle, or End.
- **Theme Colors:** This is a color picker that allows users to select the theme color for the lane.



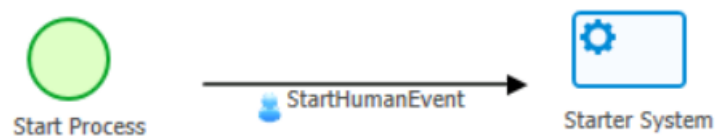
It is mandatory for users to select a name for any pools/lanes added to the diagram. In cases of non-observance, the publication will not succeed.

## 11.1.2.2.2. Start Event

All process maps require a start event—that is, a trigger that causes the process to begin.

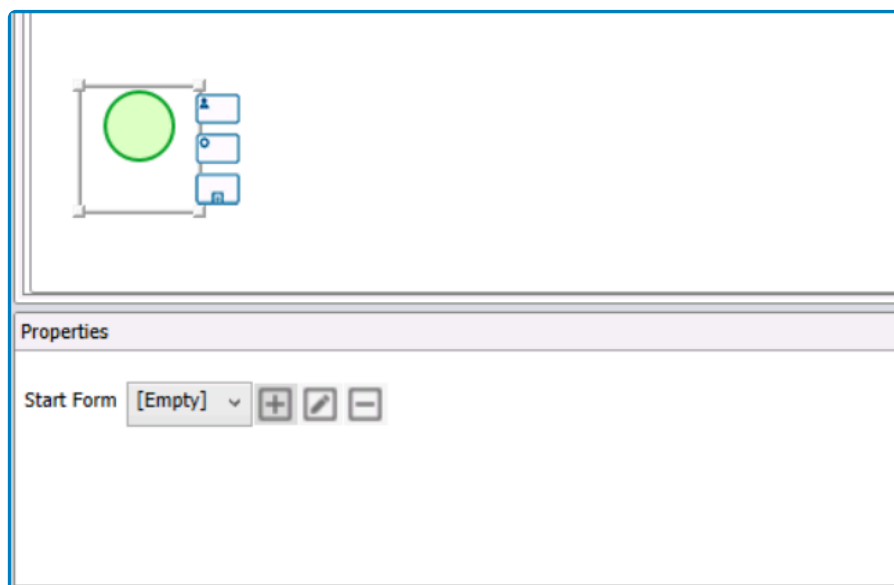
! It is mandatory for users to insert at least one start object into the diagram. Otherwise, publication will fail. Likewise, if there is more than one start object in the diagram, it is mandatory for the paths belonging to the start object to be connected using a system state/activity. Otherwise, publication will fail.

When a start object is linked to another element, the transition between them (known in the Designer as an [event](#)) is automatically defined as “StartHumanEvent” on the process map.




\* Start objects can only be linked to human states, systems states, and/or subprocesses.

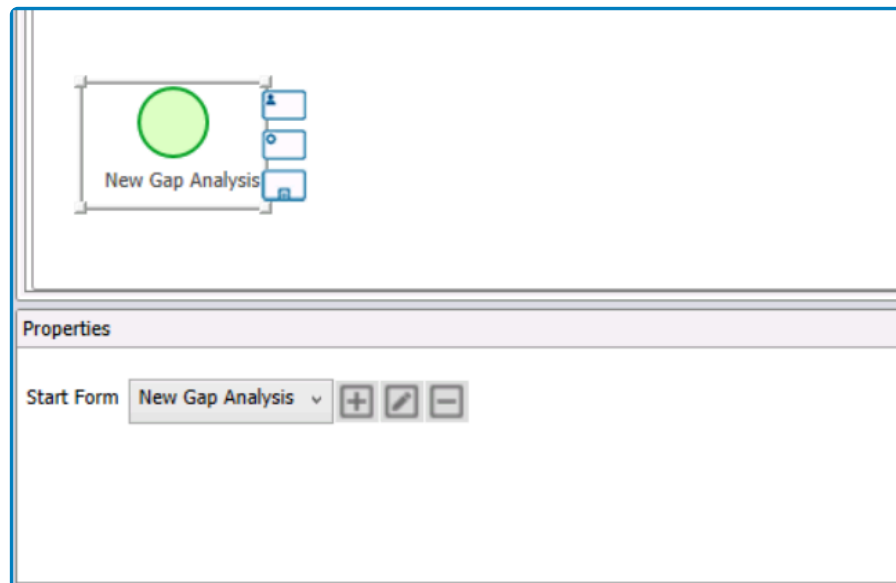
Selecting a start object on the design canvas will display its associated properties in the Properties panel.



- **Start Form:** This is a single-select dropdown field that retrieves the list of start form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to assign a start form behavior to the object.
  - To define a new start form behavior, select the button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing start form behavior, select the button.

- To delete a start form behavior, select the  button.

**Note:** Once a start form behavior has been defined/selected, its name will be displayed under the start object.



- ✿ Users must assign a start form behavior in order to determine the responsible process initiator, the physical form, and the location of the form in the menu.

## 11.1.2.2.3. End Event

All process maps conclude with an end event—that is, a final stage signaling the completion of the workflow.

! It is mandatory for users to insert at least one end object into the diagram. Otherwise, publication will fail.

Selecting an end object on the design canvas will display its associated properties in the Properties panel.

The screenshot shows the 'End Process' event in the DBP Designer. The event is represented by a red circle icon with the text 'End Process' below it. Below the icon is the 'Properties' panel. The 'Properties' panel has the following fields and controls:

- Name:** A text field containing 'End\_Process'.
- Title:** A text field containing 'End Process'.
- Data Item:** A dropdown menu showing '[Empty]' with three buttons: '+', a pencil icon, and a minus icon.
- Rule:** A dropdown menu showing '[Empty]' with the same three buttons: '+', a pencil icon, and a minus icon.
- Notifications:** A button with a calendar icon.
- Active:** A checkbox.

- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Data Item:** This is a single-select dropdown field that retrieves the list of data items (from the [Task Data Items](#) section of the **Notification** module). It allows users to assign a data item to the object.
  - To define a new data item, select the button.
    - It will display the [Data Item Properties](#) window.
  - To modify an existing data item, select the button.
  - To delete a data item, select the button.

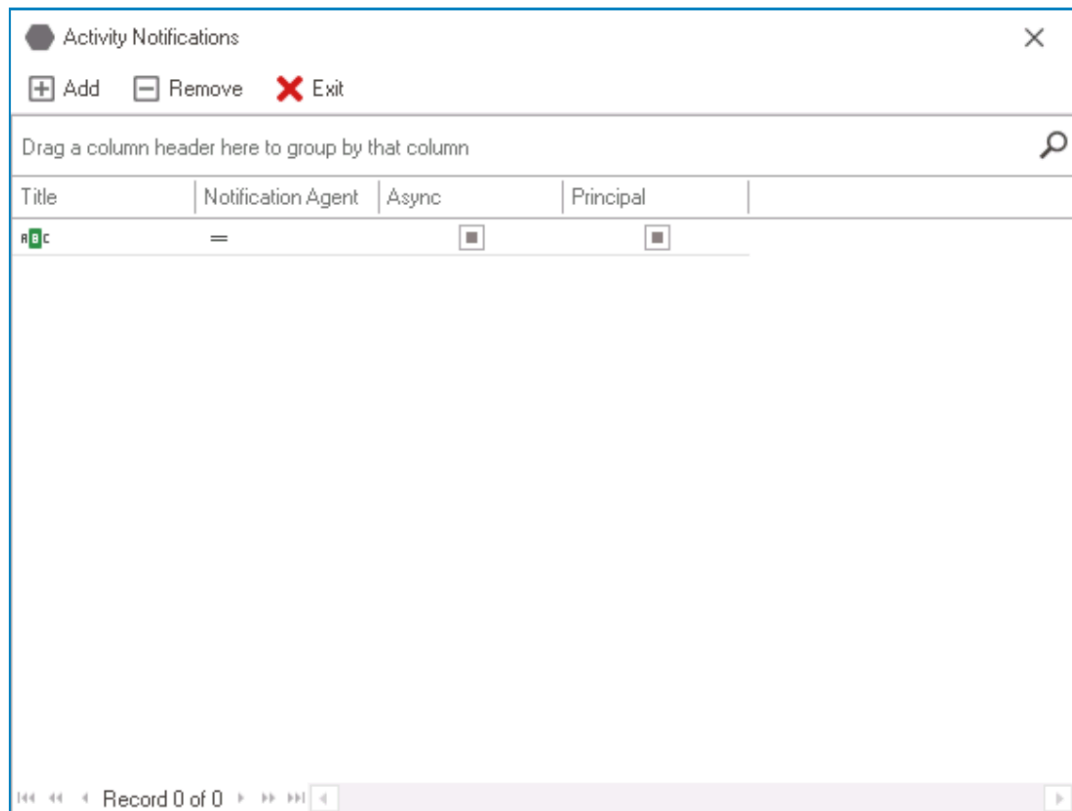
\* By selecting or defining a data item in this field, in the history and process interception,

the text message inside parameter “num” is displayed. The parameter is defined inside the template.

- **Active:** This is a checkbox which, if selected, allows users to mark out the object as active.
  - The term *active*, in this context, means that notifications have been enabled for the activity.
  - Selecting this checkbox will display a small envelope on the object in the diagram, signifying that a notification will be sent at that stage.



- **Notifications:** This button allows users to define the required notifications for the activity. It is only enabled once the **Active** checkbox has been selected and the process has been published.
  - Once selected, it will display the Activity Notifications window.






- **Add:** This button allows users to add a new notification.
  - Selecting this button will display the Notifications window, in which users can either define new notifications or select existing ones (if applicable) from the list.

- **New:** This button allows users to create a new notification item.
  - It will display the [Notification Item Properties](#) form.
- **Delete:** This button allows users to delete a notification item.
- **Edit:** This button allows users to edit an existing notification item.
- **Select:** This button allows users to select a notification item (to be triggered during the process).
  - Selecting this button will redirect users back to the [Activity Notifications](#) window—the notification item now listed therein.
    - Users can also select a notification item by double-clicking on it.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the [Notifications](#) window..
- **Remove:** This button allows users to remove a notification.
- **Exit:** This button allows users to exit the [Activity Notifications](#) window.



These notifications, once set up, will then be executed at the applicable stage in the process.

- **Rule:** This is a single-select dropdown field that retrieves the list of rules (from the [Rules](#) section of the **Business Logics** module). It allows users to assign a rule to the object.
  - To define a new rule, select the  button.
    - It will display the [Rule Properties](#) window.
  - To modify an existing rule, select the  button.
  - To delete a rule, select the  button.



This rule, once set up, will then be executed per its conditions at the applicable stage in the process.



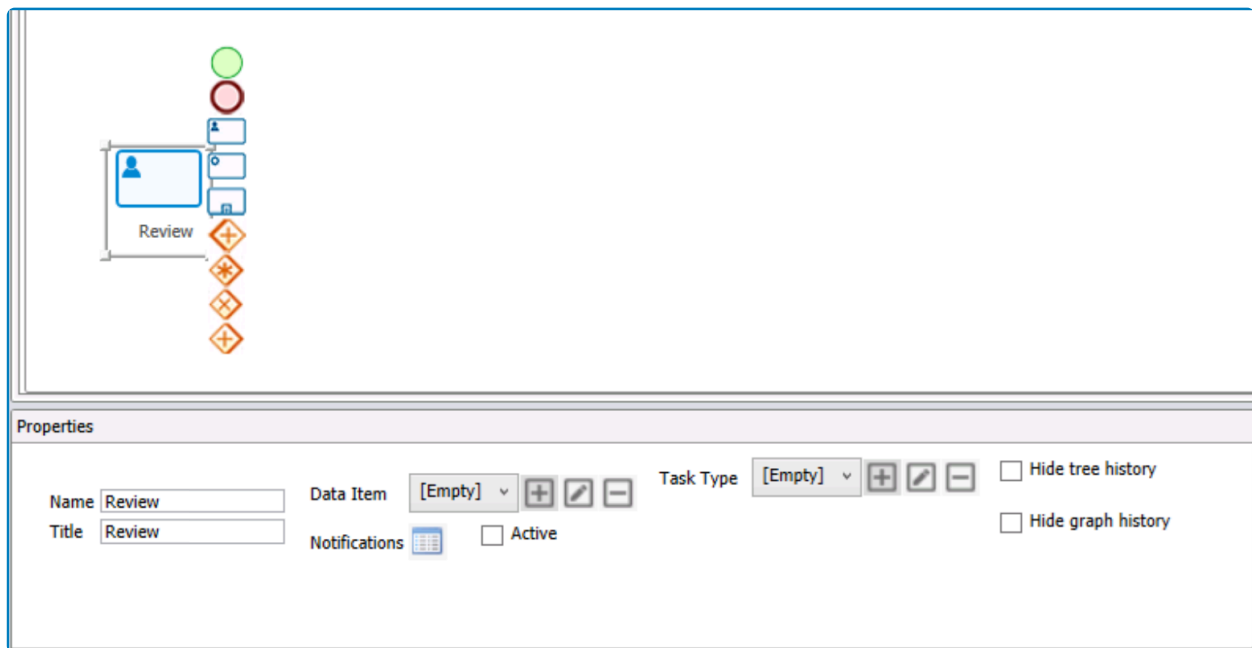
It is mandatory for users to select a name for any end events added to the diagram. In cases of non-observance, the publication will not succeed.

## 11.1.2.2.4. Human State

If individuals play a role in the execution of the process, use the human state/activity.

- \* The individuals—users, participants, etc.—involved in the workflow are determined either among the responsible staff or inside the task type properties.

Selecting a human activity on the design canvas will display its associated properties in the Properties panel.



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Data Item:** This is a single-select dropdown field that retrieves the list of data items (from the [Task Data Items](#) section of the **Notification** module). It allows users to assign a data item to the object.
  - To define a new data item, select the button.
    - It will display the [Data Item Properties](#) window.
  - To modify an existing data item, select the button.
  - To delete a data item, select the button.
- **Active:** This is a checkbox which, if selected, allows users to mark out the object as active.
  - The term *active*, in this context, means that notifications have been enabled for the activity.

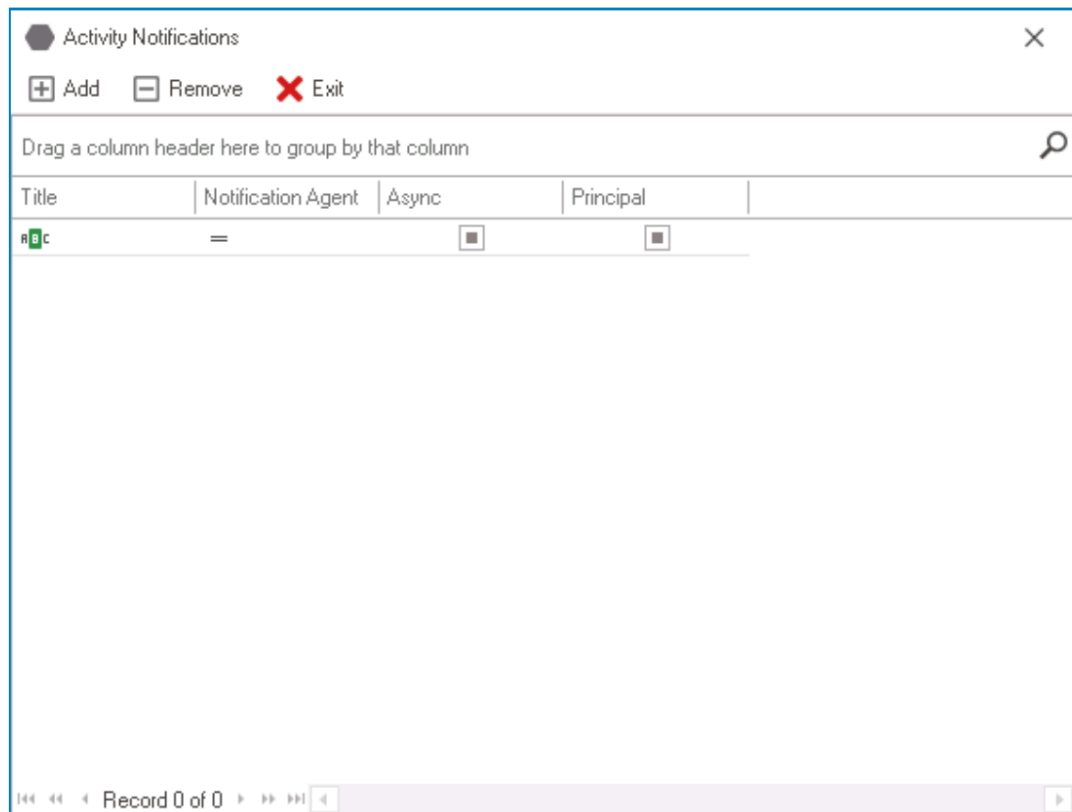


- Selecting this checkbox will display a small envelope on the object in the diagram, signifying that a notification will be sent at that stage.



Review

- **Notifications:** This button allows users to define the required notifications for the activity. It is only enabled once the **Active** checkbox has been selected and the process has been published.
  - Once selected, it will display the Activity Notifications window.






The screenshot shows the 'Activity Notifications' window. It has a title bar with a close button. Below the title bar are three buttons: 'Add' (with a plus icon), 'Remove' (with a minus icon), and 'Exit' (with a red X icon). Below these buttons is a search bar with the placeholder text 'Drag a column header here to group by that column' and a magnifying glass icon. Below the search bar is a table with the following columns: 'Title', 'Notification Agent', 'Async', and 'Principal'. The first row of the table contains the text 'RUC', '=', and two checkboxes. Below the table is a large empty area. At the bottom of the window is a status bar with the text 'Record 0 of 0' and navigation icons.

- **Add:** This button allows users to add a new notification.
  - Selecting this button will display the Notifications window, in which users can either define new notifications or select existing ones (if applicable) from the list.

- **New:** This button allows users to create a new notification item.
  - It will display the [Notification Item Properties](#) form.
- **Delete:** This button allows users to delete a notification item.
- **Edit:** This button allows users to edit an existing notification item.
- **Select:** This button allows users to select a notification item (to be triggered during the process).
  - Selecting this button will redirect users back to the [Activity Notifications](#) window—the notification item now listed therein.
    - Users can also select a notification item by double-clicking on it.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the [Notifications](#) window..
- **Remove:** This button allows users to remove a notification.
- **Exit:** This button allows users to exit the [Activity Notifications](#) window.



These notifications, once set up, will then be executed at the applicable stage in the process.

- **Task Type:** This is a single-select dropdown field that retrieves the list of task types (from the [Task Types](#) section of **Human Activities**). It allows users to assign a task type to the object.
  - To define a new task type, select the  button.
    - It will display the [Task Type Properties](#) window.
  - To modify an existing task type, select the  button.
  - To delete a task type, select the  button.

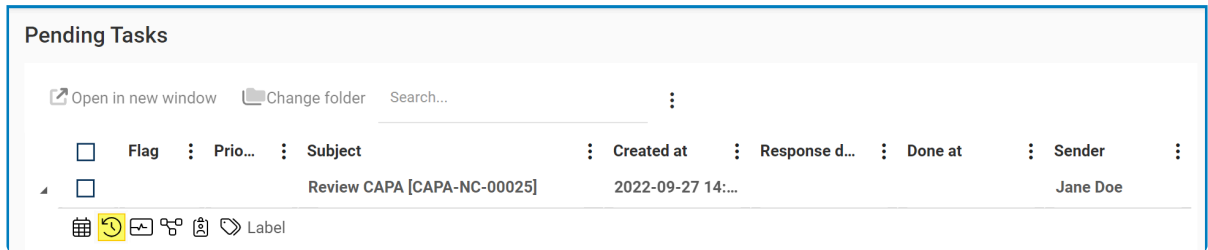


The task type determines the task for the human state, the form behavior, the receiver(s) of the form, and the dynamic task type.

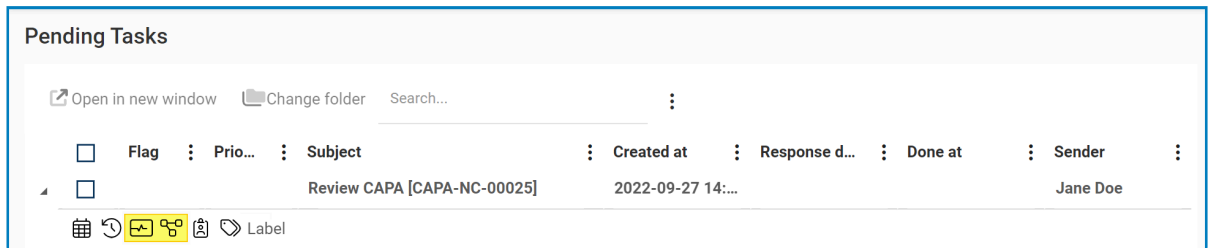
- **Hide Tree History:** This is a checkbox which, if selected, allows users to hide this stage of the

process from the tree history.

- As a result, in the **DBP** web application, the [History Tree](#) icon will not be shown for the task.



- **Hide Graph History:** This is a checkbox which, if selected, allows users to hide this stage of the process from the graph history.
  - As a result, in the **DBP** web application, both the [History Graph](#) and [Process History](#) icons will not be shown for the task.



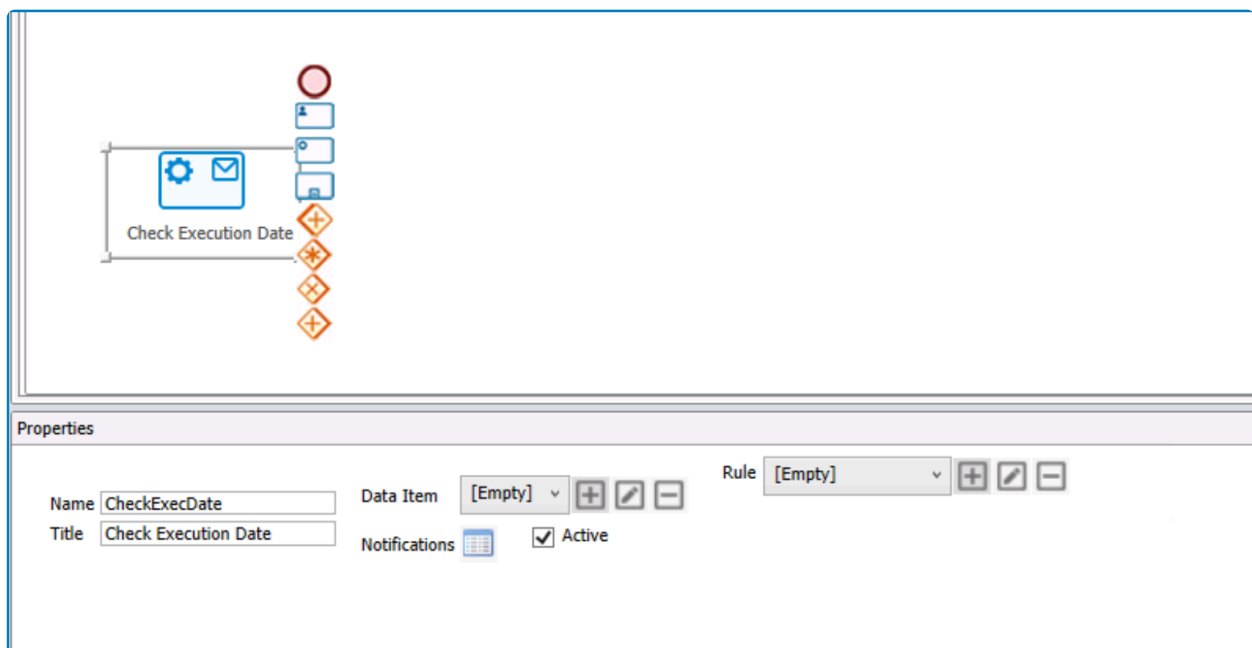
It is mandatory for users to select a name and a task type for any human states added to the diagram. In cases of non-observance, the publication will not succeed.




## 11.1.2.2.5. System State

If a process requires system rule enforcement, use the system state/activity.

- ✳ When transitioning from a system state to another object in the process, users are limited to three event types: auto, system, and scheduled. For more details on events and their types, [click here](#).

Selecting a system activity on the design canvas will display its associated properties in the Properties panel.



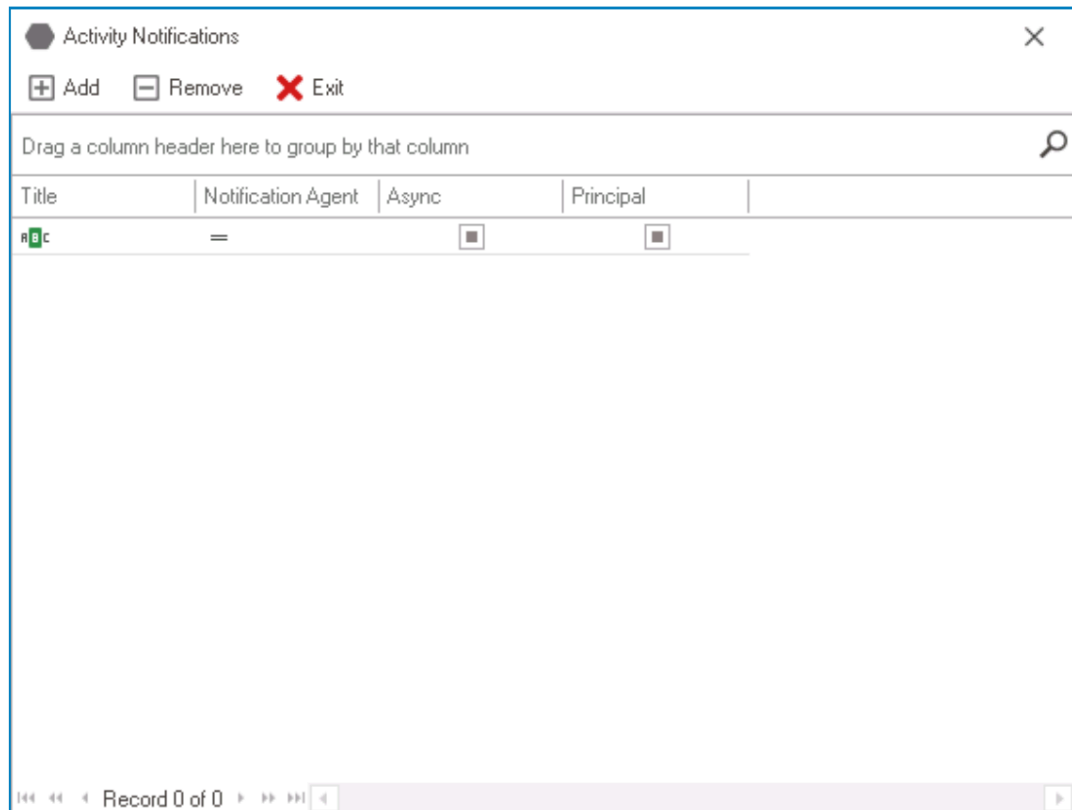
- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Data Item:** This is a single-select dropdown field that retrieves the list of data items (from the [Task Data Items](#) section of the **Notification** module). It allows users to assign a data item to the object.
  - To define a new data item, select the  button.
    - It will display the [Data Item Properties](#) window.
  - To modify an existing data item, select the  button.
  - To delete a data item, select the  button.
- **Active:** This is a checkbox which, if selected, allows users to mark out the object as active.

- The term *active*, in this context, means that notifications have been enabled for the activity.
- Selecting this checkbox will display a small envelope on the object in the diagram, signifying that a notification will be sent at that stage.



Check Execution Date

- **Notifications:** This button allows users to define the required notifications for the activity. It is only enabled once the **Active** checkbox has been selected and the process has been published.
  - Once selected, it will display the Activity Notifications window.






- **Add:** This button allows users to add a new notification.
  - Selecting this button will display the Notifications window, in which users can either define new notifications or select existing ones (if applicable) from the list.

- **New:** This button allows users to create a new notification item.
  - It will display the [Notification Item Properties](#) form.
- **Delete:** This button allows users to delete a notification item.
- **Edit:** This button allows users to edit an existing notification item.
- **Select:** This button allows users to select a notification item (to be triggered during the process).
  - Selecting this button will redirect users back to the [Activity Notifications](#) window—the notification item now listed therein.
    - Users can also select a notification item by double-clicking on it.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the [Notifications](#) window..
- **Remove:** This button allows users to remove a notification.
- **Exit:** This button allows users to exit the [Activity Notifications](#) window.



These notifications, once set up, will then be executed at the applicable stage in the process.

- **Rule:** This is a single-select dropdown field that retrieves the list of rules (from the [Rules](#) section of the **Business Logics** module). It allows users to assign a rule to the object.
  - To define a new rule, select the  button.
    - It will display the [Rule Properties](#) window.
  - To modify an existing rule, select the  button.
  - To delete a rule, select the  button.



This rule, once set up, will then be executed per its conditions at the applicable stage in the process.



It is mandatory for users to select a name for any system states added to the diagram. In cases of non-observance, the publication will not succeed.

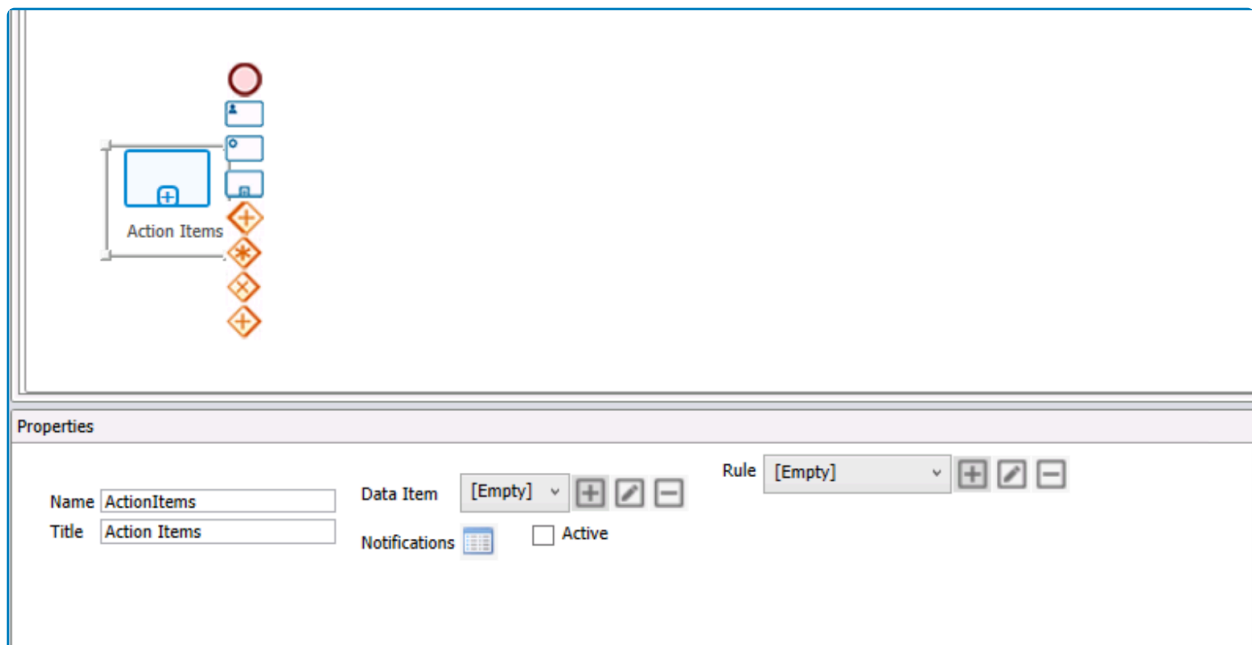
## 11.1.2.2.6. Subprocess


If you need to call upon another process during the workflow, use the subprocess.

\* Calling upon a process requires calling upon its API in the rule attributed to the subprocess. [For more details on APIs and processes, click here.](#)



\* When transitioning from a subprocess to another object in the process, users are limited to three event types: auto, system, and scheduled. [For more details on events and their types, click here.](#)

Selecting a subprocess on the design canvas will display its associated properties in the Properties panel.



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Data Item:** This is a single-select dropdown field that retrieves the list of data items (from the [Task Data Items](#) section of the **Notification** module). It allows users to assign a data item to the object.
  - To define a new data item, select the  button.

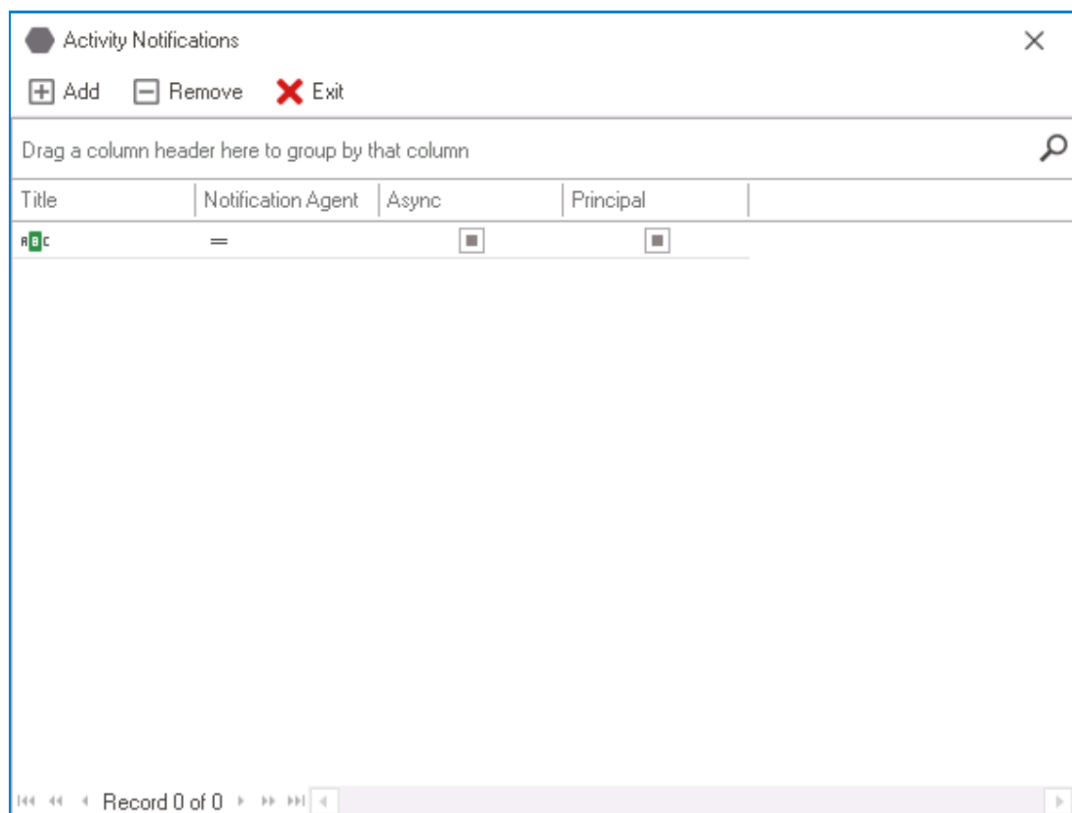


- It will display the [Data Item Properties](#) window.
- To modify an existing data item, select the  button.
- To delete a data item, select the  button.
- **Active:** This is a checkbox which, if selected, allows users to mark out the object as active.
  - The term *active*, in this context, means that notifications have been enabled for the activity.
  - Selecting this checkbox will display a small envelope on the object in the diagram, signifying that a notification will be sent at that stage.



Action Items

- **Notifications:** This button allows users to define the required notifications for the activity. It is only enabled once the **Active** checkbox has been selected and the process has been published.
  - Once selected, it will display the [Activity Notifications](#) window.






- **Add:** This button allows users to add a new notification.
  - Selecting this button will display the [Notifications](#) window, in which users can either define new notifications or select existing ones (if applicable) from the list.

- **New:** This button allows users to create a new notification item.
  - It will display the [Notification Item Properties](#) form.
- **Delete:** This button allows users to delete a notification item.
- **Edit:** This button allows users to edit an existing notification item.
- **Select:** This button allows users to select a notification item (to be triggered during the process).
  - Selecting this button will redirect users back to the [Activity Notifications](#) window—the notification item now listed therein.
    - Users can also select a notification item by double-clicking on it.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the [Notifications](#) window..
- **Remove:** This button allows users to remove a notification.
- **Exit:** This button allows users to exit the [Activity Notifications](#) window.



These notifications, once set up, will then be executed at the applicable stage in the process.

- **Rule:** This is a single-select dropdown field that retrieves the list of rules (from the [Rules](#) section of the **Business Logics** module). It allows users to assign a rule to the object.
  - To define a new rule, select the  button.
    - It will display the [Rule Properties](#) window.
  - To modify an existing rule, select the  button.
  - To delete a rule, select the  button.



This rule, once set up, will then be executed per its conditions at the applicable stage in the process.



It is mandatory for users to select a name for any subprocesses added to the diagram. In cases of non-observance, the publication will not succeed.

## 11.1.2.2.7. Parallel Gateway

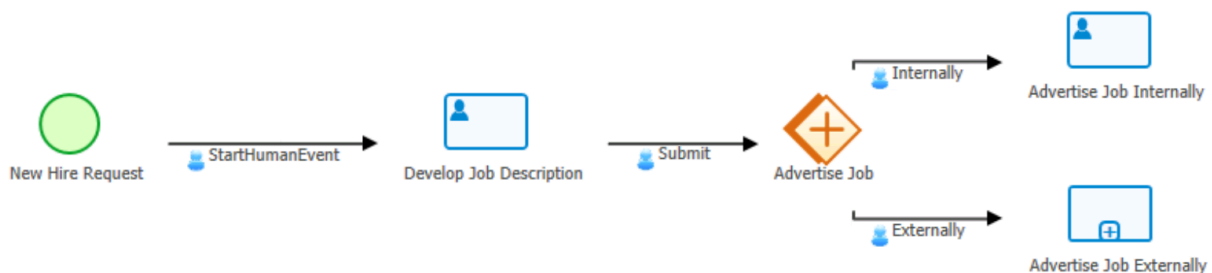
If multiple tasks need to be executed concurrently during the workflow, use the parallel gateway.

! A gateway is used to control the divergence and convergence of multiple flow objects. It can have both fork and join behavior. In the Process Designer, it is mandatory that the number of forks and joins be equal. In other words, the number of branches exiting the fork must be the same as the number of branches entering the join. Otherwise, publication will fail.

There are two types of parallel gateways in the toolbox: output parallel gateway and input parallel gateway.

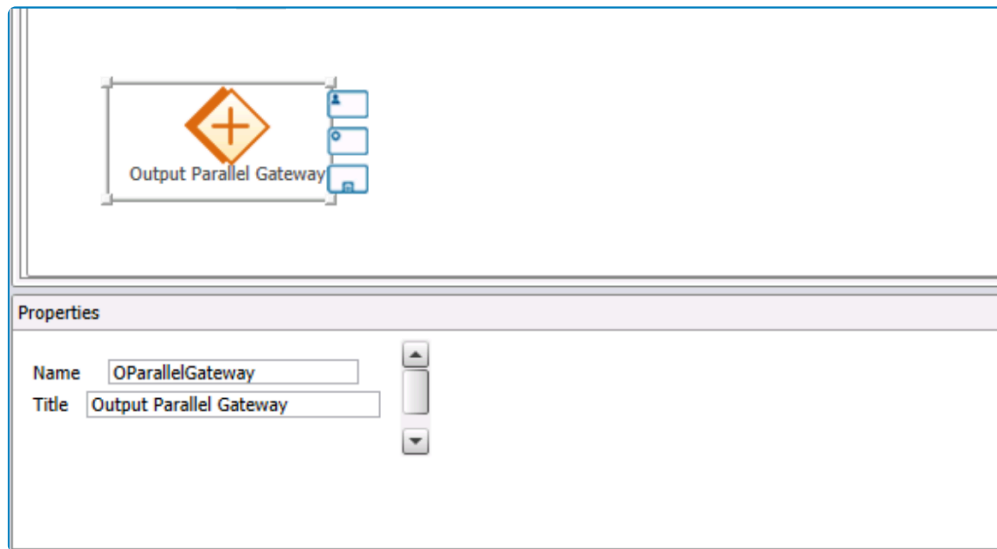
### 1. Output Parallel Gateway

An output parallel gateway is used to fork the process into multiple, concurrent paths of execution. It can only be linked to human tasks, system tasks, and subprocesses.



\* When a parallel gateway is in use, the process continues only when all of its branches have been executed

Selecting an output parallel gateway on the design canvas will display its associated properties in the Properties panel.

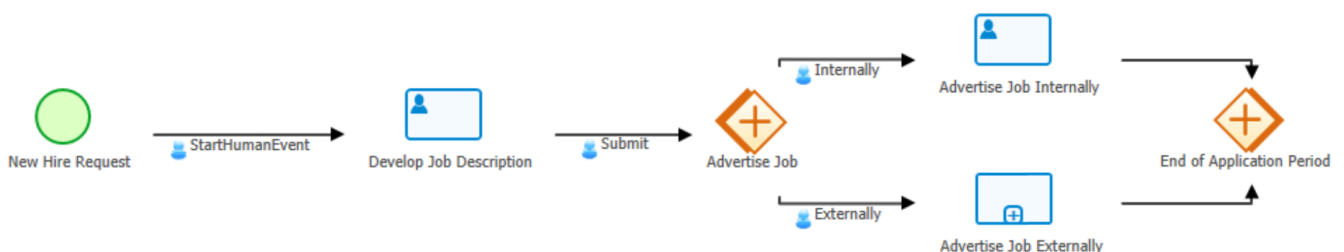


- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.

**!** If the output parallel gateway is used during the process, users must regroup the flow by using either the complex, exclusive, or input parallel gateways.

## 2. Input Parallel Gateway

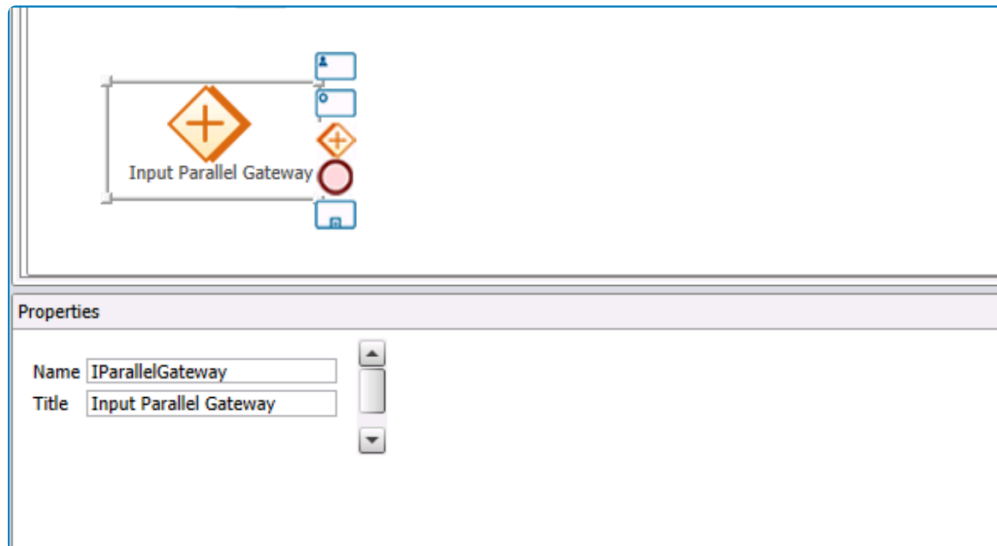
An input parallel gateway is used to join back together multiple incoming paths of execution.



**✿** When a parallel gateway is in use, the process continues only when all of its branches have been executed

Selecting an input parallel gateway on the design canvas will display its associated properties in the

Properties panel.



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.

✿ When transitioning from a parallel gateway to another object in the process, users are limited to three event types: auto, system, and scheduled. **For more details on events and their types, [click here](#).**

! It is mandatory for users to select a name for any gateways added to the diagram. In cases of non-observance, the publication will not succeed.

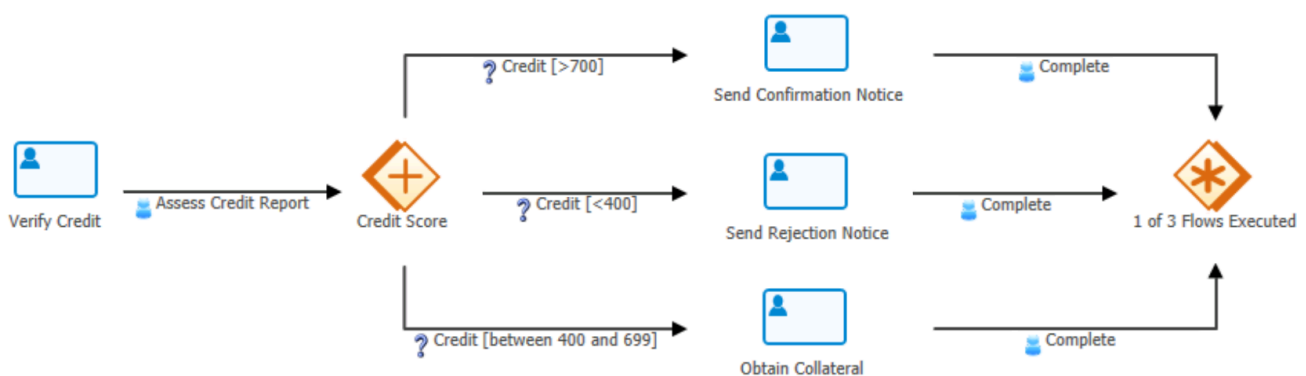
## 11.1.2.2.8. Complex and Exclusive Gateway

Apart from parallel gateways, there are two other types of gateways included in the toolbox: complex gateways and exclusive gateways.

! A gateway is used to control the divergence and convergence of multiple flow objects. It can have both fork and join behavior. In the Process Designer, it is mandatory that the number of forks and joins be equal. In other words, the number of branches exiting the fork must be the same as the number of branches entering the join. Otherwise, publication will fail.

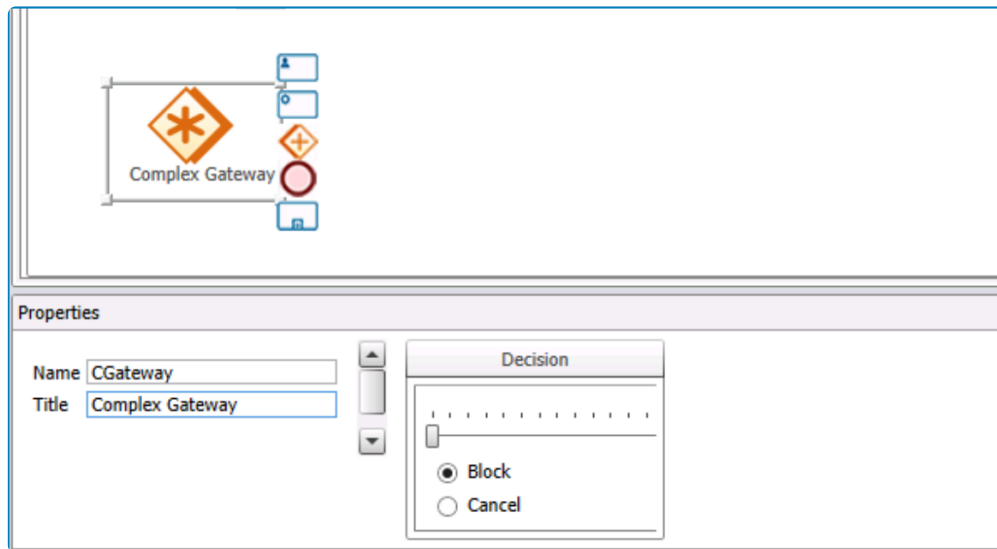
### 1. Complex Gateway

A complex gateway is used to divide execution when the possible paths are more complex. When a process requires for multiple decisions to be made in quick succession, a complex gateway allows users to simplify these complex decision or divergence points.



\* These divergence points typically cannot be applied or efficiently reflected using other gateways. A complex gateway acts as a simplification tool, enabling users to shrink the process map and streamline the model workflow.

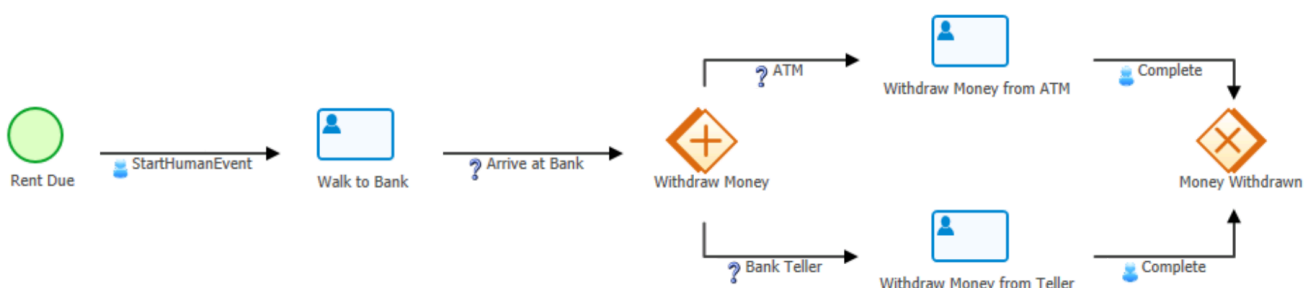
Selecting a complex gateway on the design canvas will display its associated properties in the Properties panel.



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Decision:** This is a slider that allows users to define the minimum number of branches that need to be executed before the process can continue to the next stage.
  - When the number of branches reaches the specified value, the status of existing tasks in other branches can be defined using the radio button:
    - **Block:** This option, if selected, will cause for the tasks to remain in the same state. That is to say, they will remain in the end-user's list of tasks and the end-users will be required to complete them.
    - **Cancel:** This option, if selected, will cause for the existing tasks to be removed from the user's list of tasks.

## 2. Exclusive (XOR-Join) Gateway

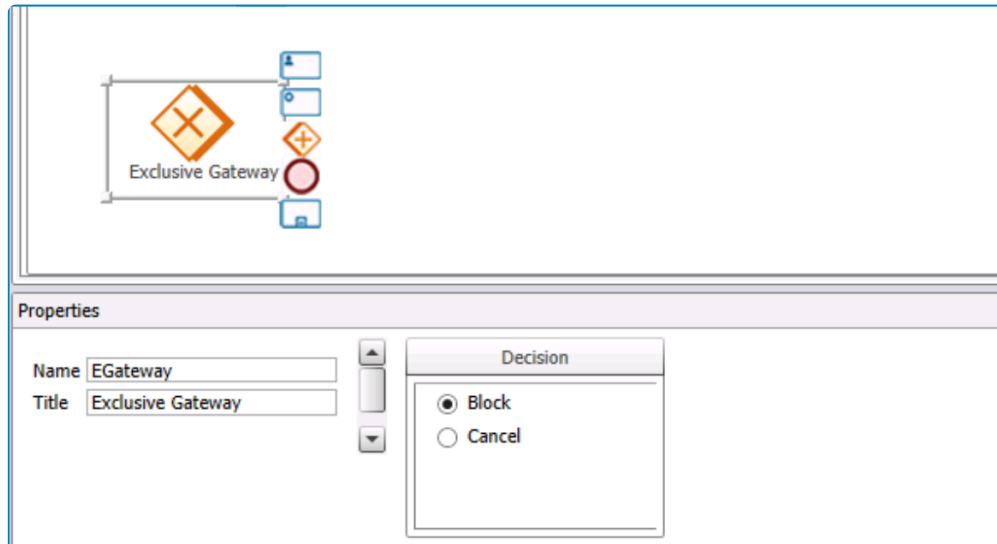
An exclusive gateway is used to divide the process into one of two or more mutually exclusive paths.





- ✿ When an exclusive gateway (also known as an XOR-Join state) is in use, the process will continue by finishing only one of its branches.

Selecting an exclusive gateway on the design canvas will display its associated properties in the Properties panel.



- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Block:** This is a checkbox which, if selected, causes for the tasks to remain in the same state. That is to say, they will remain in the end-user's list of tasks and the end-users will be required to complete them.
- **Cancel:** This is a checkbox which, if selected, causes for the existing tasks to be removed from the user's list of tasks.

- ✿ When transitioning from either a complex or exclusive gateway to another object in the process, users are limited to three event types: auto, system, and scheduled. [For more details on events and their types, click here.](#)

! It is mandatory for users to select a name for any gateways added to the diagram. In cases of non-observance, the publication will not succeed.

## 11.1.2.2.9. Common State

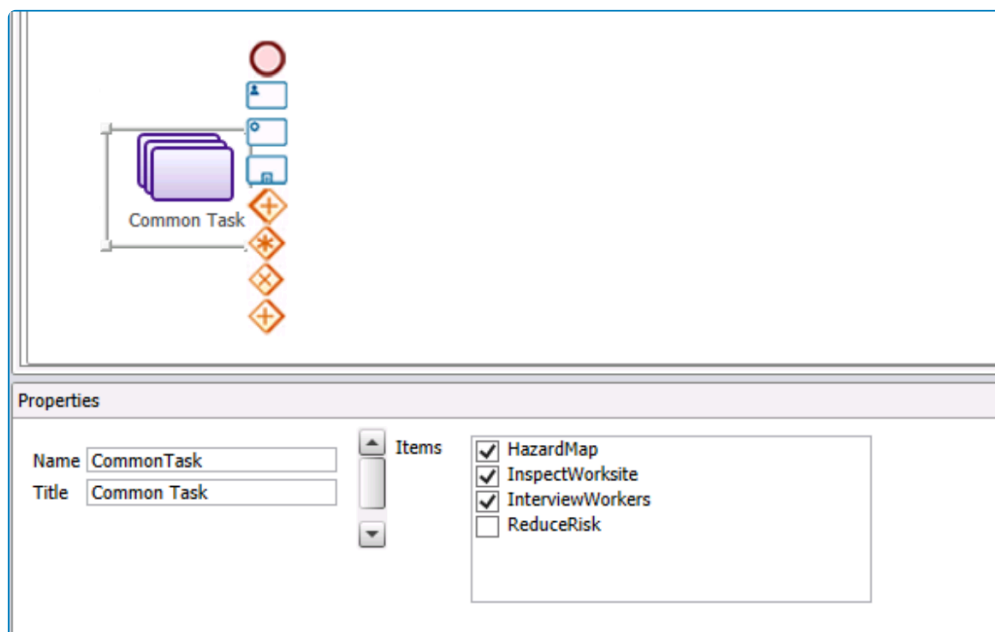
If an event ultimately causes the same result to occur in different parts of the process, use the common state/activity to make the process clearer.

✿ A common state is typically used to simplify and streamline the process. When applied following an event that repeatedly brings about the same result, the common state allows the process to flow in a more smooth and orderly manner.

Placing a common activity and establishing a link between it and the beginning of the repetitive path in the process is equivalent to connecting that event to all of the selected items in the common activity. For more information on these items, see the common activity properties (described below).

! When establishing this relationship between common states and other objects, it is important to note the direction of the link. An event must connect *from* the common state to the applicable object—not vice versa. A human state, for example, cannot be linked to a common state, but a common state can be linked to a human state.

Selecting a common activity on the design canvas will display its associated properties in the Properties panel.



✿ A common state cannot be linked to a start object or another common state.

- **Name:** This is a text field that allows users to input the name for the element.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a

number.

- An element must have a unique name. It cannot have the same name as another element in the process.
  - Users can only input up to a maximum of 17 characters in this field.
- **Title:** This is a text field that allows users to input the title for the element. This title will be displayed in the diagram.
  - Unlike with names, there are no rules determining the validity of titles.
- **Items:** This field displays the list of states and/or subprocesses currently being used in the process.
  - If there are no states or subprocesses in the diagram, this field will be empty.
    - Items listed inside this field are identified by their name (rather than their title).
      - In the Process Designer, items are also sometimes referred to as activities.
  - By selecting an item in this field, users are essentially connecting the following event in the process to it.



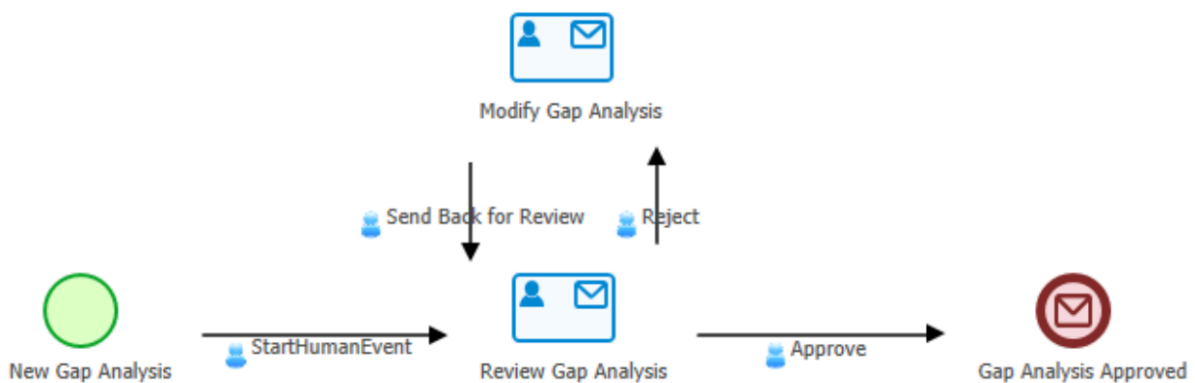
It is mandatory for users to select a name for any common states added to the diagram. These common states must also have items assigned to them. In cases of non-observance, the publication will not succeed.

## 11.1.2.2.10. Event

An essential part of designing a process involves connecting its different elements. In the Designer, these connections (or transitions) between elements are known as events.

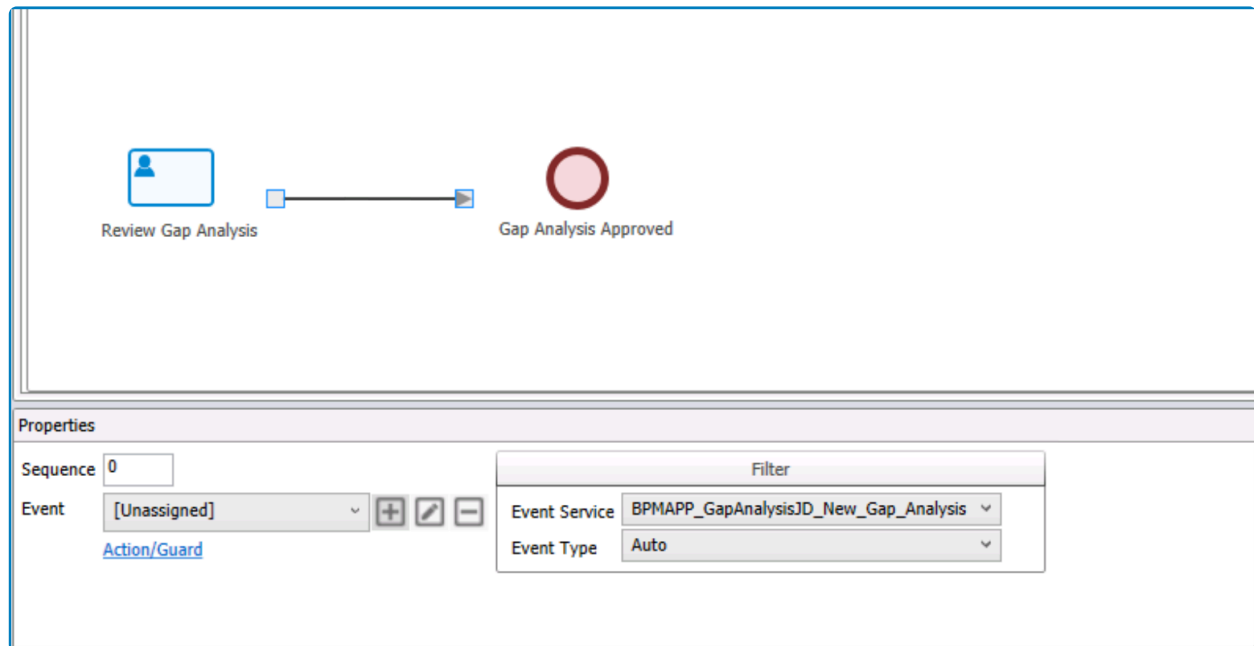
- \* An event is essentially a type of action that occurs during the execution of a process. By acting as point of communication between flow objects, an event enables the process to move from one stage to another. It is represented in the diagram by the transition arrow (→) between the flow objects.

As a symbol in the diagram, an event functions to indicate the direction or sequence of the process.



- \* Users can insert an event in the diagram either automatically or manually. For more details on how to insert and adjust events in the diagram, [click here](#).

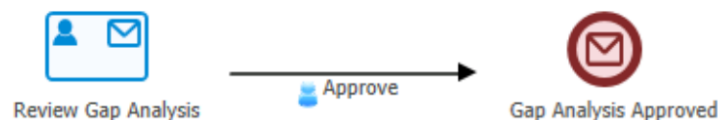
Selecting an event on the design canvas will display its associated properties in the Properties panel.



1. **Sequence:** This is a numeric field that allows user to input the order of the event in the sequence.
  - When the **Event Order** button of the [menu bar](#) is selected, it will use the values inputted in these fields to arrange the sequence of events in the process.

✿ If a task or activity has been linked to multiple events, the sequence of the buttons on the form can be set to direct the workflow to a particular event.

2. **Event:** This is a single-select dropdown field that displays the list of events (from the [Events](#) section of the **Processes** module). It allows users to assign an event to the transition.
  - Depending on the selected event service and type, the items listed in this dropdown may differ.
    - To define a new event, select the button.
      - It will display the [Event Properties](#) window.
    - To modify an existing event, select the button.
    - To delete an event, select the button.
  - Once an event has been defined/selected, its name will be displayed under the transition arrow.



3. **Action / Guard:** This is a hyperlink which, if selected, directs users to the [Guard](#) window, in which users can define new event actions and guards.

- **New Event Action:** This button allows users to add a new event action.
  - If selected, it will display the [Action Properties](#) window.
- **New Guard:** This button allows users to add a new guard.
  - If selected, it will display the [Guard Properties](#) window.
- **Delete:** This button allows users to delete an event action or guard.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the window.
  - If selected, users will be returned to the [Process Designer](#).
- For more information on the other features in this menu bar, [click here](#).



These actions or guards, once set up, will then be executed at the time of the event's occurrence in the process.

4. **Event Service:** This is a single-select dropdown field that displays the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to link an entity to the event.
  - When an event service is selected, it will automatically filter the list of items in the **Event** dropdown. Only events related to the selected entity will be displayed.
    - By default, the event service is set to the entity selected when the process was created.
5. **Event Type:** This is a single-select dropdown field that displays the list of event types.
  - [Event Types:](#) Auto, Human, System, or Scheduled.
  - When an event type is selected, it will automatically filter the list of items in the **Event** dropdown. Only events of the selected type will be displayed.
    - By default, the event type is set to [Auto](#).

\* When a start object is linked to another element in the process, the transition between them is automatically defined as “StartHumanEvent” on the map. Users cannot modify the properties of this event type.

! It is mandatory for users to assign an event to the transitions in the process.

\* For information on how to set up conditional events using code, [click here](#).

## 11.1.2.2.10.1. Insert and Adjust Events

In the Process Designer, users can insert events into the diagram either automatically or manually.

### 1. Automatic Method

To insert an event automatically, take the following steps:

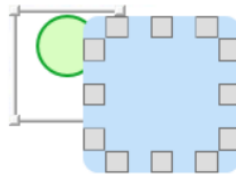
- First, drag and drop an object (start event, state, gateway, etc.) from the toolbox onto the design canvas.



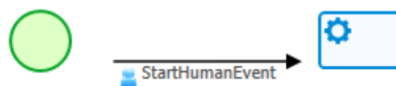
- Next, select the object now situated on the canvas. It will display to its right a list of thumbnails, each representing a different element that the object can be linked to.



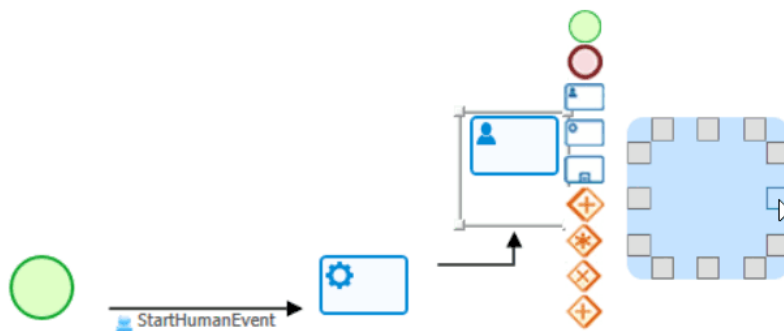
- Select the appropriate thumbnail—that is, the one representing the next object needed for your process—and from the resulting blue field, click on the grey box that most closely aligns with your desired location for the second object.



- Once you have determined its location, the second object will appear in the diagram with an arrow linking it to the first object. This arrow is known as an event.



- In this manner, users can design the rest of the process, automatically adding events as they go.





## 2. Manual Method

To insert an event automatically, take the following steps:

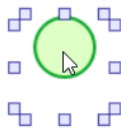
- First, drag and drop an object (start event, state, gateway, etc.) from the toolbox onto the design canvas.



- Next, drag and drop another object to which the first object can be connected. (Users can look to the thumbnails associated with the first object for reference.)



- Once you have both objects situated on the canvas, hover over the first object. It will display a set of small boxes along its borders.



- Place the cursor over the box that most closely aligns with your desired direction for the event. Click and drag the cursor across to the corresponding box for the second object.



- In this manner, users can design the rest of the process, manually adding events as they go.

**Note:** To adjust or reroute an event, simply click and drag the handle on either end of the arrow.

## 11.1.2.2.10.2. Conditional Events: Code Configuration

The `formObject.DisabledEvents` property provides access to the list of events that are hidden from the user by the system. The script can utilize this property to add or remove events from the list based on specific conditions.

Here are examples of how to use this property to set up conditional events:

- **Code Sample A**

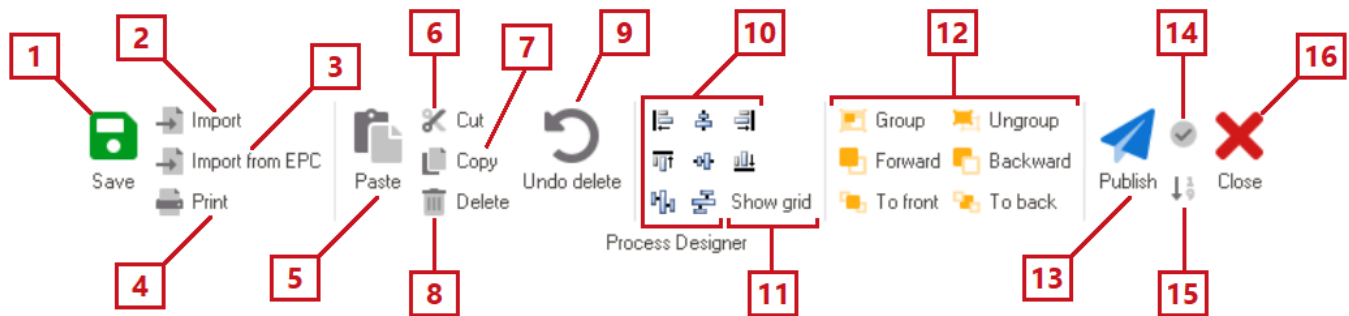
```
// Get all events
var events = bpmAppService.BPMSServices.DomainObjectService.GetEvents(domainObject);
// Find specific event
var e = events.Events.FirstOrDefault(x => x.Name == "EventName");
if (e != null)
{
    var eventId = e.Id;
    // Init collection of disabled events
    if (formObject.DisabledEvents == null)
        formObject.DisabledEvents = new List();
    // if event already in the disabled list
    var exists = formObject.DisabledEvents.Exists(x => x == eventId);
    if (NeedToEnableCondition)
    {
        if (exists)
            formObject.DisabledEvents.Remove(eventId);
    }
    else
        if (NeedToDisableCondition)
        {
            if (!exists)
                formObject.DisabledEvents.Add(eventId);
        }
}
```

- **Code Sample B**

```
formObject.DisabledEvents.Add(Ray.BPMApp.QMS.Events.BPMAPP_My_DomainObject.HumanEvents.MyEventName.Id);
```

## 11.1.2.3. Menu Bar

The menu bar of the Process Designer consists of the following items:

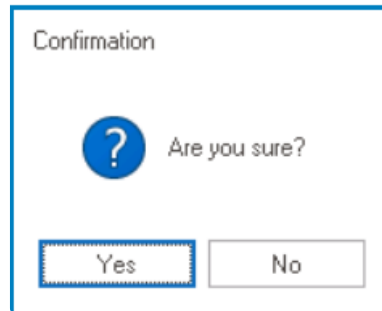


1. **Save:** This button allows users to save the process.
  - If selected, a message will appear in the lower left corner of the diagram stating “Process Saved.”
2. **Import:** This button allows users to import a process into the program. Users can import processes that are designed in the XPDL format.
  - If selected, a file dialog will appear, enabling users to select the process.
3. **Import from EPC:** This button allows users to import processes from EPC into the program.
  - If selected, a pop-up window will appear allowing users to log into EPC and select the process.
4. **Print:** This button allows users to print the process.
  - If selected, the Print dialog box will appear, enabling users to specify the printer and other printing preferences.
5. **Paste:** This button allows users to insert any data that has been stored in the clipboard into the diagram.
  - If selected, any element that was previously copied or cut in the diagram will be pasted into it.
6. **Cut:** This button allows users to remove selected data from the diagram.
  - To use this function, select an element in the diagram and then click on the **Cut** button.
7. **Copy:** This button allows users to duplicate selected data.
  - To use this function, select an element in the diagram and then click on the **Copy** button.
8. **Delete:** This button allows users to delete selected data.
  - To use this function, select an element in the diagram and then click on **Delete** button.
9. **Undo Delete:** This button allows users to undo a previous deletion.
10. **Ordering:** These are a set of functions that allow users to alter the alignment of elements in the diagram.
  - Users can group elements from the right, left, middle, upper, and lower boundary by selecting the elements and choosing the alignment.
11. **Show Grid:** This button allows users to superimpose a grid onto the design canvas.
  - To remove the grid, simply re-select the button.
12. **Grouping:** These are a set of functions that allow users to alter the display priority of elements in the diagram.
  - Users can change the display priority (front, back, etc.) of an element by selecting it from the diagram and choosing the desired option. It is possible to change the display priority of

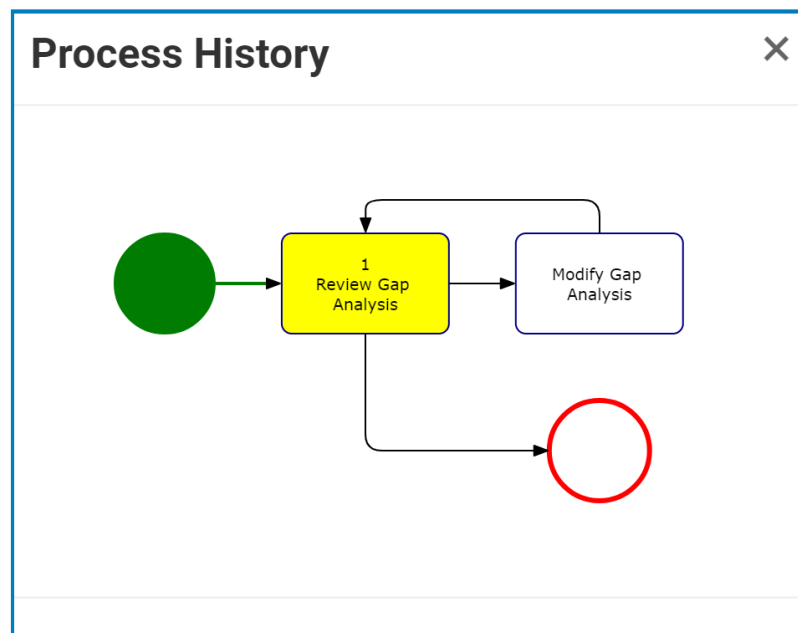
multiple elements.

13. **Publish:** This button allows users to publish the process.

- If this button is selected, users will be alerted of the action with the following confirmation message:



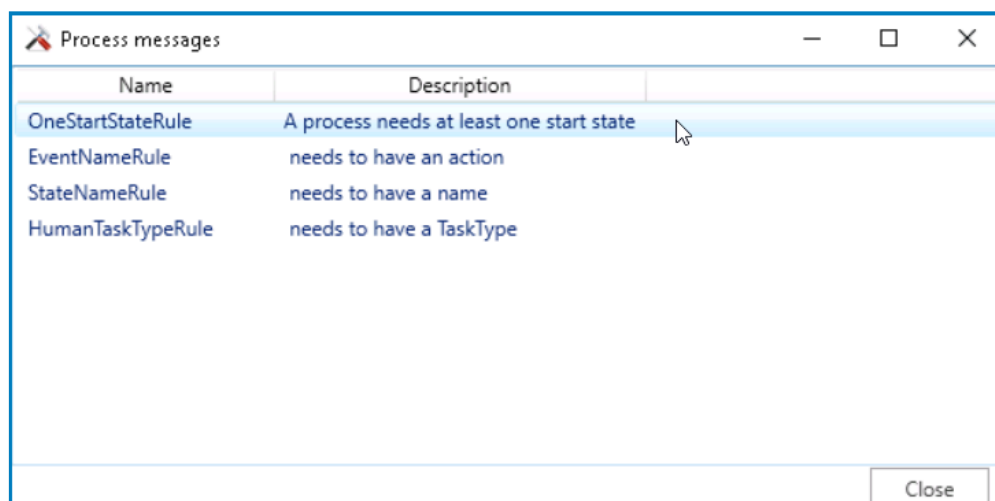
- **Yes or No:** To confirm the publication, click on the **Yes** button. To cancel, click on the **No** button.
- Once a process has been published, it can be accessed in the **DBP** web application via the [Process History](#) button.



A validation check is performed automatically upon request of publication. If there are errors in the process design, users will be prevented from publishing the process. They will instead be directed to the Process Messages window (described below).

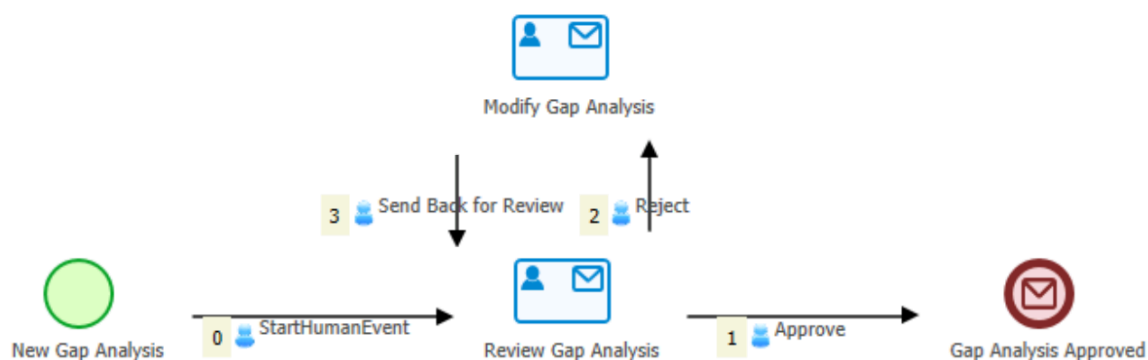
14. **Validate:** This button allows users to check the validity of the process design.

- If the process design is valid, a message will appear in the lower left corner of the diagram stating “Process Validated Successfully.”
- If the process design is invalid, a pop-up window will appear identifying the errors for the user.
  - To be redirected to the location of an error, double-click on it.



! If a process has not passed the validation check, it cannot be published.

15. **Event Order:** This button allows users to label events in the diagram by the order in which they occur in the process.
- To remove the event order, simply re-select the button.



\* Events are ordered according to the sequence specified in the [Event Properties](#).

16. **Close:** This button allows users to close the [Process Designer](#).

## 11.1.3. Process Alteration

Over time, as organization grow or their needs change, their processes and policies may require alteration.

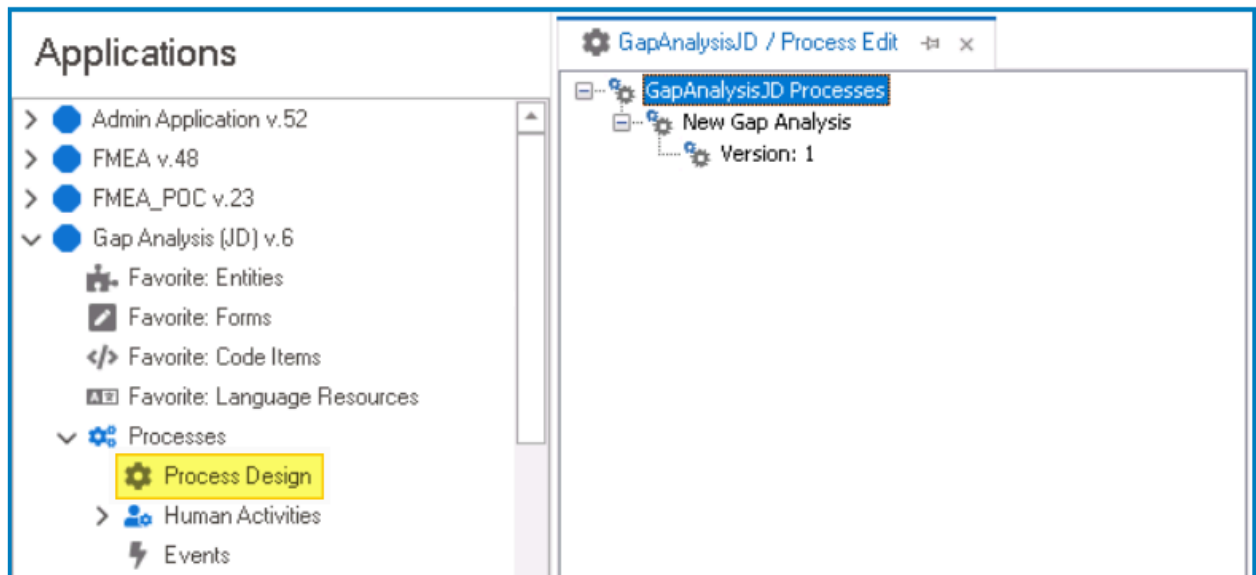
There are two ways to change processes in the program:

### 1. Add New Version

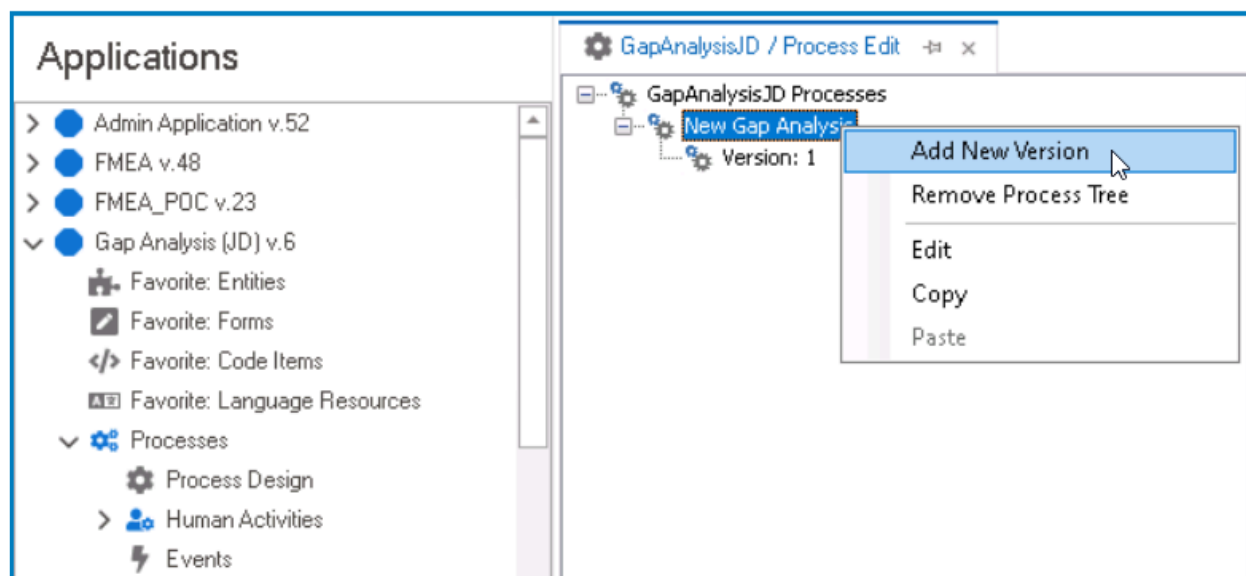
When a new version of a process is added, all current processes (existing in the application) will retain the content of the previous version. All future processes, however, will adopt the content of the new version.

To create a new version of a process, take the following steps:

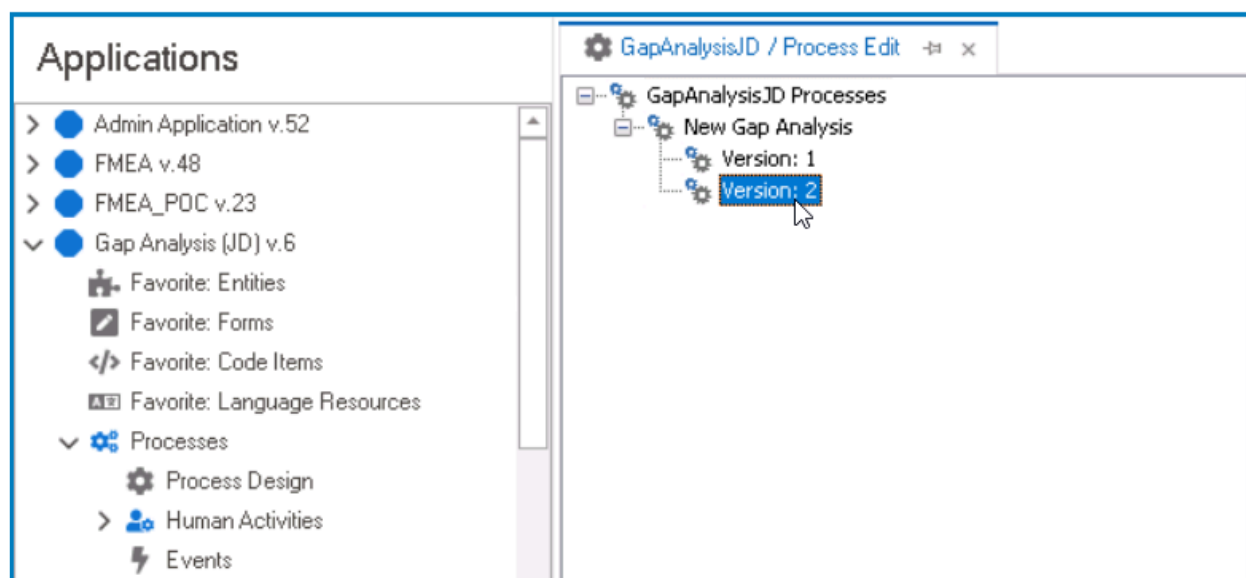
- From the **Processes** module (inside the [Application Tree](#)), double-click on **Process Design**. It will open a new tab called Process Edit.



- In the Process Edit tab, right-click on the desired process and select the **Add New Version** option.



- Selecting **Add New Version** will generate a new version below the previous one. Double-click on the newly generated version. It will launch the [Process Designer](#) in a new tab.



✿ Whenever a new version is generated, it will contain a copy of the previous version of the process. Users can thus review the previous version and apply modifications as needed.

## 2. **Process / State Mapping**

In the [Process Mapper](#) window, users can change the state of previously published content in the process.

**Process Mapper**

**State Mapping**

Map deleted activities from the previously published process to other activities.

| Removed States   | New States |
|------------------|------------|
| AnalysisApproved | [Empty] ▼  |
| ModifyAnalysis   | [Empty] ▼  |

☐ **Remove unused dependencies**

It will remove the unused objects since changing the process.

Apply Cancel

✿ To access this window, users will need to first modify the previously published process and then attempt to republish it. Once the **Publish** button has been selected, the Process Mapper window will be displayed.

As mentioned in this window, state mapping refers to the practice of connecting deleted activities from the previously published process to new activities in the modified process. By selecting a new state for a removed one, the content of the removed state will be transferred to the new state.

- **Removed States:** This column displays the states or activities that have been removed from the process. These states belong to the previous publication of the process.
  - States in this field are identified by their name (rather than title).
- **New States:** This column (using individual dropdown fields) displays the new states or activities in the process. These states belong to the modified process, set up following its previous publication.
  - States in this field are identified by their name (rather than title).

! For each removed state, users can match a new state from the corresponding dropdown. All removed states need to be mapped.

In this window, users can also choose to remove any unused objects in the modified process.

- **Remove Unused Dependencies:** This is a checkbox which, if selected, allows users to remove unused objects and their dependencies from the modified process.



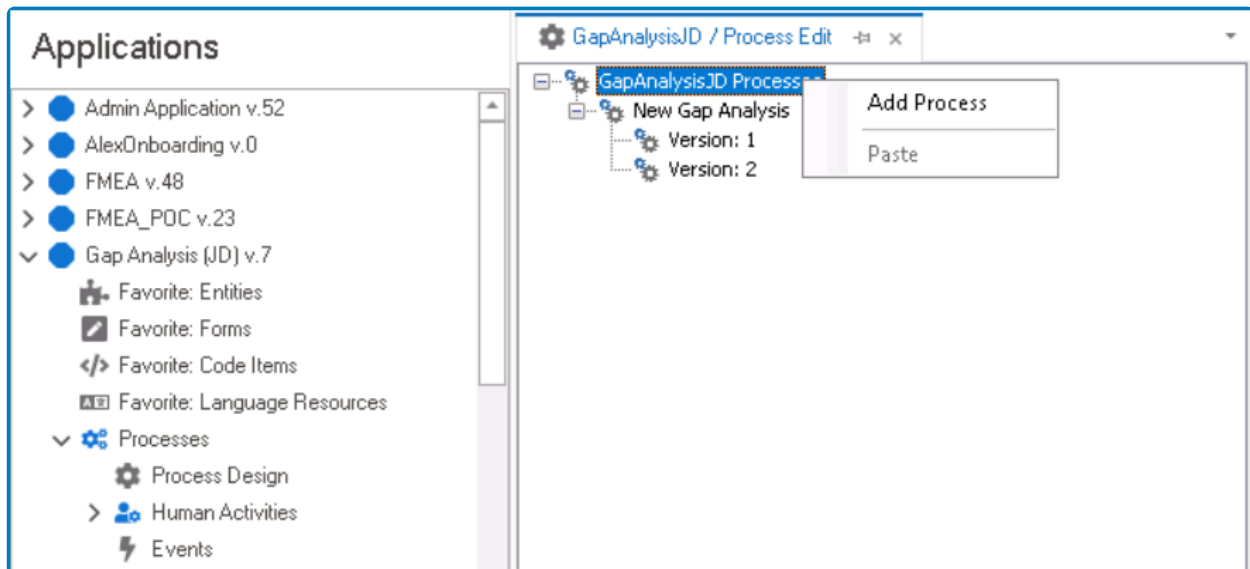
Once all removed states have been mapped and changes are applied, the modified process will be published.

## 11.1.4. Process Context Menus

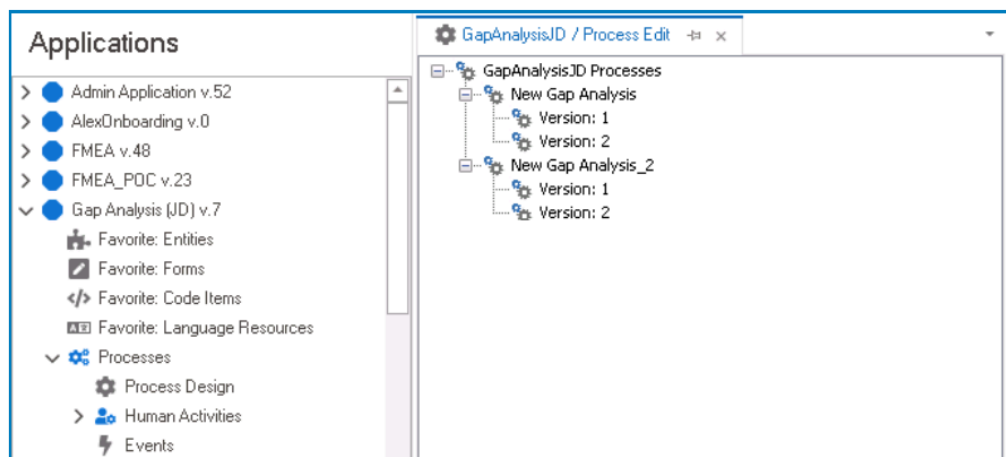
In the Process Edit tab, users are provided access to three sets of context menus:

### 1. Application Processes Menu

To access this menu, right-click on “[AppName] Processes.” It will display the following options:



- **Add Process:** This button allows users to add a new process.
  - [For more details on this option, click here.](#)
- **Paste:** This button allows users to paste a process into the set.
  - To enable the **Paste** button, first copy the desired process in the application.
    - Once pasted, the process will be displayed below the last process tree in the set.

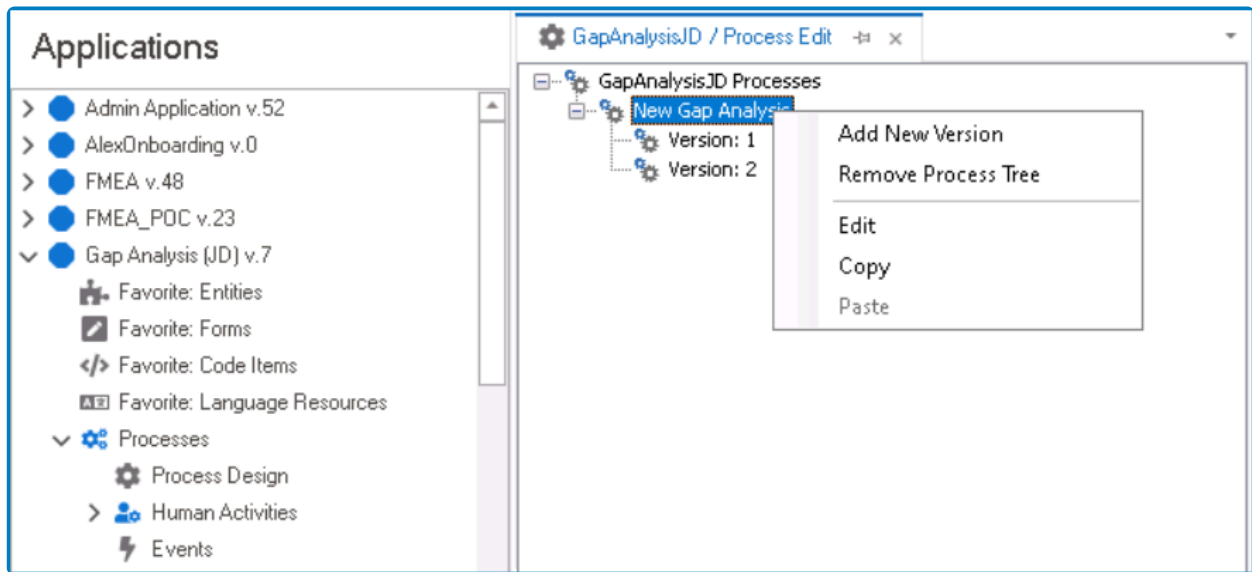


- Users can only paste processes from the same application.

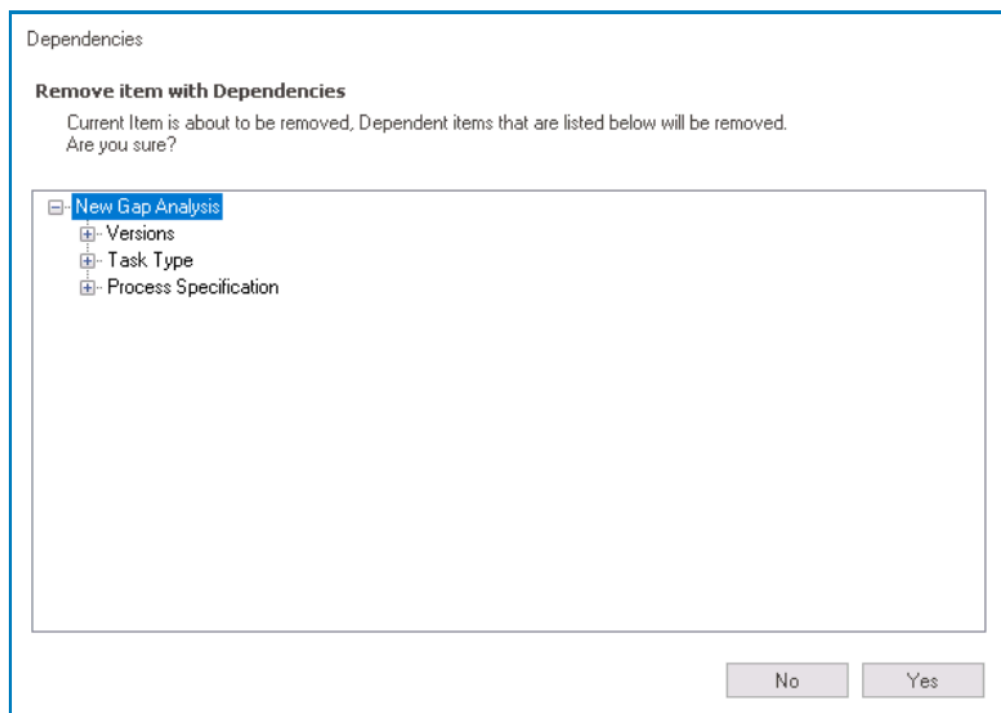
\* Under the “[AppName] Processes” superset, users can access a list of all defined processes for the application. Inside it, processes are organized into subsets of tree-like diagrams. These diagrams are referred to as process trees.

## 2. Process Tree Menu

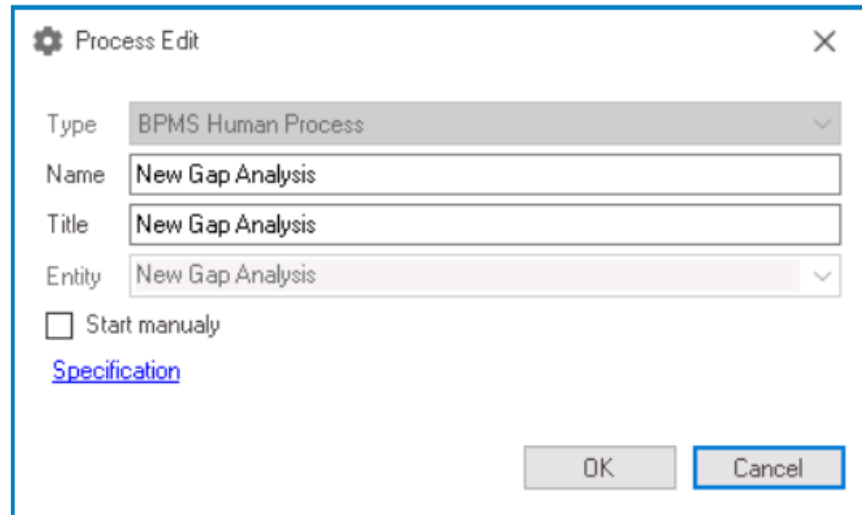
To access this menu, right-click on a process tree. It will display the following options:



- **Add New Version:** This button allows users to add a new version of the process.
  - For more details on this option, [click here](#).
- **Remove Process Tree:** This button allows users to remove a process tree.
  - If selected, it will display the Dependencies window. In this window, users can access an overview of the selected process, including its dependent objects.



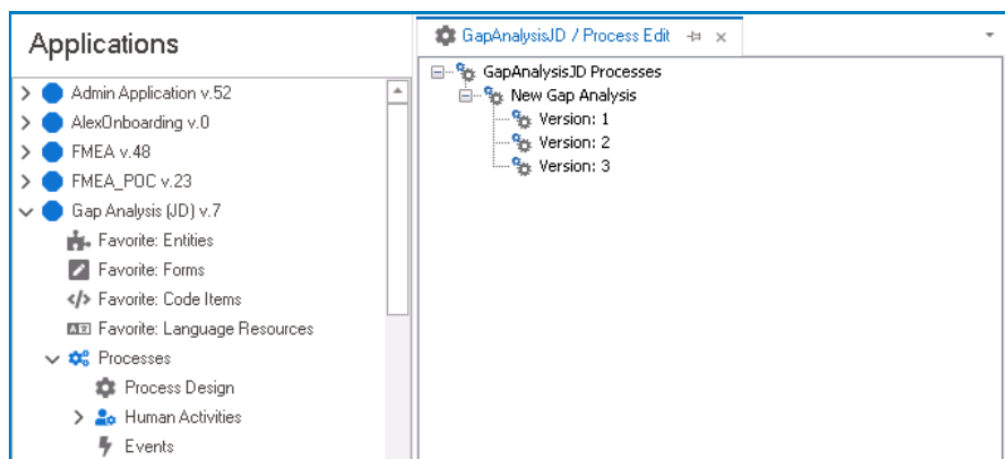
- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
- **Edit:** This button allows users to edit the process details.
  - If selected, it will display the Process Edit window. In this window, users can make modifications to previously inputted information.



The **Process Edit** dialog box contains the following fields and controls:

- Type:** A dropdown menu set to **BPMS Human Process**.
- Name:** A text field containing **New Gap Analysis**.
- Title:** A text field containing **New Gap Analysis**.
- Entity:** A dropdown menu set to **New Gap Analysis**.
- Start manually:** An unchecked checkbox.
- Specification:** A blue hyperlink.
- Buttons:** **OK** and **Cancel** buttons at the bottom right.

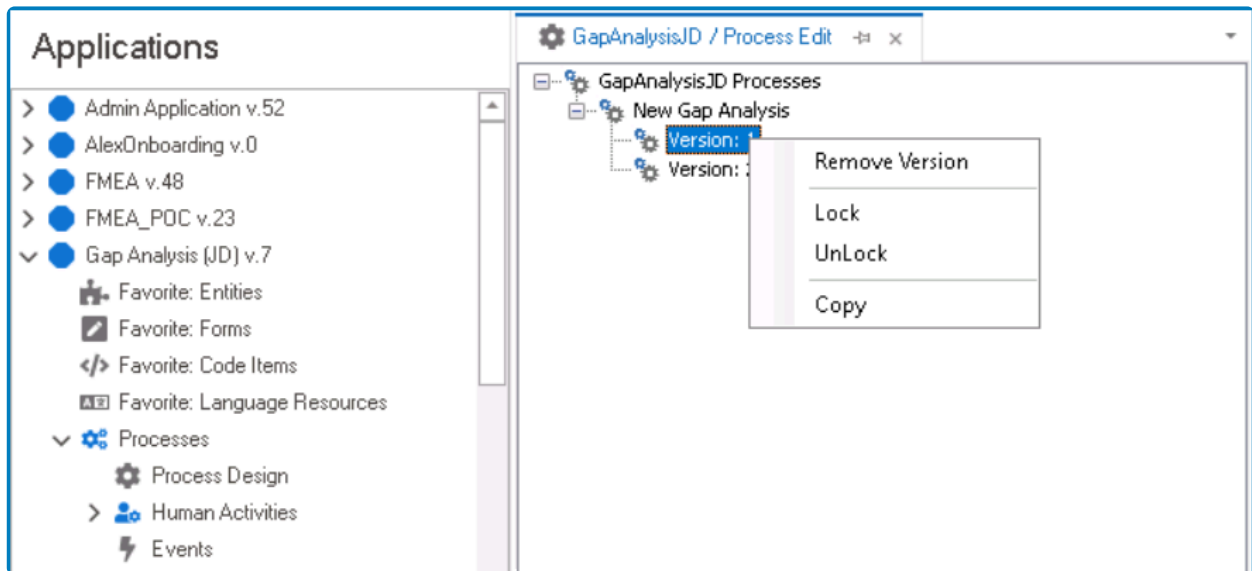
- For more details on this window, [click here](#).
- **Copy:** This button allows users to copy a process.
  - Once a process has been copied, it can be pasted via the **Application Processes** menu.
- **Paste:** This button allows users to paste a process version into the tree.
  - To enable the **Paste** button, first copy the desired process version in the application.
    - Once pasted, the process version will be displayed below the last version in the tree.



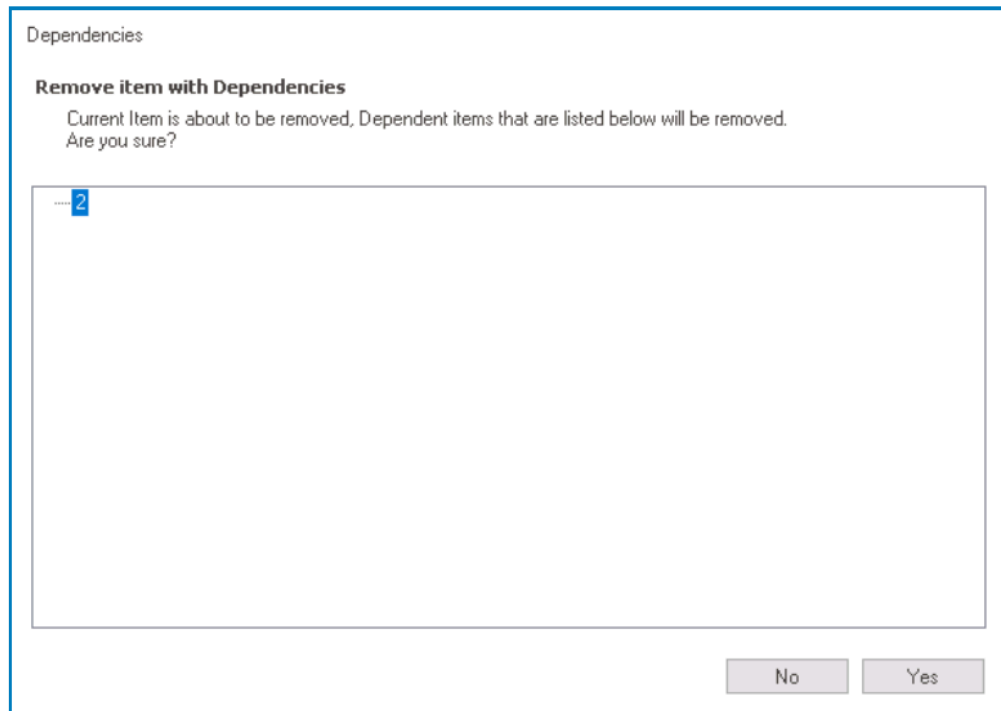
- Users can only paste process versions from the same application.

### 3. Process Version Menu

To access this menu, right-click on a process version. It will display the following options:



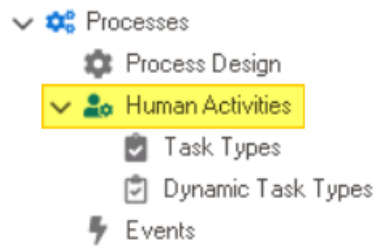
- **Remove Version:** This button allows users to remove a process version.
  - If selected, it will display the Dependencies window. In this window, users can access an overview of the selected version, including its dependent objects.



- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.
- **Lock:** This button allows users to lock the process version, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the process version.
  - If selected, access restriction on the process version will be lifted.
- **Copy:** This button allows users to copy a process version.
  - Once a process version has been copied, it can be pasted via the **Process Tree** menu.

## 11.2. Human Activities

In the **Human Activities** section of this module, users can create static and dynamic task types.



\* For each human activity or state in the process, users must define a task type. Task types, along with helping to determine aspects of how tasks function and are featured in the web application, enable for the determination of user responsibility. **For more information on human activities and their properties, [click here](#).**

1. [Static Task Types](#)
2. [Dynamic Task Types](#)

## 11.2.1. Task Types

Static task types—known simply as task types in the program—are task types that are fixed and predefined.

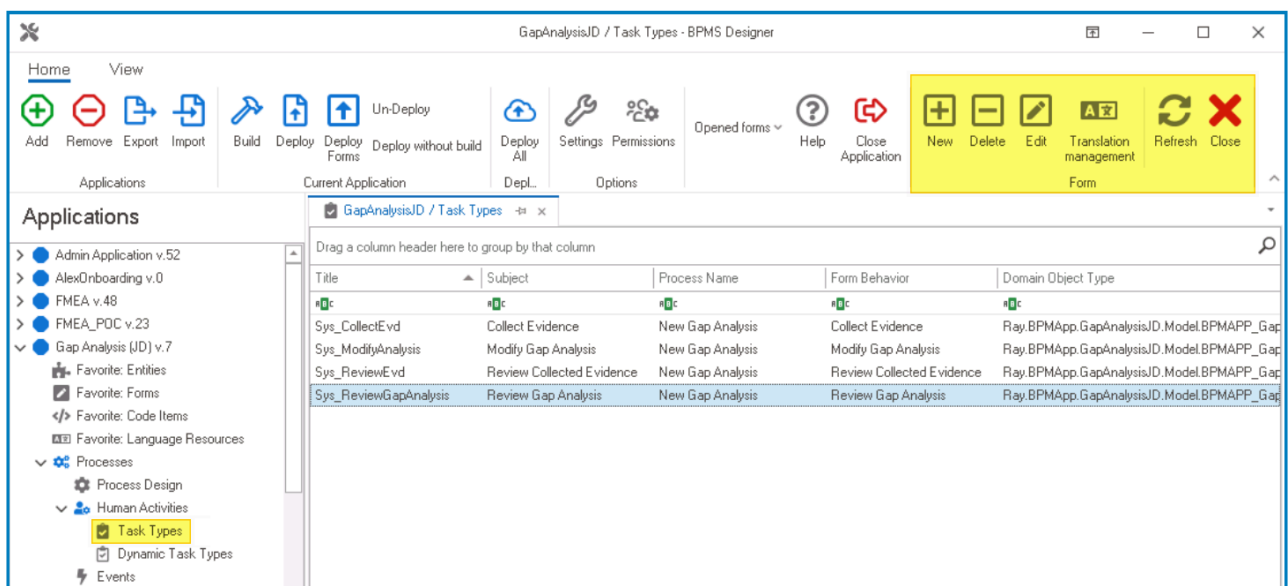
- ✿ Unlike dynamic task types, which are created or modified based on certain conditions, static task types remain the same throughout the execution of the workflow.

There are two main ways by which static task types can be defined, edited, and deleted in the program:

### 1. Menu Bar

To manage task types using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Processes > Human Activities > Task Types** (double-click to open). It will display the Task Types tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined task types for the application. For [guidance on how to manipulate the data presented in this tab](#), [click here](#).

- From the menu bar, select the desired option.



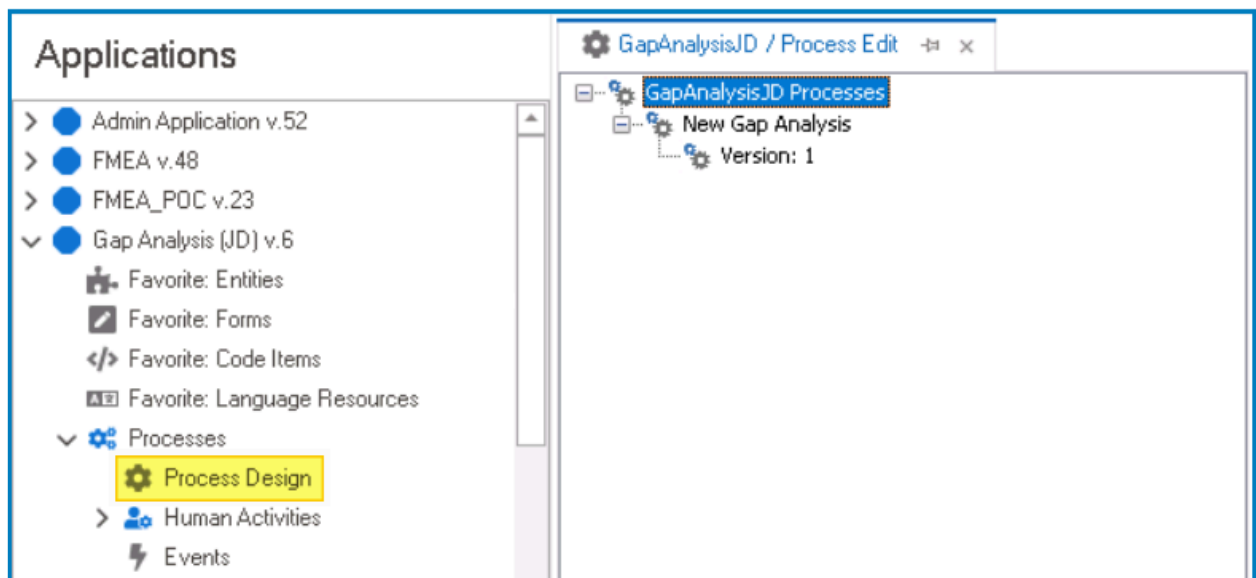
- **New:** This button allows users to define a new task type.

- If selected, it will display the [Task Type Properties](#) window.
- **Delete:** This button allows users to delete an existing task type.
  - To use this function, first select a task type from the list.
- **Edit :** This button allows users to edit an existing task type.
  - To use this function, first select a task type from the list.
- **Translation Management:** This button allows users to manage the translation of task titles.
  - If selected, it will display the [Translation Management](#) window, in which users will find a table listing all defined task types in the application.
- **Refresh :** This button allows users to refresh the [Task Types](#) tab.
- **Close:** This button allows users to close the [Task Types](#) tab.

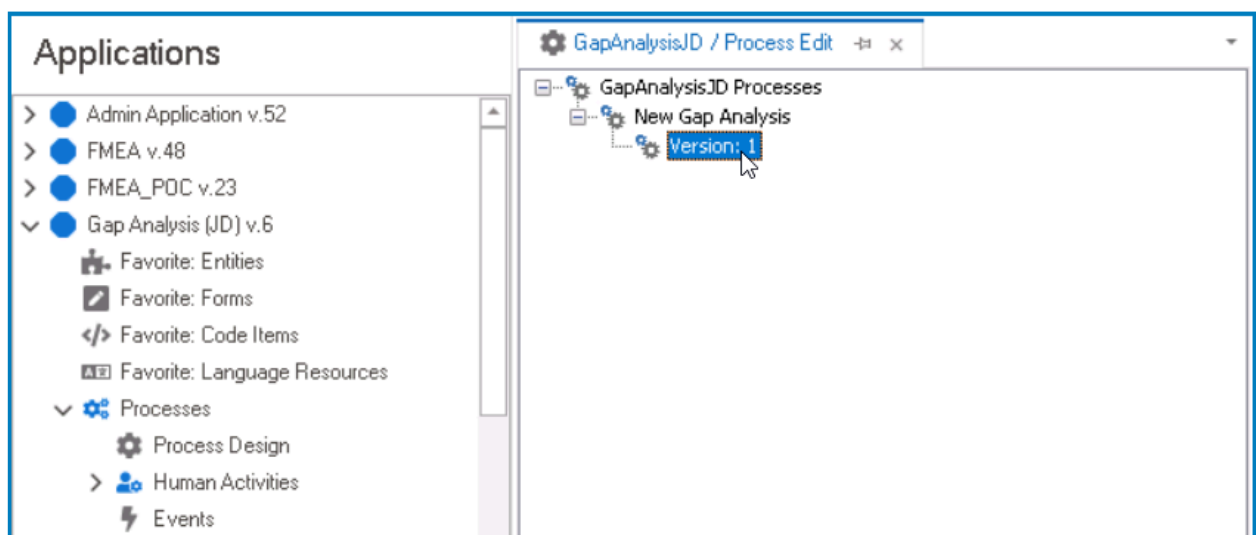
## 2. Human Activity Properties

To manage task types using the human activity properties, take the following steps:

- From the [Application Tree](#), navigate to **Processes > Process Design** (double-click to open). It will display a new tab called [Process Edit](#).

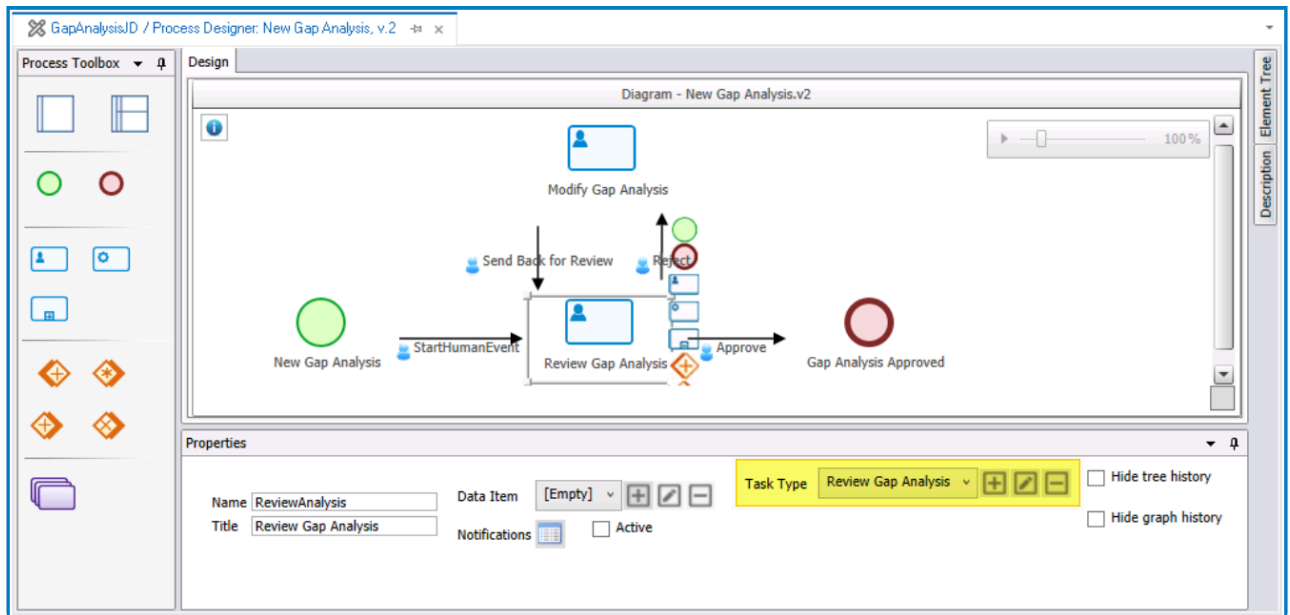


- In the [Process Edit](#) tab, double-click on a process version. It will launch the [Process Designer](#) in a new tab.





- In the diagram of the Process Designer, select a human activity. It will display its associated properties in the Properties panel.



- From the Properties panel, select the desired option next to **Task Type**.

Task Type Review Gap Analysis + ✎ -

- To define a new task type, select the + button.
  - It will display the [Task Type Properties](#) window.
- To edit an existing task type, select the ✎ button.
- To delete a task type, select the - button.

# 11.2.1.1. Translation Management

In the Translation Management window, users can either automatically or manually translate items into different languages.

Translation Management

Auto Translate

Enter text to search...

Find

Drag a column header here to group by that column

|   | Task              | English               | French | Arabic | German | Hebrew |
|---|-------------------|-----------------------|--------|--------|--------|--------|
| ▶ | Sys_ReviewGap...  | Review Gap Analysis   |        |        |        |        |
|   | Sys_ModifyAnal... | Modify Gap Analysis   |        |        |        |        |
|   | Sys_CollectEvd    | Collect Evidence      |        |        |        |        |
|   | Sys_ReviewEvd     | Review Collected E... |        |        |        |        |

✓

Ok

✗

Cancel

✿ Users can determine which languages are available for translation in the Settings of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

To automatically translate items into all available languages, click on the **Auto Translate** button.

Translation Management

Auto Translate

Enter text to search... Find

Drag a column header here to group by that column

| Task                      | English               | French                   | Arabic                | German                  | Hebrew            |
|---------------------------|-----------------------|--------------------------|-----------------------|-------------------------|-------------------|
| <b>Sys_ModifyAnal...</b>  | Modify Gap Analysis   | Modifier l'analyse de... | تعديل تحليل الفجوة    | Lückenanalyse änd...    | שנה ניתוח פערים   |
| <b>Sys_CollectEvd</b>     | Collect Evidence      | Recueillir des preuves   | جمع الأدلة            | Sammeln Sie Beweise     | אסוף ראיות        |
| <b>Sys_ReviewEvd</b>      | Review Collected E... | Examiner les preuve...   | مراجعة الأدلة المجمعة | Überprüfen Sie die g... | סקור ראיות שנאספו |
| ▶ <b>Sys_ReviewGap...</b> | Review Gap Analysis   | Examiner l'analyse d...  | مراجعة تحليل الفجوة   | Überprüfen Sie die ...  | סקירת ניתוח פערים |

✓ Ok ✗ Cancel

! It is important to note that this feature may not be as accurate or contextually relevant as manual translations, so it is recommended to review the results before finalizing.

To manually translate items into the available languages, select the corresponding column and enter the translation.


Translation Management


Auto Translate

Enter text to search...


Find

Drag a column header here to group by that column

|   | Task                     | English               | French | Arabic | German | Hebrew |
|---|--------------------------|-----------------------|--------|--------|--------|--------|
|  | <b>Sys_ReviewGap...</b>  | Review Gap Analysis   |        |        |        |        |
|   | <b>Sys_ModifyAnal...</b> | Modify Gap Analysis   |        |        |        |        |
|   | <b>Sys_CollectEvd</b>    | Collect Evidence      |        |        |        |        |
|   | <b>Sys_ReviewEvd</b>     | Review Collected E... |        |        |        |        |



Ok



Cancel

 Users can manipulate the data presented in this window using certain techniques. For more information on these techniques, [click here](#).

## 11.2.1.2. Task Type Properties

---

In the Task Type Properties window, users can create a new task type for the application.

There are three tabs to consider in this window:

1. [General](#)
2. [Advanced](#)
3. [Analytic Data](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Task Type Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 11.2.1.2.1. General Tab

In the General tab, users can define the basic properties of the task.

**Task Type Properties**

Apply OK Cancel

General Advanced Analytic Data

1 Name  ☐ Save to Staff 2

English Title

French Title

German Title  3

Arabic Title

Hebrew Title

4 ☐ Timesheet

5 Description

6 Dynamic Task Type

7 Confidentiality Normal  8 Priority Normal

9 Receiver

☐ Specific receiver(s)

☐ All

☐ Random

☐ Sequential

☒ Defined receiver(s)

☐ Current Role

☐ Starter Role

☒ Current User

☐ Starter User

☐ Current Role Department Management

☐ Starter Role Department Management

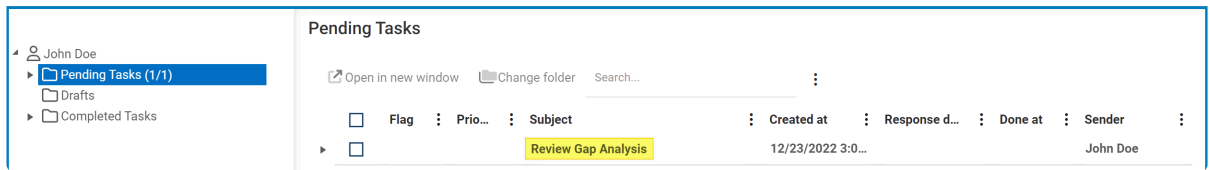
☐ Dynamic receiver(s)

Field Name :

☐ Department Management of specific/dynamic receiver

1. **Name:** This is a text field that allows users to input the name of the task. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A task type must have a unique name. It cannot have the same name as another task type in the application.

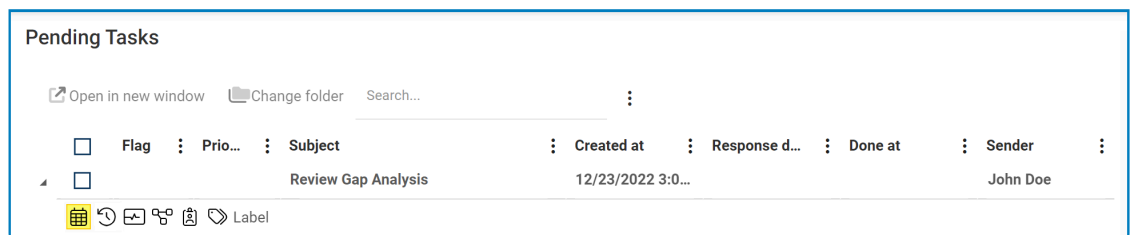
2. **Save to Staff:** This is a checkbox which, if selected, allows users to change the defined receiver of the task to the current role.
3. **Title:** This is a text field that allows users to input the title of the task.
  - The title inputted in this field will be used to identify the task in the **DBP** web application.



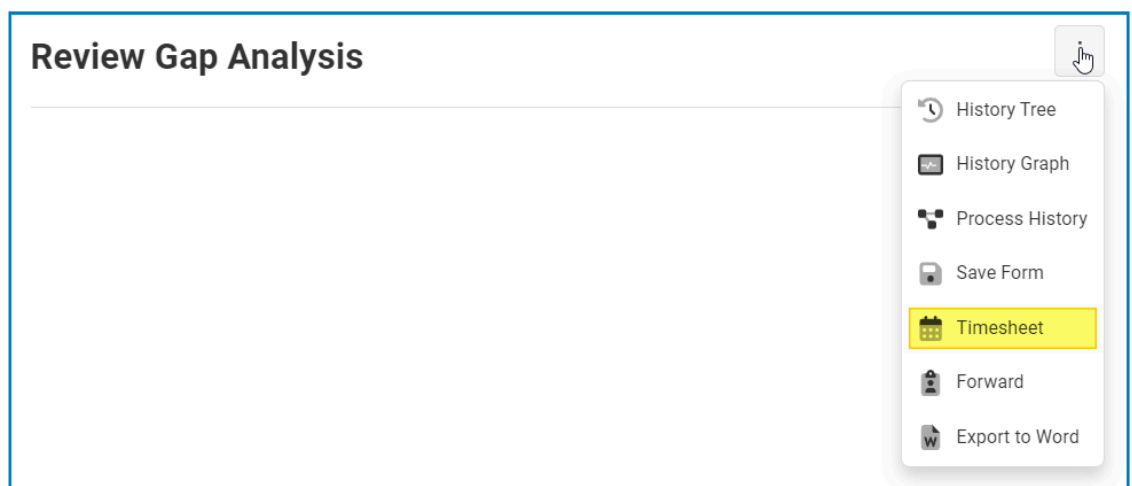
- Users, however, can also change this displayed title using the **Subject Info Template** field in the [Advanced](#) tab.
- Unlike with names, there are no rules determining the validity of titles.

✿ Users can input the title in English, French, German, Arabic, and Hebrew. Currently, these are the only languages supported on the web server.

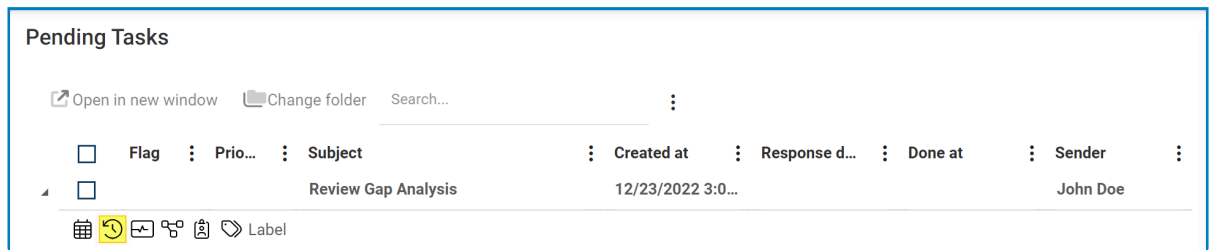
4. **Timesheet:** This is a checkbox which, if selected, allows users to enable a timesheet for the task.
  - In the web application, there are two ways to access the timesheet for a given task:
    1. From the task context menu:




2. From the task form:




- For more information on the features of the Timesheet, [click here](#).
5. **Description:** This is a text field that allows users to input a description for the task.
    - In the web application, the task description can be accessed via the [History Tree](#) tool.



6. **Dynamic Task Type:** In this section, users can manage the dynamic task types linked to the task.

- To create a dynamic task type, select the  button.
  - It will display the Dynamic Task Proxy window, in which users can define the name and subject of the dynamic task type.

- **Save or Exit:** To save the dynamic task type, click on the **Save** button. To exit the window, click on the **Exit** button.
  - To further define the dynamic task type, users can then switch to the [Advanced](#) tab. Users, however, must first apply their changes and close the Task Type Properties window.

- To delete a dynamic task type, select the  button.

**!** Only one dynamic task type can be created per task.

7. **Confidentiality:** This is a single-select dropdown field that allows users to select the confidentiality level of the task.

- Confidentiality Levels: Normal or Confidential.
  - If a task is set as normal, it will be shown in the web application as a standard task without any marker or indication of its confidentiality level.
  - If a task is set as confidential, there will be a marker at the top of the form.



Review Gap Analysis

⋮

Confidential

8. **Priority:** This is a single-select dropdown field that allows users to select the priority level of the task.

- **Priority Levels:** Normal, Low, Medium, High, or Urgent.
  - If a task is set as normal, it will be shown in the web application as a standard task without any marker or indication of its priority level.
  - If a task is not set as normal, its priority level will be indicated in the task list.

John Doe  
 Pending Tasks (1/1)  
 Drafts  
 Completed Tasks

Pending Tasks

Open in new window Change folder Search...

| <input type="checkbox"/> | Flag | Prio... | Subject             | Created at        | Response d... | Done at | Sender   |
|--------------------------|------|---------|---------------------|-------------------|---------------|---------|----------|
| <input type="checkbox"/> |      | High    | Review Gap Analysis | 12/23/2022 3:0... |               |         | John Doe |


✿ Users can customize which columns are displayed in the task list. For more details on this feature, [click here](#).


9. **Receiver:** In this section, users can define the receiver(s) of the task. Users can choose to set either specific receivers, defined receivers, or dynamic receivers.

- **Specific Receiver(s):** These receivers are individual users, roles, or groups that are explicitly identified by the user.

☒ Specific receiver(s)

☒ All  
☐ Random  
☐ Sequential

-  : This button allows users to add a new receiver.
  - It will display the **Receivers** window, from which users can select the specific receiver.
    - For more details on the **Receivers** window, see [previous instance](#).
  - Once a receiver has been added, it will populate the **Specific Receiver(s)** field.


-  : This button allows users to remove a receiver.
- **All**: This option allows for the task to be sent to all selected recipients.
- **Random**: This option allows for the task to be sent randomly to one of the selected recipients.
- **Sequential**: This option allows for the task to be sent to sequentially to the selected recipients.
  - For example, there is a process with four tasks. For each task, the same four receivers are selected. If users opt for these task to be sent sequentially, the tasks will be divided as follows.
    - Task A will be sent to the first receiver in the list.
    - Task B will be sent to the second receiver in the list.
    - Task C will be sent to the third receiver in the list.
    - Task D will be sent to the fourth receiver in the list.
  - Essentially, this option acts as way for work to be distributed equally among receivers.
- **Defined Receiver(s)**: These receivers are not explicitly identified by the user. Instead, they are defined by the system based on certain criteria or conditions.

- Users can choose one of the following criteria:
  - **Current Role**: This option allows for the task to be sent to the current role.
  - **Starter Role**: This option allows for the task to be sent to the starter role.
  - **Current User**: This option allows for the task to be sent to the current user.
  - **Starter User**: This option allows for the task to be sent to the starter user.
  - **Current Role Department Management**: This option allows for the task to be sent to the current department manager.
    - A task can only be assigned to the current department manager if the previous step in the workflow was assigned to a role.
  - **Starter Role Department Management**: This option allows for the task to be sent to the starter department manager.
- Depending on the criteria selected, the system will automatically assign the task to the

appropriate receiver(s).

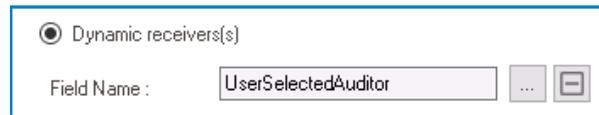
\* The system will determine the role of the user and position of the manager by referring to the information set up in the [Organization Management \[Admin\]](#) application.

- For general reference:
  - A **starter user or role** refers to the person or role responsible for initiating a workflow or process.
  - A **current user or role** refers to the person or role currently working on or interacting with the workflow or process.
- **Dynamic Receiver(s):** These receivers are explicitly identified by the user in the web application. From a field in the form, the user can select the receiver to whom they want the task to be sent. The user must select a receiver before they can submit the form.


-  : This button allows users to select the field or attribute in which the receiver(s) will be identified.
  - It will display the [Object Browser](#) window. In this window, users can access a list of the associated entity's attributes.


\* Users can select the associated entity in the [Advanced](#) tab of the [Task Type Properties](#) window.

- Once an attribute has been selected, it will populate the **Field Name** box.



Dynamic receivers(s)

Field Name : UserSelectedAuditor ... 

-  : This button allows users to remove the field.
- **Department Management of Specific/Dynamic Receiver:** This is a checkbox which, if selected, allows for the department manager of the receiving unit to be included in the task assignment.



It is mandatory for users to select a receiver (whether specific, defined, or dynamic) for the task.

## 11.2.1.2.2. Advanced Tab

In the Advanced tab, users can define the advanced properties of the task.

**Task Type Properties**

Apply OK Cancel

General Advanced Analytic Data

1 Process [Empty]

2 Entity Type New Gap Analysis (New Gap Analysis)

3 Dynamic Task Type [ ] Edit Code Designer Remove

4 ☒ Assignable 5 ☐ Hide Save Button

6 ☐ Formless Events 7 ☐ Hide Tracking

8 Application Name [ ]

9 Subject Info Template [Empty] + [ ] -

10 Form Title [Empty] + [ ] -

1. **Process:** This is a single-select dropdown field that retrieves the list of processes (from the [Process Design](#) section of the **Processes** module). It allows users to select the process associated with the task type.

✿ If a task type is created using the [Process Designer](#), this field will automatically be populated with the current process associated with the human activity.

2. **Entity Type:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the task type. It is a **mandatory** field.

\* If a task type is created using the [Process Designer](#), this field will automatically be populated with the current entity associated with the process.

3. **Dynamic Task Type:** In this section, users can further define the details of the dynamic task type.

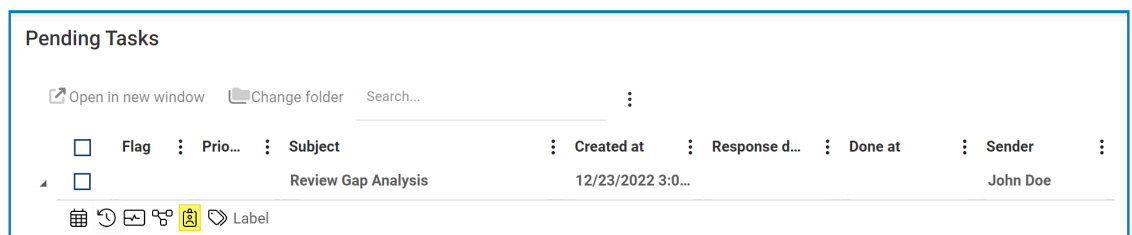
- Once a dynamic task type has been created in the [General](#) tab (and the [Task Type Properties](#) window has been saved and closed), it will populate this field.

Dynamic Task Type  [Edit Code](#) [Designer](#) [Remove](#)

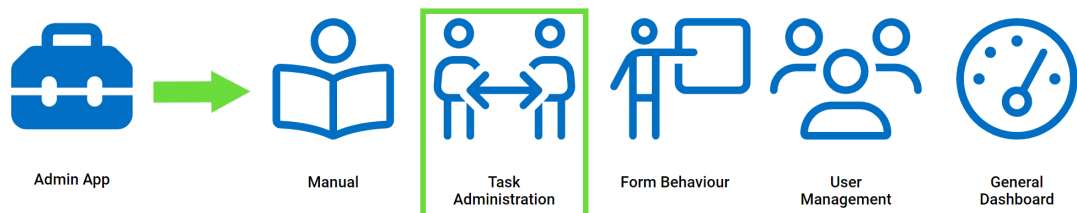
- Edit Code:** This button allows users to launch the [Code Designer](#), in which users can write code specifying the behavior and properties of the dynamic task type.
  - For more details on how to define dynamic task types using the [Code Designer](#), [click here](#).
- Designer:** This button allows users to launch the [Rule Designer](#), in which they can define rules that specify the conditions and actions for the dynamic task type.
  - For more details on how to define dynamic task types using the [Rule Designer](#), [click here](#).
- Remove:** This button allows users to remove the dynamic task type.

4. **Assignable:** This is a checkbox which, if selected, allows for assignees to forward the task to other users.

- In the **DBP** web application, there are two ways to forward a given task:
  - Using the **Assign** tool in the task context menu:



2. Using the **Task Administration** application:

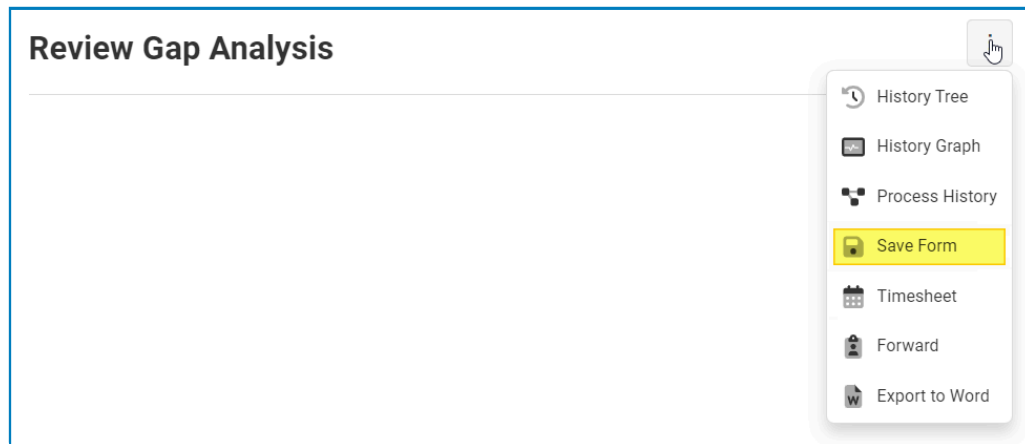


! For this feature to work, the web server must be on SQL 2019.

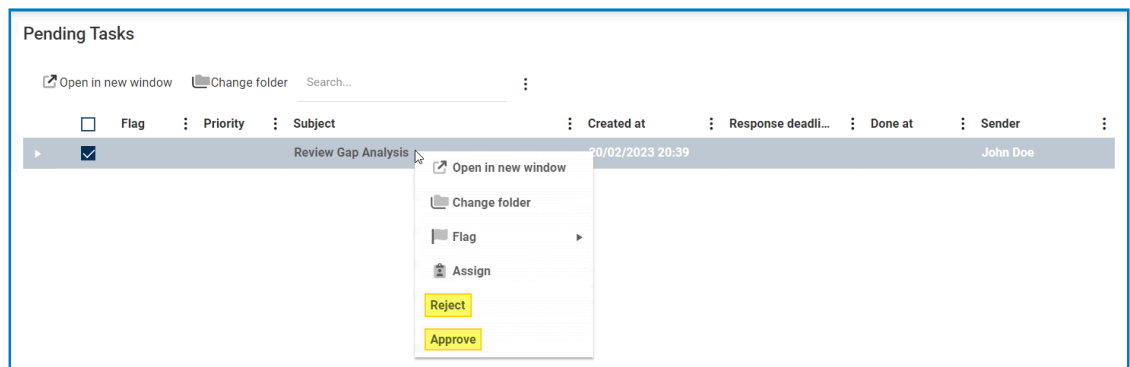
\* For more information on the **Assign** tool, [click here](#). For more information on the **Task Administration** application, [click here](#).

5. **Hide Save Button:** This button allows users to hide the save button (from the kebab menu located

in the task form).



6. **Formless Events:** This is a checkbox which, if selected, allows users to change the status of a task in the task list without opening the form associated with it.
  - Users can, in other words, execute events directly from the task list.
    - To execute formless events, simply right-click on the task in the task list and select the desired event from the context menu.



For this feature to work, users must also select the **Can Run Background** checkbox in the [Event Properties](#) form.

7. **Hide Tracking:** This is a checkbox which, if selected, allows users to prevent task tracking via the [Tracking](#) form.
8. **Application Name:** This field displays the name of the application associated with the task type.
9. **Subject Info Template:** This is a single-select dropdown field that retrieves the list of task data items (from the [Task Data Items](#) section of the **Notification** module). It allows users to select the subject info template for the task type.
  - The template selected in this field will override the title inputted in the [General](#) tab. In other words, it will be used instead as the title of the task in the application.
10. **Form Title:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated with the task type.

## 11.2.1.2.3. Analytic Data Tab

In the Analytic Data tab, users can define the metrics and data that will be used to track and analyze the task's performance.

✿ By defining metrics such as time and cost estimates for task completion, users can generate reports to evaluate the organization's activities and overall performance.

The screenshot shows the 'Task Type Properties' dialog box with the 'Analytic Data' tab selected. The dialog has three tabs: 'General', 'Advanced', and 'Analytic Data'. The 'Analytic Data' tab contains the following fields:

- 1. **Response Duration**: A text field with the value '0' and a unit label 'Minutes'.
- 2. **Has Analytic Data**: A checkbox.
- 3. **Predicted Operative Time**: A text field.
- 4. **Predicted Time**: A text field.
- 5. **Predicted Cost Per Hour**: A text field.
- 6. **Predicted Operative Cost Per Hour**: A text field.

At the top of the dialog, there are buttons for 'Apply', 'OK', and 'Cancel'.

1. **Response Duration:** This is a text field that allows users to input the time (in minutes) taken to respond to the task.
2. **Has Analytic Data:** This is a checkbox which, if selected, allows users to indicate that there is available data for the chosen analytics.
3. **Predicted Operative Time:** This is a text field that allows users to input the estimated amount of time it will take to complete the task (i.e., the time interval between the creation of the task and its completion).
4. **Predicted Time:** This is a text field that allows users to input the remaining time for the task.



5. **Predicted Cost Per Hour:** This is a text field that allows users to input the estimated cost for each hour of work required to complete the task, taking into account real-life forecasting.
  - When inputting this value, users should consider the cost imposed on the organization per hour of work remaining in the task list
6. **Predicted Operative Cost Per Hour:** This is a text field that allows users to input the estimated cost incurred for each hour during which work is stopped or delayed.
  - When inputting this value, users should consider the cost imposed on the organization per hour of productive work.

## 11.2.2. Dynamic Task Types

Dynamic task types are task types that are created based on certain conditions or rules met during the execution of the workflow.

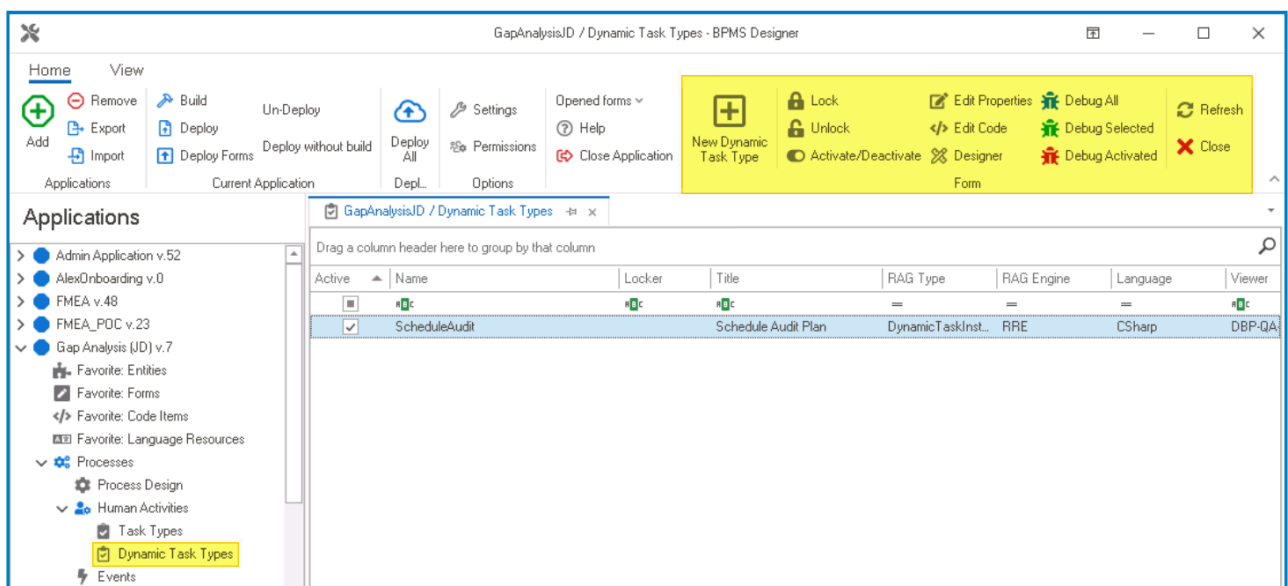
- ✿ Unlike static task types, which are fixed and predefined, dynamic task types are specific to and change based on the workflow execution context.

There are two ways by which dynamic task types can be managed in the program:

### 1. Menu Bar

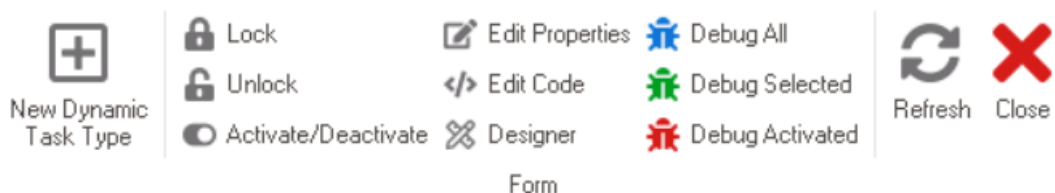
To manage dynamic task types using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Processes > Human Activities > Dynamic Task Types** (double-click to open). It will display the Dynamic Task Types tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined dynamic task types for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



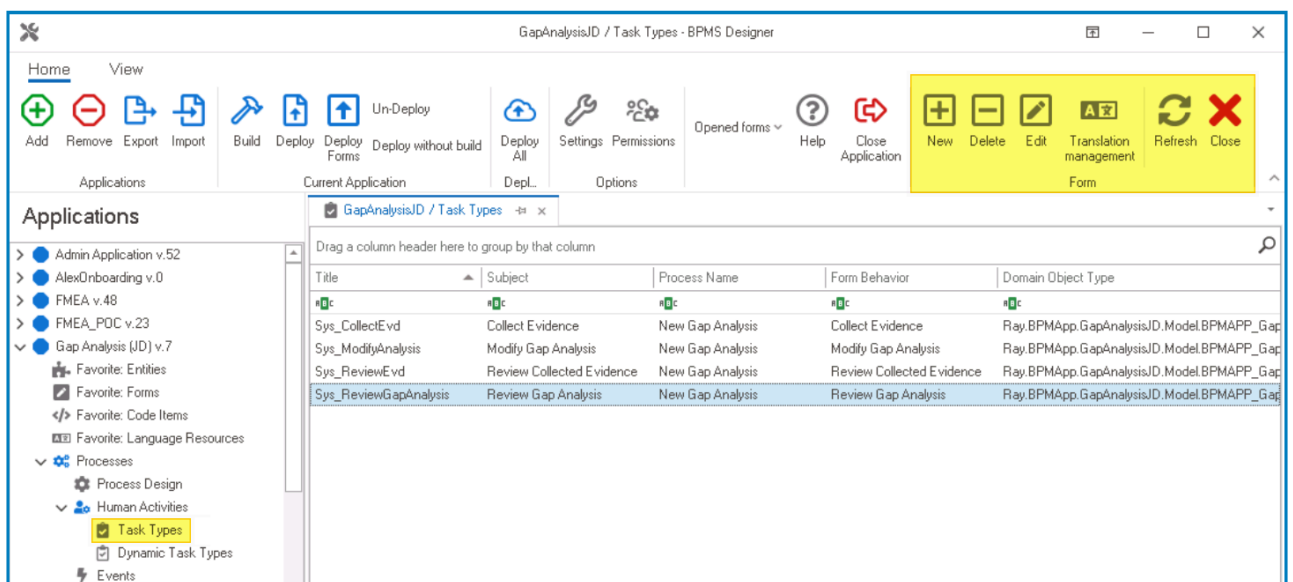
- **New Dynamic Task Type:** This button allows users to define a new dynamic task type.

- If selected, it will display the [Dynamic Task Type Properties](#) window.
- **Lock:** This button allows users to lock the dynamic task type, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the dynamic task type.
  - If selected, access restriction on the dynamic task type will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the dynamic task type.
- **Edit Properties:** This button allows users to edit the properties of the dynamic task type.
  - If selected, it will display the [Dynamic Task Type Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
- **Edit Code:** This button allows users to launch the [Code Designer](#), in which users can write code specifying the behavior and properties of the dynamic task type.
  - For more details on how to define dynamic task types using the [Code Designer](#), [click here](#).
- **Designer:** This button allows users to launch the [Rule Designer](#), in which they can define rules that specify the conditions and actions for the dynamic task type.
  - For more details on how to define dynamic task types using the [Rule Designer](#), [click here](#).
- **Debug All:** This button allows users to debug all dynamic task types in the list.
- **Debug Selected:** This button allows users to debug only the selected dynamic task type in the list.
- **Debug Activated:** This button allows users to debug only the dynamic task types that are tagged as active.
- **Refresh :** This button allows users to refresh the [Dynamic Task Types](#) tab.
- **Close:** This button allows users to close the [Dynamic Task Types](#) tab.

## 2. Task Type Properties

To manage dynamic task types using the task type properties, take the following steps:

- From the [Application Tree](#), navigate to **Processes > Human Activities > Task Types** (double-click to open). It will display the [Task Types](#) tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined task types for the application.

- From the menu bar, select the **New** button. It will display the [Task Type Properties](#) window.



- In the [General](#) tab of this window, users can create (and/or delete) the dynamic task type.



- Once you have created the task type, apply your changes, and close the [Task Type Properties](#) window.

! Only one dynamic task type can be created per task.

- Next, select the defined task type from the [Task Types](#) tab and click on the **Edit** button in the menu bar. (You can also just double-click on the task type). It will reopen the [Task Type Properties](#) window.



- In the [Advanced](#) tab of the window, users can further define the details of the dynamic task type.



## 11.2.2.1. Dynamic Task Type Properties

In the [Dynamic Task Type Properties](#) window, users can create a new dynamic task type for the application.

\* To save changes, click on the **Apply** button. To save changes and exit the [Dynamic Task Type Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

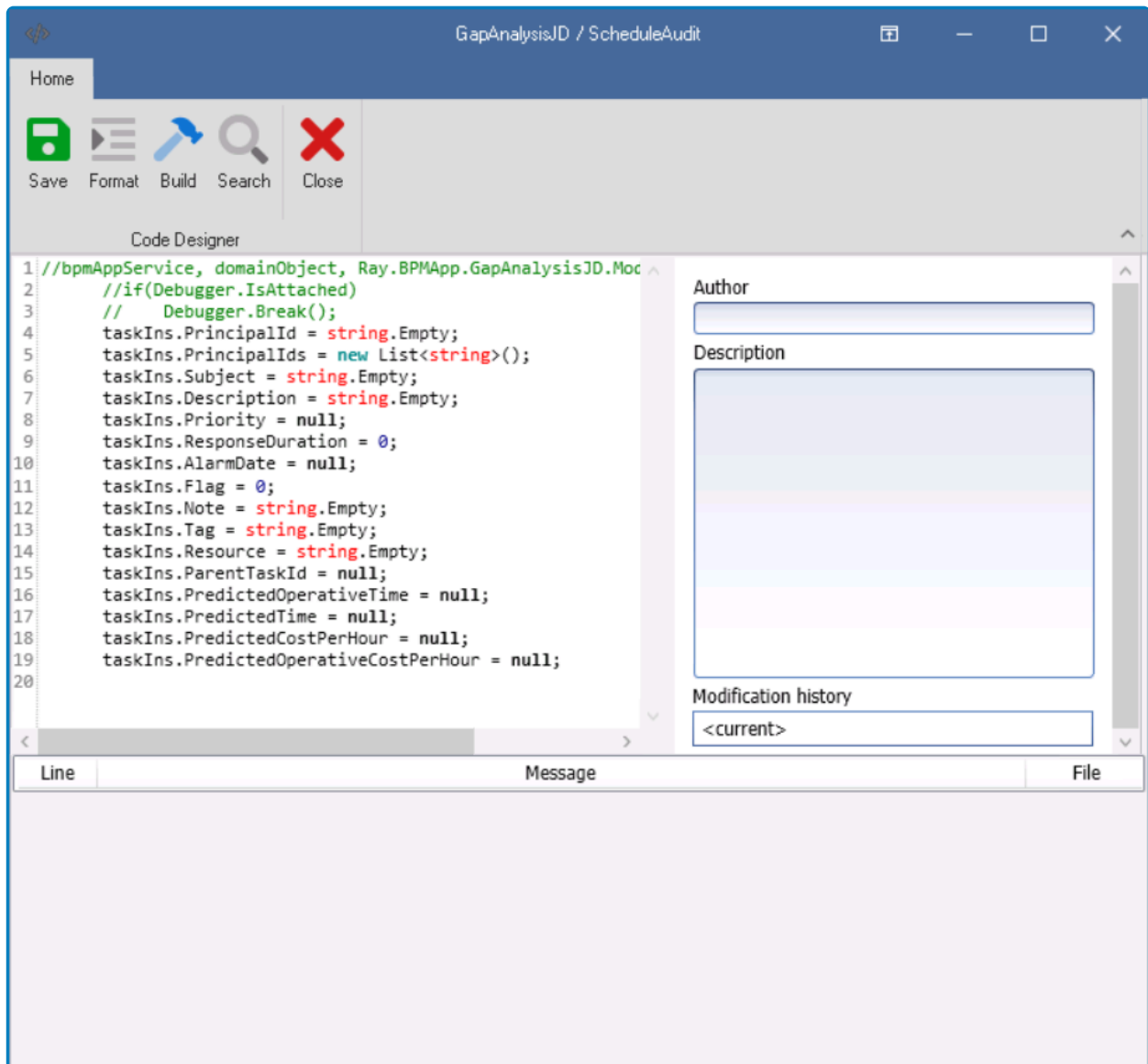
1. **Name:** This is a text field that allows users to input the name of the dynamic task type. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A dynamic task type must have a unique name. It cannot have the same name as another dynamic task type in the application.
2. **Title:** This is a text field that allows users to input the title of the dynamic task.
  - Unlike with names, there are no rules determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to tag the dynamic task type as active.
4. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the dynamic task type.

\* Once users have created the dynamic task type, they can define its behavior using either the [Code Designer](#) or the [Rule Designer](#).

## 11.2.2.2. Edit Code

In the Code Designer, users can write and build code specifying the behavior of the dynamic task type. This code is scripted in C#.

- ✿ Once users have created the dynamic task type, they can use an information script to customize the task properties (such as the task assignee, title, description, priority, tag, etc.) based on the specific workflow information. These dynamic properties are calculated and implemented during the execution of the workflow.



- ✿ For more information on the general features of the Code Designer, [click here](#).

To define the dynamic task type properties, apply the following steps:

1. In the script, type the required condition to execute the code—that is, define the specific

circumstances under which the code should be triggered.

- For more information on script form, [click here](#).

2. Enter the API in the content of the script in order to change the values of the dynamic task information.

## 11.2.2.3. Designer

In the Rule Designer, users can define rules that specify the conditions and actions for the dynamic task type.

Rule Designer: Schedule Audit Plan

Conditions

☒ Active

( Audit Plan ... Is Equal To Audit Plan ... ) AND

( taskIns ... Is Not Equal T ... ) OR

( bpmAppService ... Is Equal To ... )

Actions

Call Method Set Property

taskIns.PrincipalId ... = ...

taskIns.Subject ... = ...

taskIns.Description ... = ...

taskIns.Priority ... = ...

Errors Output

✿ For more information on the general features of the Rule Designer, [click here](#).

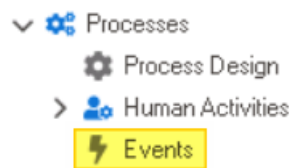
To define the dynamic task type properties, apply the following steps:

1. In the **Conditions** section of the window, set up the required conditions for the dynamic task type.
2. In the **Actions** section of the window, set up the required actions (Call Method and Set Property).
3. Click on the **Validate** button to assess the validity of the information.
  - If there are any errors, they will appear in the **Error** tab.
  - If the information is valid, the results will appear in the **Output** tab.



## 11.3. Events

In the **Events** section of this module, users can create events for the application.



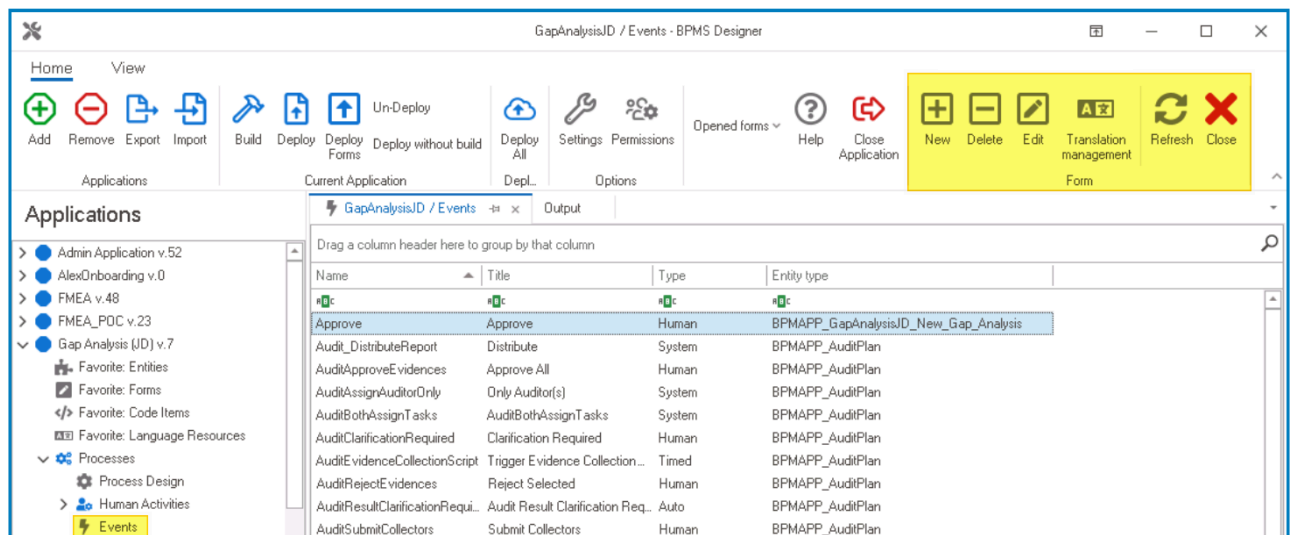
\* An event is essentially a type of action or trigger that occurs during the execution of a process. Although it can affect the process flow and be affected by it, an event remains independent of the process. For more information on how events function in a process, [click here](#).

There are two ways by which events can be defined, edited, and deleted in the program:

### 1. Menu Bar

To manage events using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Processes > Events** (double-click to open). It will display the **Events** tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined events for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.

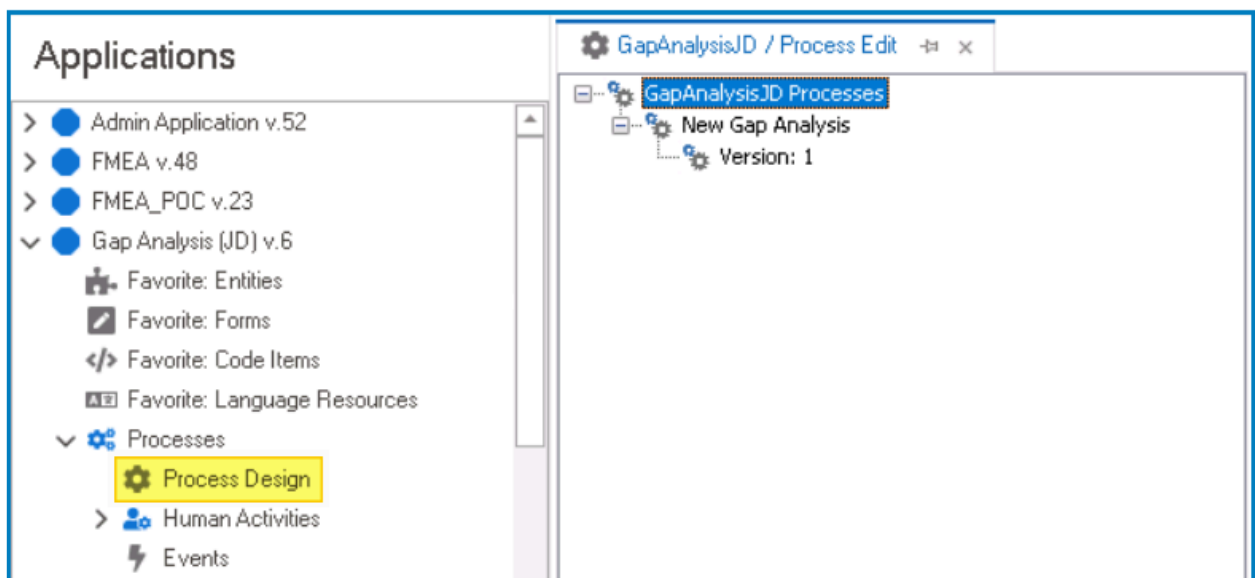


- **New:** This button allows users to define a new event.
  - If selected, it will display the [Event Properties](#) window.
- **Delete:** This button allows users to delete an existing event.
  - To use this function, first select an event from the list.
- **Edit :** This button allows users to edit an existing event.
  - To use this function, first select an event from the list.
- **Translation Management:** This button allows users to manage the translation of event titles.
  - If selected, it will display the [Translation Management](#) window, in which users will find a table listing all defined events in the application.
- **Refresh :** This button allows users to refresh the [Events](#) tab.
- **Close:** This button allows users to close the [Events](#) tab.

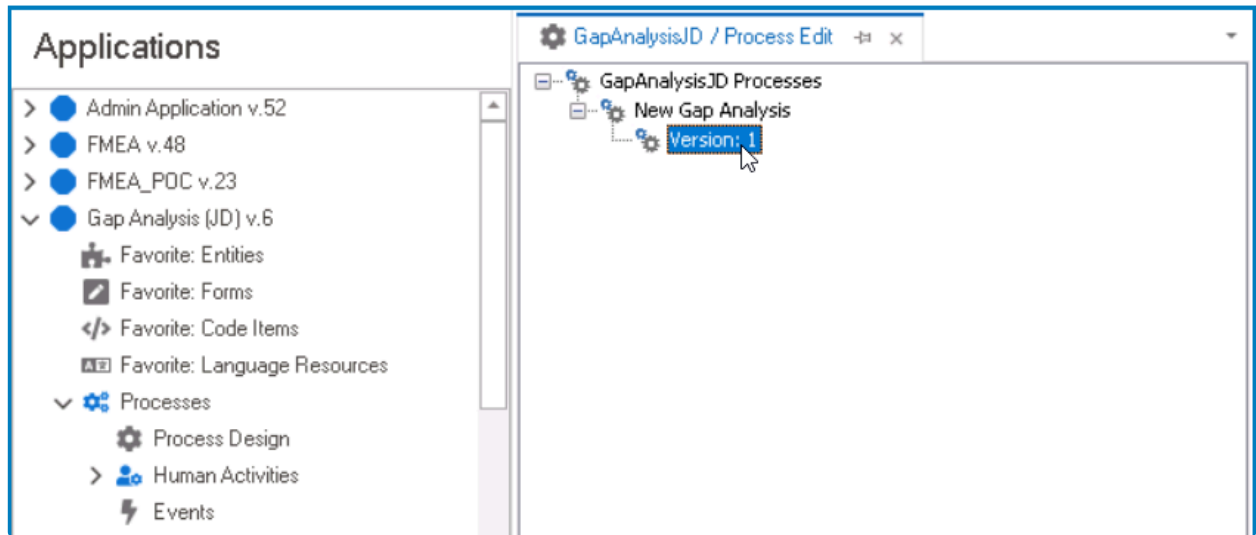
## 2. Event Properties

To manage task types using the event properties, take the following steps:

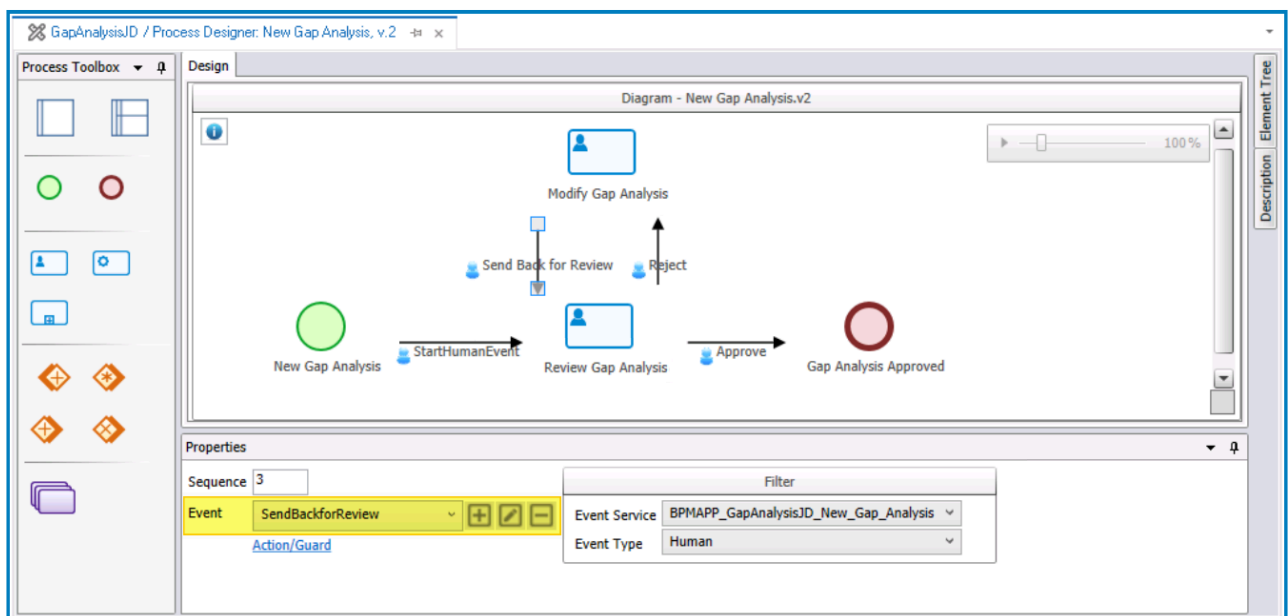
- From the [Application Tree](#), navigate to **Processes > Process Design** (double-click to open). It will display a new tab called [Process Edit](#).



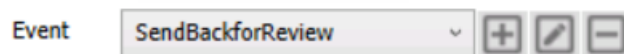
- In the [Process Edit](#) tab, double-click on a process version. It will launch the [Process Designer](#) in a new tab.






- In the diagram of the Process Designer, select an event. It will display its associated properties in the Properties panel.



- From the Properties panel, select the desired option next to **Event**.



- To define a new event, select the  button.
  - It will display the [Event Properties](#) window.
- To edit an existing event, select the  button.
- To delete an event, select the  button.

## 11.3.1. Event Properties

---

In the [Event Properties](#) window, users can create a new event for the application.



In the program, users can create four types of events: auto, human, system, and timed. The 'timed' event type is referred to as 'scheduled' in the [Process Designer](#).

Depending on the event type selected in this window, different fields may be enabled or disabled. To view how the window looks for a specific event type, please refer to the corresponding section listed below.

1. [Auto and System Event Properties](#)
2. [Human Event Properties](#)
3. [Timed Event Properties](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the [Event Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 11.3.1.1. Auto and System Event Properties

When selecting an event type in the [Event Properties](#) window, the available fields may vary depending on the type chosen. In this case, the 'auto' and 'system' event types share the same set of enabled and disabled features.

✿ An 'automated' (or 'auto') event is an event that automatically advances the process to the next step. A 'system' event, on the other hand, is an event that is triggered by the system.

The [Event Properties](#) window is composed of two tabs:

1. [General](#)
2. [Advanced](#)

✿ As long as there is no system event in the scripting environment (action after form, rule, etc.), the process will remain in the same state. **For more information about event invocation API, [click here](#).**

## 11.3.1.1.1. General Tab

In the General tab, users can define the basic properties of the event.

\* If 'auto' or 'system' are selected as event types, users will be provided access to a limited set of features in this tab.

1. **Event Type:** This is a single-select dropdown field that displays the list of event types.
  - Depending on the event type selected, different fields may be enabled or disabled in the window.
2. **Name:** This is a text field that allows users to input the name of the event. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An event must have a unique name. It cannot have the same name as another event linked to the entity.
3. **Title:** This is a text field that allows users to input the title of the event.

- Unlike with names, there are no rules determining the validity of titles.



Users can input the title in English, French, German, Arabic, and Hebrew. Currently, these are the only languages supported on the web server.

4. **Can Run Background:** This is a checkbox which, if selected, allows for the event to run in the background. A background-capable event is an event that can be executed without requiring the associated form to be viewed.



For this feature to work, users must also select the **Formless Events** checkbox in the Task Type Properties form.

5. **Has Timesheet:** This is a checkbox which, if selected, allows for a timestamp of the event to be recorded.

## 11.3.1.1.2. Advanced Tab

In the Advanced tab, users can define the advanced properties of the event.

- \* If 'auto,' 'system,' or 'timed' are selected as event types, users will be provided access to a limited set of features in this tab.

The screenshot shows the 'Event Properties' dialog box with the 'Advanced' tab selected. The 'Entity' field is highlighted with a red circle and the number 1. The 'Form Behaviour' field is set to '[Empty]'. The 'Access' section has a checked box for 'Access by principal task'. Below this is a large empty rectangular area. At the bottom, there are three buttons: 'Button color', 'Button color on mouse hover', and 'Set default color'.

1. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the event. It is a **mandatory** field.



## 11.3.1.2. Human Event Properties

---

When selecting an event type in the [Event Properties](#) window, the available fields may vary depending on the type chosen. Human events have a distinct set of enabled and disabled features.



A 'human' event is an event that requires manual input. This type of event is initiated by a human being and triggered by clicking a button on the form.

The [Event Properties](#) window is composed of two tabs:

1. [General](#)
2. [Advanced](#)

## 11.3.1.2.1. General Tab

In the General tab, users can define the basic properties of the event.

\* If 'human' is selected as an event type, users will be provided access to a wider (albeit still limited) set of features in this tab.

1. **Event Type:** This is a single-select dropdown field that displays the list of event types.
  - Depending on the event type selected, different fields may be enabled or disabled in the window.
2. **Name:** This is a text field that allows users to input the name of the event. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An event must have a unique name. It cannot have the same name as another event linked to the entity.
3. **Title:** This is a text field that allows users to input the title of the event.

- The title inputted in this field will be the title of the event in the **DBP** web application.

Reject


Approve

- Unlike with names, there are no rules determining the validity of titles.




Users can input the title in English, French, German, Arabic, and Hebrew. Currently, these are the only languages supported on the web server.

4. **Event Icon:** This field allows users to upload an icon for the event.

- To add an icon, select the  button.
  - It will display the [Application Files](#) window, in which users can add and select the image file.
    - Once uploaded and saved, the event icon will be displayed in the form as the background of the button.

Approve

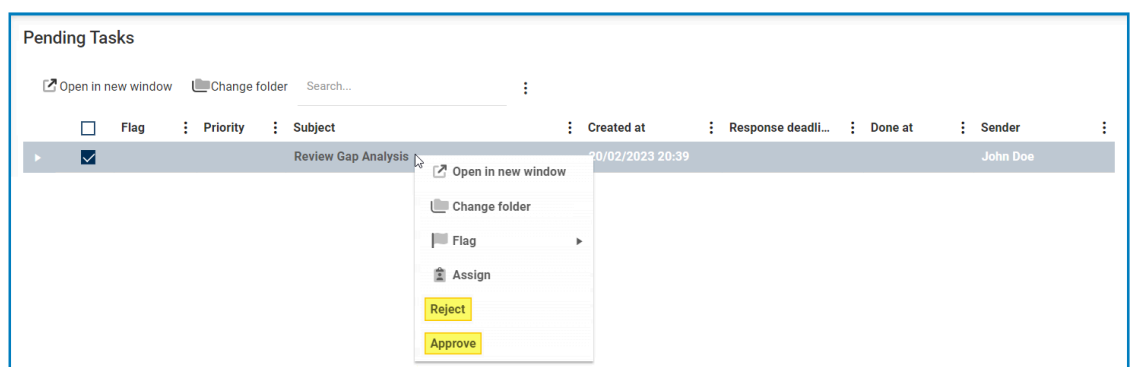
- To remove an icon, select the  button.



The system restricts image files to a maximum size of 20 KB.

5. **Can Run Background:** This is a checkbox which, if selected, allows for the event to run in the background. A background-capable event is an event that can be executed without requiring the associated form to be viewed.

- In other words, rather than having to open and interact with the form, users can execute events directly from the task list.
  - To execute formless events, simply right-click on the task in the task list and select the desired event from the context menu.

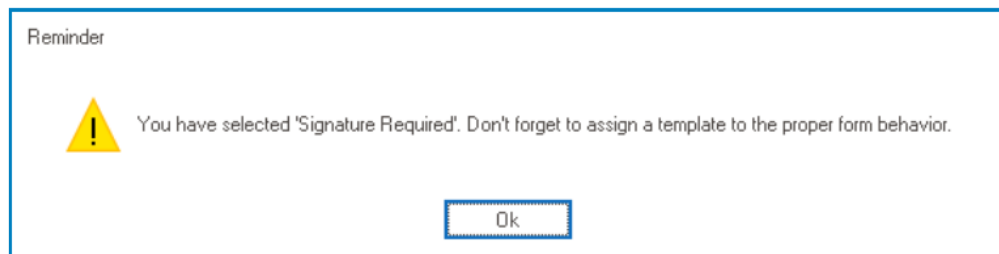


For this feature to work, users must also select the **Formless Events** checkbox in the [Task Type Properties](#) form.

6. **Has Timesheet:** This is a checkbox which, if selected, allows for a timestamp of the event to be recorded.
- If this feature is enabled, the remaining time on the timesheet must be zero before users can execute the event.

| All current works                |            |          |                |                     |
|----------------------------------|------------|----------|----------------|---------------------|
| Date                             | Start time | End time | Remaining time | Description         |
| Wednesday, February 22, 10:08:00 |            | 17:00    | 03:00          | Review Gap Analysis |

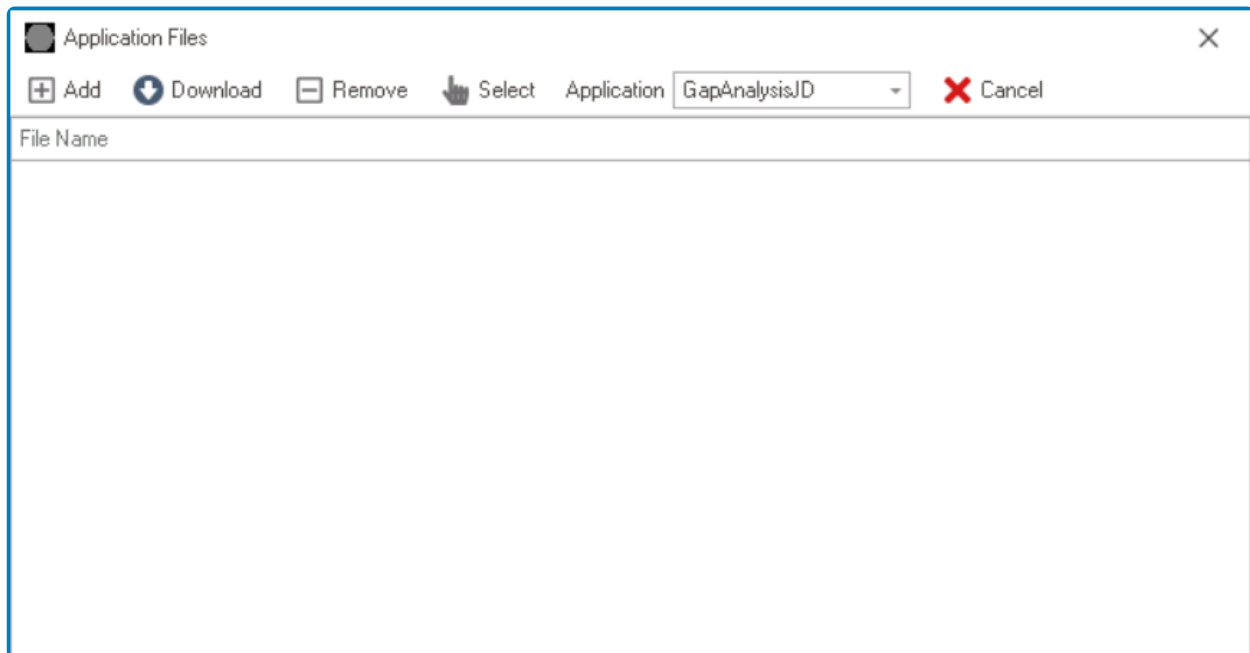
7. **Signature Required:** This is a checkbox which, if selected, allows for the event to be digitally signed by authorized user(s) before it can be executed.
- In other words, to execute the event in the web application, users will be required to provide their signature.
    - For more information on how digital signatures work in the web application, [click here](#).
  - Once this checkbox has been selected and changes are saved in the window, the following message will be displayed:



- \* To enable digital signatures for an event, users will need to both select the above checkbox and assign a template to the appropriate form behavior. Users must specifically assign a PDF template. For more information on how to assign templates to form behaviors, [click here](#).

## 11.3.1.2.1.1. Application Files

In the Application Files window, users can perform various file management tasks, including adding, downloading, removing, and selecting files.



1. **Add:** This button allows users to add a new file.
  - If selected, a file dialog will appear, enabling users to browse and choose a file from their local system.
2. **Download:** This button allows users to download a file.
  - If selected, a Save As dialog will appear, enabling users to choose a location on their local system to save the file.
    - Users can also specify the file name and type.
  - To use this function, users must first select a file with their cursor.
3. **Remove:** This button allows users to remove a file.
  - To use this function, users must first select a file with their cursor.
4. **Select:** This button allows users to select a file for upload.
  - Once a file has been selected, users will be returned to the parent window where they will find the file populating the upload field.
5. **Application:** This is a single-select dropdown field that displays the list of applications. It allows users to select the application associated with the file or file set.
  - Users can choose between the currently active application and the ray system.
    - If the current application is selected from the dropdown, the window will remain unchanged.
      - Users can add, download, remove, select, and cancel files as desired.
    - If the ray system is selected from the dropdown, a file set will populate the window.
      - Users can download or select these files. They cannot, however, remove them or add to them.
  - By default, this field will be populated with the application currently being used in the program.
6. **Cancel:** This button allows users to cancel the file management process, returning users to the


parent window.

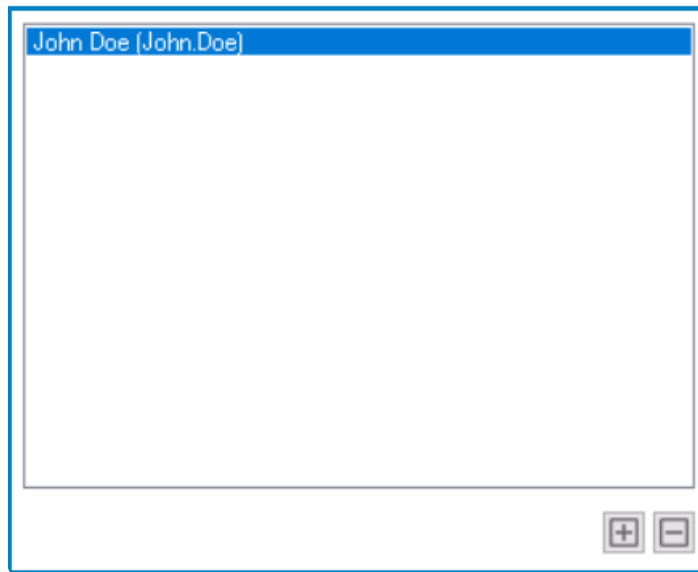
## 11.3.1.2.2. Advanced Tab


In the Advanced tab, users can define the advanced properties of the event.

- \* If 'human' is selected as an event type, users will be provided access to the full set of features in this tab.

1. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the event. It is a **mandatory** field.
2. **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated with the event.
3. **Access:** In this section, users can manage access to the event.
  - **Access By Principal Task:** This is a checkbox which, if selected, allows for the event to be visible in the form.
    - If this checkbox is left unselected, the event will only be visible to the receivers (users/ roles/groups) added in the field.

-  : This button allows users to add a new receiver.
  - It will display the Receivers window, from which users can select the specific receiver.
    - [For more details on the Receivers window, see previous instance.](#)
  - Once a receiver has been added, it will populate the field.



-  : This button allows users to remove a receiver.
4. **Button Color:** This field allows users to set the color of the button.
- If selected, it will display the [Select Color](#) window, in which users can access a color picker.
  - In the **DBP** web application, the button representing the event will take on the selected color.



5. **Button Color on Mouse Hover:** This field allows users to set the color of the button when a mouse hovers over it.
- If selected, it will display the [Select Color](#) window, in which users can access a color picker.
  - In the **DBP** web application, the button representing the event will take on the selected color when the user interacts with it.




- If a button color has not been selected yet, this field will remain disabled.



When selecting a button color, users must both choose a color from the color field and determine its attributes (such as its luminance, hue, and/or opacity). Otherwise, the selection process will not work and the button's color will remain transparent.

6. **Set Default Color:** This field allows users to reset the button color to its default state, removing the mouse hover color as well.
- In the **DBP** web application, the button representing the event will take on the default color.



A blue rectangular button with rounded corners and the text "Reject" in white.A blue rectangular button with rounded corners and the text "Approve" in white.

- If a button color has not been selected yet, this field will remain disabled.

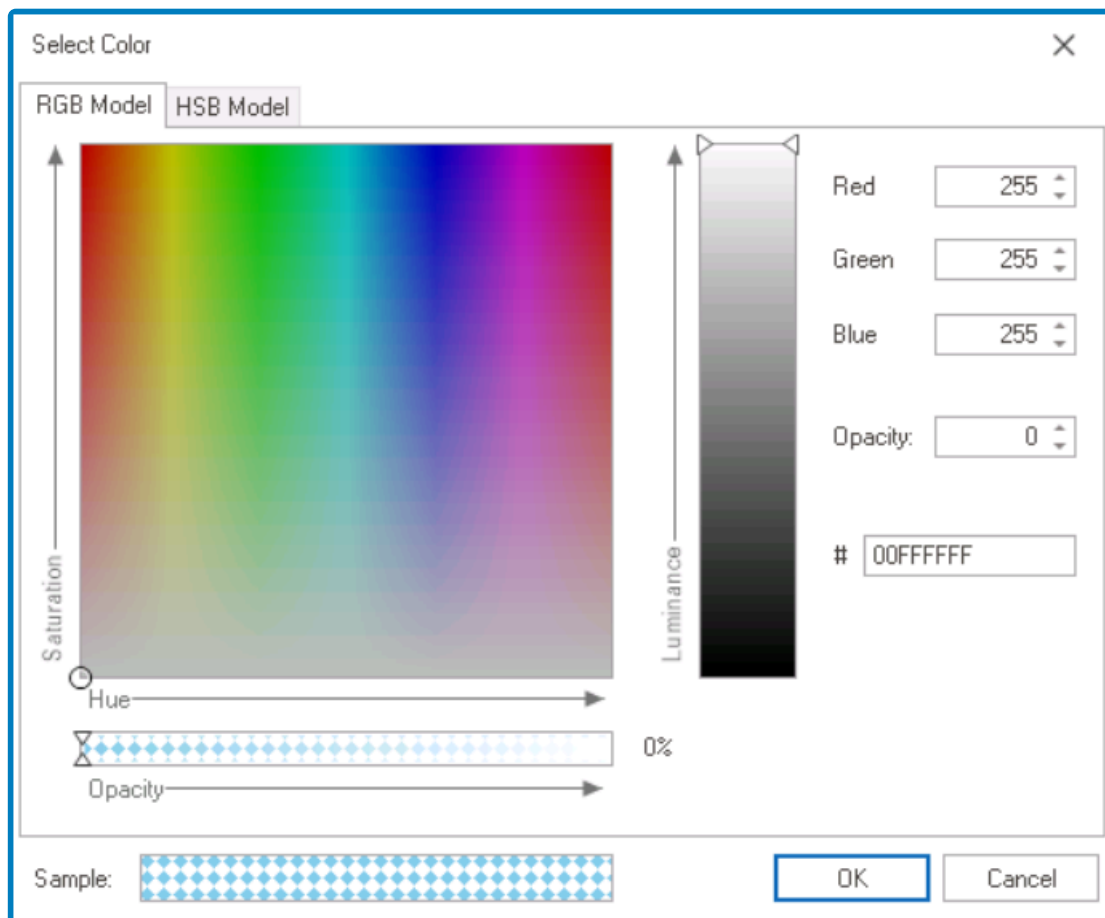
## 11.3.1.2.2.1. Select Color

In the Select Color window, users are provided access to a color picker which allows them to choose a color from a range of options.

There are two tabs in this window: RGB Model and HSB Model.

### 1. RGB Model

In the RGB Model tab, users can select a color using the color field, which displays a range of colors that can be chosen by clicking on them. Alternatively, users can adjust the levels of red, green, and blue components by entering numeric values for each component.



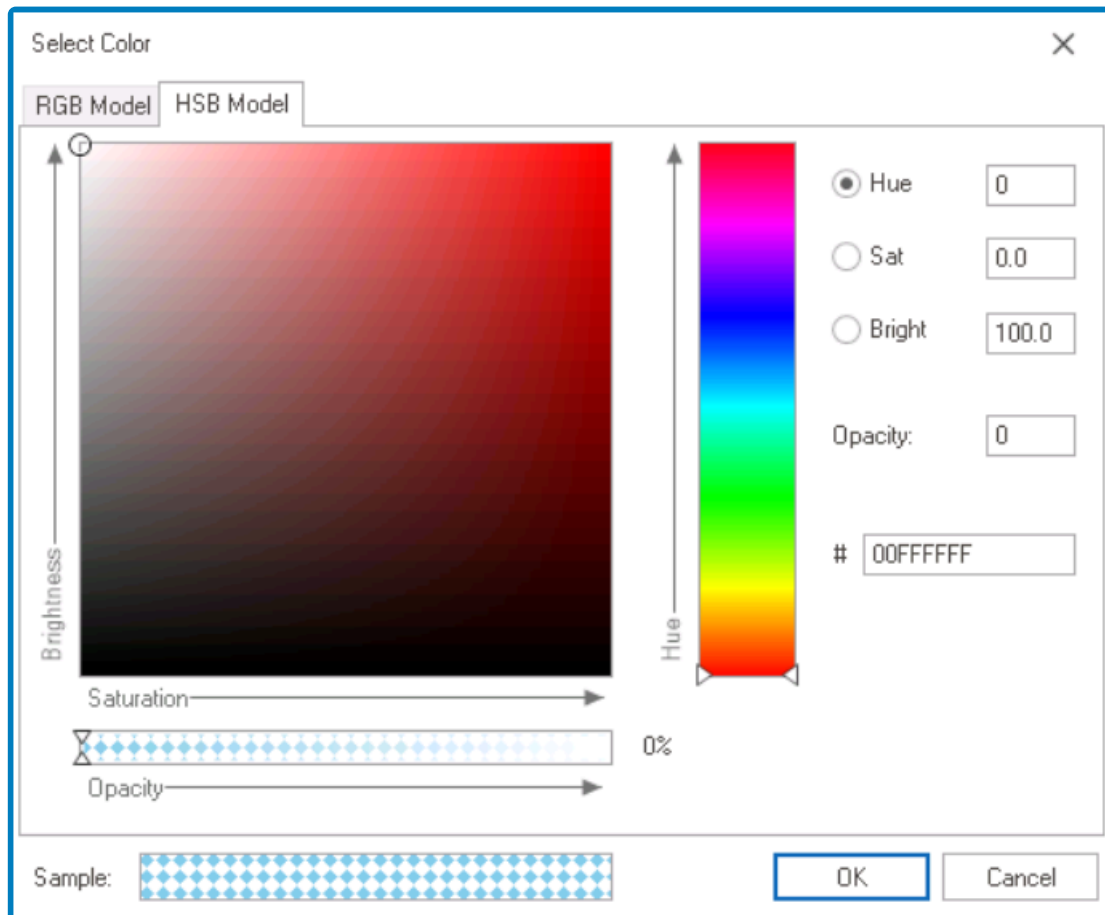
- To adjust the luminance of a selected color, use the slider.
- To adjust the levels of red, green, and blue components, use the numeric updown fields.
  - In these fields, users can enter numbers either manually or by clicking the increment/decrement arrows.
- To adjust the opacity, use either the numeric updown field or the slider.
- To input or modify the hexadecimal code of a specific color, use the # field.

! To select a color in the RGB Model tab, users must both choose a color from the color field and determine the luminance and opacity levels. The selected color will then appear in the **Sample** field. It is important to follow these steps correctly; otherwise, the

selection process will not work and the button's color will remain transparent.

## 2. HSB Model

In the HSB Model tab, users can choose a color by adjusting the hue, saturation, and brightness values using sliders or by entering numeric values for each parameter.



- To adjust the hue of a selected color, use either the numeric updown field or the slider.
- To adjust the hue, saturation, and brightness values, use either the numeric updown fields or the sliders.
- To adjust the opacity, use either the numeric updown field or the slider.
- To input or modify the hexadecimal code of a specific color, use the # field.

! To select a color in the RGB Model tab, users must both choose a color from the color field and determine the hue and opacity levels. The selected color will then appear in the **Sample** field. It is important to follow these steps correctly; otherwise, the selection process will not work and the button's color will remain transparent.

## 11.3.1.3. Timed Event Properties

When selecting an event type in the [Event Properties](#) window, the available fields may vary depending on the type chosen. Timed events have a distinct set of enabled and disabled features.

- \* A 'timed' (or 'scheduled') event is an event that is triggered by the system at certain times or over a period of time.

The [Event Properties](#) window is composed of two tabs:

1. [General](#)
2. [Advanced](#)

- \* In order to utilize timed events, it is necessary to install and run the BPMS Scheduler Service in the system services section. For more information on how to install the scheduler service, [click here](#).

- \* Once a timed event has been created, it can be assigned as the scope for the corresponding event action. For more information on how to set up an event action, [click here](#).

## 11.3.1.3.1. General Tab

In the General tab, users can define the basic properties of the event.

\* If 'timed' is selected as an event type, users will be provided access to a limited set of features in this tab.

1. **Event Type:** This is a single-select dropdown field that displays the list of event types.
  - Depending on the event type selected, different fields may be enabled or disabled in the window.
2. **Name:** This is a text field that allows users to input the name of the event. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An event must have a unique name. It cannot have the same name as another event linked to the entity.
3. **Title:** This is a text field that allows users to input the title of the event.

- Unlike with names, there are no rules determining the validity of titles.

✿ Users can input the title in English, French, German, Arabic, and Hebrew. Currently, these are the only languages supported on the web server.

4. **Can Run Background:** This is a checkbox which, if selected, allows for the event to run in the background. A background-capable event is an event that can be executed without requiring the associated form to be viewed.

✿ For this feature to work, users must also select the **Formless Events** checkbox in the [Task Type Properties](#) form.

5. **Has Timesheet:** This is a checkbox which, if selected, allows for a timestamp of the event to be recorded.
6. **Scheduling:** In this section, users can configure when the event will occur.
- **In Process:** This is a checkbox which, if selected, allows for the scheduled event to be triggered and executed automatically during the process.
    - If this checkbox is left unselected, the scheduled event will be executed outside of the process and can only be triggered by selecting the “Deploy TimedEvent” option from the event subsection.
  - **Run Automatically (Without API):** This is a checkbox which, if selected, allows for the scheduled event to be executed automatically (without API).
    - The event will be triggered directly after it has been created in or synced with the Scheduler service.
  - **Add/Edit:** This button allows users to add or edit the event schedule.
    - If selected, it will display the [Scheduler](#) window.
  - **Delete:** This button allows users to delete the event schedule.

✿ If an operation is required with each scheduled event, define its code in the event action. For more information on scheduled event APIs, [click here](#).



For this event type, it is mandatory to set up the event schedule.

## 11.3.1.3.1.1. Scheduler

In the Scheduler window, users can set up and configure the event schedule.

The screenshot shows the 'Scheduler' window with the following components:

- Start Time (1):** Contains three radio buttons: 'Run Time' (selected), 'After Run Time', and 'Start Time'. To the right of 'After Run Time' is a numeric field with '0' and a 'Second' dropdown menu.
- End Time (2):** Contains two radio buttons: 'None' (selected) and 'End Time'. To the right of 'End Time' is an empty text field.
- Simple (3):** Contains a 'Repeat counter' field with '0', a 'Forever' checkbox, a 'Repeat interval' field with '1', and a 'Hour' dropdown menu. There is also an 'According to work calander' checkbox.
- Cron (4):** Contains fields for 'Seconds', 'Minutes', 'Hours', 'Day of month', 'Month', 'Day of week', and 'Year (optional)'.
- Ok (5):** A button at the bottom right.

1. **Start Time:** In this section, users can configure the start time of the event. Users can choose whether they want the event to be triggered at run time, after run time, or at a specific start time.
  - **Run Time:** This option allows for the event to be triggered at run time.
    - If selected, the scheduler's start time for deploying scheduled events will be independent of the process—that is, if the scheduled event can occur independently of the process.
  - **After Run Time:** This option allows for the event to be triggered after a specified amount of time.
    - If selected, it will enable the two following fields:

This close-up shows the 'Start Time' section with the 'After Run Time' radio button selected. To its right is a numeric field with '0' and a 'Second' dropdown menu.

- **Amount:** This is a numeric updown field that allows users to specify the amount of time that must pass before the event can be triggered.
  - Users can enter this number either manually or by clicking the increment/decrement arrows.
- **Unit:** This is a single-select dropdown field that displays the list of time units. It

allows users to select the unit of time for the amount specified.

- **Units:** Second, Minute, Hour, or Day.
- **Start Time:** This option allows for the event to be triggered at a specific start time.
  - If selected, it will enable the following field, from which users can access a date picker.

2. **End Time:** In this section, users can configure the end time of the event. Users can choose whether they want the event to continue indefinitely or conclude at a specific end time.

- **None:** This option allows for the event to continue indefinitely without an end time.
- **End Time:** This option allows for the event to conclude at a specific end time.
  - If selected, it will enable the following field, from which users can access a date picker.

3. **Simple:** This option, if selected, allows users to set up the simple scheduler. In this form of scheduling, users can specify the number and spacing of repetitions.

- **Repeat Counter:** This is a numeric updown field that allows users to specify the number of



repetitions.

- **Forever:** This is a checkbox which, if selected, allows for the scheduled event to repeat indefinitely.
- **Repeat Interval:** This is a numeric updown field that allows users to specify the time interval between scheduled repetitions.
- **Unit:** This is a single-select dropdown field that displays the list of time units. It allows users to select the unit of time for the interval specified.
  - Units: Second, Minute, Hour, or Day.
- **According to Work Calendar:** This is a checkbox which, if selected, allows for the repetitions to be arranged according to the work calendar.



To enable the **Repeat Interval** and **Unit** fields, either input a value in the **Repeat Counter** or select the **Forever** checkbox.

4. **Cron:** This option, if selected, allows users to set up the cron scheduler. In this form of scheduling, users can define a pattern of when the event should occur.
  - By inputting values in the corresponding fields, users can specify the second, minute, hour, day of week, day of month, month, and year (optional).
5. **OK:** This button allows users to save the schedule and return to the [Event Properties](#) window, whether they will find the event schedule populating the **Scheduling** section.

Scheduling

☐ In Process ☒ Run automatically (without API)

**Add/Edit** Delete

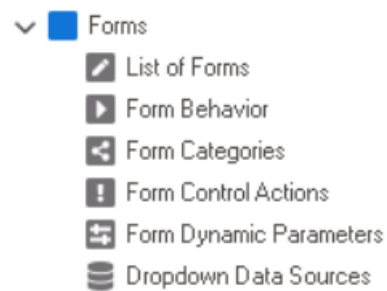
Start time: Wednesday, February 22, 2023 at 12:00 AM

Simple Schedule  
Repeat Count: 1  
Repetitions Interval: 01:00:00  
End time: unlimited

# 12. Forms

---

In the **Forms** module, users are provided access to the following elements:

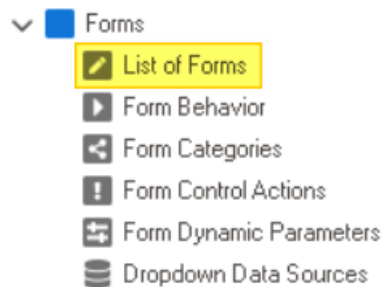


1. [List of Forms](#) \*\*
2. [Form Behavior](#) \*\*
3. [Form Categories](#)
4. [Form Control Actions](#)
5. [Form Dynamic Parameters](#)
6. [Dropdown Data Sources](#)

\*\* Both the **List of Forms** and the **Form Behavior** sections of the **Forms** module are essential to developing an application.

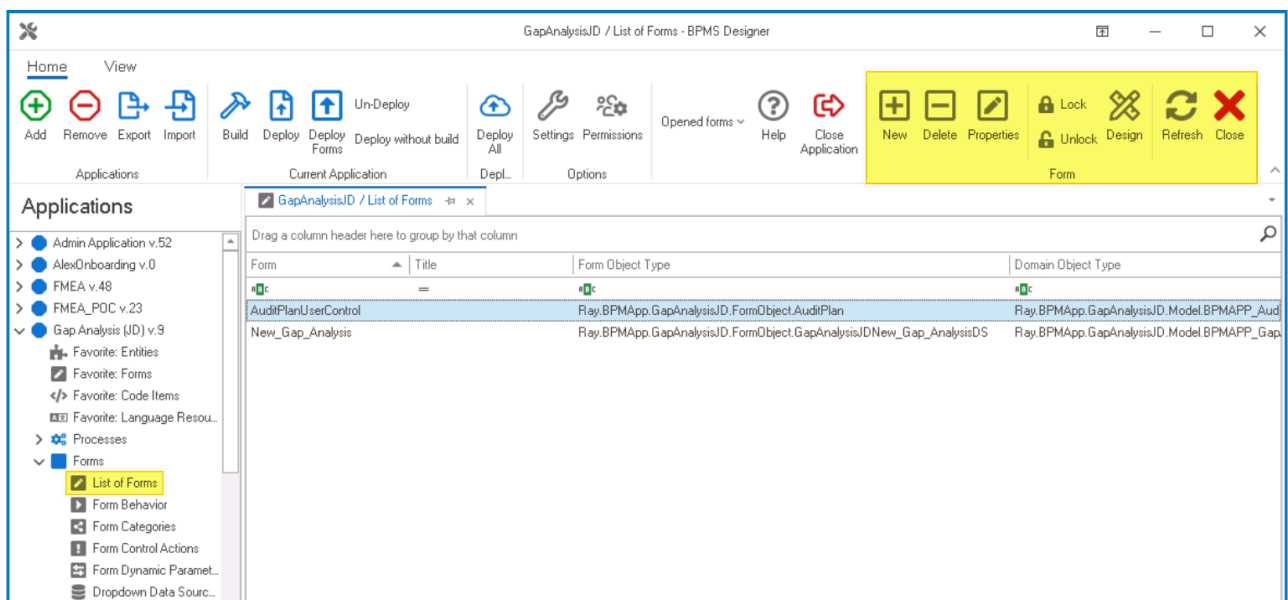
## 12.1. List of Forms

In the **List of Forms** section of this module, users can create and design forms for the application.



To manage forms in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined forms for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



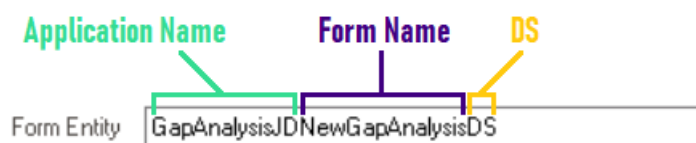
- **New:** This button allows users to define a new form.
  - If selected, it will display the [Form Properties](#) window.

- **Delete:** This button allows users to delete an existing form.
  - To use this function, first select a form from the list.
- **Properties :** This button allows users to edit the properties of an existing form.
  - If selected, it will display the Form Properties window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a form from the list.
- **Lock:** This button allows users to lock the form restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the form.
  - If selected, access restriction on the form will be lifted.
- **Design:** This button allows users to launch the Form Designer, in which they can design the features of the form.
  - For more information on the Form Designer, [click here](#).
- **Refresh :** This button allows users to refresh the List of Forms tab.
- **Close:** This button allows users to close the List of Forms tab.

## 12.1.1. Form Properties

In the Form Properties window, users can create a new form for the application.

1. **Name:** This is a text field that allows users to input the name of the form. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A form must have a unique name. It cannot have the same name as another form in the application.
2. **Form Entity:** This is a text field that allows users to input the entity of the form.
  - The system will automatically generate a form entity based on the name entered by the user, but it can be modified if necessary.
    - To generate form entities, the system adheres to the [Application Name][Form Name][DS] format.



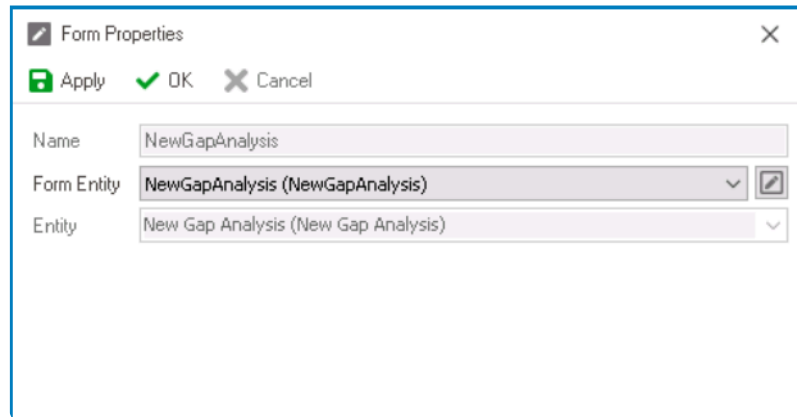
- If this field is left blank, the system will assign a default name of “NewFormModel (NewFormTableMeta)” to the form entity.



Users are restricted from adding spaces to the form entity.


3. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the form.
  - Users can choose to leave this field empty in either of the following scenarios:
    - They want to use the **Quick Forms** feature, which automatically generates entities, attributes, and form behaviors, streamlining the form development process.
      - For more information on the **Quick Forms** feature, [click here](#).

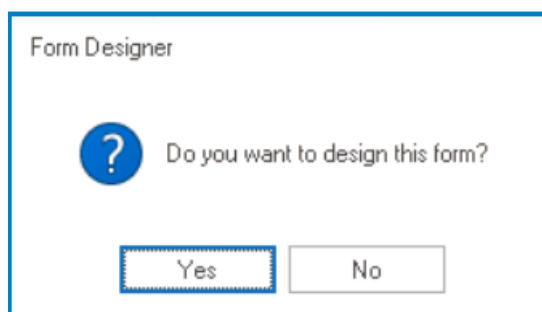
- They want to create a form with a minimalistic design, which includes only a grid for displaying information from another form and a search panel.
    - For more information on **grids** and **search panels**, [click here](#).
  - Otherwise, however, to store and retrieve form information in a database, users are required to select an entity associated with the form.
4. **Apply:** This button allows users to save their changes.
- If selected, the **Name** and **Entity** fields will be disabled and the form entity will be updated to its final format.



The image shows a 'Form Properties' dialog box. At the top, there are three buttons: 'Apply' (with a green square icon), 'OK' (with a green checkmark icon), and 'Cancel' (with a red X icon). Below these buttons, there are three input fields:
 

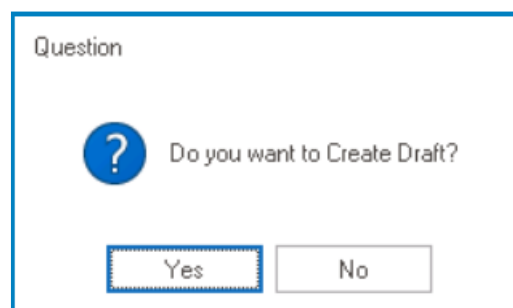
- Name:** A text box containing 'NewGapAnalysis'.
- Form Entity:** A dropdown menu showing 'NewGapAnalysis (NewGapAnalysis)' with a small square icon containing a pencil to its right.
- Entity:** A dropdown menu showing 'New Gap Analysis (New Gap Analysis)'.

- To select a different form entity within the application, click on the dropdown menu and select from the available options.
  - To edit the selected form entity, click on the  button.
    - It will display the [Form Entity Properties](#) form.
5. **OK:** This button allows users to save their changes and exit the window.
- As with the **Apply** button, selecting **OK** will disable the **Name** and **Entity** fields and update the form entity to its final format.
  - When closing the window, a prompt will appear asking users if they want to design the form.



The image shows a 'Form Designer' dialog box. It has a title bar that says 'Form Designer'. Inside, there is a blue circle with a white question mark icon, followed by the text 'Do you want to design this form?'. At the bottom, there are two buttons: 'Yes' and 'No'.

- If **Yes** is selected, a follow-up question will be presented, giving users the option to create a draft of the form.



The image shows a 'Question' dialog box. It has a title bar that says 'Question'. Inside, there is a blue circle with a white question mark icon, followed by the text 'Do you want to Create Draft?'. At the bottom, there are two buttons: 'Yes' and 'No'.

- If **Yes** is selected, the form will open in the [Form Designer](#) with all associated

entity attributes linked to the corresponding control.

- If **No** is selected, the form will open in the Form Designer without any of attributes included.
- If **No** is selected, the window will close and users will be returned to the [List of Forms](#) tab, where they will find the newly created form displayed in the table.



When dealing with numerous attributes, designing a form may become confusing. In such instances, we suggest designing the form without including the attributes and instead linking them to the relevant controls during the process.

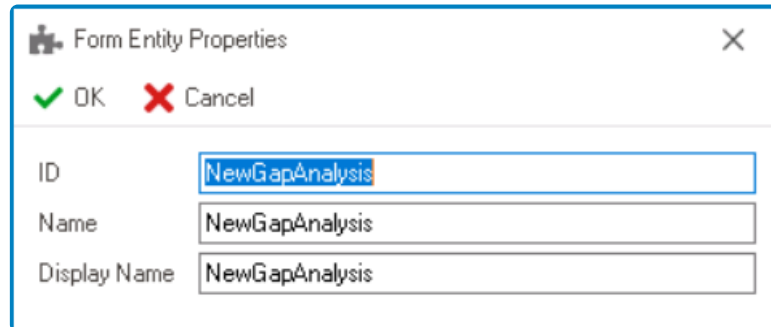
6. **Cancel** This button allows users to cancel their changes and exit the Form Properties window.



Once users have created the form, they can design it using the Form Designer.

## 12.1.1.1. Form Entity Properties

In the [Form Entity Properties](#) window, users can modify the properties of the form entity.



The screenshot shows a dialog box titled "Form Entity Properties". At the top left is a small icon of a building. To the right of the title is a close button (X). Below the title bar are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon. The main content area has three text input fields. The first field is labeled "ID" and contains the text "NewGapAnalysis". The second field is labeled "Name" and also contains "NewGapAnalysis". The third field is labeled "Display Name" and also contains "NewGapAnalysis".

1. **ID:** This field shows the ID of the form entity.
  - Any changes made to this field will directly impact the **Name** field
    - To ensure proper naming, users must follow certain rules:
      - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
2. **Name:** This field shows the current name of the form entity.
  - To modify this field, users should make changes to the **ID** field.
3. **Display Name:** This field shows the display name of the form entity.
  - There are no rules determining the validity of the display name.
4. **OK:** This button allows users to save their changes and return to the [Form Properties](#) window.
5. **Cancel:** This button allows users to cancel their changes and exit the [Form Entity Properties](#) window.



## 12.1.2. Form Designer

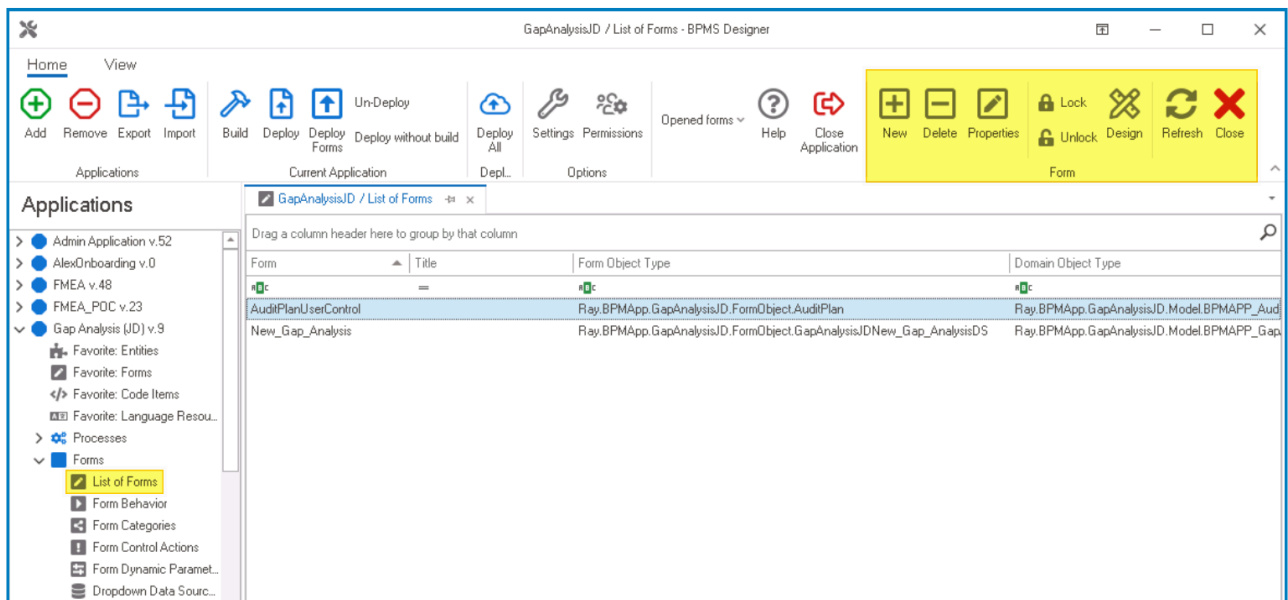
The Form Designer is a tool which enables users to customize the layout and controls of their web application. In addition to customization and design, the Form Designer allows users to establish connections between controls and attributes, as well as specify the behavior of a control during a specific stage of a process.

\* Controls encompass various physical elements of the web application, including textboxes, radio buttons, and dropdown menus.

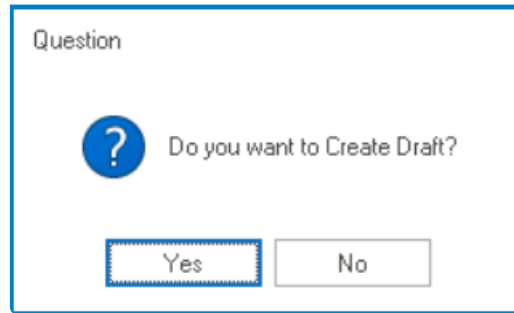
Once users have created the required form, they can launch the Form Designer.

To access the Form Designer, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



2. Choose the form you want to design. You can either select the desired form and click on the **Design** button in the menu bar, or right-click on the form and select the **Design** option in the context menu.
3. Upon selecting **Design**, a prompt window will appear asking users whether they want to create a draft of the form.



- If **Yes** is selected, the form will open in the Form Designer with all associated entity attributes linked to the corresponding control.
- If **No** is selected, the form will open in the Form Designer without any of attributes included.

! When dealing with numerous attributes, designing a form may become confusing. In such instances, we suggest designing the form without including the attributes and instead linking them to the relevant controls during the process.

Once in the Form Designer, the menu bar will expand to show additional options.



- **Save:** This button allows users to save the form.
- **Translation Management:** This button allows users to manage the translation of controls.
  - If selected, it will display the [Translation Management](#) window, in which users will find a table listing all the controls in the form.
    - Inside this table, users will find columns for Control ID, Control Type, Property, as well as the available languages for translation.

\* Users can also manually translate controls in the [General Control Properties](#).

- **Generate Word Template:** This button allows users to automatically generate a Word template of the form.
  - If selected, a Save As dialog will appear, enabling users to choose a location on their local system to save the file.

\* This generated template will include all labels, captions, and attributes associated with each control in the form. To customize the layout and look of the template, users only need to move the tags to their preferred location. For general tips and tricks on customizing templates, [click here](#).

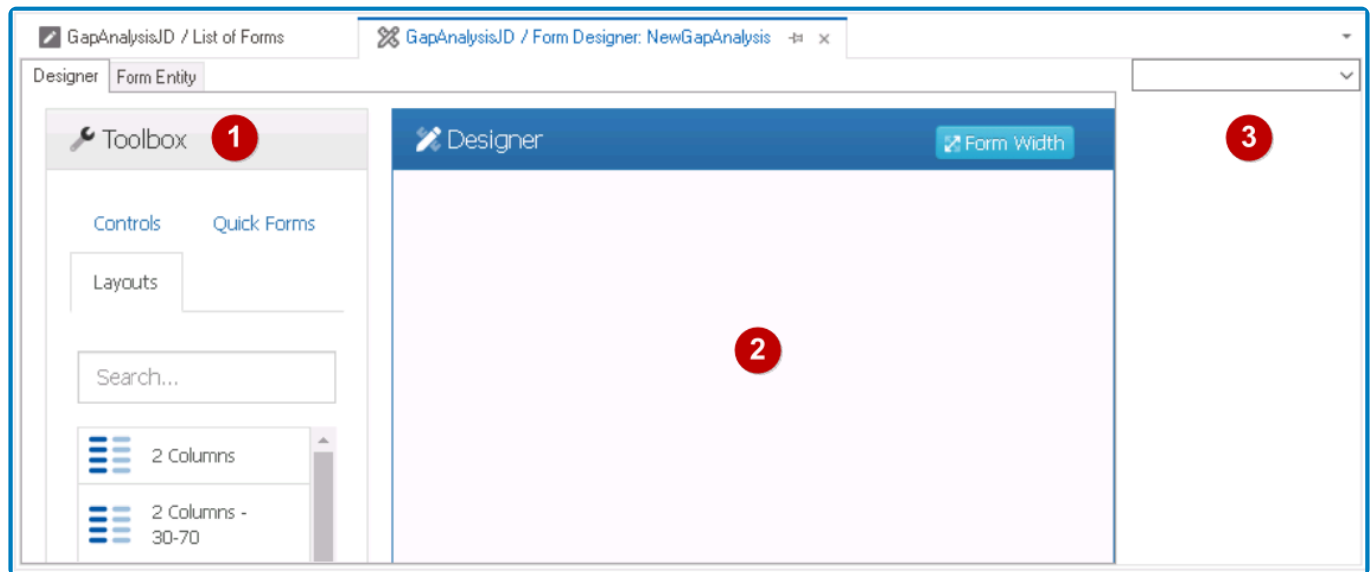
Within the Form Designer, there are two tabs available to users: Designer and Form Entity.

- To design the form, use the [Designer](#) tab.
- To modify the form properties and create local attributes, use the [Form Entity](#) tab.

## 12.1.2.1. Designer Tab

In the Designer tab, users can personalize the appearance and functionality of their web application by selecting its layout and controls.

- ✿ Beyond customization and design, users can also establish connections between controls and attributes, as well as determine control behavior during a specific process stage.



There are three primary components to the Designer tab:

### 1. **Toolbox**

In the toolbox, users can access the layouts and controls needed to build and design a form. There are three tabs comprising the toolbox:

- [Controls](#)
- [Quick Forms](#)
- [Layouts](#)

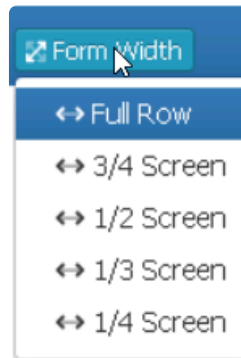
### 2. **Designer**

In the Designer, users can construct and customize the form by dragging and dropping various layouts and controls from the toolbox onto the canvas. As users add and arrange a layout or control, they configure its properties in the Control Properties panel.

- ✿ If users opt to create a draft for the form, this section will contain all controls that correspond to the selected entity's attributes.

To adjust the width of the form in the web application, click on the **Form Width** button. It will display the

following context menu:



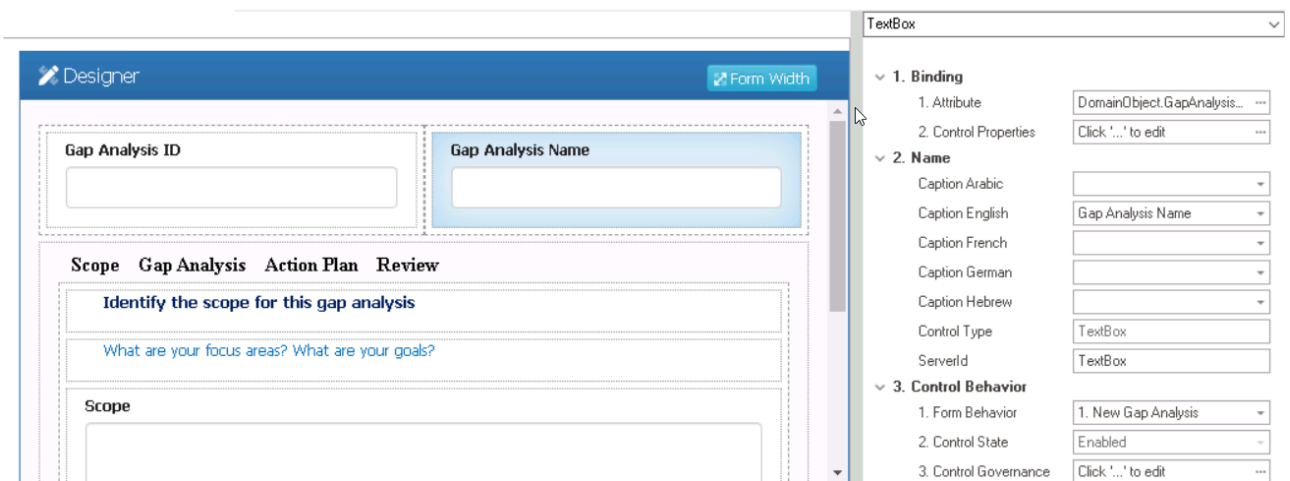
### 3. Control Properties

In the Control Properties panel, users can view and modify the properties of a selected layout or control. To select a layout or control, either click on it from the design environment or choose it from the dropdown menu in the panel.

✿ Depending on the item selected, the properties displayed in this panel may differ.

To resize this panel, take the following steps:

- Hover the cursor over the edge of the panel's left-side border (as shown below).



- Click and drag the cursor until you get the desired size.

Designer

Form Width

Gap Analysis ID

Gap Analysis Name

TextBox

Scope

Gap Analysis

Action Plan

Review

Identify the scope for this gap analysis

What are your focus areas? What are your goals?

Scope

TextBox

1. Binding

1. Attribute

DomainObject.GapAnalysis...

2. Control Properties

Click '...' to edit

2. Name

Caption Arabic

Caption English

Gap Analysis Name

Caption French

Caption German

Caption Hebrew

Control Type

TextBox

ServerId

TextBox

3. Control Behavior

1. Form Behavior

0. Modify Gap Analysis

2. Control State

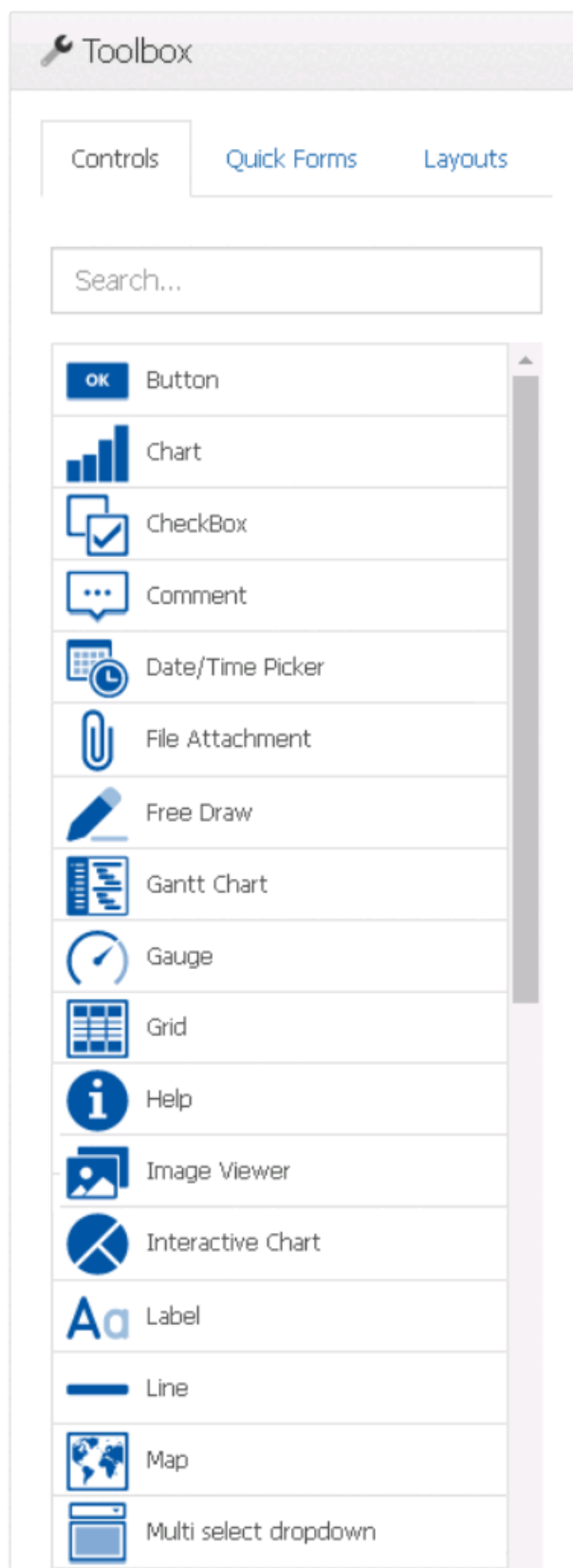
Enabled

3. Control Governance

Click '...' to edit

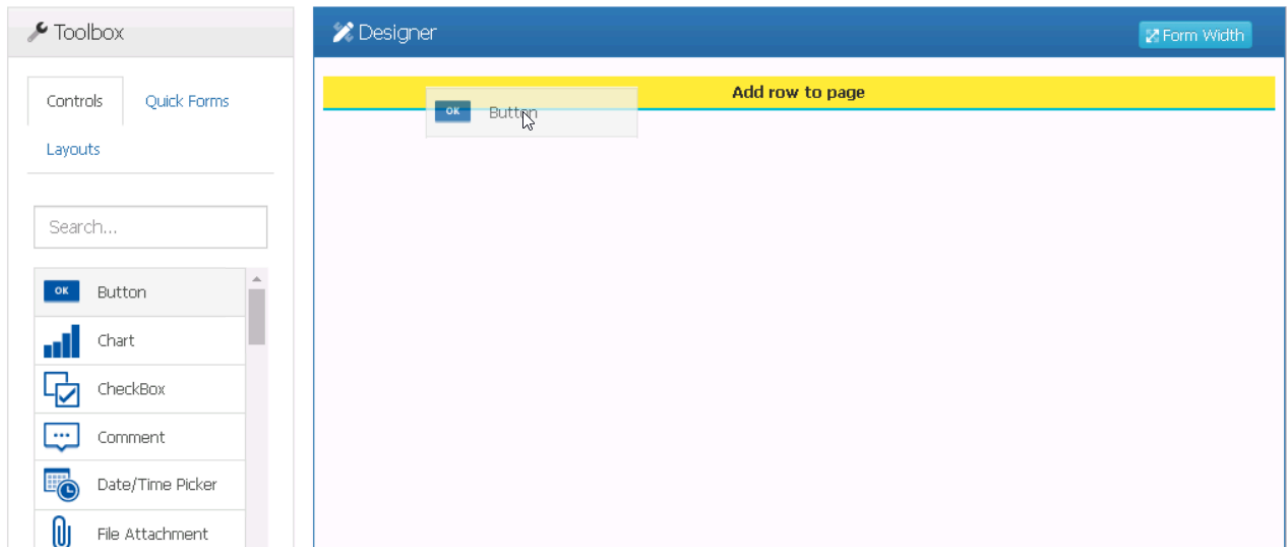
## 12.1.2.1.1. Controls

In the Controls tab of the toolbox, users are provided access to a variety of controls that can be used for designing a form.



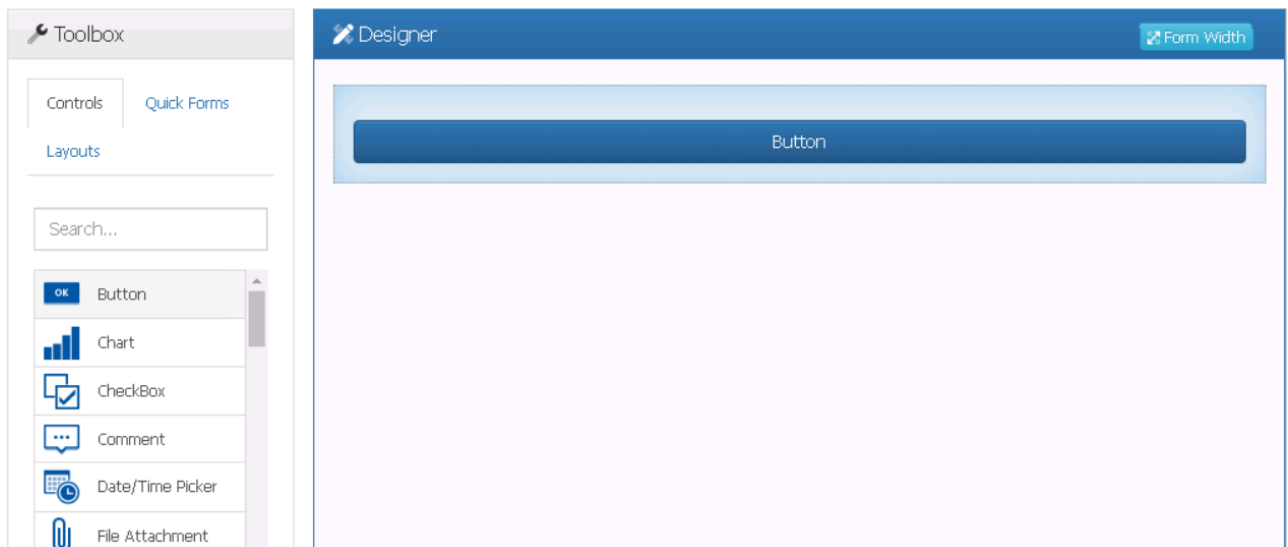
To add controls to the design environment, take the following steps:

1. From the toolbox, select a control and drag it towards a viable drop location in the design environment.



\* As users drag an object towards the design environment, the viable drop locations will be indicated by the **Add Row to Page** fields.

2. Drop the control onto the viable area.



\* Make sure the **Add Row to Page** field is highlighted in yellow before dropping the object. Otherwise, the placement operation will not succeed.

3. In this manner, users can design the functionality of the form, adding as many controls as needed.

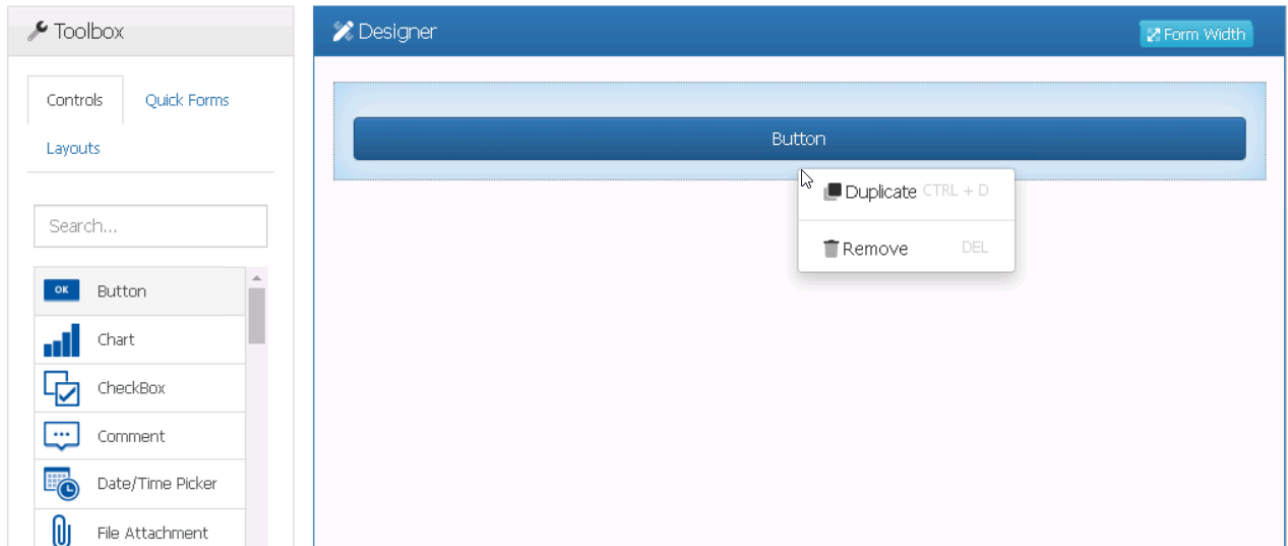
\* When adding an object to an existing column in the design environment, the viable drop



locations will be highlighted in yellow. When adding an object in between existing controls, the viable drop locations will be indicated by the **Insert Here** fields.

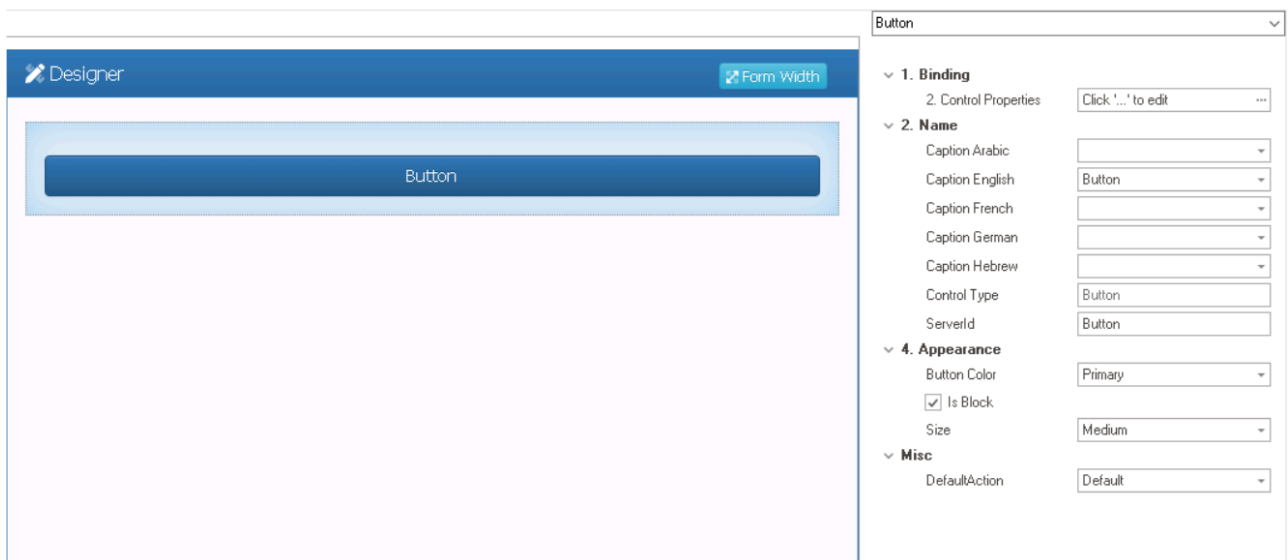
Once a control has been added to the design environment, users can perform the following actions:

### 1. Duplicate or Remove Control



- For more information on the options in this context menu, [click here](#).

### 2. Configure Properties



- To view the properties for a specific control, please refer to the corresponding section listed below.
  - **Controls**
    1. [Button](#)
    2. [Checkbox](#)
    3. [Comment](#)
    4. [Date/Time Picker](#)
    5. [File Attachment](#)
    6. [Free draw](#)

7. [Gantt Chart](#)
8. [Gauge](#)
9. [Grid](#)
10. [Help](#)
11. [Image Viewer](#)
12. [Interactive Chart](#)
13. [Label](#)
14. [Line](#)
15. [Map](#)
16. [Multi-Select Dropdown](#)
17. [Multi-Line Textbox](#)
18. [Nested Form Link](#)
19. [Numeric Textbox](#)
20. [Password Box](#)
21. [Picture](#)
22. [Radio Button List](#)
23. [Rich Textbox](#)
24. [Search Panel](#)
25. [Single-Select Dropdown](#)
26. [Static Image](#)
27. [Tab Control](#)
28. [Table](#)
29. [Textbox](#)
30. [Time Duration](#)
31. [Tree](#)
32. [User Picture](#)
33. [User Signature](#)

- For general control properties that apply to most controls, however, please refer to the [dedicated section](#) in the manual.

## 12.1.2.1.1.1. General Control Properties

---

This section of the manual provides an overview of the general properties that are commonly used to configure controls.

The general control properties include the following categories:

1. [Binding](#)
2. [Name](#)
3. [Control Behavior](#)
4. [Appearance](#)
5. [Caption Appearance](#)

# 12.1.2.1.1.1. Binding

In the **Binding** category, users can link a control to its corresponding attribute and define its binding properties.

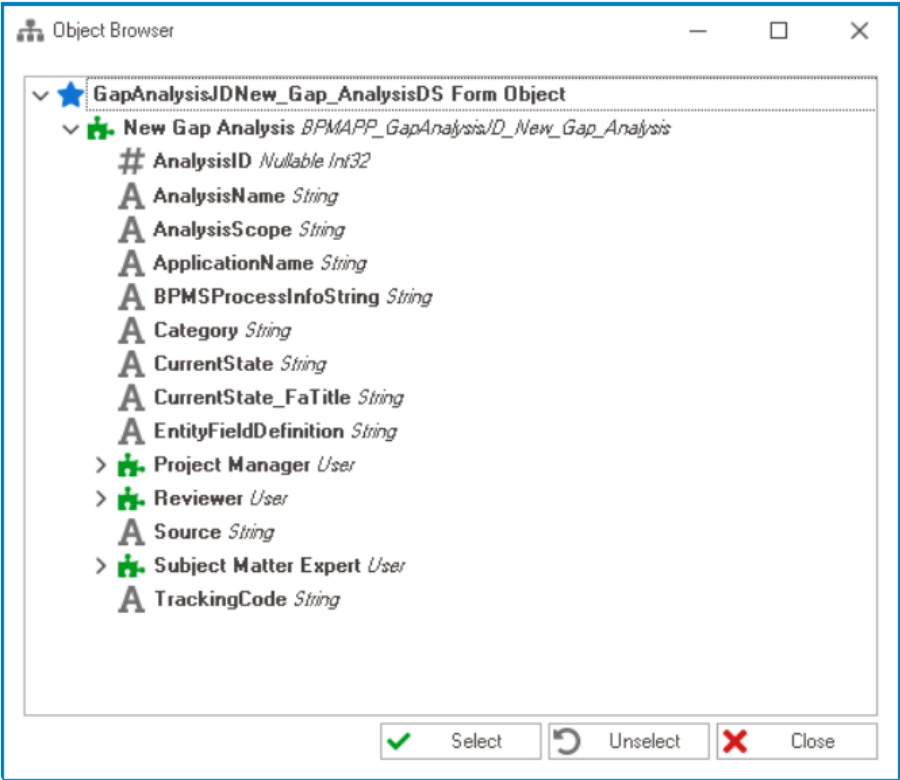
▼ 1. Binding

1. Attribute

2. Control Properties

Click '...' to edit

1. **Attribute:** This field allows users to select the attribute associated to the control.
- To access the list of relevant attributes, click on the ellipses ( ... ) inside the field.



✿ In the Object Browser window, users can access a list of the associated entity’s attributes—specifically, the attributes relevant to the control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

DomainObject.AnalysisName ...

✿ Controls can be linked to attributes created in either the [Entity Designer](#) or the [Form Entity](#) tab.

## 2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (\*\*\*) inside the field.

**Control Properties**

Apply OK Cancel

**Form Control** SelectBusinessDivision  
**Entity and Attribute** DomainObject.PrincipalBusinessDivision.ID  
**Object type** System.String

**General**

1 ☐ Call at loading page  
 2 ☐ Active  
 3 ☐ AutoPostBack (Update all controls)

Code item list 4

| Select                              | Rule index | Name   | Form Control                  | Form Control Type       |
|-------------------------------------|------------|--|-------------------------------|-------------------------|
| <input checked="" type="checkbox"/> | =          | +  |                               |                         |
| <input type="checkbox"/>            | 100000     | QMS_QualityEventa0915e258467453a9943ad810131b6f9 | EventType                     | RayAutoCompleteDropDown |
| <input type="checkbox"/>            | 100000     | QMS_QualityEvent57934ea33f5d4a6f9897ad8600elf2ed | GridActionItems               | RayGrid                 |
| <input type="checkbox"/>            | 100000     | QMS_QualityEventd21c799578494d45a99bad860150b713 | GridCustomQuestionsAssessment | RayGrid                 |
| <input type="checkbox"/>            | 100000     | QMS_QualityEvent2e3265b1282644e28d88ad870029a8e9 | InspectionId                  | RayTextBox              |

Record 1 of 29

Reload Dropdown/Grid/Table 5

| Select                              | Rule index | Form Control                  |
|-------------------------------------|------------|-------------------------------|
| <input checked="" type="checkbox"/> | =          | +                             |
| <input type="checkbox"/>            | 100000     | EventType                     |
| <input type="checkbox"/>            | 100000     | Severity                      |
| <input type="checkbox"/>            | 100000     | GridActionItems               |
| <input type="checkbox"/>            | 100000     | GridCustomQuestionsAssessment |
| <input type="checkbox"/>            | 100000     | SingleSelectDropdown          |

Record 1 of 30

- Call at Loading Page:** This is a checkbox which, if selected, allows for the linked code to run automatically whenever the form is loaded in the **DBP** web application.
  - In the web application, the running of the code will be indicated by the loading icon displayed when the form is first opened.



This feature is typically used to perform various tasks when a form is first loaded, such as initializing controls or updating visibility settings. For example, users can apply this feature to set the initial values for a dropdown or date picker.

- Active:** This is a checkbox which, if selected, allows users to activate the code linked to the control, enabling it to be executed.
  - If this checkbox is selected, the adjacent field will be enabled. In this field, users can define and modify the linked code.

☒ Active


Edit Code Designer Remove

- Edit Code:** This button allows users to launch the Code Designer, in which users can write code specifying the behavior and properties of the control.
  - For more details on how to configure controls using the Code Designer, [click here](#).
- Designer:** This button allows users to launch the Rule Designer, in which


they can define rules that specify the conditions and actions for the controls.

- For more details on how to configure controls using the [Rule Designer](#), [click here](#).
- **Remove:** This button allows users to remove the code.
- If this checkbox is not selected, the linked code will not be executed when the form is opened.



A control that has a linked code will be visually identified in the design environment by the  symbol.

◦ **Control Preview: Linked Code Design View**






3. **Autopostback (Update All Controls):** This is a checkbox which, if selected, allows for the automatic updating of all controls in the form.

- To enable a control to be updated according to the linked code, simply select it from the **Code Item List** field.

☒ AutoPostBack (Update all controls)

Code item list

| Select                              | Rule index | Name  | Form Control  | Form Control Type   |
|-------------------------------------|------------|---|---|---|
| <input checked="" type="checkbox"/> | =          |  |  |  |
| <input type="checkbox"/>            | 0          | QMS_QualityEventf58adc00e55a434b8824aea7013eaa28                                    | SelectBusinessDivision  | RayAutoCompleteDropDown   |
| <input type="checkbox"/>            | 100000     | QMS_QualityEvent372b8792734f4db0921aacc901093f64                                    | BtnImportAllControl   | RayButton   |
| <input type="checkbox"/>            | 100000     | QMS_QualityEvent9d6b64f8cf12490bbf4baec9010947f0                                    | BtnImportSelectedControl  | RayButton   |
| <input type="checkbox"/>            | 100000     | QMS_QualityEvent8be413be7ba4c5bba3daec901095810                                     | Button5   | RayButton   |

- To enable a control (specifically a dropdown, grid, or table) to be reloaded according to the linked code, simply select it from the **Reload Dropdown/Grid/Table** field.

Reload Dropdown/Grid/Table

| Select                              | Rule index | Form Control  |
|-------------------------------------|------------|---|
| <input checked="" type="checkbox"/> | =          |  |
| <input type="checkbox"/>            | 100000     | SelectBusinessDivision  |
| <input checked="" type="checkbox"/> | 100000     | SelectBusinessLine  |
| <input type="checkbox"/>            | 100000     | SingleSelectDropdown2   |
| <input type="checkbox"/>            | 100000     | SelectControlFolder   |
| <input type="checkbox"/>            | 100000     | SelectRuleFolder  |

- Once the linked code has been executed, any change made to the control to which the code is linked will automatically trigger an update of all controls selected in the **Code Item List** and **Reload Dropdown/Grid/Table** fields.



The autopostback feature ensures that connected controls, such as those that are hidden or shown based on user selection, are properly updated after the code has been executed.

◦ Autopostback Feature Preview: Web View

New CAPA

CAPA Details

Actions

CAPA ID

CAPA Name

CAPA Start Date

CAPA Initial Closure Date

Event Type

CAPA-00002

24/08/2023

Select...

Business Division

Business Line

Category

Select...

RCA Owner

CAPA Reviewer

Eff. Check Responsible

Site

Select...

Select...

Select...

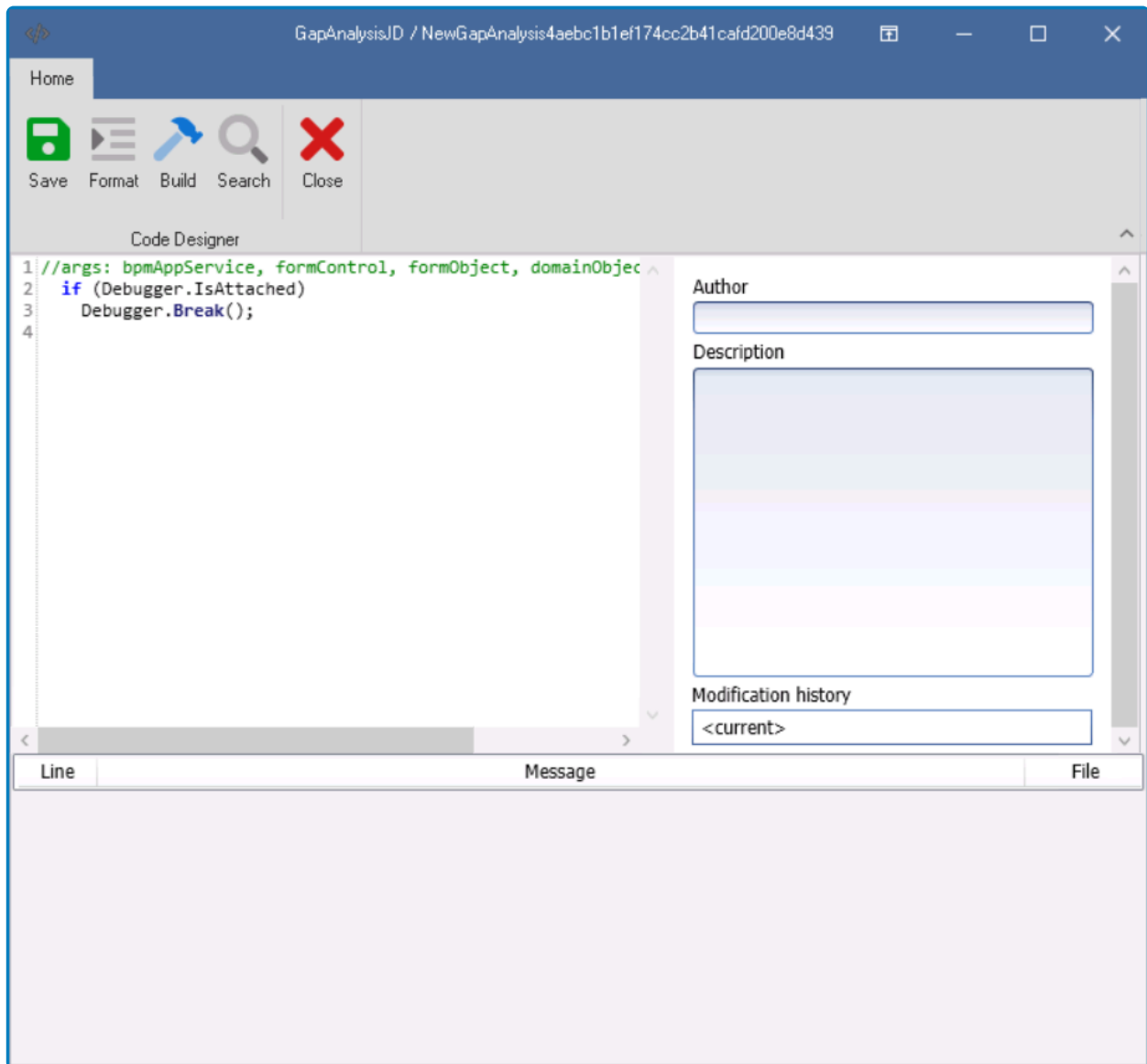
Select All

Deselect All

4. **Code Item List:** This table displays all codes that are currently linked to controls in the form.
  - Once the linked code for the control has been executed, all selected items in this list are updated accordingly.
5. **Reload Dropdown/Grid/Table:** This table displays all dropdowns, grids, and tables that are being used in the form.
  - Once the linked code for the control has been executed, all selected controls in this grid are reloaded accordingly.

## 12.1.2.1.1.1.1. Edit Code

In the Code Designer, users can write and build code specifying the behavior of the control. This code is scripted in C#.



For more information on the general features of the Code Designer, [click here](#).

In the code environment, there are comments written in the code that suggest which objects or variables can be used to accomplish certain tasks. These comments are intended to help programmers understand how the code works and to make modifications or additions to the code more easily.

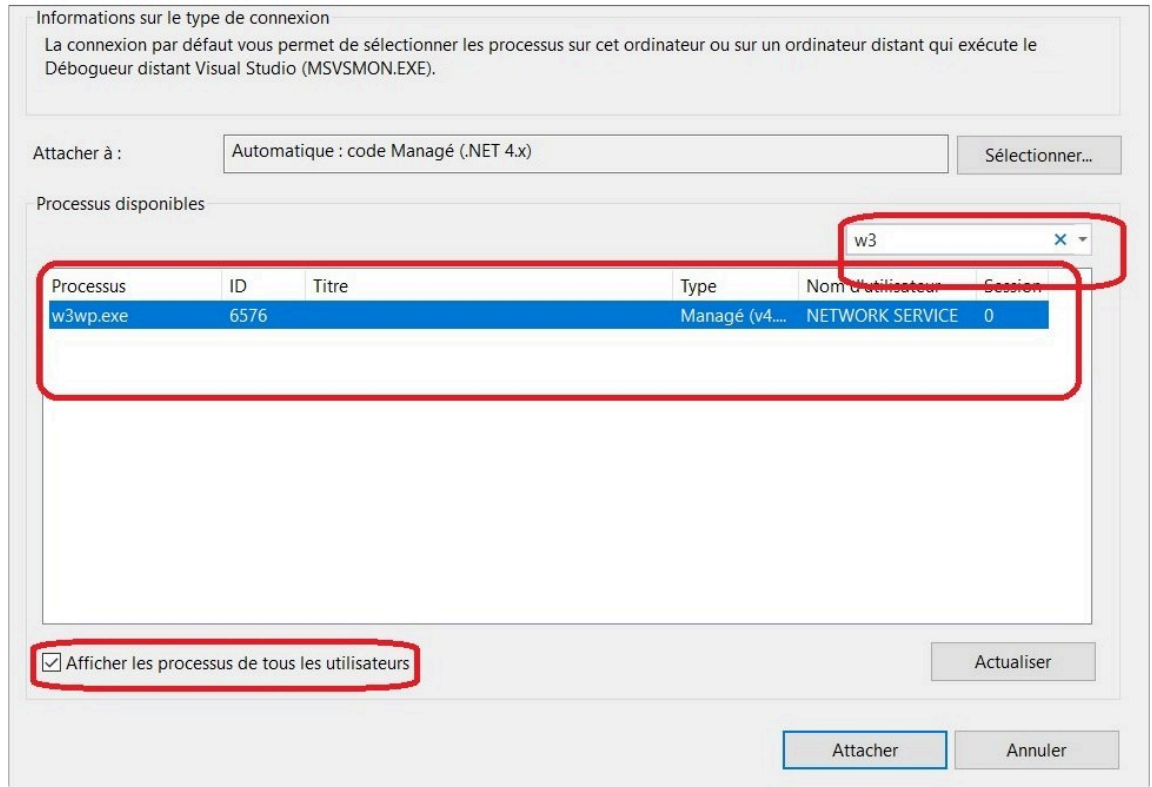
### Debugging Code with Debugger.Break()

- “if (Debugger.IsAttached) — Debugger.Break()” is a line of code that is currently commented out in the code environment. When uncommented, it causes the code to break at that point when the code is run in Visual Studio’s debugger.



✿ This can be useful for troubleshooting and identifying errors in the code.

- In order for the code to break at that point when using the Visual Studio debugger, the programmer needs to have Visual Studio open and running, and they need to have attached the debugger to the process that is running the code.



There are two main types of objects used in form and application building: DomainObjects and FormObjects.

### 1. DomainObjects

A DomainObject is an object that represents an entity within the domain of the application. It is typically used to manage business logic and data, and it represents the encapsulated data and behavior associated with a specific entity in the domain.

✿ In simpler terms, it is a programming construct used to organize and manage information related to a particular entity within an application.

Here are some examples of code for creating, retrieving, updating, and saving DomainObjects:

#### Create DomainObject

- To create a DomainObject, the following method should be called together with the application name and entity name:
  - `var project = bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject();`

### Retrieve DomainObject

- To retrieve a domain object for a specific entity, use the code below, where “Id” is the identifier of the entity.
  - `var pCostSummary = bpmAppService.BPMSServices.DomainObjectService.GetDomainObject(Id);`

### Save DomainObject

- To save a domain object, use:
  - `bpmAppService.BPMSServices.DomainObjectService.SaveDomainObject(domainObjectName);`

### Update DomainObject

- To update a domain object, use:
  - `bpmAppService.BPMSServices.DomainObjectService.UpdateDomainObject(domainObjectName);`

### Commit All Saved and Updated DomainObjects

- To commit all saved and updated domain objects, use:
  - `bpmAppService.UnitOfWork.CommitTransaction();`
  - `bpmAppService.UnitOfWork.CommitTransaction(false);`



It is recommended to add “false” as a parameter so as to avoid clearing the current session.

## 2. FormObjects

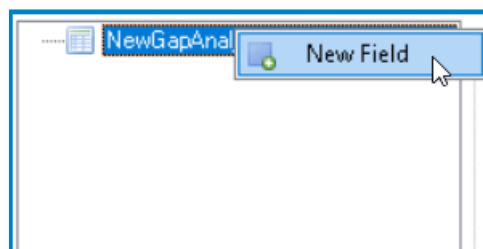
A FormObject is an object that represents a user interface (UI) element, such as a label, textbox, button, or dropdown. It provides a way for users to interact with the data stored in the DomainObjects. Typically, FormObjects are used to capture user input or display information to the user.



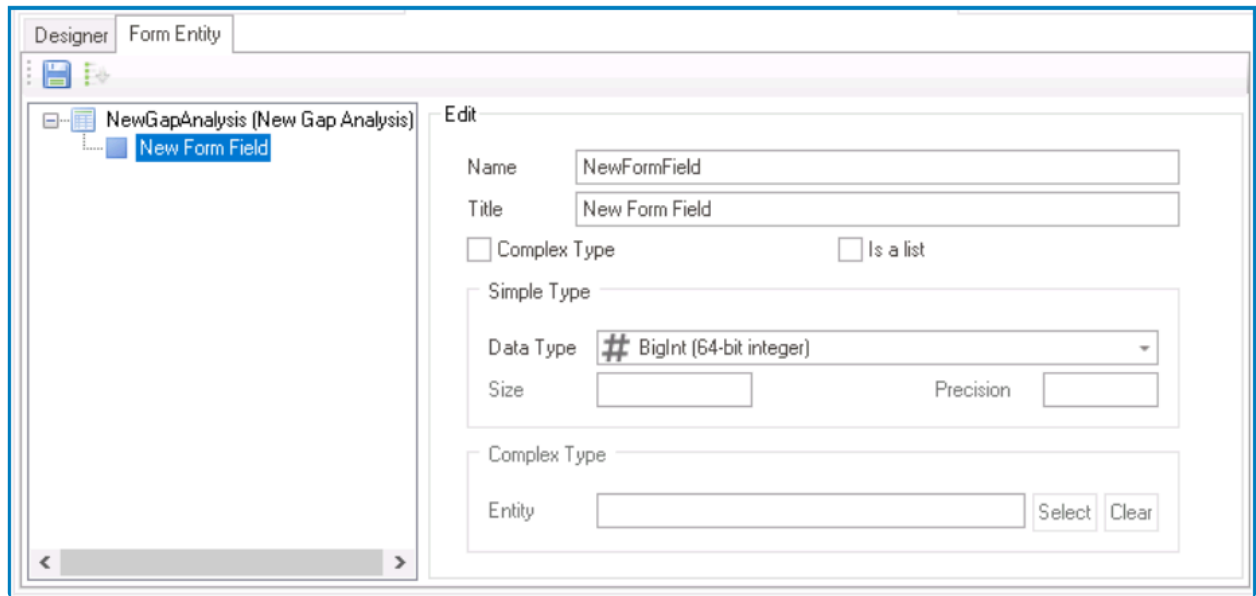
FormObjects provide a way for users to interact with the data stored in the DomainObjects. Their values are often used to update or retrieve data from the corresponding DomainObjects.

To create a form object, take the following steps:

- In the [Form Entity](#) tab of the [Form Designer](#), add a new field by right-clicking on the form entity and select **New Field** from the context menu.



- In the properties of the newly added field, provide the **Name** (which is used to reference the field in code), the **Title** (which is displayed to the user), and the **Data Type** (which defines the type of data that the field will hold, such as text, numbers, dates, etc.).



- If the FormObject will reference data from another form or entity, select the **Complex Type** checkbox.
  - This will indicate that the field is more than a simple data type and needs to be linked to other data.
- If the FormObject will hold a list of data, select the **Is a List** checkbox.
  - This will indicate that the field needs to be able to hold multiple instances of the data type, effectively creating a list of data.

✿ To access a more detailed description of the available functions and methods, please refer to the [Application Programming Interface](#) section.

## 12.1.2.1.1.1.2. Designer

In the Rule Designer, users can define rules that specify the conditions and actions for the control.

The screenshot shows the 'Rule Designer' window titled 'Rule Designer: NewGapAnalysis9ac3b9f77745490d942eafba00f27884'. It has a toolbar with a checkmark, a save icon, and a close icon. The window is divided into two main sections: 'Conditions' and 'Actions'.

**Conditions:** This section contains a list of conditions. The first condition is 'Active' with a checked checkbox. Below it, there are three conditions:
 

- 'New Gap Analysis' (dropdown) followed by 'Is Equal To' (dropdown) and an empty text field.
- 'formControl' (dropdown) followed by 'Is Not Equal T' (dropdown) and an empty text field.
- 'bpmAppService' (dropdown) followed by 'Is Equal To' (dropdown) and an empty text field.


 Each condition is separated by a logical operator: 'AND' for the first two and 'OR' for the third. There are also delete icons for each condition.

**Actions:** This section contains two buttons: 'Call Method' and 'Set Property'. Below these buttons, there are two actions:
 

- 'bpmAppService.BPMSServices.DomainObjectService.UpdateObject(Object obj)' (dropdown) followed by an empty text field.
- 'NewGapAnalysis' (dropdown) followed by an equals sign and an empty text field.

 Each action has a delete icon.

At the bottom, there are two tabs: 'Errors' and 'Output'. The 'Errors' tab is currently selected, showing an empty text area.

 For more information on the general features of the Rule Designer, [click here](#).

To configure the control properties, apply the following steps:

1. In the **Conditions** section of the window, set up the required conditions for the control.
2. In the **Actions** section of the window, set up the required actions (Call Method and Set Property).
3. Click on the **Validate** button to assess the validity of the information.
  - If there are any errors, they will appear in the **Error** tab.
  - If the information is valid, the results will appear in the **Output** tab.

# 12.1.2.1.1.2. Name

In the **Name** category, users can access basic properties required for the control's identification and proper referencing in the code.

▼ 2. Name

Caption Arabic

Caption English

Single select dropdown

Caption French

Caption German

Caption Hebrew

Control Type

Single select dropdown

ServerId

SingleSelectDropdown

1. **Caption:** This field allows users to input the name or caption of the control.
- Users can also directly add a caption to the control by simply clicking on its caption in the design environment.



- As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the control, regardless of other language options that may be available.

◦ Similarly, any caption entered in the design environment will automatically populate the English caption field.

✿

Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

2. **Control Type:** This field displays the control type. It cannot be modified.
3. **Server ID:** This is a text field that allows users to input the Server ID for the control.
- Server ID refers to the name used to call the control in the code.

## 12.1.2.1.1.3. Control Behavior

In the **Control Behavior** category, users can configure the behavior of the control and define how it interacts with the form or application.



| 3. Control Behavior   |                     |
|-----------------------|---------------------|
| 1. Form Behavior      | 0. Collect Evidence |
| 2. Control State      | Enabled             |
| 3. Control Governance | Click '...' to edit |
| 4. Initial Value      |                     |
| 5. State Matrix       | Click '...' to edit |

✿ To access the Control Behavior category, users must first create the form behavior.

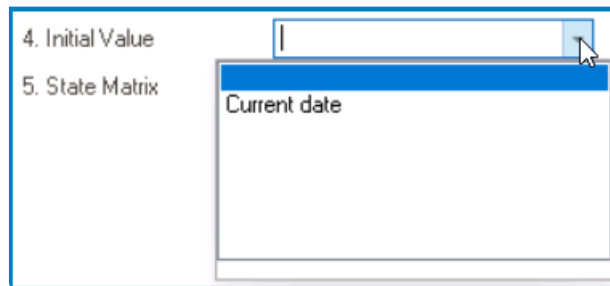
1. **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
  - Any state or validation properties set for the control will apply only to the selected form behavior.

✿ Only form behaviors linked to the associated form will be displayed in this dropdown.

2. **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
3. **Control Governance:** This field allows users to configure various properties that govern the functionality of controls, including validation rules and other control-specific settings.
  - To access the control governance properties, click on the ellipses (...) inside the field. It will display the [Form / Behavior / Control](#) window.

✿ Although the available tabs and options in this window may vary depending on the control, typically users can configure validation properties for controls such as required field and range validation. [To learn more about configuring control validation, click here.](#)

4. **Initial Value:** This field allows users to set an initial value for the control, which can either be input manually or selected from a dropdown menu.
- For example, when configuring the initial value for the date picker control, users can either set a specific date or select the Current Date option from the dropdown menu.



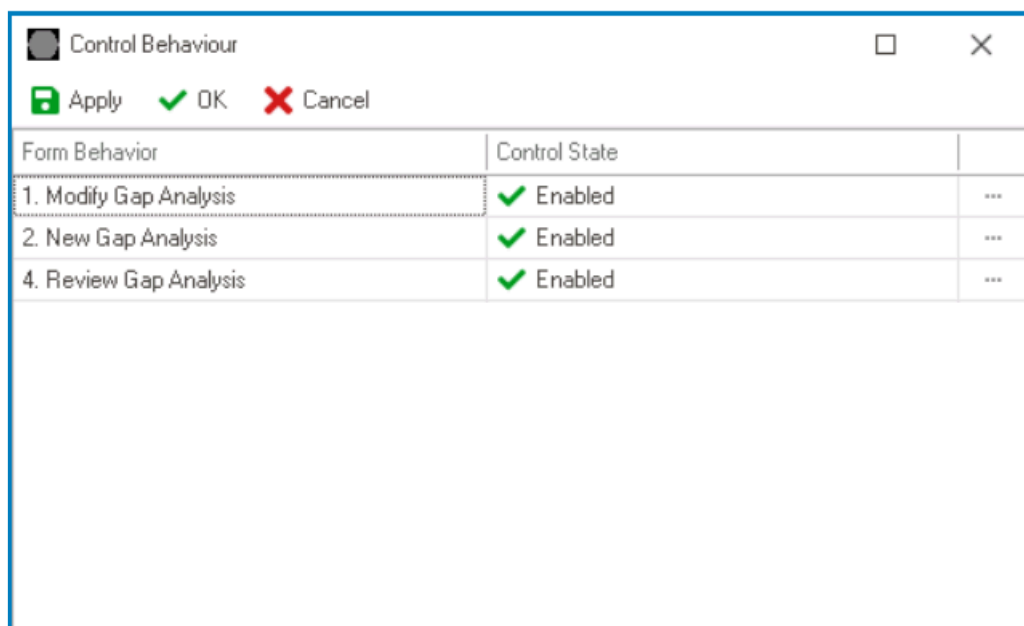
• **Control Initial Value Preview: Web View**

A control's initial value is the default value that is assigned to it when the form is first opened or loaded.

**Analysis ID**

AN-00001

5. **State Matrix:** This field allows users to configure the control state properties.
- To access the control state properties, click on the ellipses (...) inside the field.



- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.



By modifying the control state per form behavior, users can determine which controls are made enabled, disabled, or invisible at different stages of the workflow.



Depending on the individual properties of the control, the available columns and options in this window can vary. For example, some controls may allow users to set an initial value or configure control governance properties, while others may not. These properties, where applicable, can be managed in this window on a per-form-behavior basis, allowing users to tailor the control's behavior and functionality for different stages of the workflow. To gain an understanding of how these properties can be configured in this window, [click here](#).



## 12.1.2.1.1.3.1. Control Governance – Validation

Users can configure validation properties for controls, such as required field and range validation.




The screenshot shows a configuration window titled "Form [QuickForm] / Behaviour [Quick Form] / Control [DefectsPerUnit]". At the top, there are buttons for "Apply" (with a green square icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon). Below these buttons, the "Control Name" is set to "DefectsPerUnit". A "Validation" tab is selected. Inside the tab, there are two checkboxes: "Has Required Field Validator" and "Has Range Validator", both of which are currently unchecked. Below these checkboxes, there are two sub-tabs: "Required Field Validator" (which is active) and "Range Validator". Under the "Required Field Validator" sub-tab, there is an "Error message" section containing a table with columns for "Language", "Is Default", and "Text".

| Language | Is Default                          | Text        |
|----------|-------------------------------------|-------------|
| English  | <input checked="" type="checkbox"/> | Enter value |
| French   | <input type="checkbox"/>            |             |
| German   | <input type="checkbox"/>            |             |
| Arabic   | <input type="checkbox"/>            |             |
| Hebrew   | <input type="checkbox"/>            |             |

**!** To configure validation for a control, users must first create a form behavior.

1. **Required Field Validator:** This is a checkbox which, if selected, allows users to enforce a control as mandatory and trigger a message to appear if the mandatory field is left unfilled.
  - If this checkbox is selected, it will enable the Required Field Validator tab.

Form [QuickForm] / Behaviour [Quick Form] / Control [DefectsPerUnit]

 Apply  OK  Cancel

Control Name **DefectsPerUnit**

**Validation**

☒ Has Required Field Validator  
☐ Has Range Validator

Required Field Validator Range Validator

Error message

| Language       | Is Default                          | Text        |
|----------------|-------------------------------------|-------------|
| <b>English</b> | <input checked="" type="checkbox"/> | Enter value |
| <b>French</b>  | <input type="checkbox"/>            |             |
| <b>German</b>  | <input type="checkbox"/>            |             |
| <b>Arabic</b>  | <input type="checkbox"/>            |             |
| <b>Hebrew</b>  | <input type="checkbox"/>            |             |

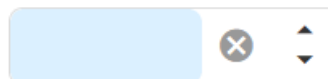
- In this tab, users can customize the error message to be displayed when the mandatory field is not filled out. Users can specify this message in different languages as well as designate a default language.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Required Field Validation Preview: Web View**

- By default, in the **DBP** web application, mandatory fields are visually distinguished with a light-blue background color.

#### Defects Per Unit



✿ Users can customize the color of mandatory fields in the [Settings](#) of the web application.

- In the event that a user attempts to submit the form without filling out a

mandatory field, the specified error message will appear towards the bottom of the page.

Sample Form

Product ID

Product Name

Category

Source

Severity

Date of Corrective Action

Defects Per Unit

Project Manager

Subject Matter Expert

Reviewer

Issue Statement




Please enter a value

Submit form

✿ When the **Required Field Validator** checkbox is selected, the form will prevent submission until the required field is completed, ensuring that important information is not missing.

2. **Range Validator:** This is a checkbox which, if selected, allows users to restrict a control to a specific range of values and trigger a message to appear if the range is not met.
- If this checkbox is selected, it will enable the Range Validator tab.

Form [QuickForm] / Behaviour [Quick Form] / Control [DefectsPerUnit]

 Apply  OK  Cancel

Control Name **DefectsPerUnit**

**Validation**

☐ Has Required Field Validator

☒ Has Range Validator

**Required Field Validator** **Range Validator**


**Error message**

| Language       | Is Default                          | Text |
|----------------|-------------------------------------|------|
| <b>English</b> | <input checked="" type="checkbox"/> |      |
| <b>French</b>  | <input type="checkbox"/>            |      |
| <b>German</b>  | <input type="checkbox"/>            |      |
| <b>Arabic</b>  | <input type="checkbox"/>            |      |
| <b>Hebrew</b>  | <input type="checkbox"/>            |      |

Minimum value

Maximum value

- In this tab, users can customize the behavior of range validation. They can specify the minimum and maximum values for the range, and define a custom error message to appear when the input value falls outside of the specified range. Users can specify this message in different languages as well as designate a default language.

 Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Note:** For date picker controls, users will need to specify the maximum and minimum date values using a date picker. (In other words, users must specify the earliest and latest dates that can be selected in the control.)

#### • **Range Validation Preview: Web View**

In the event that a user attempts to submit the form without entering a value within the required range, the specified error message will appear towards the bottom of the page.

Sample Form

Product ID

Product Name

Category

Select... ▾



Source

Select... ▾




Severity

Select... ▾

Date of Corrective Action

01/04/2025  

Defects Per Unit

11   

Project Manager

Select... ▾

Subject Matter Expert

Select... ▾

Reviewer

Select... ▾

Issue Statement

Paragraph ▾

A ▾

A1 ▾

A ▾

A ▾

B

I

U

↺

≡ ▾

≡ ▾

≡ ▾

≡ ▾

🔗

“ ”

🖼️

📅 ▾

🎥 ▾

⋮

The date selected needs to be between 2023-05-03 and 2024-05-03

The value entered must be between 0 and 10.

Submit form

**Note:** For date picker controls, dates outside the specified range will be grayed out and inaccessible on the calendar.



When the **Range Validator** checkbox is selected, the form will prevent submission until the required range is met, ensuring that important information is not entered incorrectly.

3. **Apply:** This button allows users to save their changes.
4. **OK:** This button allows users to save their changes and close the window.
5. **Cancel:** This button allows users to cancel their changes and close the window.

## Fields with Configurable Validation

- To ensure the accuracy and completeness of the data entered into the form, certain controls have options for required field and range validation.
  - Required field validation is available for the following controls:
    - Comment
    - Date/Time Picker
    - File Attachment
    - Free Draw
    - Multi-Select Dropdown
    - Multi-Line Textbox
    - Numeric Textbox

- Password Box
- Radio-Button List
- Rich Textbox
- Single-Select Dropdown
- Textbox
- Time Duration
- Range validation is available for the following controls:
  - Date/Time Picker
  - Numeric Textbox

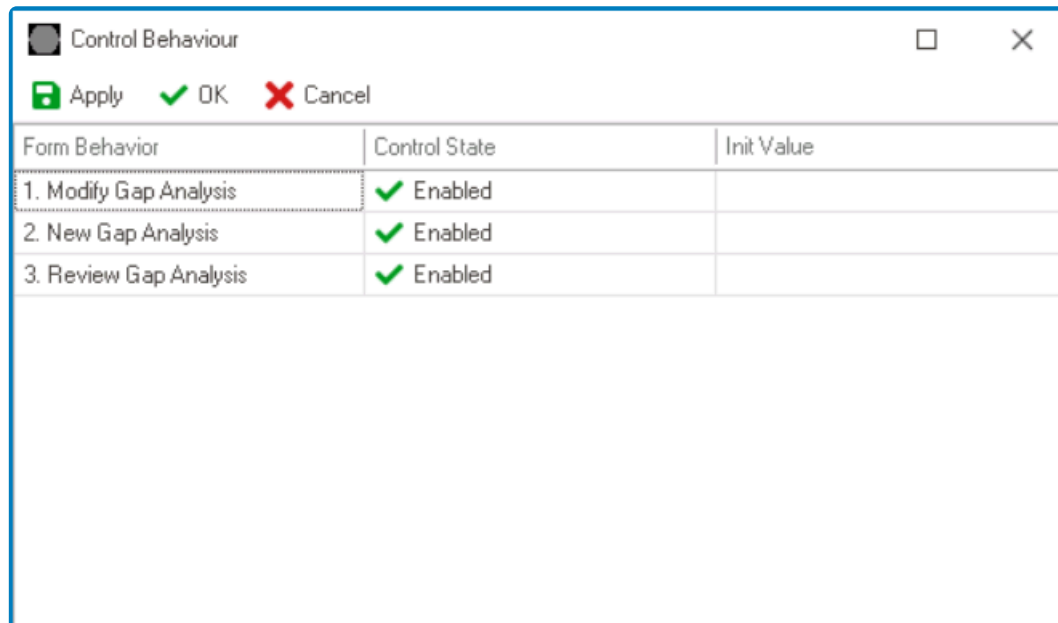


Users can apply field validation to a specific task in the workflow by changing the control form behavior.

## 12.1.2.1.1.3.2. State Matrix – Control-Specific Options

To illustrate how users can configure control-specific properties in the window linked to the **State Matrix** field, consider the two following examples:

### Example 1: State Matrix for Checkbox Control: Expanded Configuration Options

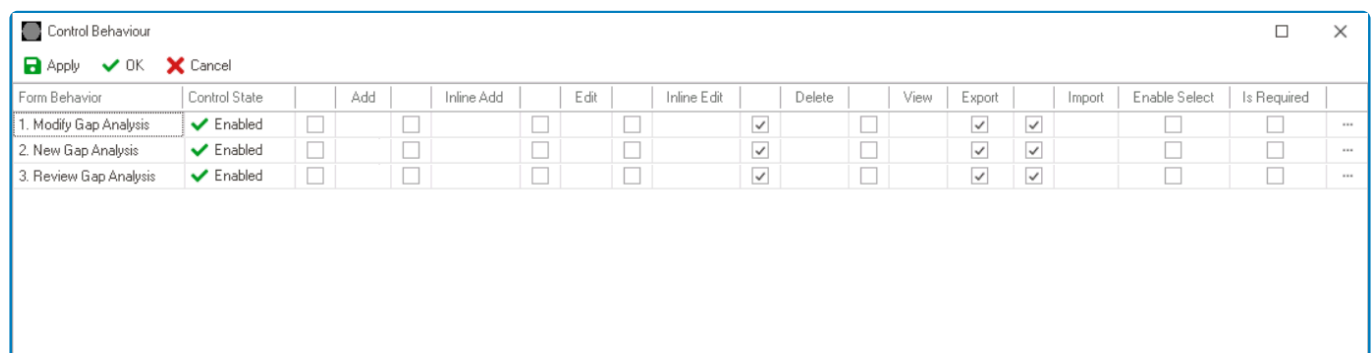


| Form Behavior          | Control State | Init Value |
|------------------------|---------------|------------|
| 1. Modify Gap Analysis | ✓ Enabled     |            |
| 2. New Gap Analysis    | ✓ Enabled     |            |
| 3. Review Gap Analysis | ✓ Enabled     |            |

Along with configuring the control state for each form behavior, users can also perform the following action:

- Set initial value for the control on a per-form-behavior basis

### Example 2: State Matrix for Grid Control – Expanded Configuration Options



| Form Behavior          | Control State | Add                      | Inline Add               | Edit                     | Inline Edit              | Delete                              | View                     | Export                              | Import                              | Enable Select            | Is Required              |
|------------------------|---------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Along with configuring the control state for each form behavior, users can also perform the following actions:

- Enable control properties or functions on a per-form-behavior basis

- To enable a specific function, select the corresponding checkbox (located either directly under the function or to its left).

| Form Behavior          | Control State |                          | Add |                          | Inline Add |
|------------------------|---------------|--------------------------|-----|--------------------------|------------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |     | <input type="checkbox"/> |            |

- Set the corresponding form behavior for each enabled function (where applicable)
  - Once a function has been enabled for the control, users can select the form behavior associated with it.
- Set validators for the control on a per-form-behavior basis
  - Depending on the control, users can enable required field and/or range validation.

- **Range Validation**

In cases where users can enable range validation for the control, they must also set a minimum and maximum value.

| Form Behavior          | Control State | Init Value | Is Required              | Range Valida...                     | Min Value | Max Value |     |
|------------------------|---------------|------------|--------------------------|-------------------------------------|-----------|-----------|-----|
| 1. Modify Gap Analysis | ✓ Enabled     |            | <input type="checkbox"/> | <input type="checkbox"/>            |           |           | --- |
| 2. New Gap Analysis    | ✓ Enabled     |            | <input type="checkbox"/> | <input checked="" type="checkbox"/> |           |           | --- |
| 3. Review Gap Analysis | ✓ Enabled     |            | <input type="checkbox"/> | <input type="checkbox"/>            |           |           | --- |

- For example, when setting up range validation for the date/time picker control, users must enable the **Range Validator** checkbox as well as select the relevant date values.



By setting these properties for the control across multiple form behaviors, users can tailor its behavior to each stage of the workflow and ensure that it functions as intended.



## 12.1.2.1.1.1.4. Appearance

In the **Appearance** category, users can customize the visual appearance of the control.

1. **Placeholder:** This field allows users to input the placeholder text for the control.

- Users can also directly add a placeholder text to the control simply by clicking on the inside of the control in the design environment.

- As English is set as the default language in the system, any placeholder inputted in the English placeholder field will be used as the display placeholder for the control, regardless of other language options that may be available.
  - Similarly, any placeholder text entered in the design environment will automatically populate the English placeholder field.

\* Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Control Placeholder Text Preview: Web View**

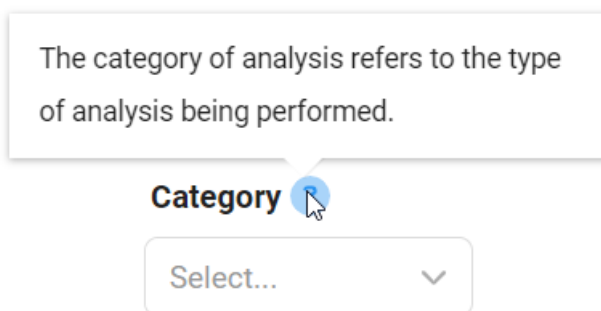
**Category**

Select...

\* A placeholder text is the default value or text that is displayed inside the control until the user enters their own value or selects an option.

2. **Tooltip:** This field allows users to input a tooltip for the control.

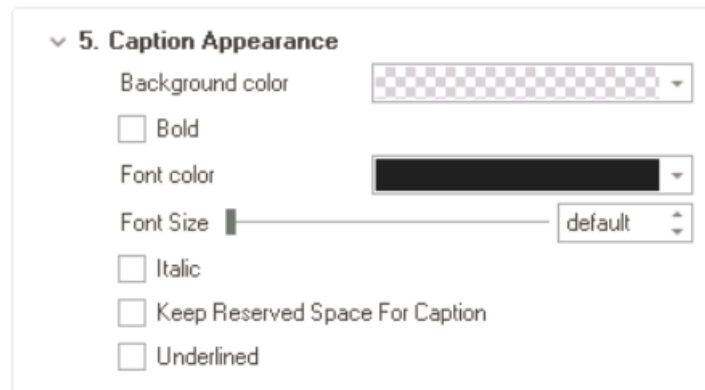
- In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
- **Control Tooltip Preview: Web View**



\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

## 12.1.2.1.1.1.5. Caption Appearance

In the **Caption Appearance** category, users can customize the visual appearance of the control's caption.

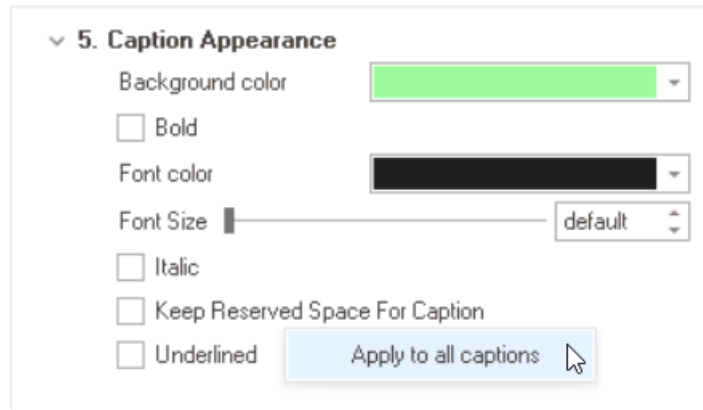


1. **Background Color:** This is a single-select dropdown field that allows users to choose the background color of the caption text.
  - It contains two tabs: Custom and Web.
    - In the Custom tab, users can choose a background color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
    - In the Web tab, users can select a background color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
  - After choosing a background color, the control will update to reflect the new color scheme.
2. **Bold:** This is a checkbox which, if selected, allows users to apply a bold effect to the caption text.
3. **Font Color:** This is a single-select dropdown field that allows users to choose the font color of the caption text.
  - It also has two tabs, similar to the **Background Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.
4. **Font Size:** This field allows users to set the font size of the caption text.
  - By default, the font size is set to 12. To adjust this size, use either the slider or the numeric updown field.
    - In the numeric updown field, users can enter numbers either manually or by clicking the increment/decrement arrows.
5. **Italic:** This is a checkbox which, if selected, allows users to apply an italic or slanted effect to the caption text.
6. **Keep Reserved Space for Caption:** This is a checkbox which, if selected, allows users to reserve space for the control's caption even when there is no caption set.
7. **Underlined:** This is a checkbox which, if selected, allows users to apply an underline effect to the caption text.

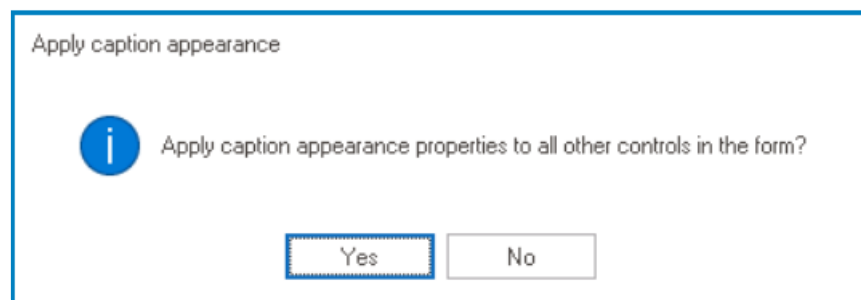
**Note:** When designing a form, users can choose to either apply the same caption appearance properties to all controls, or customize each control's caption individually.

- To apply the same caption appearance properties to all controls in the form, right-click on the

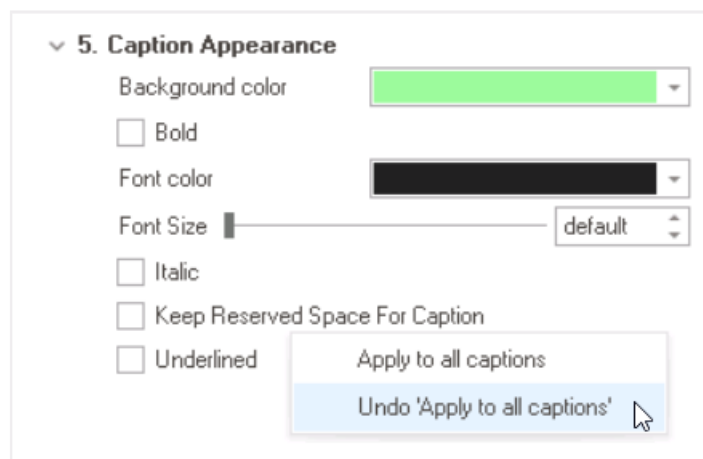
category section and select the **Apply to All Captions** option.



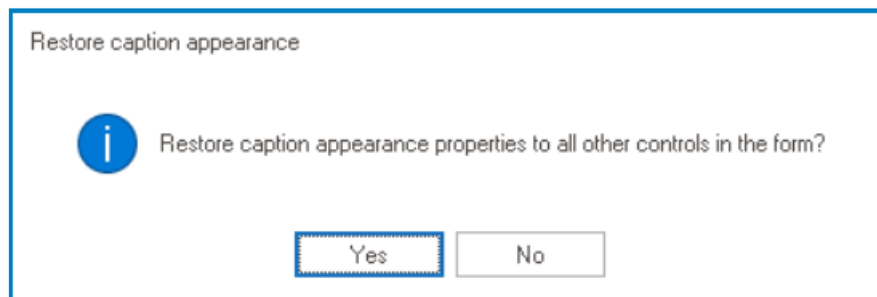
- If this option is selected, users will be alerted of the action with the following confirmation message:



- **Yes or No:** To confirm the application, click on the **Yes** button. To cancel, click on the **No** button.
- To revert back to individual caption properties, simply right-click on the category section and choose the **Undo Apply to All Captions** option.



- If this option is selected, users will be alerted of the action with the following confirmation message:



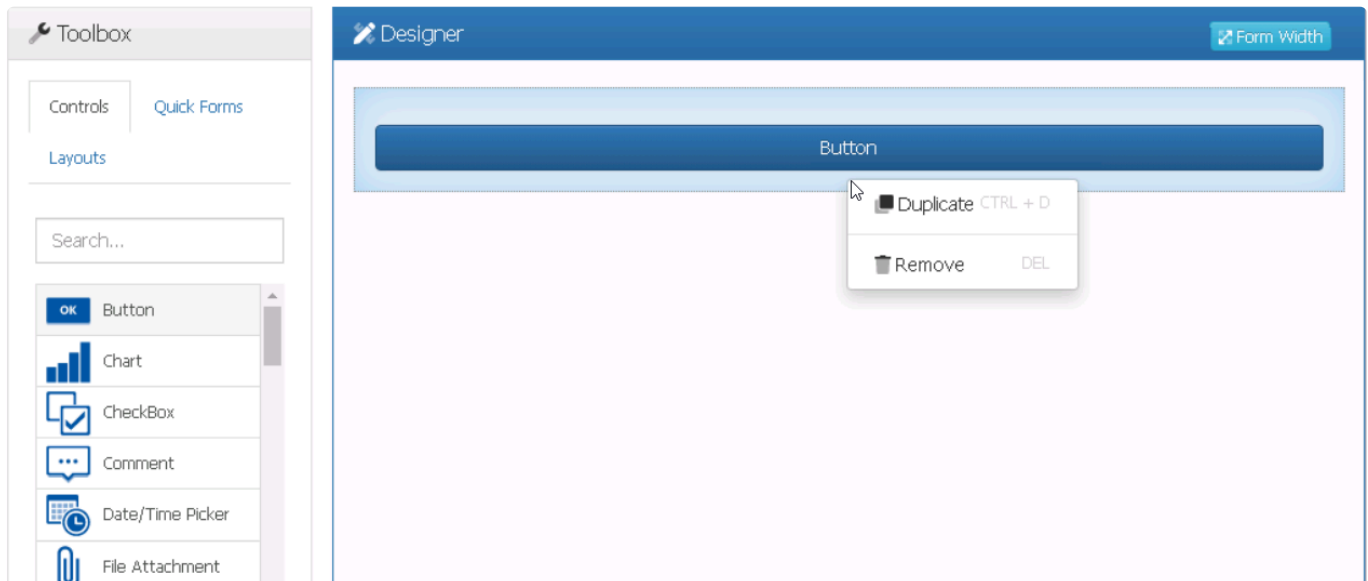
- **Yes or No:** To confirm the restoration, click on the **Yes** button. To cancel, click on the **No** button.

#### Caption Text Style Preview: Web View

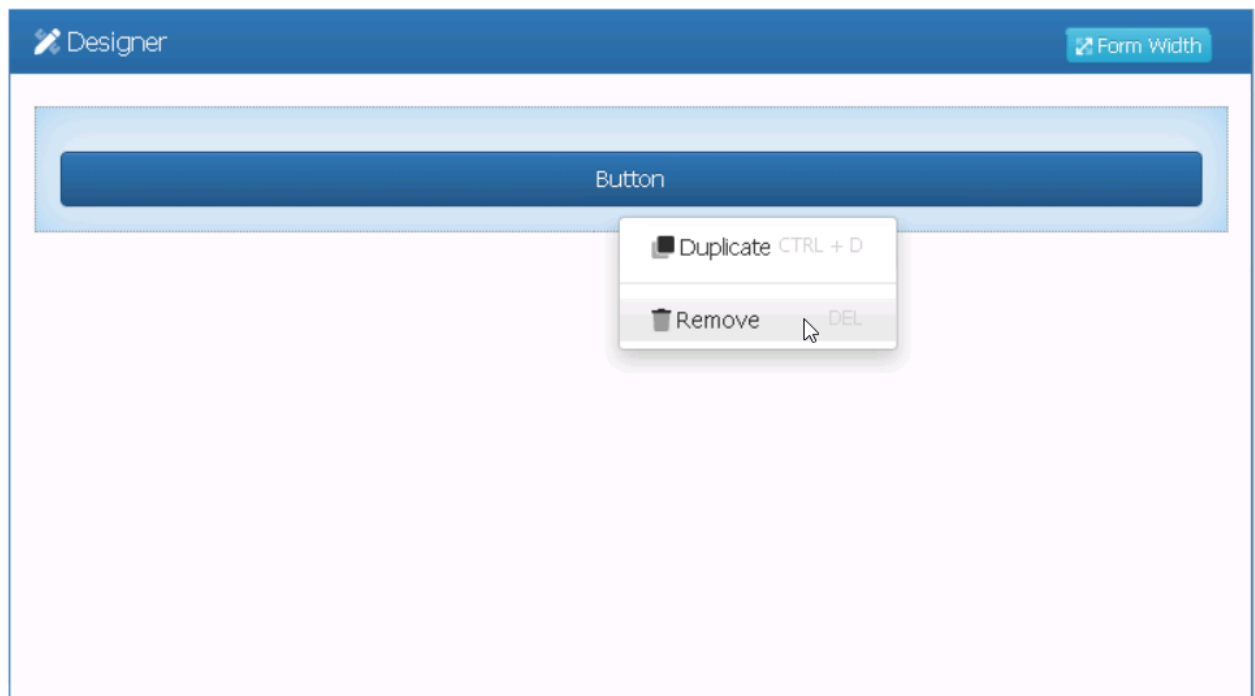
Analysis Name

## 12.1.2.1.1.2. Control Context Menu

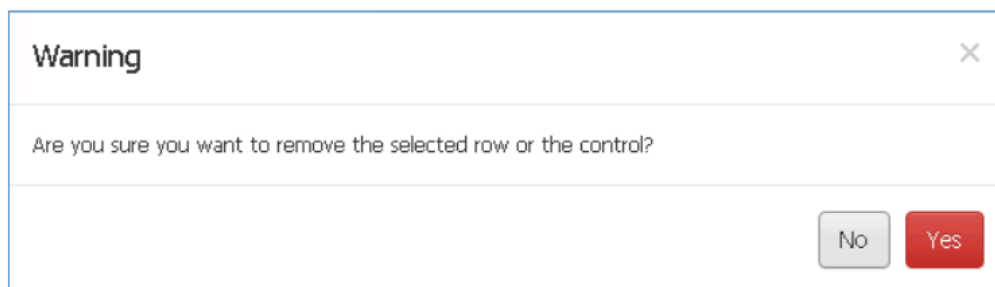
To duplicate or remove a control, right-click on the control and select the corresponding option from the context menu.



1. **Duplicate Control:** This option allows users to duplicate the selected control, creating a copy below it.
2. **Remove Control:** This option allows users to remove the selected control.



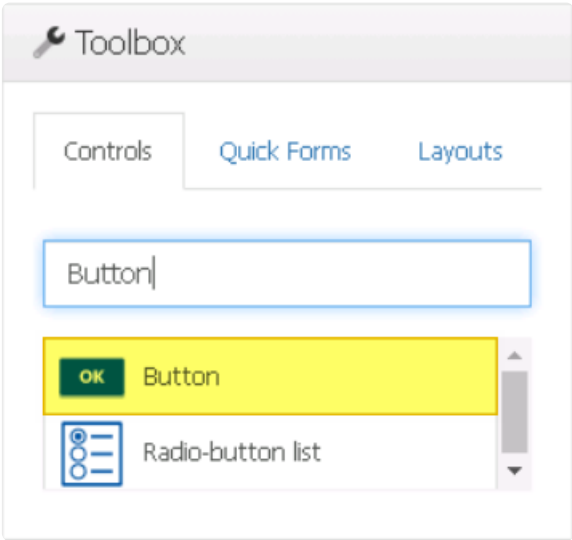
- If this option is selected, users will be alerted of the action with the following confirmation message:



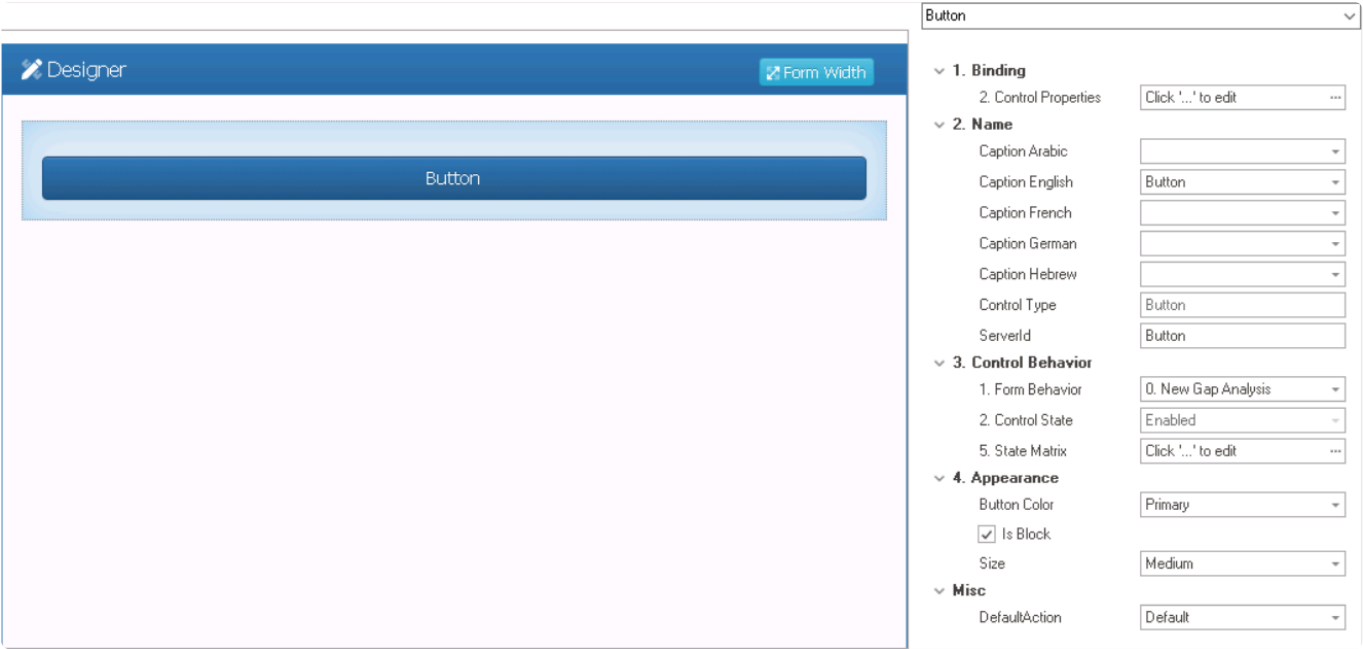
- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.

# 12.1.2.1.1.3. Button

Buttons are an essential element in form development, allowing users to interact with forms in a variety of ways. With buttons, users can store and send form information, trigger actions and events, and execute custom code to perform specific tasks.



When a button control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

- 1. **Binding:** In this category, users can define the binding properties of the control.
  - **Control Properties:** This field allows users to configure the control properties.
    - For more information on these properties and how to configure them, [click here](#).



! Users do not need to create or link an attribute for button controls.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.

- For more information on these properties and how to configure them, [click here](#).

! When inputting a name for a button control, users can only use the **Caption** field. They cannot set a name by double-clicking on the control.

3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.

- For more information on these properties and how to configure them, [click here](#).

! For button controls, users can select the form behavior, view the current control state, and modify the control state using the **State Matrix** field. Please note that the linked section contains information on other fields that do not apply to button controls and can be disregarded.

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Button Color:** This is a single-select dropdown field that allows users to choose the color of the button.
  - Button Colors: Default, Primary, Success, Info, Warning, Danger, or Link.
    - **Button Color Preview: Design View vs. Web View**
- **Is Block:** This is a checkbox which, if selected, allows the button to expand to fill the available space within the layout box.
  - If this checkbox is left unselected, the button will be sized according to its content, and may not take up the full width of the box.
    - **Button Width Preview: Is Block Checkbox (Unselected/Selected)**
- **Size:** This is a single-select dropdown field that allows users to choose the size of the button.
  - Button Sizes: Large, Medium, Small, or Tiny.
    - **Button Size Preview: Web View**



5. **Misc:** In this category, users can define additional properties for the control.

- **Default Action:** This is a single-select dropdown field that allows users to choose the default action of the button.
  - Button Actions: Default, Submit, or Save.
    - Default: This action allows the button to perform a specific action that can be

configured by code.

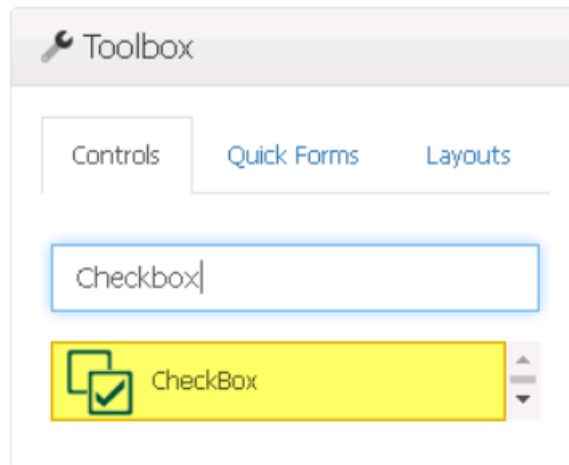
- Users, for example, can configure the button to perform tasks such as selecting all items in a given grid, importing items from one grid to another, or copying a previous form's content.
- Submit: This action allows the button to submit the form.
- Save: This action allows the button to save the form.



A default action defines the default behavior of the button when it is clicked in the form.

## 12.1.2.1.1.4. Checkbox

Checkboxes are a common feature in user interfaces, providing users with a simple way to make selections.



- \* Checkboxes are particularly effective in implementing business rules, where specific actions can be triggered based on the user's selection. For example, by selecting this checkbox, XYZ can be triggered, whereas by selecting a different checkbox, ABC can be initiated. For more information on business logic and rules, [click here](#).

When a checkbox control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

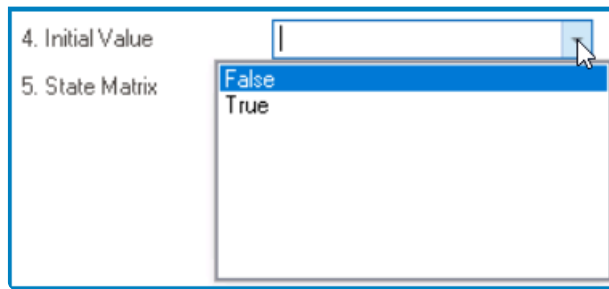
- \* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

- \* To create attributes for checkbox controls, users must use the **Bit (Boolean)** data type.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Checkbox Initial Value**  
When configuring the initial value for the checkbox control, users can select either the

False (unchecked) or True (checked) option from the dropdown menu.

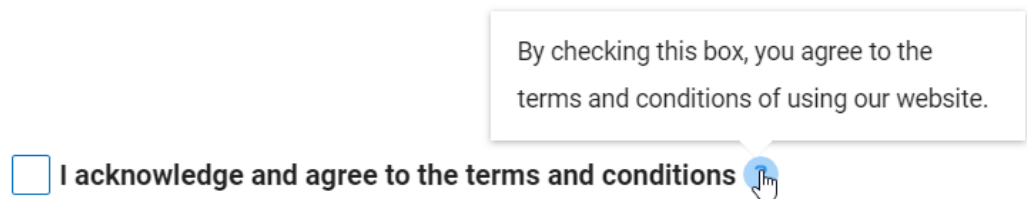


- For more information on these properties and how to configure them, [click here](#).

! Please note that the linked section contains information on control governance properties that are not applicable to checkbox controls and can be disregarded.

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
  - **Checkbox Tooltip Preview: Web View**



\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

6. **Misc:** In this category, users can define additional properties for the control.

- **Align:** This is a single-select dropdown field that allows users to choose the alignment of the control.
  - Alignment Options: Auto By Culture, Right, or Left.
    - Auto By Culture: This option aligns the control based on the language or culture setting in the web application.
    - Right: This option aligns the control to the right side of the screen.
    - Left: This option aligns the control to the left side of the screen.

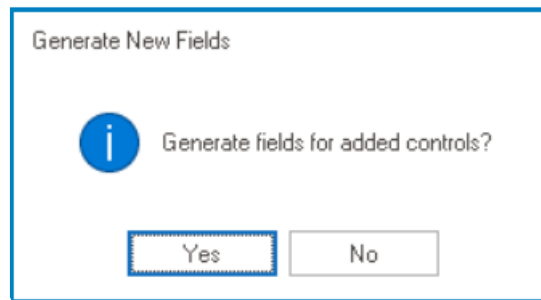
## Checkbox Control Preview: Web View



### Quick Forms Application

To apply the **Quick Forms** feature to checkbox controls, take the following steps:

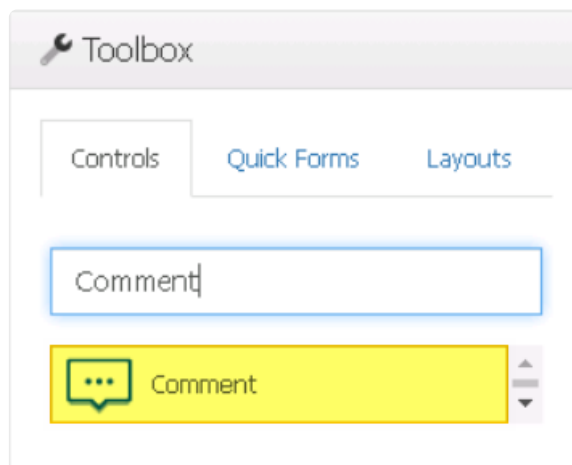
- Drag a checkbox control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:



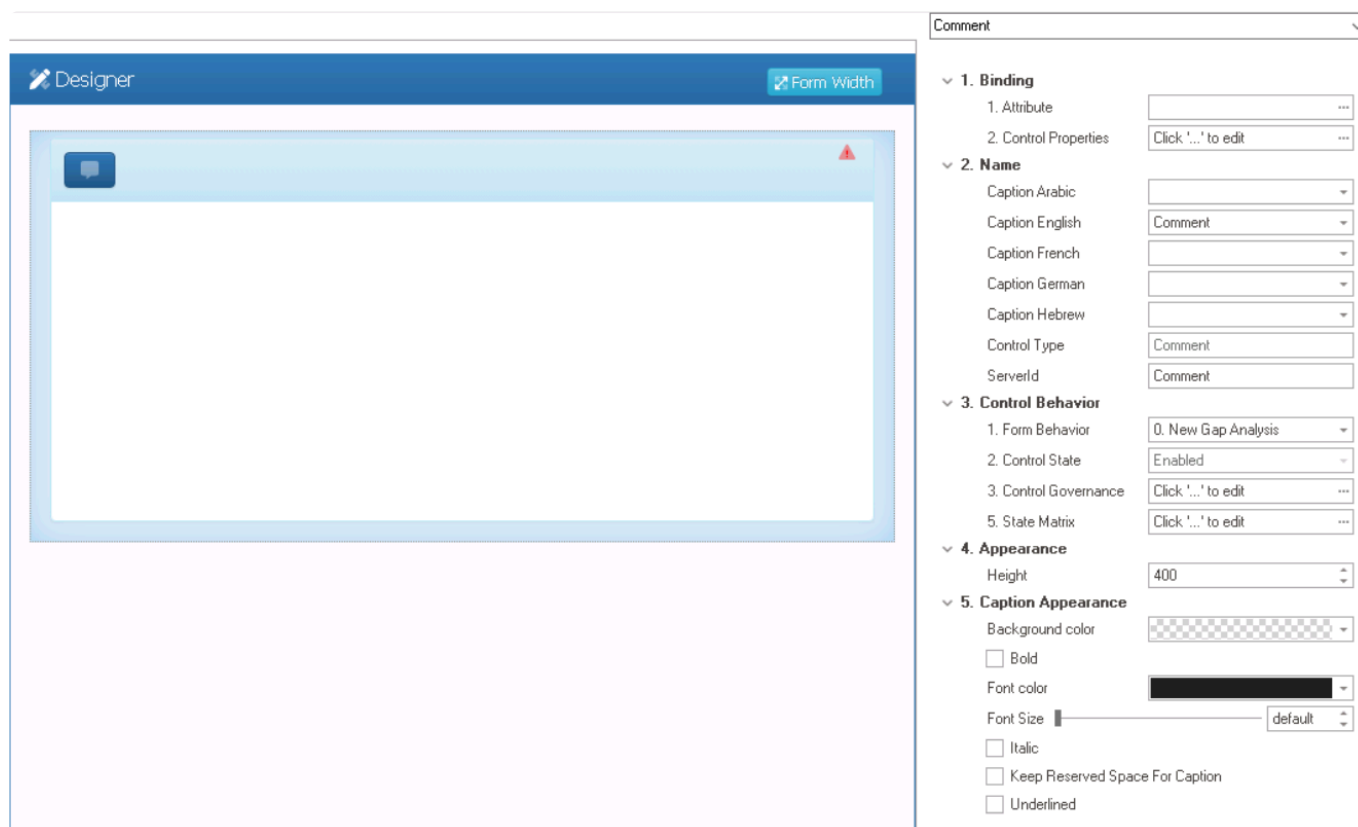
\* For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.5. Comment

Comment controls enable users to add comments to a particular form.



When a comment control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.

- For more information on the fields in this category, [click here](#).



For more information on the required attributes for comment controls, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
      - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
        - This window consists of two tabs:
          - [Comment](#)
          - [Validation](#)



Please note that users are limited to configuring only the required field validation properties for comment controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).



Please note that the linked section contains information on setting an initial value, which is not applicable to comment controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 400. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - For more information on the fields in this category, [click here](#).

Comment Control Preview: Web View

Comment



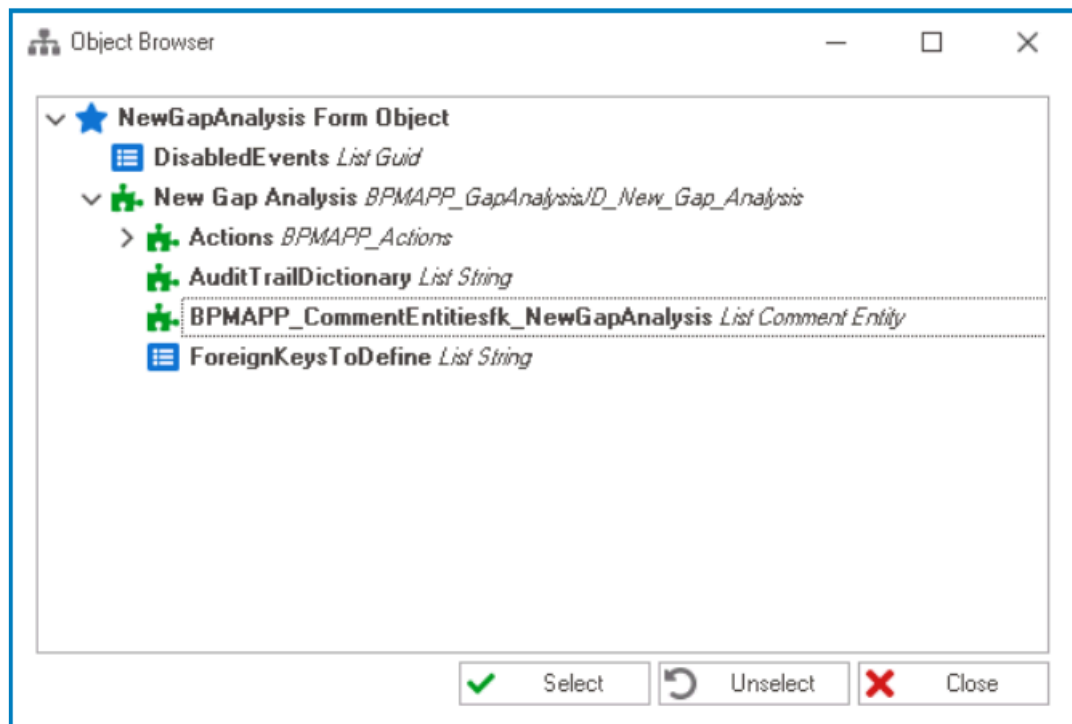
For more information on how to use this field in the web application, [click here](#).



## 12.1.2.1.1.5.1. Comment – Binding

In the **Binding** category, users can link a comment to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the comment control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the comment control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

! Users must bind the comment control to the object that references the foreign key associated with the comment entity. [For more information on this matter, click here.](#)

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. In the displayed window, navigate to the Comments tab.

The screenshot shows the 'Control Properties' dialog box with the 'Comments' tab selected. The dialog has a title bar with a close button (X) and three buttons: 'Apply' (green square), 'OK' (green checkmark), and 'Cancel' (red X). Below the buttons, there is a section for 'Form Control' and 'Entity and Attribute Object type'. The 'Form Control' is 'Comment' and the 'Entity and Attribute Object type' is 'DomainObject.BPMAPP\_CommentEntitiesfk\_NewGapAnalysis' and 'Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_CommentEntity'. Below this, there are two tabs: 'General' and 'Comments'. The 'Comments' tab is active, showing a list of properties with dropdown menus:

| Property    | Value |
|-------------|-------|
| Foregin Key |       |
| Text        |       |
| Parent      |       |
| Vote        |       |
| Date        |       |
| Sender user |       |
| Sender role |       |
| Form        |       |

- From the set of dropdowns, select the following options:

The screenshot shows the 'Control Properties' dialog box with the 'Comments' tab selected, showing the selected values for the properties:

| Property    | Value  |
|-------------|--|
| Foregin Key | BPMAPP_GapAnalysisJD_New_Gap_Analysisfk_NewGapAnalysis |
| Text        | Text   |
| Parent      | BPMAPP_CommentEntityfk_CommentEntity                   |
| Vote        |  |
| Date        | Date   |
| Sender user | UserUser   |
| Sender role | StaffRole  |
| Form        | Comment_NewGapAnalysis                                 |

- **Foreign Key:** Foreign Key Towards Main Entity
- **Text:** Text Attribute
- **Parent:** Foreign Key Towards Comment Entity
- **Date:** Date Attribute
- **Sender User:** UserUser
- **Sender Role:** StaffRole
- **Form:** Comment Form



In the General tab of this window, users can configure business rules for the comment control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.1.5.1.1. Comment – Attributes

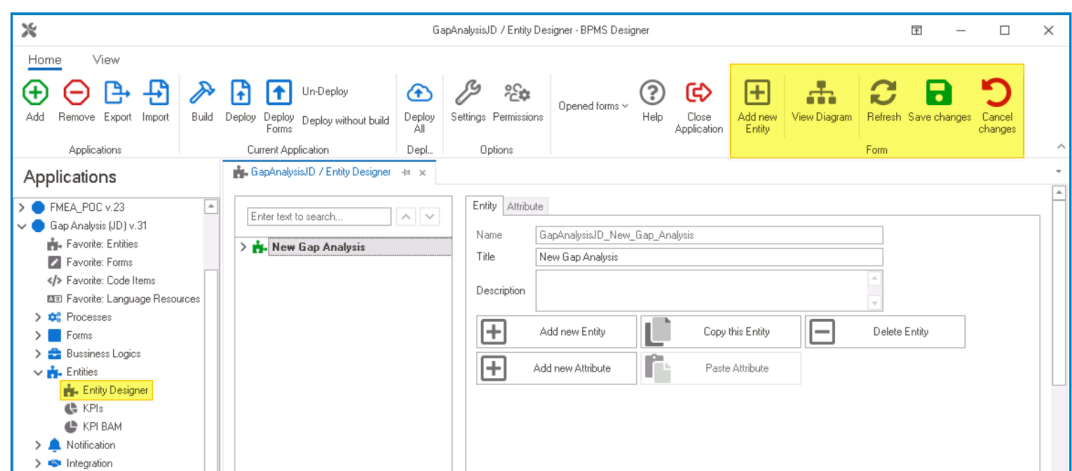
To create comment controls and their associated attributes, follow these steps:

- ✿ Before proceeding with the following steps, please ensure you have created the main entity with its attributes, and linked the main entity to the main form.

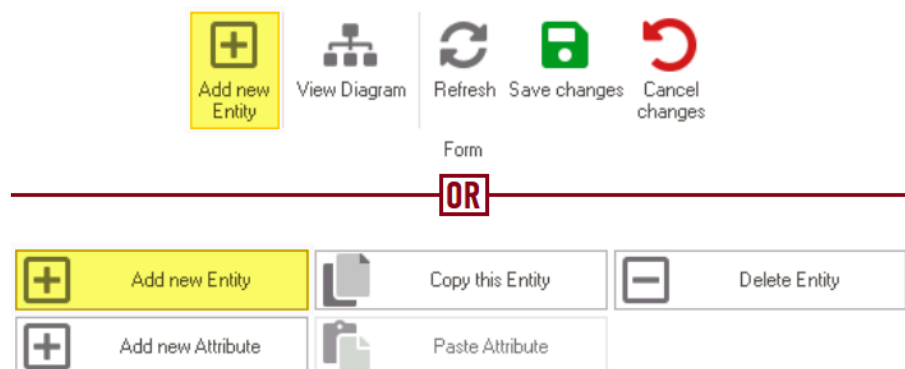
### 1. [Create Comment Entity](#)

- **Access Entity Properties**

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

 Regardless of whether you have created an entity for the form, comment controls require their own entities.

## 2. Create Attributes and Foreign Keys

### • Access Attribute Properties

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' application window. On the left is a sidebar with a search bar and a tree view containing 'Comment Entity', 'NewAttribute Id / UniqueIdentifier', and 'New Gap Analysis'. The main area has two tabs: 'Entity' and 'Attribute'. The 'Attribute' tab is active, showing a form for a new attribute named 'NewAttribute'. The form includes fields for 'Name', 'Title', and 'Description'. Below these are radio buttons for 'Data field' (selected) and 'Reference'. Under the 'Data field' section, there are tabs for 'Data field' and 'Reference', with the 'Data field' tab active. This tab contains a 'Data Type' dropdown set to 'UniquelIdentifier (Globally unique identifier)', and input fields for 'Size' and 'Precision'. Below this is a section 'Show as a column in the task list' with radio buttons for 'Hide', 'Hidden by default', and 'Show by default' (selected). At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', and three buttons: 'Add new Attribute', 'Copy Attribute', and 'Delete Attribute'.

\* For comment controls, users must create two data attributes (text and date) and two reference attributes (user and role). In addition, two foreign keys are required—one to link to the main entity associated with the form, and the other to link to the entity associated with the comment control.

- **Create Data & Reference Attributes**

1. **Text:** This attribute is used to capture and store the textual content of a comment.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Date Field** option.
  - From the **Data Type** dropdown, select **NVarChar\_Max** (to allow storage of any characters, including letters, numbers, and special characters).

Entity Attribute

Name

Title

Description

☒ Data field ☐ Reference

Data field Reference

Data Type

Size  Precision

Show as a column in the task list

☒ Hide ☐ Hidden by default ☐ Show by default

☐ Read Only ☐ Defines field ☐ Track changes

Add new Attribute Copy Attribute Delete Attribute

2. **Date:** This attribute is used to capture the date and time when a comment was saved
- In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Date Field** option.
  - From the **Data Type** dropdown, select **DateTime (Date with Time)**.

Entity Attribute

Name

Title

Description

☒ Data field ☐ Reference

Data field Reference

Data Type


Size  Precision

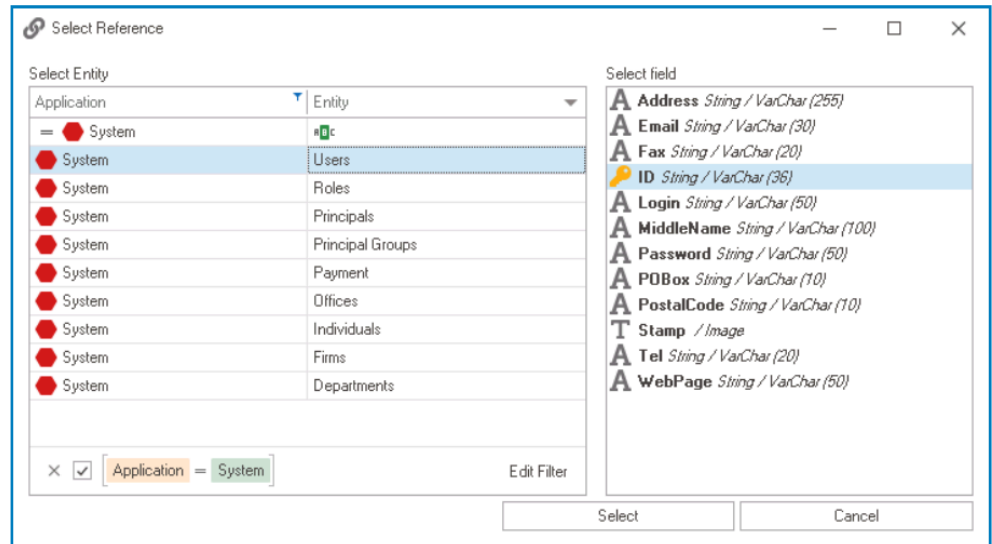
Show as a column in the task list

☒ Hide ☐ Hidden by default ☐ Show by default

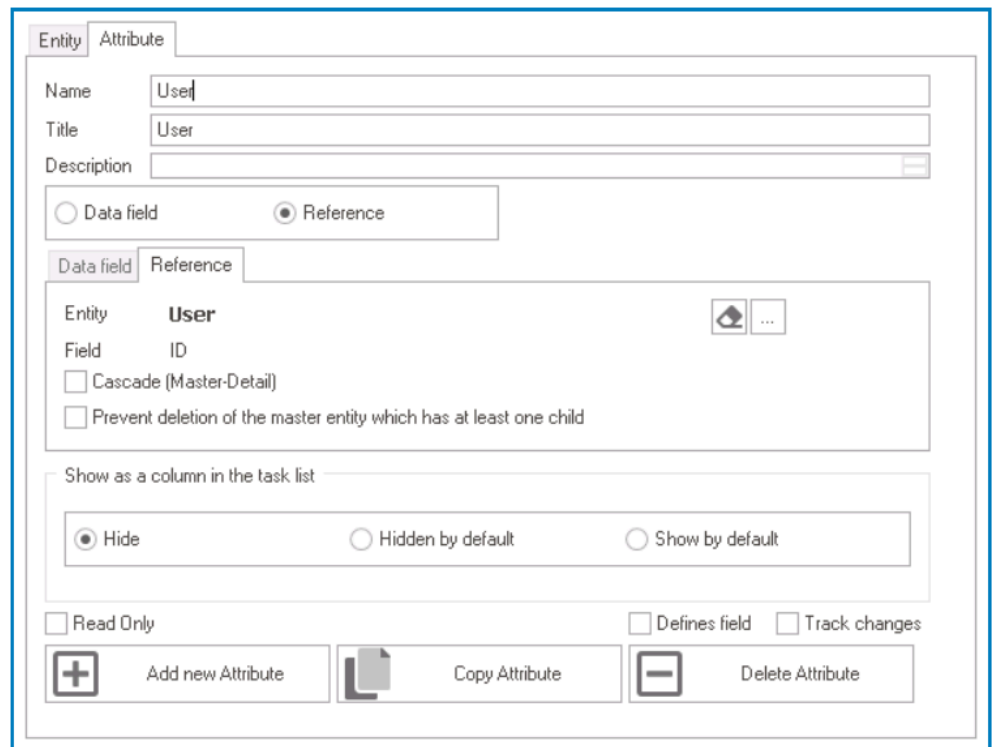
☐ Read Only ☐ Defines field ☐ Track changes

Add new Attribute Copy Attribute Delete Attribute

3. **User:** This attribute is used to capture and store the identity of the user who left the comment.
- In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the  button. It will display the Select Reference window. From the entity section, select **System – Users**. From the field section, select **ID**.




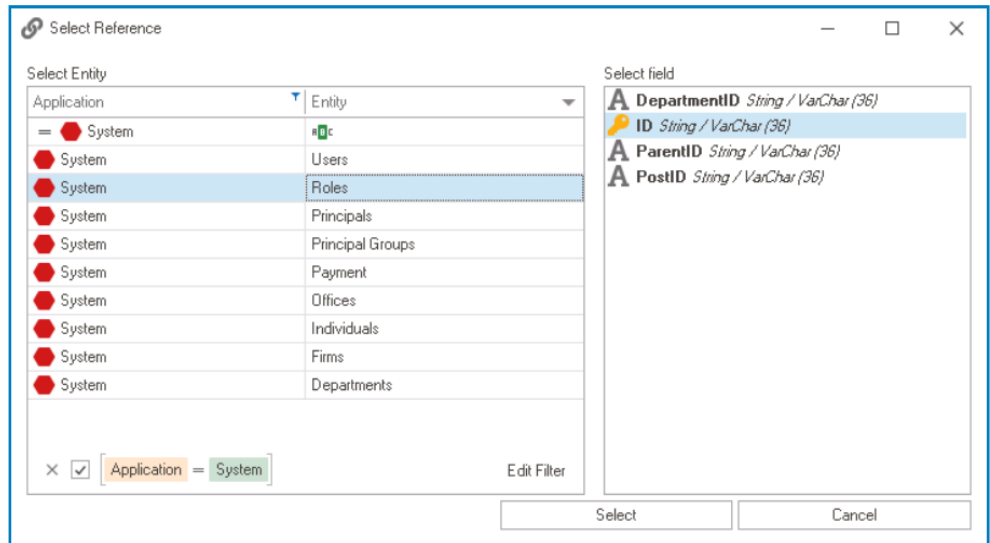
- Selecting the system reference will redirect to the Attribute tab, where the entity section will display **User** and the field section will display **ID**.



4. **Role:** This attribute is used to capture and store the user's designated role in the organization.
- In the **Name** field, provide a name for the attribute.



- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Reference** option.
- Click on the  button. It will display the Select Reference window. From the entity section, select **System – Role**. From the field section, select **ID**.



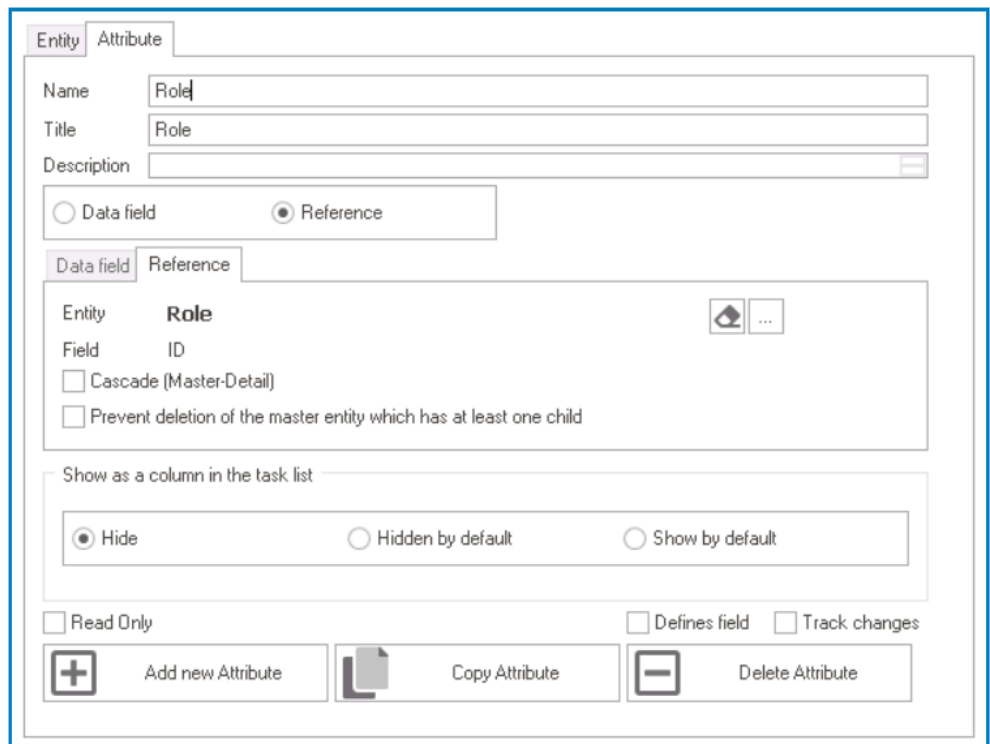
The **Select Reference** dialog box is shown. It has two main sections: **Select Entity** and **Select field**.

**Select Entity:** A table with columns **Application** and **Entity**. The **Application** column has a dropdown menu set to **System**. The **Entity** column lists various entities: **System**, **Users**, **Roles** (highlighted), **Principals**, **Principal Groups**, **Payment**, **Offices**, **Individuals**, **Firms**, and **Departments**. Below the table, there is a filter section with a checkbox ☒ and the text **Application = System**. An **Edit Filter** button is also present.

**Select field:** A list of fields for the selected entity: **DepartmentID String / VarChar (36)**, **ID String / VarChar (36)** (highlighted), **ParentID String / VarChar (36)**, and **PostID String / VarChar (36)**.

At the bottom, there are **Select** and **Cancel** buttons.

- Selecting the system reference will redirect to the **Attribute** tab, where the entity section will display **Role** and the field section will display **ID**.



The **Attribute** configuration dialog box is shown. It has two tabs: **Entity** and **Attribute**. The **Attribute** tab is active.

**Name:**

**Title:**

**Description:**

**Data field** ☐ **Reference** ☒

**Data field** ☐ **Reference** ☒

**Entity:** **Role**

**Field:** **ID**

☐ Cascade (Master-Detail)

☐ Prevent deletion of the master entity which has at least one child

**Show as a column in the task list:**

☒ Hide ☐ Hidden by default ☐ Show by default


☐ Read Only ☐ Defines field ☐ Track changes

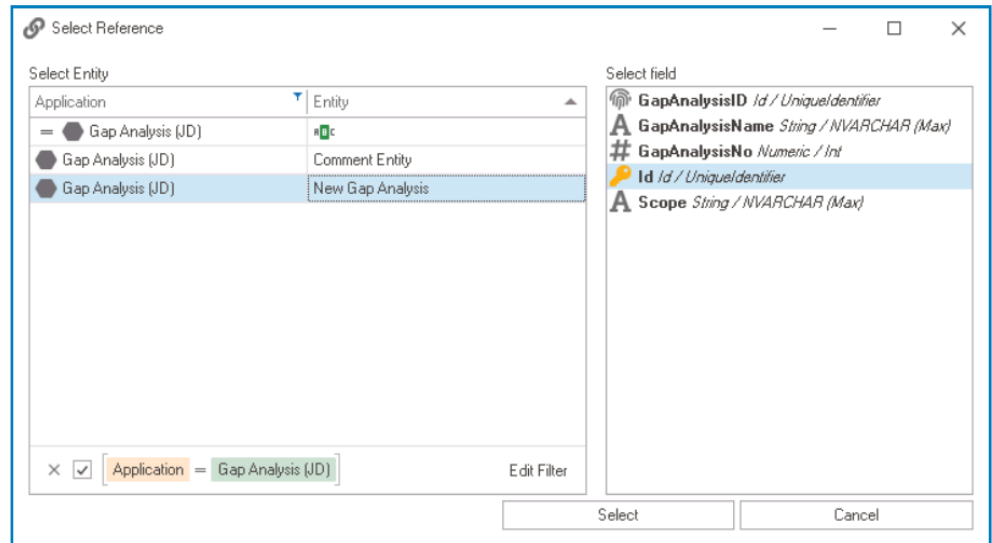
**Buttons:** **Add new Attribute** (with a plus icon), **Copy Attribute** (with a copy icon), and **Delete Attribute** (with a minus icon).

## • Create Foreign Keys

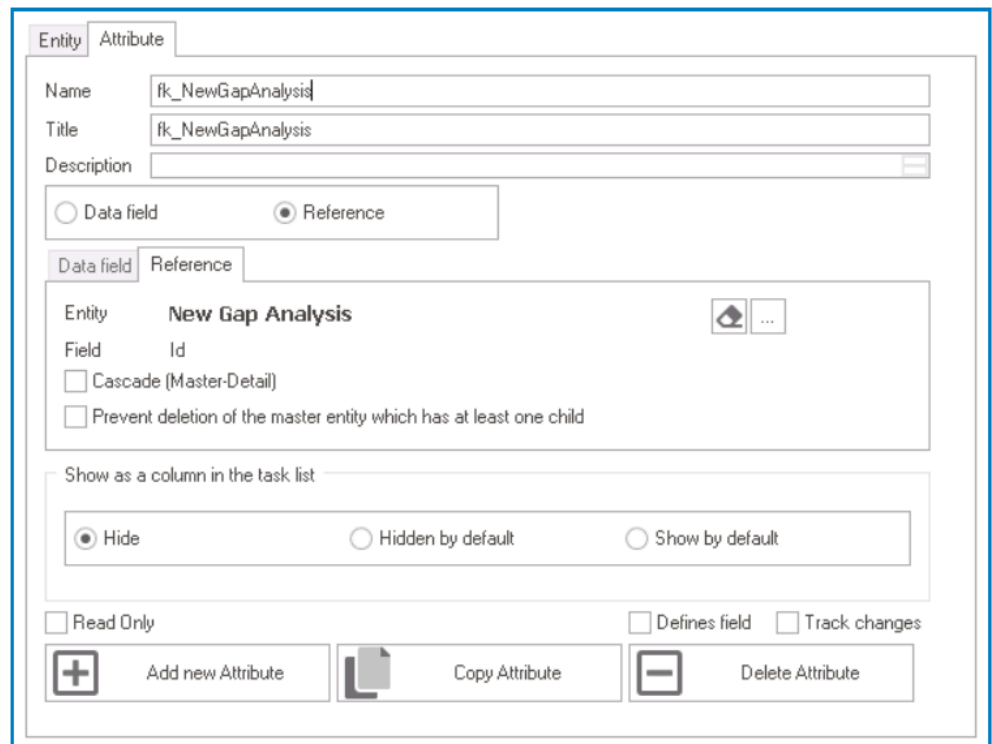
1. **Foreign Key Towards Main Entity:** This foreign key is linked to the entity associated with the form where the comment control is being implemented.
  - In the **Name** field, provide a name for the foreign key.
  - In the **Title** field, either specify a custom title for the foreign key or let the

system automatically generate it based on the provided name.


- Choose the **Reference** option.
- Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [MainEntity]**. From the field section, select **ID**.

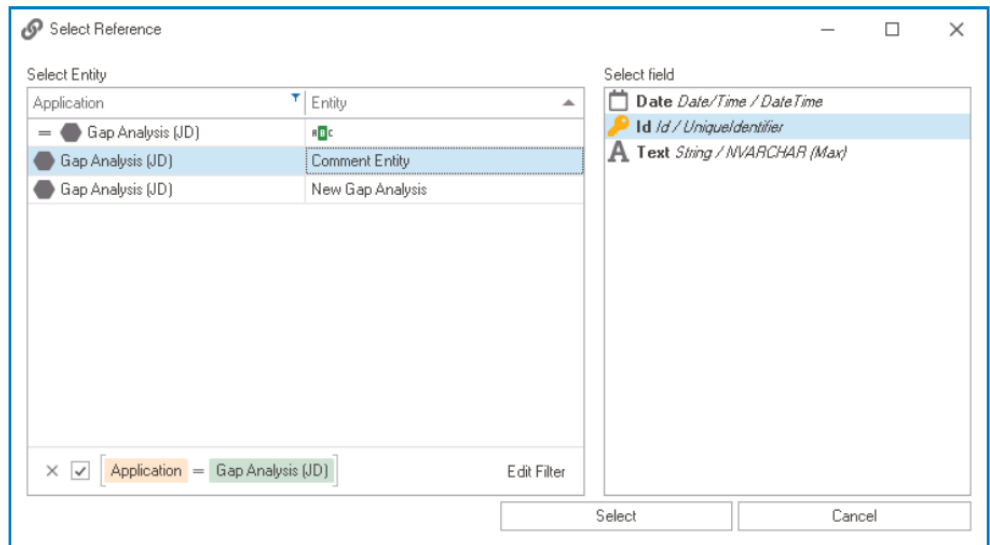


- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[MainEntity]** and the field section will display **ID**.

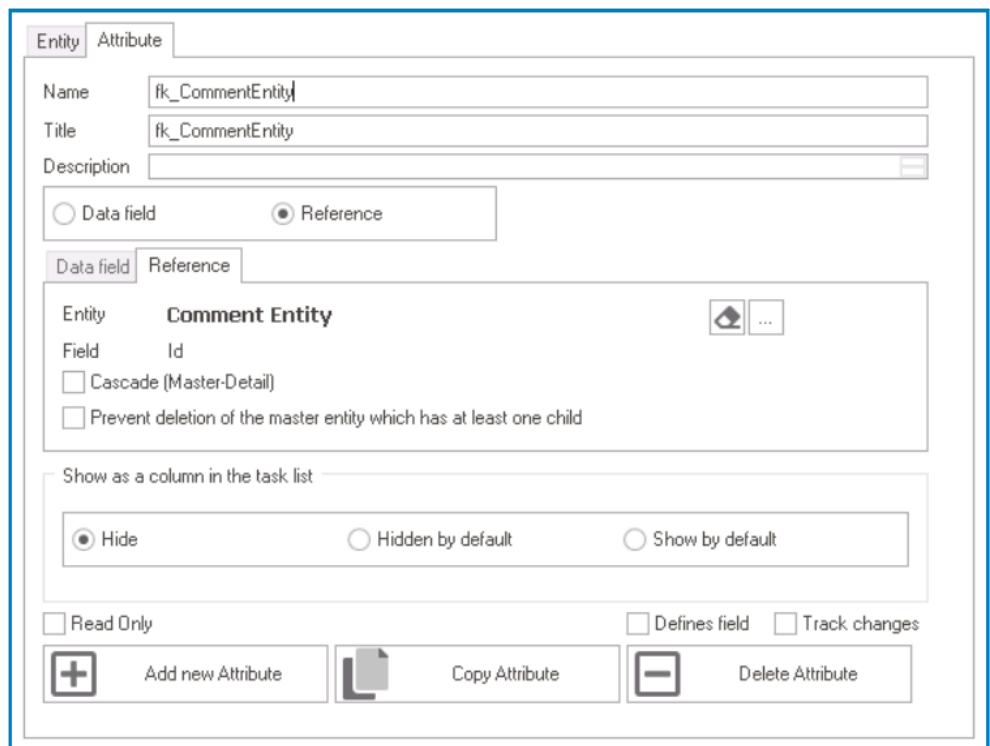


2. **Foreign Key Towards Comment Entity:** This foreign key is linked to the comment entity—that is, the entity that contains the comments.
  - In the **Name** field, provide a name for the foreign key.
  - In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.

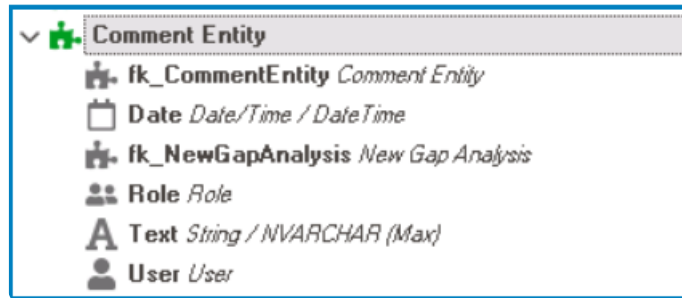
- Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [CommentEntity]**. From the field section, select **ID**.



- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[CommentEntity]** and the field section will display **ID**.



- Once users have created the necessary attributes and foreign keys for the comment control, the end-result should be the following list:

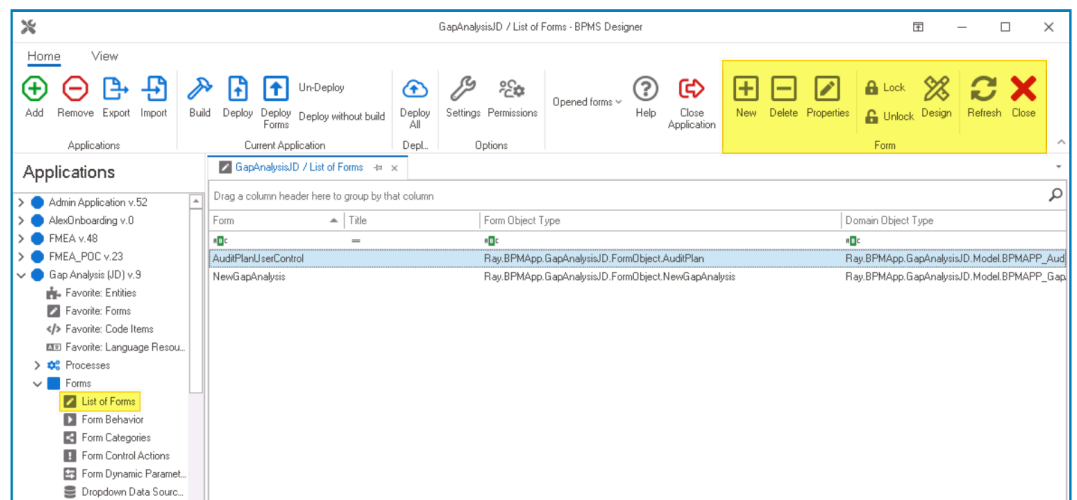


! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

### 3. [Create Comment Form](#)

- **Access Form Properties**

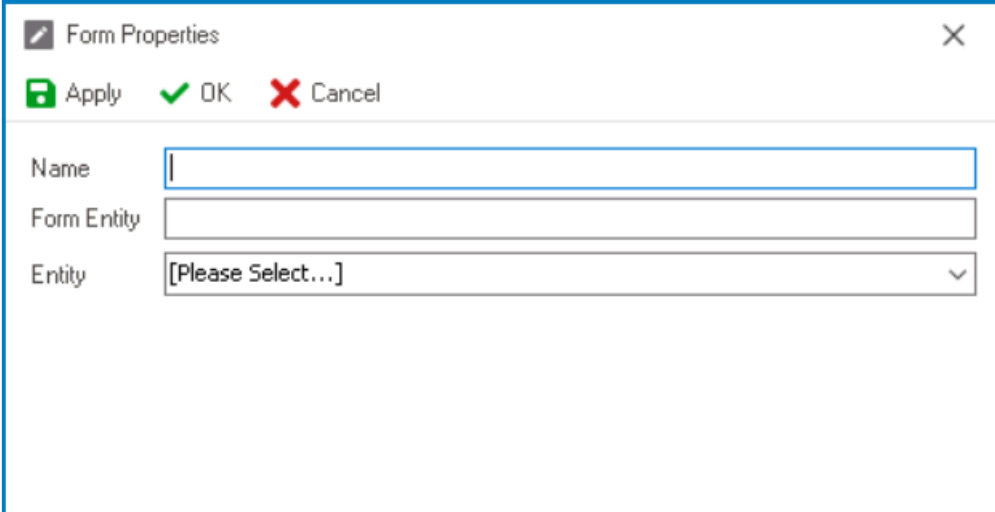
- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the [Form Properties](#) window.



- In the [Form Properties](#) window, users can create a new form for the application.



The **Form Properties** dialog box contains the following fields:

- Name:** A text input field for naming the form.
- Form Entity:** A text input field for specifying the form entity.
- Entity:** A dropdown menu currently showing "[Please Select...]" with a downward arrow.

At the top, there are buttons for **Apply** (with a floppy disk icon), **OK** (with a green checkmark icon), and **Cancel** (with a red X icon).

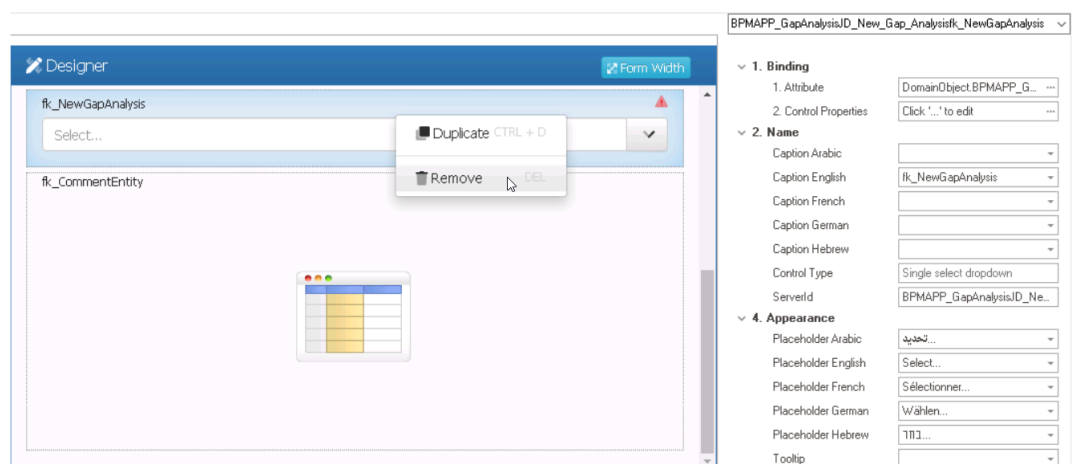
- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the comment entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.



The purpose of the comment form is to link the user and role attributes to their corresponding controls in the interface and to their respective lists in the database, enabling accurate and secure capture, storage, and display of user and role information in the comment field.

#### 4. Configure Comment Form

- Once users have created the comment form and launched the Form Designer, they can proceed to finalize the form's configuration.
- **Remove Foreign Key Controls**
  - To remove the foreign key controls, simply right-click on them in the design environment and select the **Remove** option.

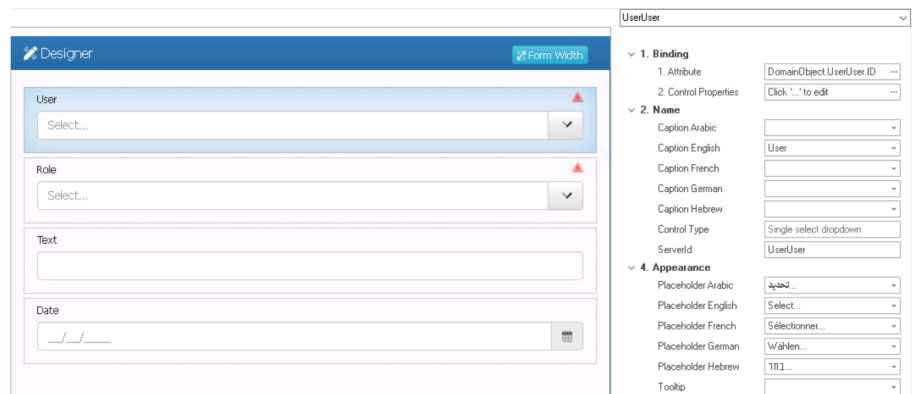


- **Link User and Role Attributes**
  - To link the user and role attributes to their respective lists in the database, take

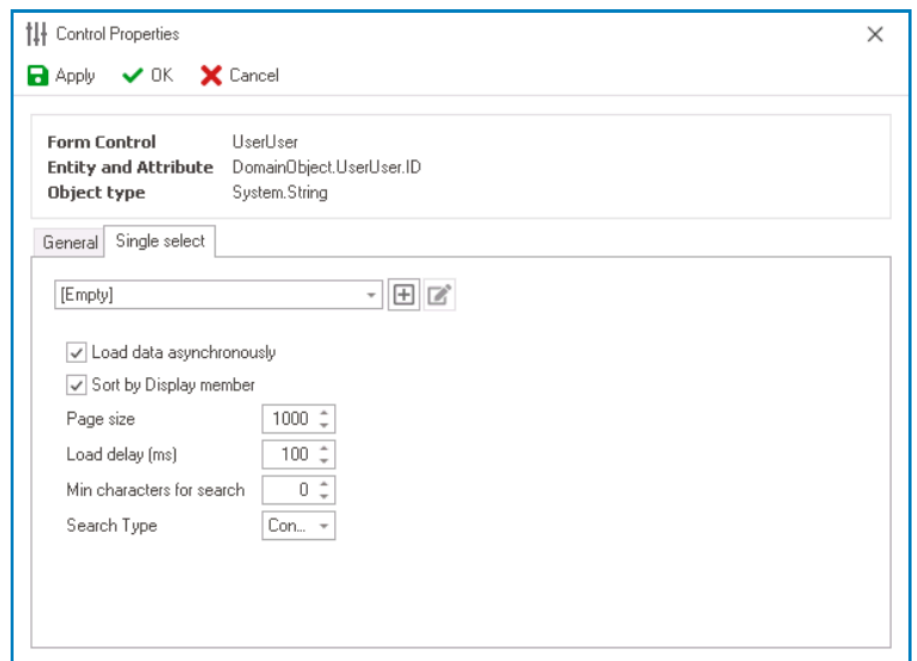
the following steps:

- **User Control**

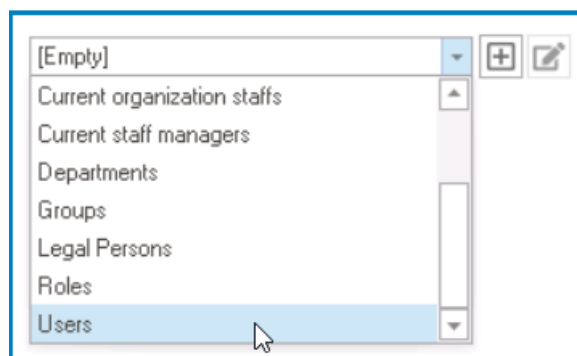
- Select the user control from the design environment. It will display its associated properties in the Control Properties panel.



- Click on the ellipses inside the **Control Properties** field, and then navigate to the Single-Select tab in the displayed window.



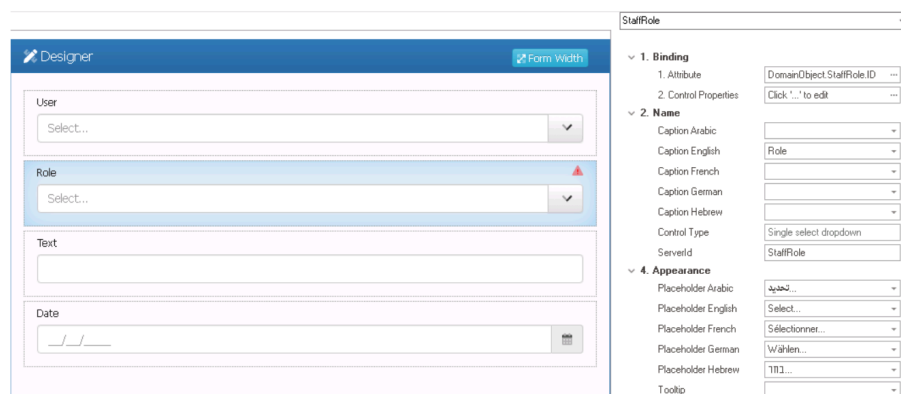
- From the dropdown, select **Users**.



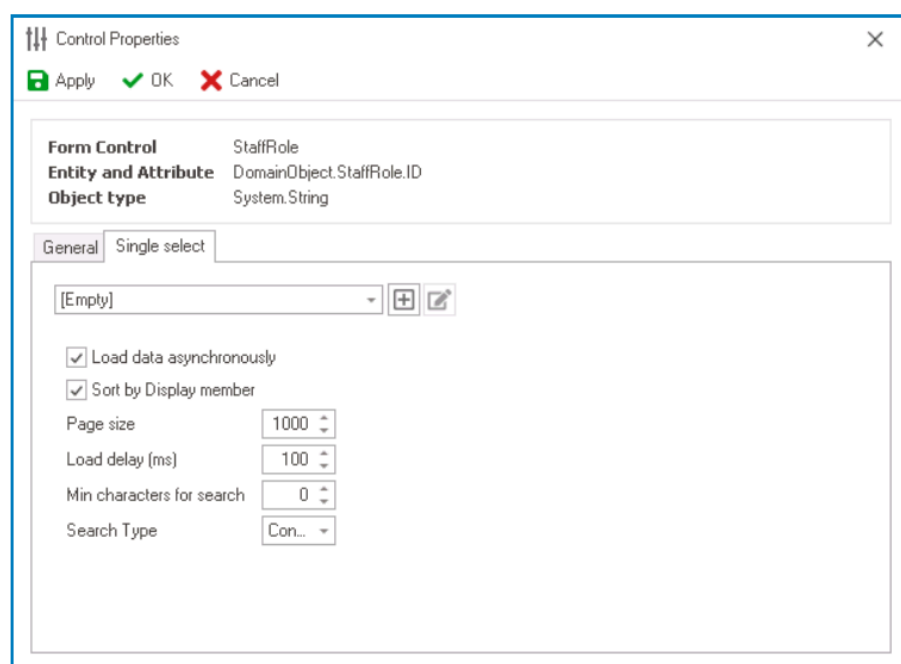
- **Role Control**

- Select the role control from the design environment. It will display its

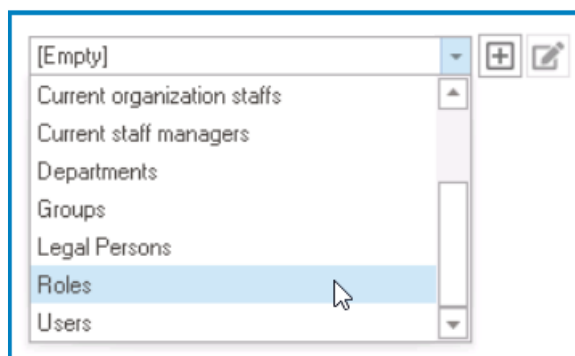
associated properties in the Control Properties panel.



- Click on the ellipses inside the **Control Properties** field, and then navigate to the Single-Select tab in the displayed window.



- From the dropdown, select **Roles**.



To improve performance, make sure to select the **Load Data Asynchronously** checkbox for both controls.

- Click **Save Form** and confirm the prompt to generate a link in the main menu.

5. [Configure Comment Control Binding Properties](#)

- Once users have created and configured the comment form, they can return to their main form and define the binding properties for the comment control.



## 12.1.2.1.1.5.2. Comment – Control Governance

In the Comment tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of comment controls.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Comment]

Apply OK Cancel

Control Name **Comment**

Comment Validation




|  |             |         |       |
|--|-------------|---------|-------|
| <input type="checkbox"/> + Enable New    | New Form    | [Empty] | + ✎ - |
| <input type="checkbox"/> - Enable Delete | Delete Form | [Empty] | + ✎ - |
| <input type="checkbox"/> ✎ Enable Edit   | Edit Form   | [Empty] | + ✎ - |

1. **Enable New:** This is a checkbox which, if selected, allows users to add new comments.
  - If this checkbox is selected, it will enable the **New Form** field.
    - **New Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'new' functionality.
      - To define a new form behavior, select the button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the button.
      - To remove a form behavior, select the button.







To set up the 'New Form', users only need to provide a name, a title, and select the **Starter** checkbox.

2. **Enable Delete:** This is a checkbox which, if selected, allows users to delete comments.
  - If this checkbox is selected, it will enable the **Delete Form** field.
    - **Delete Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'delete' functionality.


- To define a new form behavior, select the  button.
  - It will display the [Form Behavior Properties](#) window.
- To modify an existing form behavior, select the  button.
- To remove a form behavior, select the  button.

 It is not necessary for users to set up the 'Delete Form'.

3. **Enable Edit:** This is a checkbox which, if selected, allows users to edit comments.
- If this checkbox is selected, it will enable the **Edit Form** field.
    - **Edit Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'edit' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.
      - To remove a form behavior, select the  button.

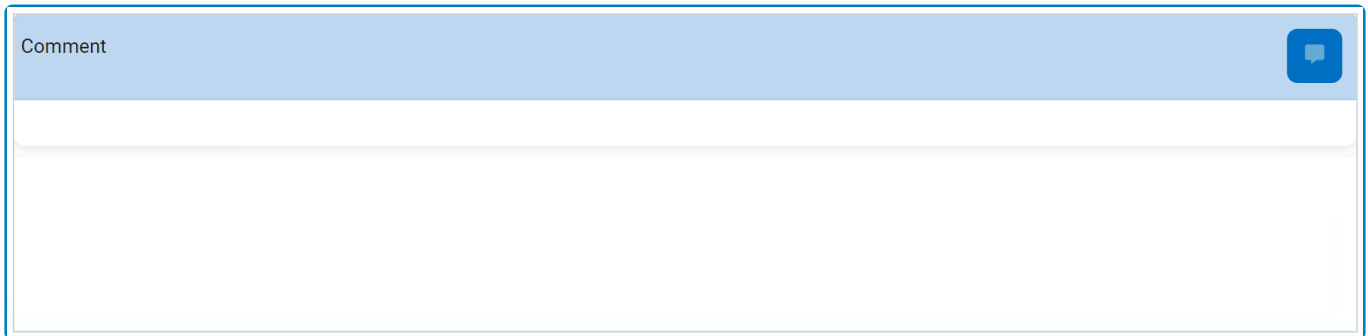
 To set up the 'Edit Form', users only need to provide a name and a title.


4. **Apply:** This button allows users to save their changes.
5. **OK:** This button allows users to save their changes and close the window.
6. **Cancel:** This button allows users to cancel their changes and close the window.

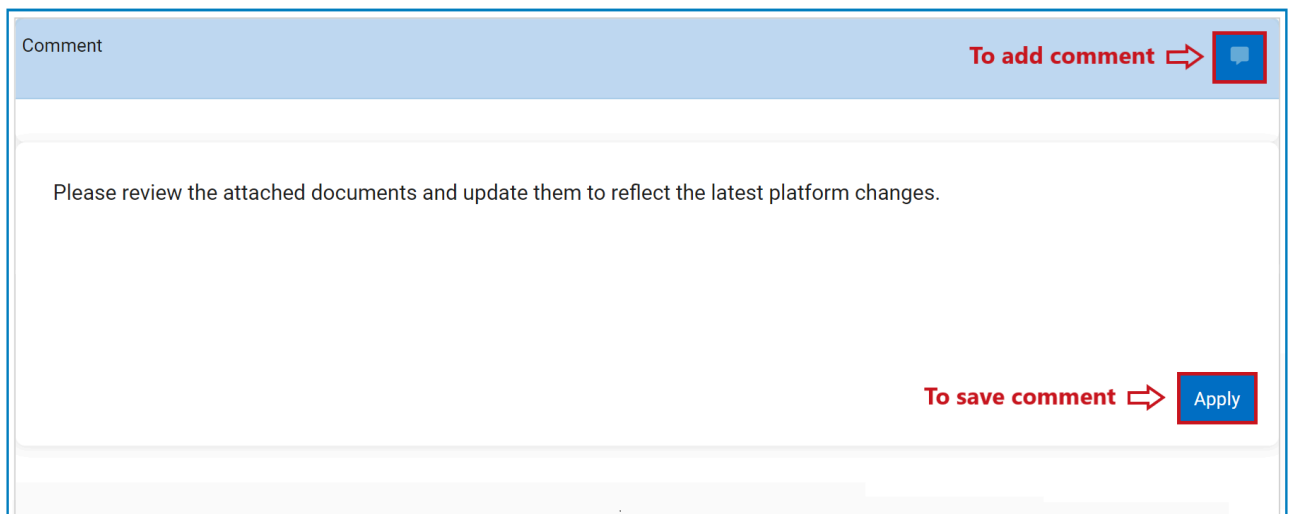
 In the Validation tab of this window, users can configure the required field validation properties for the comment control. For more information on this matter, please [click here](#) and navigate to the relevant section.


## 12.1.2.1.1.5.3. Comment – Web Display

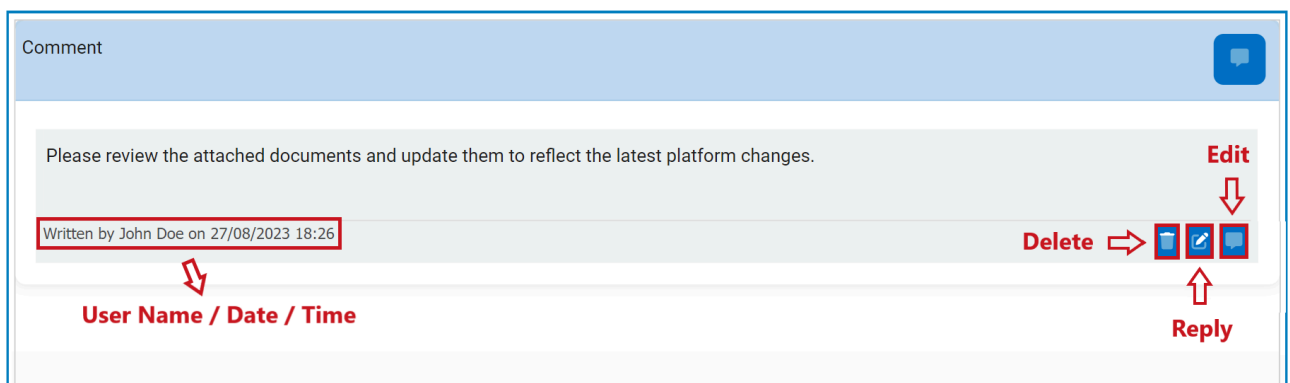
For those unfamiliar with using the comment field in the **DBP** web application, the following instructions will guide you through the process:




1. To add a comment, click on the  icon in the top right corner, input your comment(s) in the text field, and then click on the **Apply** button to save your work.



2. To edit a comment, click on the  icon in the bottom right corner, make changes as desired to the original comment, and then click on the **Apply** button to save your work.




3. To delete a comment, click on the  icon in the bottom left corner.
  - If selected, users will be alerted of the action with the following confirmation message:


Are you sure you want to delete?

Submit


Cancel

- Click on either the **Submit** button to confirm the deletion or the **Cancel** button to cancel it.




4. To reply to a comment, click on the  icon in the bottom right corner, input your reply, and then click on the **Apply** button again to save your work.

Comment 

Please review the attached documents and update them to reflect the latest platform changes.

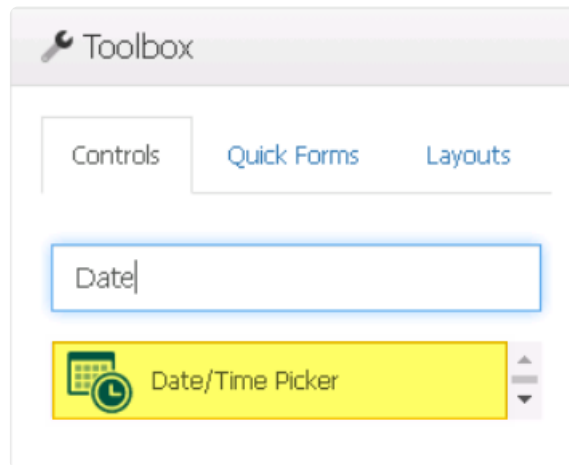
Written by John Doe on 27/08/2023 18:26 

Please review the attached documents and update them to reflect the latest platform changes.  
Will do!

Written by Jane Doe on 27/08/2023 20:50   

## 12.1.2.1.1.6. Date/Time Picker

When incorporated into a form, date/time pickers offer a simple and efficient means of entering or displaying date and time information, simplifying the process of tracking tasks or deadlines.



When a date/time picker control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

\* This section of the manual covers only the unique properties of the date/time picker control. For information on configuring properties under the categories of **Name**, **Control Behavior**, **Appearance**, and **Caption Appearance**, please [click here](#) and navigate to the relevant section.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

\* When creating attributes for date/time picker controls, users have the option to choose between the **Date**, **DateTime**, and **DateTime2** data types, each of which corresponds to a specific date and/or time format.

2. **Misc:** In this category, users can define additional properties for the control.
  - **Display Type:** This is a single-select dropdown field that allows users to choose the display type of the date/time picker control.
    - Display Types: Date/Time, Date, or Time.
      - Date/Time This option allows the control to store and display both date and time information.
      - Date: This option allows the control to store and display only the date information.
      - Time: This option allows the date/time picker control to store and display only

the time information.



When adding a date field to a grid, users should ensure that the selected format matches the display type chosen for the date/time picker control. This is because the selected format will override the display type. For more information on how to select this format, [click here](#).

- **Store as UTC:** This is a checkbox which, if selected, allows the date/time entered or displayed in the control to be stored in Coordinated Universal Time (UTC) format.

#### Date/Time Picker Control Preview: Web View

##### Date of Corrective Action




For more information on how to use this field in the web application, [click here](#).

#### Quick Forms Application

To apply the **Quick Forms** feature to date picker controls, take the following steps:

- Drag a date picker control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:

Generate New Fields



Generate fields for added controls?



For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.6.1. Date/Time Picker – Web Display

---

For those unfamiliar with using the date/time picker in the **DBP** web application, the following instructions will guide you through the process:

1. **Date/Time**

If Date/Time is selected as a display type, the date/time picker will display a calendar with time input options.

**Due Date**




Users can enter a date and time into this field either manually or by selecting it from the calendar.

### Instructions

To display the calendar, click on the  icon.

**Due Date**

| 

< MARCH 2023 >

| MON | TUE | WED | THU | FRI | SAT | SUN |
|-----|-----|-----|-----|-----|-----|-----|
| 27  | 28  | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |


11 : 42

Today OK Cancel

- To view all the months in the year, click on the month.
  - To view the previous or next year's months, use the side arrows to navigate.



**Due Date**

| 

< MARCH 2023 >

| MON | TUE | WED | THU | FRI | SAT | SUN |
|-----|-----|-----|-----|-----|-----|-----|
| 27  | 28  | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |

11 : 42

Today OK Cancel

- To view the current set of ten years, click on the year.
  - To view the previous or next decade, use the side arrows to navigate.

The image shows a 'Due Date' picker interface. At the top, there is a text input field with a vertical line cursor and a calendar icon. Below this is a calendar grid for the year 2023. The year '2023' is displayed in blue at the top of the calendar, with a mouse cursor hovering over it. The calendar shows the months Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, and Dec. The month 'Mar' is highlighted. At the bottom of the calendar, there are two spinners for the time, showing '11' and '42' with a colon between them. Below the time spinners are three buttons: 'Today', 'OK', and 'Cancel'.

- To view the decades of the current century, click on the decade
  - To view the decades of the previous or next century, use the side arrows to navigate.

- To select the time, use the numeric updown field.
  - In this field, users can enter numbers either manually or by clicking the increment/decrement arrows.

✳ To go to the current day, click on **Today**. To save your selection and exit, click on **OK**. To cancel and exit without making a selection, click on **Cancel**.

Once selected, the date and time will populate the field in the following manner:

## 2. **Date**

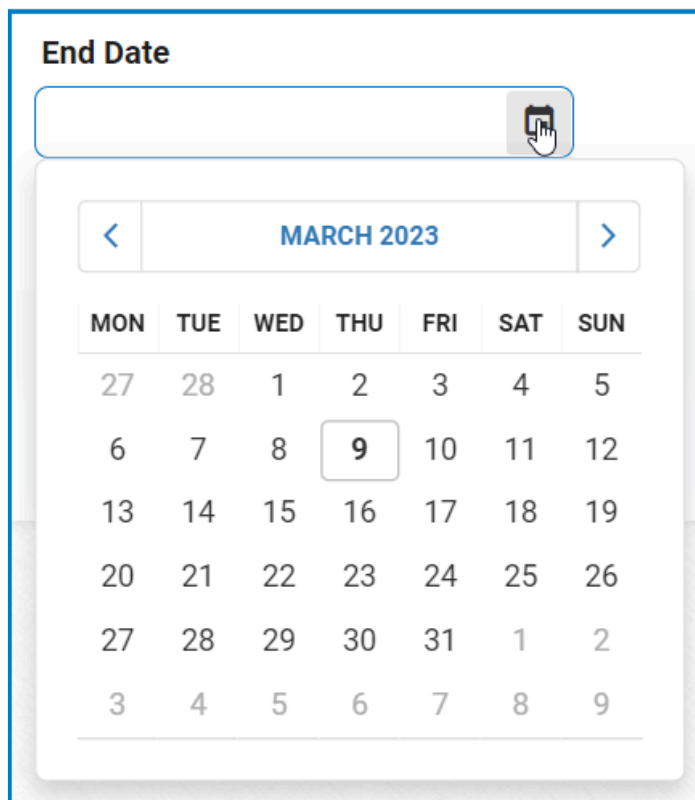
If Date is selected as the display type, the date picker will display a calendar allowing the user to select a single date.

**End Date**

Users can enter a date into this field either manually or by selecting it from the calendar.



**Instructions**

To display the calendar, click on the  icon.



Unlike the date/time picker, which also allows for the selection of a time, users cannot input a time when using the date picker. Navigation between months, years, and decades, however, works similarly in both pickers.

Once selected, the date will populate the field in the following manner:

**End Date****3. Time**


If Time is selected as the display type, the time picker will display a list of pre-defined times for the user to select from.

**Start Time**


Users can enter a time into this field either manually or by selecting it from the list.

**Instructions**

To display the list, click on the  icon.

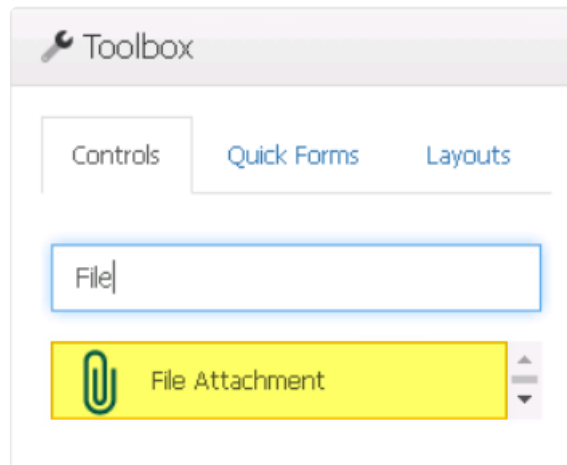
**Start Time**  
  
00:00  
00:30  
01:00  
01:30  
02:00  
02:30  
03:00  
03:30  
04:00

Once selected, the time will populate the field in the following manner:

**Start Time**

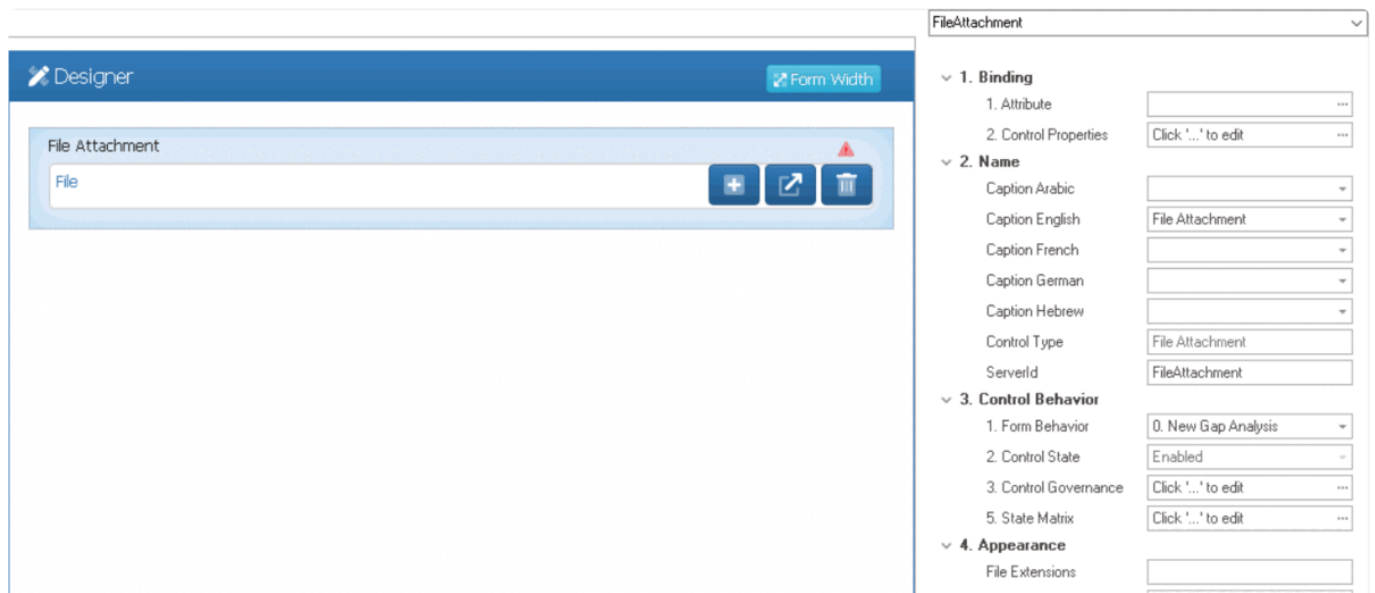
## 12.1.2.1.1.7. File Attachment

The file attachment control allows users to upload and download different types of files, providing a convenient way to share information.



✿ For more information on how to access and manage file data using scripts, [click here](#).

When a file attachment control is selected in the design environment, its associated properties are displayed in the Control Properties panel.




✿ To access the **Control Behavior** category, users must first create the form behavior.


1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

 For more information on the required attribute for file attachment controls, [click here](#).


2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
      - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
        - This window consists of two tabs:
          - [File Attachment](#)
          - [Validation](#)

 Please note that users are limited to configuring only the required field validation properties for file attachment controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).

 Please note that the linked section contains information on setting an initial value, which is not applicable to file attachment controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **File Extensions:** This field allows users to specify the types of files that can be uploaded to the control.
    - Users can input specific file extensions in this field to restrict uploading access to only those files types.
      - To include multiple file types, users need to separate each file type with a comma.

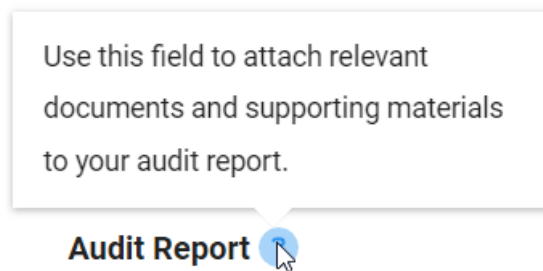
 If no file type is specified, all file types can be added to the attachment control.


- **Max File Count:** This is a numeric updown field that allows users to set the maximum number of files that can be attached to the control.
  - By default, the maximum file count is set to 1. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- **Max File Size:** This is a numeric updown field that allows users to set the maximum combined size limit, in megabytes, per file attached to the control.
  - By default, the maximum file size is set to 2 MB. To adjust this value, users can enter

numbers into the field either manually or by clicking the increment/decrement arrows.

**!** Please note that the maximum file size inputted in this field cannot exceed 2 GB (or 2048 MB), which is the limit per file attached to the control.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon (  ) next to the control.
  - **File Attachment Tooltip Preview: Web View**



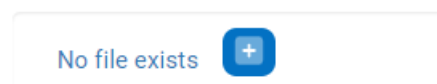
 A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.


5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - For more information on the fields in this category, [click here](#).

#### File Attachment Control Preview: Web View

- If the max file count is set to 1, the file attachment field will take on the following appearance in the web application:

#### Document



-  : This button allows users to upload a single file.
  - If selected, a file dialog will appear, enabling users to browse and choose a file from their local system.



- If the max file count is greater than 1, the file attachment field will take on the following appearance in the web application:

### Documents

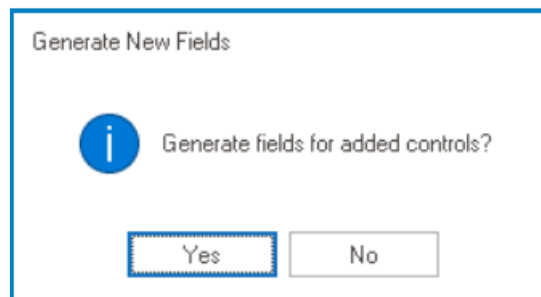


\* For more information on how to use this field in the web application, [click here](#).

### Quick Forms Application

To apply the **Quick Forms** feature to file attachment controls, take the following steps:

- Drag a file attachment control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:



\* For more information on the **Quick Forms** feature, [click here](#).

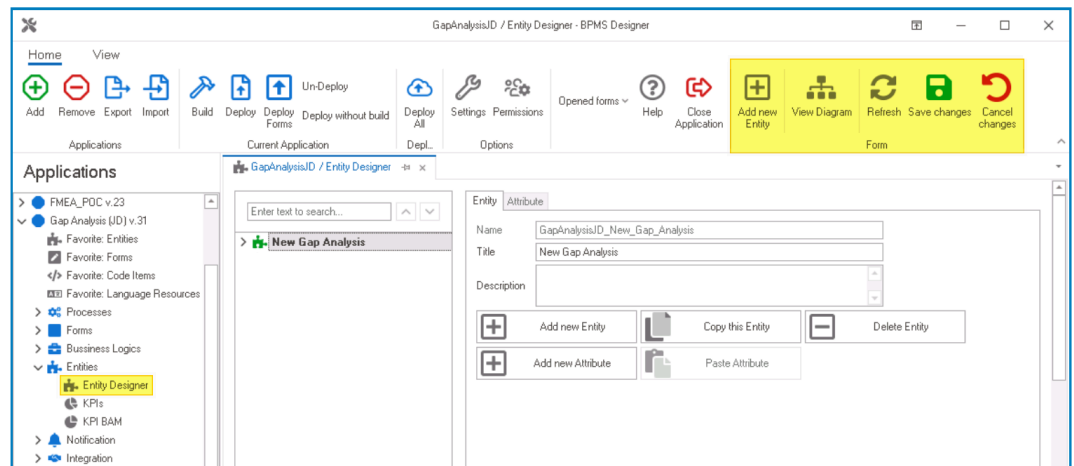
# 12.1.2.1.1.7.1. File Attachment – Attribute

To create file attachment controls and their associated attributes, follow these steps:

## 1. [Create Attribute](#)

### • Access Attribute Properties

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From the Entity tab, select the **Add New Attribute** option. It will direct you to the Attribute tab where you can define the properties of the new attribute.



- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Data Field** option.
- From the **Data Type** dropdown, select one of the following string data types:
  - **NVarChar**
  - **NVarChar\_Max**
  - **VarChar**
  - **VarChar\_Max**
- In the **Size** field, specify a maximum size limit if the chosen string data type does not have 'max' in its name (e.g. VarChar, NVarChar).
  - Typically, for the **NVarChar** data type, a size of 100 is used to accommodate the number of attachable files.

The screenshot shows the 'Attribute' configuration window. It includes fields for Name, Title, and Description. Below these are radio buttons for 'Data field' (selected) and 'Reference'. The 'Data field' section contains a 'Data Type' dropdown set to 'NVarChar (Variable-length Unicode string)', a 'Size' input set to '100', and a 'Precision' input. A section titled 'Show as a column in the task list' contains radio buttons for 'Hide' (selected), 'Hidden by default', and 'Show by default'. At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes'. Three buttons are at the very bottom: 'Add new Attribute' (with a plus icon), 'Copy Attribute' (with a document icon), and 'Delete Attribute' (with a minus icon).

\* The attribute capacity specifies the maximum number of files that can be attached, with a minimum value set. By default, attached files of the form are stored in the DMS folder within the application's installation path. This location, however, can be modified during the installation process.


## 2. [Bind Attribute to File Attachment Control](#)

- Once users have created the appropriate attribute, they can return to their form and bind the attribute to the file attachment control.


## 12.1.2.1.1.7.2. File Attachment – Control Governance

In the File Attachment tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of file attachment controls.

1. **Enable Upload:** This is a checkbox which, if selected, allows users to upload files to the file attachment control.
2. **Enable Delete:** This is a checkbox which, if selected, allows users to delete attached files.
3. **Enable Download:** This is a checkbox which, if selected, allows users to download attached files.
4. **Enable Scan:** This is a checkbox which, if selected, allows users to scan attached files.
5. **Attach Files to Process:** This field allows users to attach files to the file attachment control in advance, so that the files are readily available and displayed when the control is opened in the web.

- To upload a file, select the  button.
  - It will display the [Application Files](#) window, in which users can add and select the file.
  - **Pre-Attached Files Preview: Web View**

### Template

- To remove a file, select the  button.

\* Users can upload multiple files to this field, but each file must be added individually.

6. **Apply:** This button allows users to save their changes.
7. **OK:** This button allows users to save their changes and close the window.
8. **Cancel:** This button allows users to cancel their changes and close the window.

\* In the Validation tab of this window, users can configure the required field validation properties for the file attachment control. [For more information on this matter, please click here](#) and navigate to the relevant section.

## 12.1.2.1.1.7.3. File Attachment – Using Scripts to Manage Data

---

A file attachment control is bound to a string attribute (of the string data type). This attribute stores the unique identifier of the group (GroupID). The group contains one or more files with their own unique identifier (FileID).

To retrieve and manipulate files within a group, users can apply any of the following services in their custom code:

### 1. Get Files by GroupID

```
List<DmsFile> bpmAppService.BPMSServices.DmsService.GetFileFromControl (string GroupID, bool isWithContent)
```

### Code Example

```
var files = bpmAppService.BPMSServices.DmsService.GetFileFromControl(domainObject.VendorManualMultiple, false);
```

### 2. Get File Content by FileID

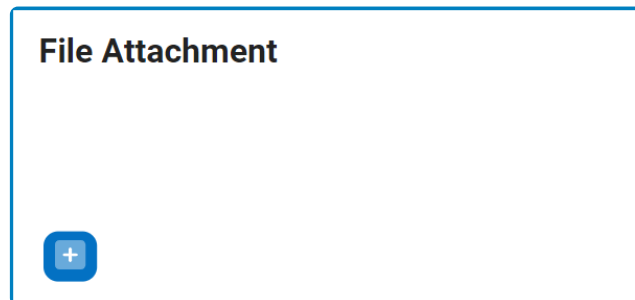
```
byte[] bpmAppService.BPMSServices.DmsService.GetFileContent(Guid FileID)
```

### 3. Add a New File to the Group

```
Guid bpmAppService.BPMSServices.DmsService.AddAttachFile(byte[] buffer, string name, string GroupID)
```

## 12.1.2.1.1.7.4. File Attachment – Web Display

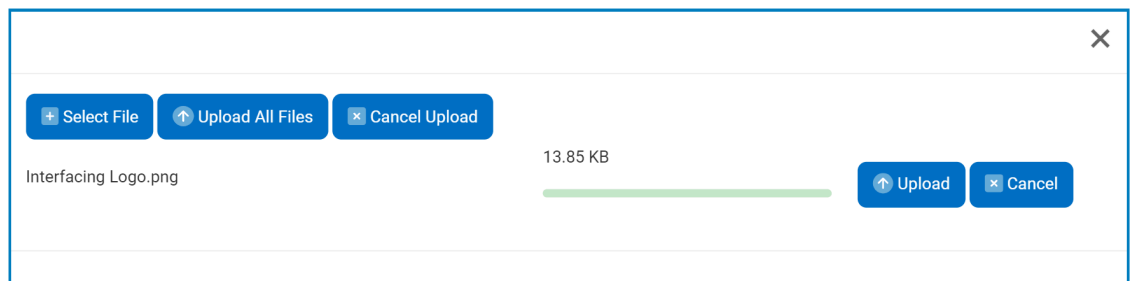
For those unfamiliar with using the file attachment field in the **DBP** web application, the following instructions will guide you through the process:



1. To attach a file, click on the  button. A pop-up window will then appear with the following functions:



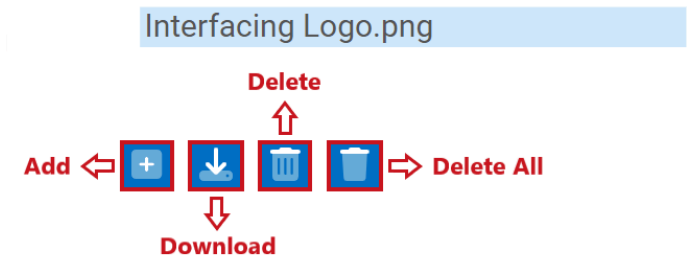
- **Select File:** This button allows users to choose files for upload.
  - If selected, a file dialog will appear, enabling users to browse and choose files from their local system.
  - Once a file has been selected, it will populate the form in the following fashion:




- **Upload:** This button allows users to upload an individual file.
    - **Cancel:** This button allows users to cancel the upload of an individual file.
  - **Upload All Files:** This button allows users to upload all files currently populating the form.
  - **Cancel Upload:** This button allows users to cancel the upload of all files currently populating the form. These files will then be cleared from the page.
2. Once all necessary files have been uploaded, users will be returned to the main form wherein the files can be found attached to the field.
    - Users can then use the following set of buttons to upload new files, download existing files, deleted selected ones, or delete them all.

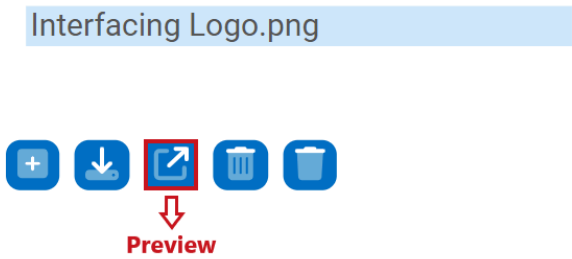


File Attachment



- Users can also preview selected files.
  - The  button will only appear once a file has been selected from the field. It will display the contents of the file in a separate tab.

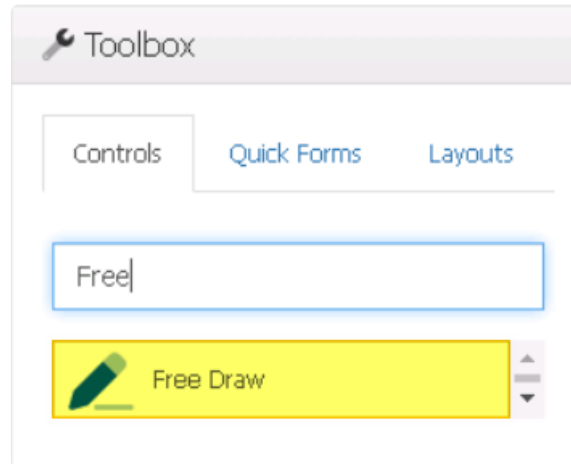
File Attachment



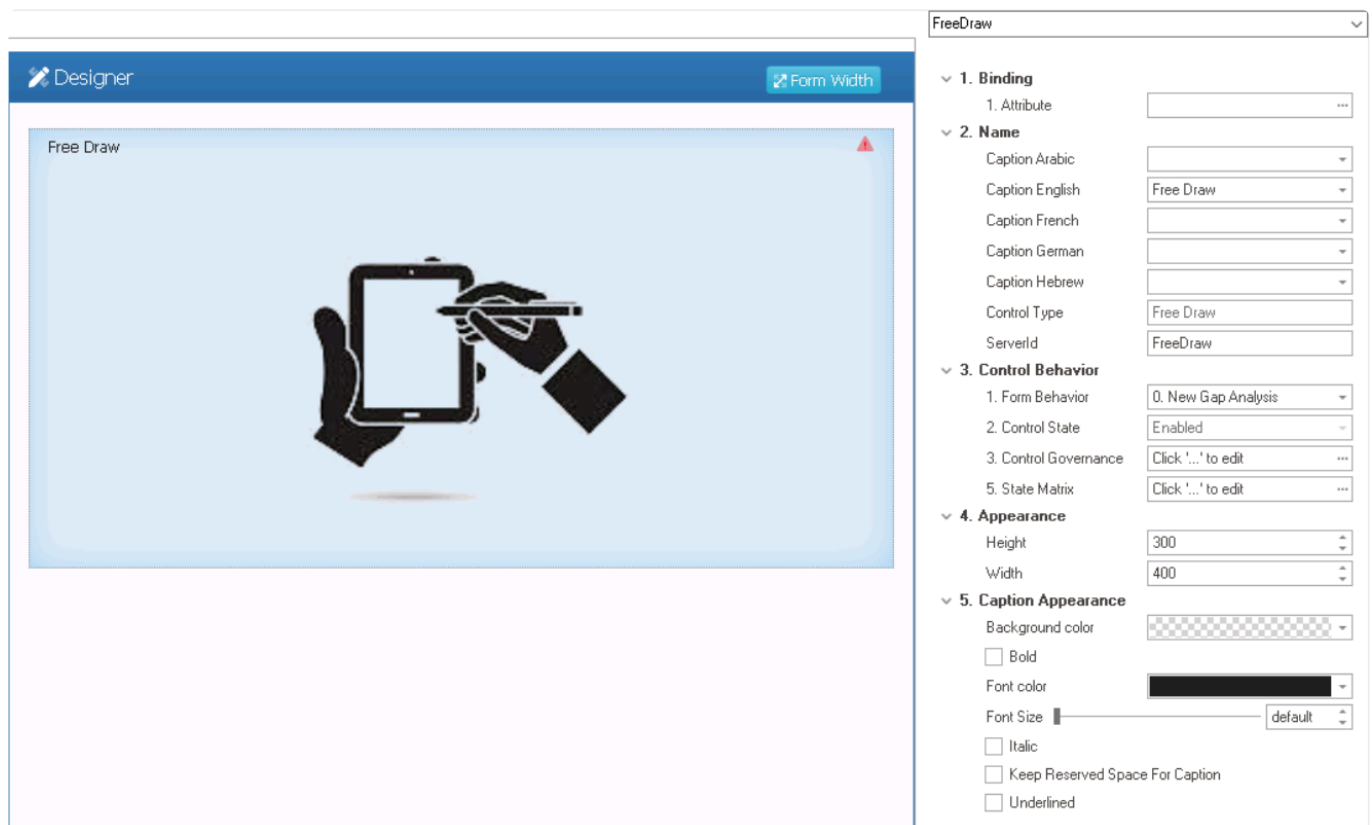
\* The option to preview file attachments is only available for JPEG, PNG, and PDF file types.

## 12.1.2.1.1.8. Free Draw

A free draw control is a digital tool for creating freehand drawings.



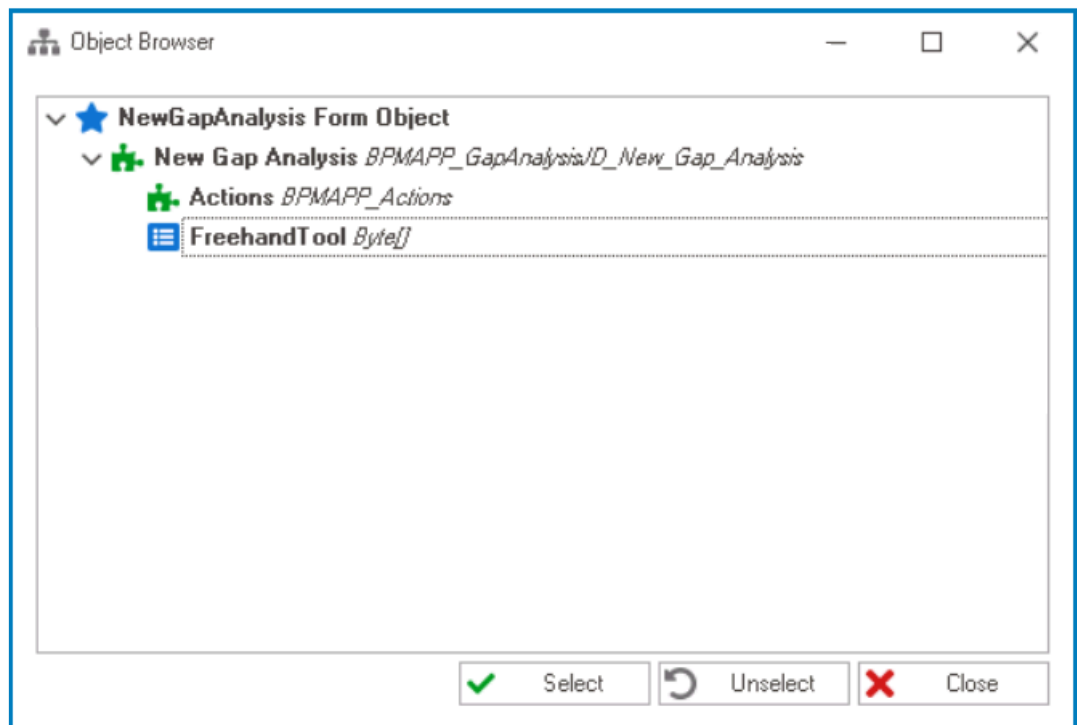
When a free draw control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute.
  - **Attribute:** This field allows users to select the attribute associated to the control.

- To access the list of relevant attributes, click on the ellipses (...) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

DomainObject.FreehandTool ...

\* To create attributes for free draw controls, users must use either the **VarBinary** or **VarBinary\_Max** data type.

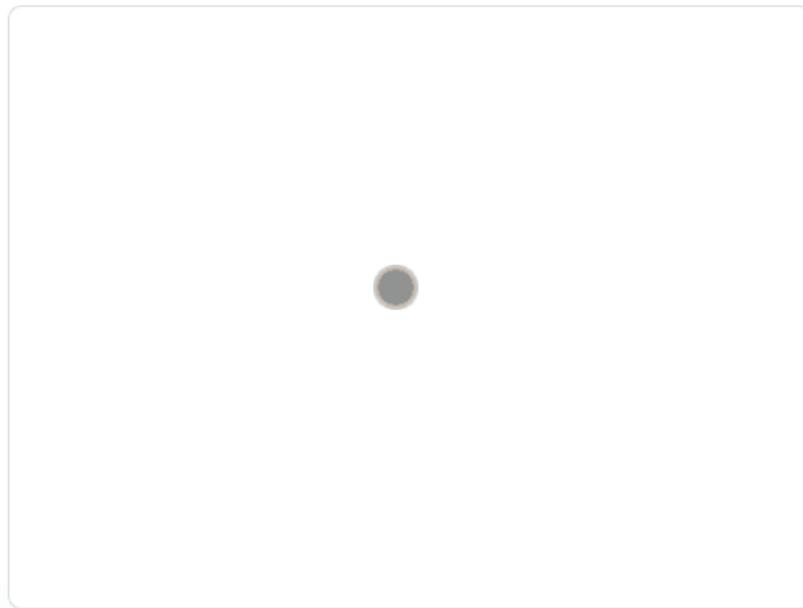
2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - For more information on these properties and how to configure them, [click here](#).


! Please note that the linked section contains information on setting an initial value, which is not applicable to free draw controls and can be ignored. In addition, regarding control governance properties, users are limited to configuring only the required field validation properties for free draw controls and cannot set up required range validations.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
- **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 300. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
  - **Width:** This is a numeric updown field that allows users to set the width of the control.
    - By default, the width is set to 400. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
- For more information on the fields in this category, [click here](#).

#### Free Draw Control Preview: Web View

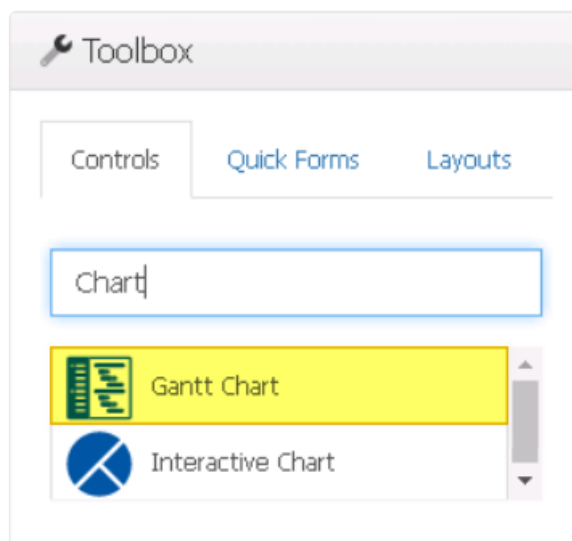
##### Freehand Tool



- To erase or undo the drawing, click on the  button.

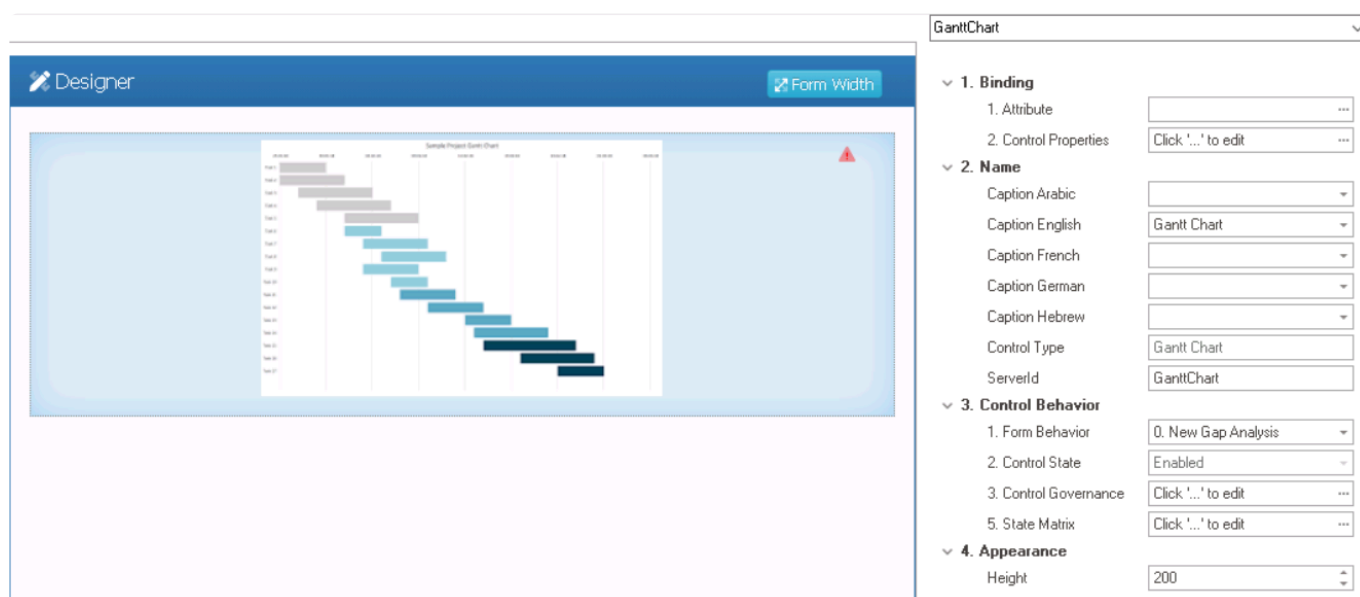
## 12.1.2.1.1.9. Gantt Chart

Charts, like tables and grids, can be powerful tools for visualizing data in a form. In this section, we will explore the capabilities of Gantt charts, one of the two chart controls available in the program. Gantt charts allow for efficient representation of project timelines, tasks, and dependencies, enabling effective project planning and tracking.



\* In the Designer, there are two types of chart controls available: interactive charts and Gantt charts. For more information on interactive charts, [click here](#).

When an Gantt chart control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



\* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

 For more information on the required attributes for Gantt chart controls, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
      - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
        - This window consists of two tabs:
          - [Grid](#)
          - [Validation](#)



Please note that users are limited to configuring only the required field validation properties for Gantt chart controls and cannot set up required range validations.

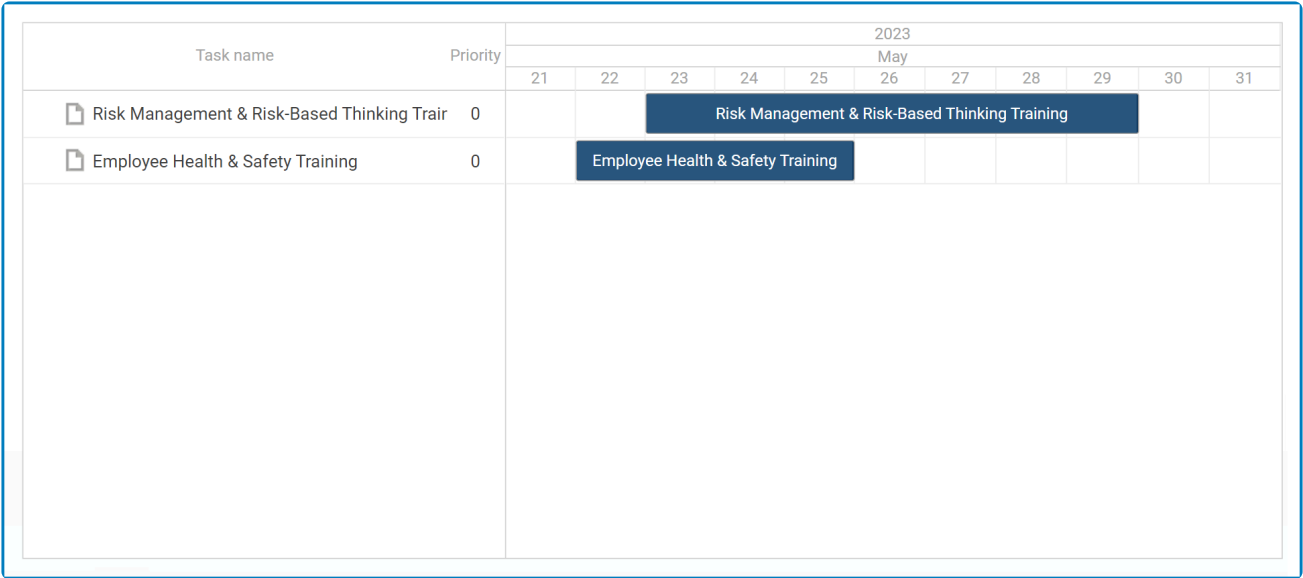
- For more information on these properties and how to configure them, [click here](#).



Please note that the linked section contains information on setting an initial value, which is not applicable to Gantt chart controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 200. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

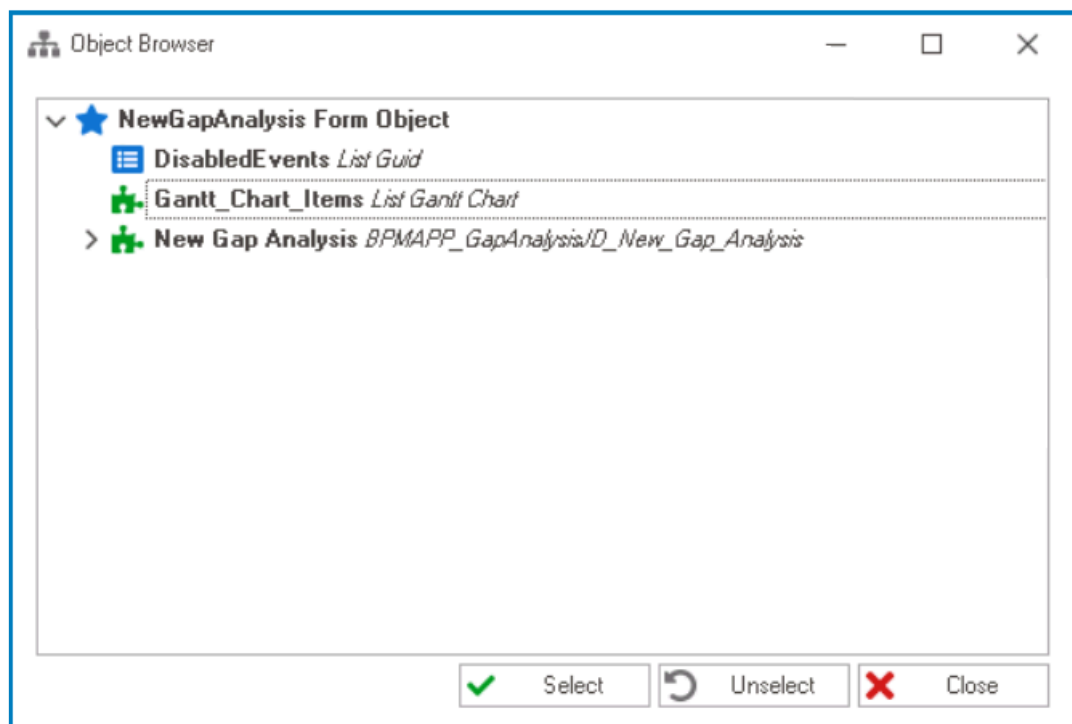
Gantt Chart Control Preview: Web View



## 12.1.2.1.1.9.1. Gantt Chart – Binding

In the **Binding** category, users can link a Gantt chart to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the Gantt chart control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the Gantt chart control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute



Users must bind the Gantt chart control to the local attribute created in the [Form Entity](#) tab. For more information on this matter, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.



- To access the control properties, click on the ellipses (...) inside the field. It will display the Control Properties window.
  - This window consists of three tabs:
    1. [General](#)
    2. [Gantt Chart](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Control Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

# 12.1.2.1.1.9.1.1. Gantt Chart – Key Components, Entity Conditions, & Binding Methods

This section of the manual outlines the key components, entity conditions, and binding methods that are integral to the functionality of the Gantt chart control.

## A. Key Components

There are several key components involved in the creation and utilization of the Gantt chart control.

These components include:

1. **Gantt Chart Entity**

This entity holds the data for the Gantt chart control. It must meet the specified requirements outlined in the **Entity Conditions** section.

✿ This entity is linked to the Gantt chart form, acting as the data source from which information is extracted to populate the control.

2. **Gantt Chart Form**

This form serves as a platform for entering data into the Gantt chart entity. It allows users to input relevant information, such as task details and scheduling parameters, which are stored in the entity for the Gantt chart control.

3. **Main Entity**

This entity is linked to the form where the Gantt chart control is being used or implemented. It represents the primary data object that the user is working with or collecting data for.

✿ In the [provided instructions](#) for building these key components, this entity is created using the [Quick Forms](#) feature. Users, however, can choose to create the main entity manually.

4. **Main Form**

This form is where the Gantt chart control is implemented. It provides the overall interface and context for incorporating the Gantt chart functionality.

- To bind the Gantt chart control to the main form, users will need to create a local attribute (as described in the **Binding Methods** section).



For more detailed instructions on how to create Gantt chart controls and their associated attributes, [click here](#).

## B. Entity Conditions

To qualify as a Gantt chart entity, three specific conditions must be met:

1. **Start Date/Time**

The entity must have a data attribute that represents the start date/time of the corresponding task or activity.

2. **End Date/Time**

The entity must have a data attribute that represents the end date/time, indicating the completion or deadline of the task.

3. **Reference to Self (ParentID)**

The entity must have a reference attribute that points to itself. This self-reference is necessary for proper hierarchical organization and visualization of tasks within the control.



Alternatively, entities can meet the criteria by having a combination of Start Date/Time and Duration or End Date/Time and Duration, in addition to the mandatory requirement of a self-reference.

## C. Binding Methods

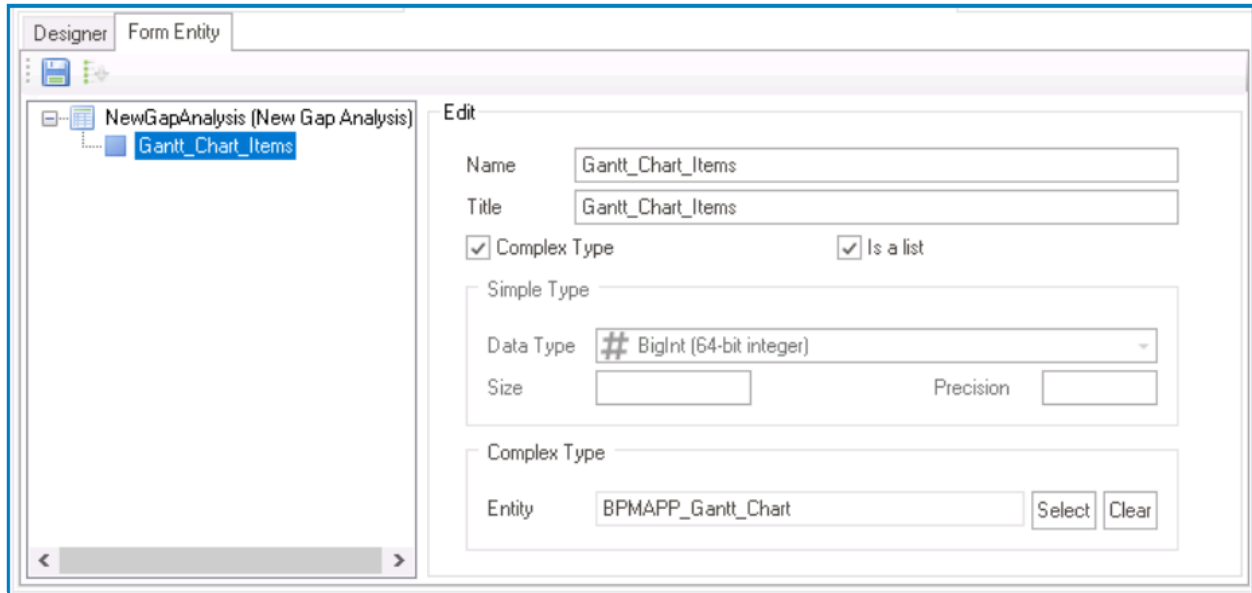
There are currently three ways by which to bind a Gantt chart control:

1. **Using List in Form Entity (Local Attribute)**

To bind the control to the form, users must create a local attribute in the [Form Entity](#) tab. This local attribute acts as a container or reference to a list of entities that meet the specified conditions outlined in the **Entity Conditions** section.



To populate this list, users can write the necessary code either in a [pre-form action](#) or within the [code of the control](#) itself, making sure to enable the **Call at Loading Page** option. In the [Control Properties](#) window, specifically in the Gantt Chart tab, it is important not to check the **Base Info** checkbox.



\* For more information on how to create a local attribute for this control, [click here](#) and navigate to the relevant section.

## 2. Using All Entities in Table

Users have the option to utilize all entities in a table by enabling the **Base Info** checkbox in the Control Properties window.

\* For more information on how to configure the properties in this window, [click here](#).

## 3. Using HQL Query

For more advanced scenarios, users can enable the **Base Info** checkbox and provide a custom HQL query statement in the Control Properties window. This allows users to define specific criteria to retrieve the desired data for the Gantt chart control.

Control Properties

Apply

OK

Cancel

Form Control

GanttChart

Entity and Attribute

Gantt\_Chart\_Items

Object type

Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_Gantt\_Chart

General

Gantt Chart

Base Info

Query (HQL only) [Empty]

Scale

BPMAPP\_Gantt\_ChartParent

Attribute

BPMAPP\_Gantt\_Ch

Condition

Task Color

Text Color

Progress Color

Add

Delete

Nested

Binding

Form

Display attribute

Start Date

End Date

Duration

Progress

Weight

© 2023 Interfacing Technologies Corporation

Page 439 of 1144

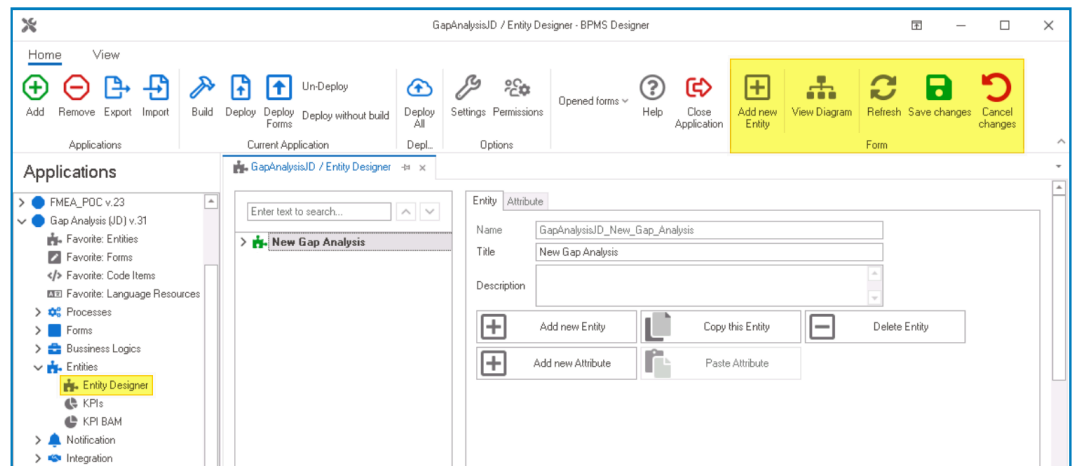
# 12.1.2.1.1.9.1.1.1. Gantt Chart – Attributes

To create Gantt chart controls and their associated attributes, follow these steps:

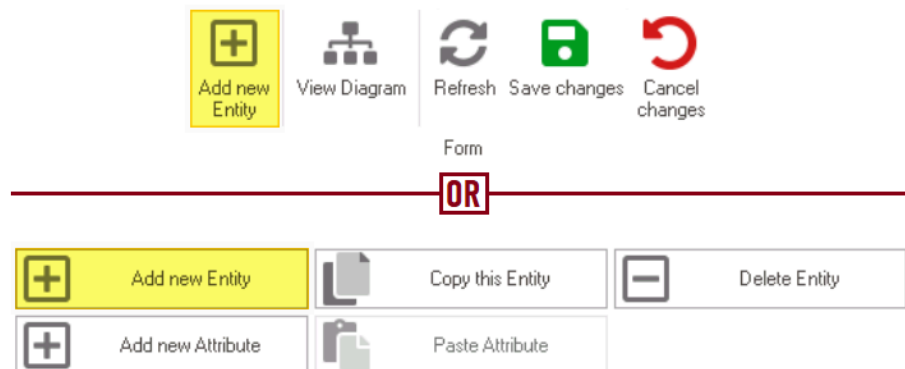
## 1. [Create Gantt Chart Entity](#)

### • Access Entity Properties

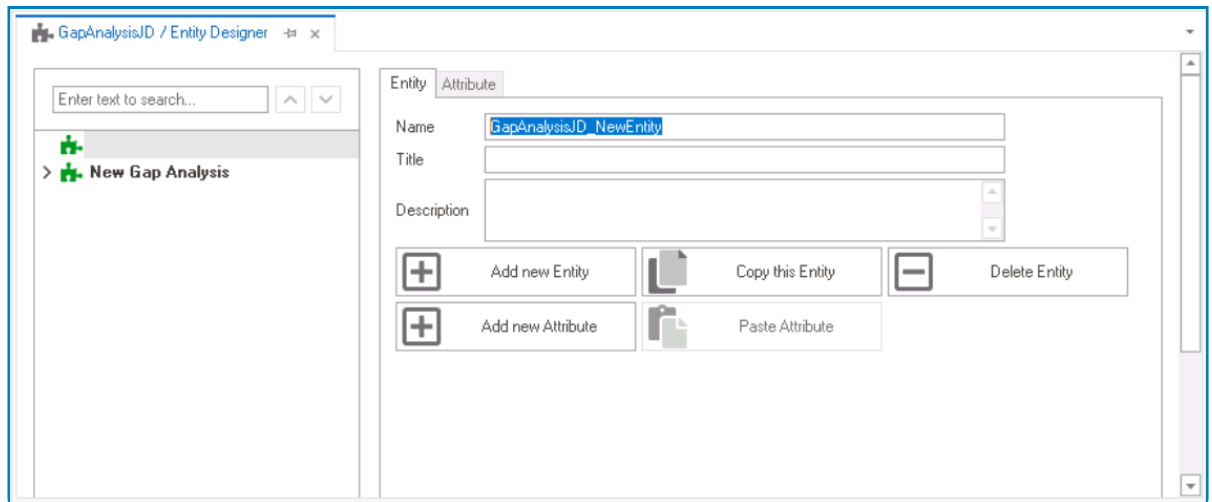
- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.



- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

 For information on the required conditions for a Gantt chart entity, [click here](#).

## 2. [Create Attributes](#)

- **Access Attribute Properties**

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.



- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' application. On the left is a sidebar with a search bar and a tree view containing 'Gantt Chart', 'NewAttribute Id / UniqueIdentifier', and 'New Gap Analysis'. The main area has two tabs: 'Entity' and 'Attribute'. The 'Attribute' tab is active, showing a form for a new attribute named 'NewAttribute'. The form includes fields for Name, Title, and Description. Below these are radio buttons for 'Data field' (selected) and 'Reference'. Under the 'Data field' section, there are tabs for 'Data field' and 'Reference'. The 'Data field' tab is active, showing a 'Data Type' dropdown set to 'UniquelIdentifier (Globally unique identifier)', and input fields for 'Size' and 'Precision'. Below this is a section 'Show as a column in the task list' with radio buttons for 'Hide', 'Hidden by default', and 'Show by default' (selected). At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', and three buttons: 'Add new Attribute', 'Copy Attribute', and 'Delete Attribute'.

✿ For Gantt chart controls, users must create two data attributes (start date/time and end date/time) and one reference attribute (to itself—parent).

- **Create Data & Reference Attributes**

1. **Start Date/Time:** This attribute is used to capture the start date/time of the task or activity.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Date Field** option.
  - From the **Data Type** dropdown, select **DateTime (Date with Time)**.




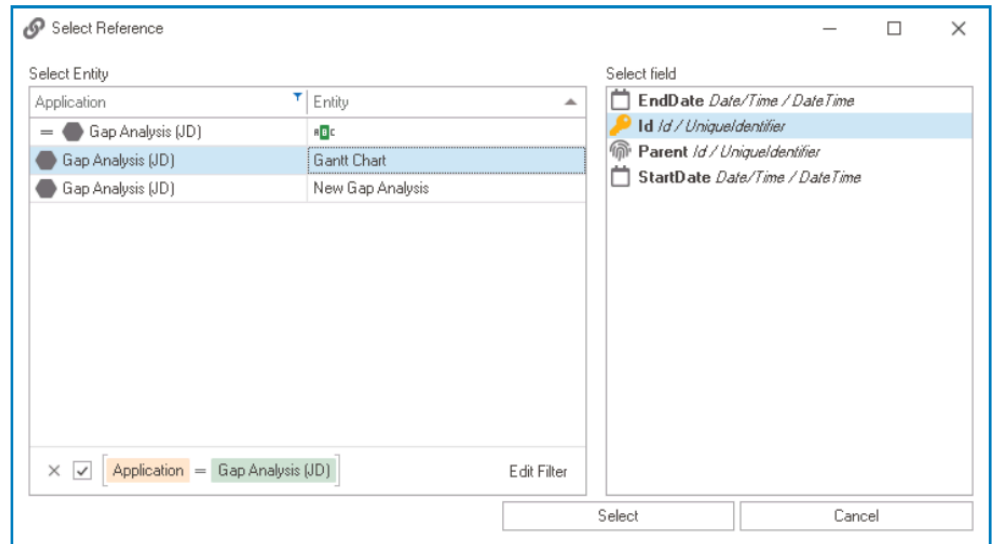
The screenshot shows the 'Attribute' configuration window for an attribute named 'StartDate'. The 'Name' field contains 'StartDate' and the 'Title' field contains 'Start Date'. The 'Description' field is empty. The 'Data field' radio button is selected. The 'Data Type' dropdown is set to 'DateTime (Date with time)'. The 'Size' and 'Precision' fields are empty. The 'Show as a column in the task list' section has the 'Hide' radio button selected. The 'Read Only', 'Defines field', and 'Track changes' checkboxes are all unchecked. At the bottom, there are three buttons: 'Add new Attribute' (with a plus icon), 'Copy Attribute' (with a copy icon), and 'Delete Attribute' (with a minus icon).

2. **End Date/Time:** This attribute is used to capture the end date/time of the task or activity.

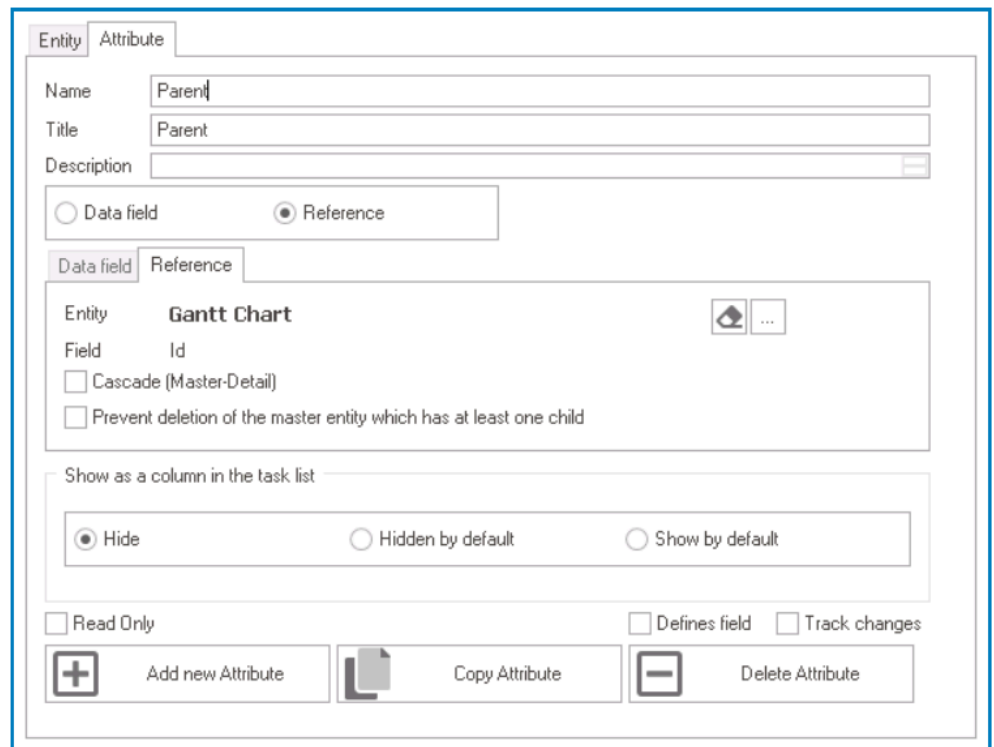
- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Data Field** option.
- From the **Data Type** dropdown, select **DateTime (Date with Time)**.

The screenshot shows the 'Attribute' configuration window for an attribute named 'EndDate'. The 'Name' field contains 'EndDate' and the 'Title' field contains 'End Date'. The 'Description' field is empty. The 'Data field' radio button is selected. The 'Data Type' dropdown is set to 'DateTime (Date with time)'. The 'Size' and 'Precision' fields are empty. The 'Show as a column in the task list' section has the 'Hide' radio button selected. The 'Read Only', 'Defines field', and 'Track changes' checkboxes are all unchecked. At the bottom, there are three buttons: 'Add new Attribute' (with a plus icon), 'Copy Attribute' (with a copy icon), and 'Delete Attribute' (with a minus icon).

3. **Parent:** This attribute is used to reference the Gantt chart entity.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [GanttChartEntity]**. From the field section, select **ID**.

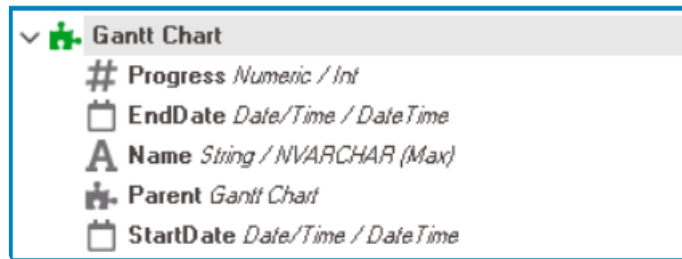


- Selecting the reference will redirect to the Attribute tab, where the entity section will display **[GanttChartEntity]** and the field section will display **ID**.



- Once users have created the necessary attributes for the Gantt chart control, the end-result

should be along the lines of the following list:

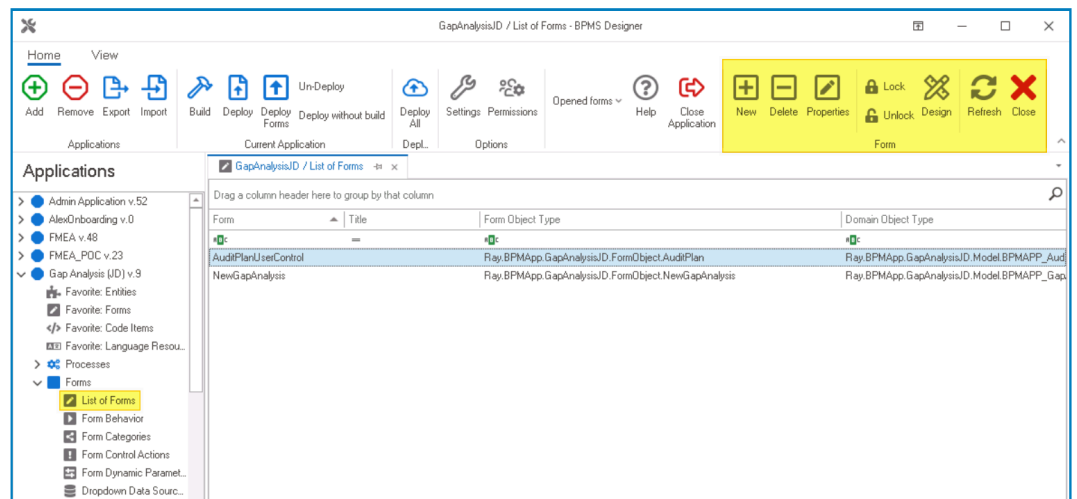


! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

### 3. [Create & Design Gantt Chart Form](#)

- **Access Form Properties**

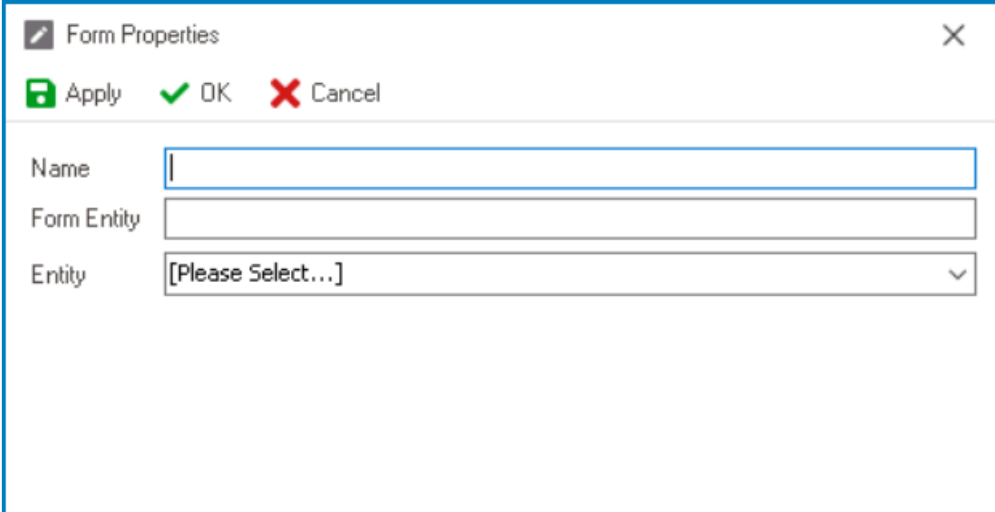
- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the [Form Properties](#) window.



- In the [Form Properties](#) window, users can create a new form for the application.



The **Form Properties** dialog box contains the following fields and controls:

- Name:** A text input field for naming the form.
- Form Entity:** A text input field for specifying the form entity.
- Entity:** A dropdown menu currently showing "[Please Select...]" with a downward arrow.
- Buttons:** "Apply" (with a floppy disk icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon).

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the Gantt chart entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

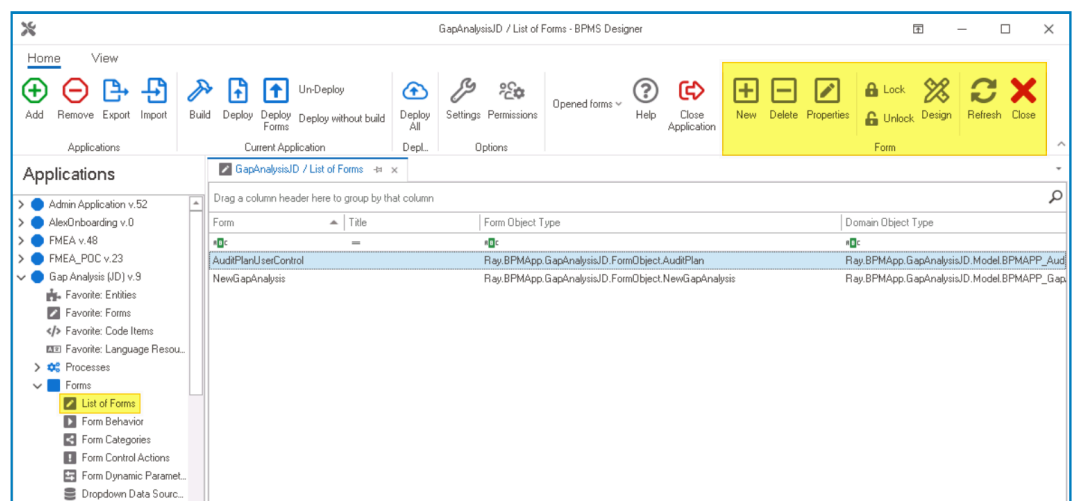


The purpose of this form is to allow data entry into the entity that holds data for the Gantt chart. Users simply need to add the required controls and link them to their corresponding attributes.

#### 4. [Create & Design Main Form](#)

- **Access Form Properties**

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the [Form Properties](#) window.



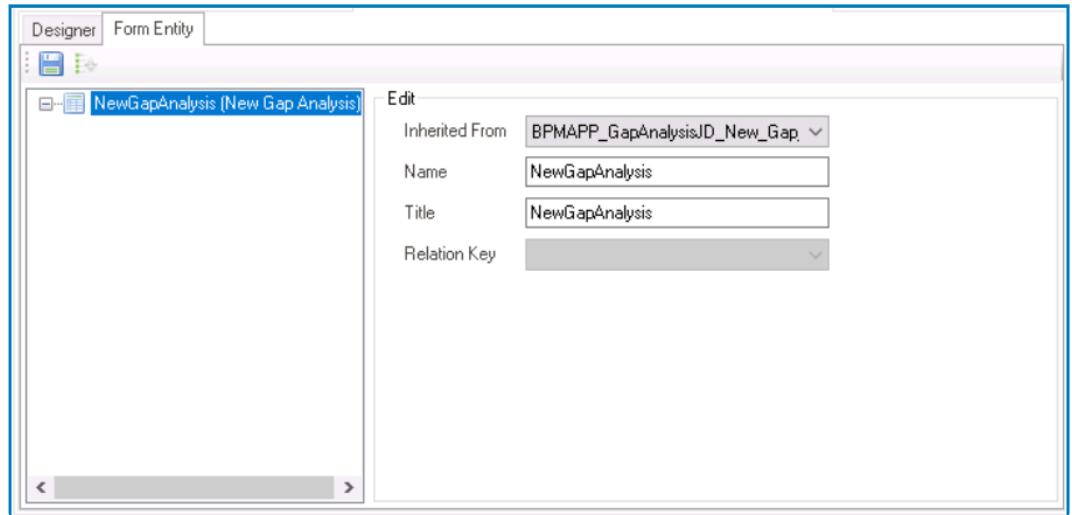
- In the Form Properties window, users can create a new form for the application.

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- Leave the **Entity** field empty.
- Click **OK** and confirm the prompt to start designing the form.

✿ The main form is where the Gantt chart control is used or implemented. After creating the main form and launching the Form Designer, users should simply drag the Gantt chart control from the toolbox and drop it onto the design environment.

## 5. Create Local Attribute

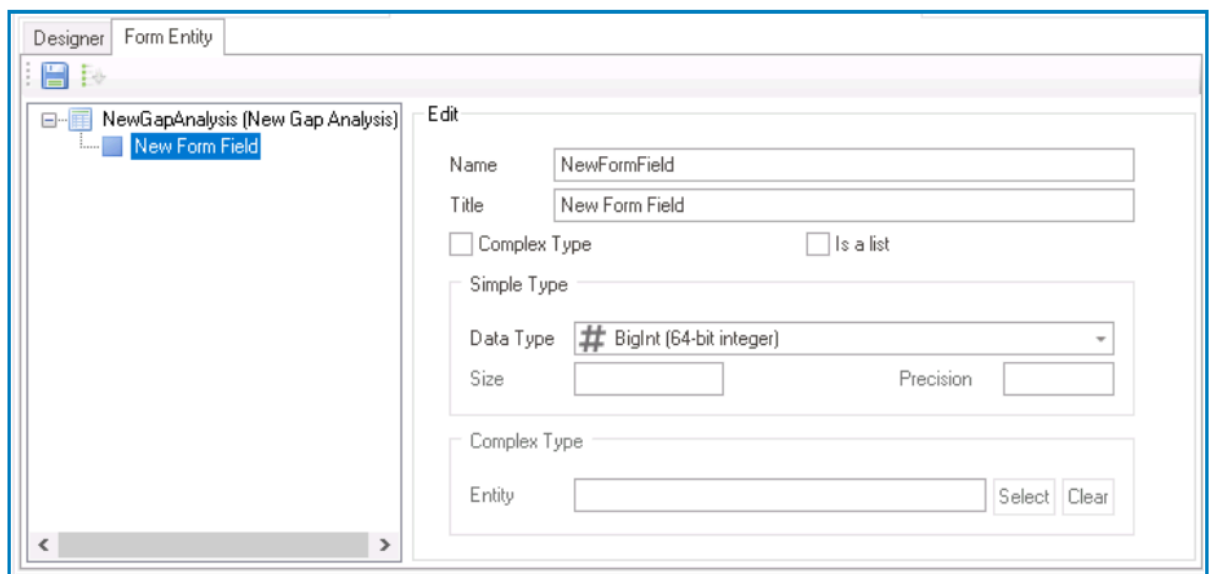
- **Access Attribute Properties**
  - From the Form Designer, navigate to the Form Entity tab.



- Right-click on the form entity and select **New Field** from the context menu. It will generate a new field below the form entity, which can then be customized and defined to suit the needs of the form.



- In the Edit Properties pane, users can define the properties of the newly created attribute.



- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Select the **Complex Type** checkbox (because the control will reference data from another form).
- Select the **Is a List** checkbox (because the control will hold multiple instances of the attribute's value).

- In the **Complex Type** section, select the Gantt chart entity from which the attribute will reference data.

| Application         | Entity                                  |
|---------------------|---|
| = Gap Analysis (JD) | #c                                      |
| Gap Analysis (JD)   | Actions                                 |
| Gap Analysis (JD)   | Comment Entity                          |
| Gap Analysis (JD)   | Comment_NewGapAnalysis                  |
| Gap Analysis (JD)   | Data Analysis Methods                   |
| Gap Analysis (JD)   | Data Analysis Methods Mid               |
| Gap Analysis (JD)   | DataAnalysisMethods_Items               |
| Gap Analysis (JD)   | Gantt Chart                             |
| Gap Analysis (JD)   | GapAnalysisJDDataAnalysisMethods_ListDS |
| Gap Analysis (JD)   | GapAnalysisJDGapAnalysisListDS          |
| Gap Analysis (JD)   | GapAnalysisJDNewGapAnalysisDS           |
| Gap Analysis (JD)   | NewGapAnalysis                          |
| Gap Analysis (JD)   | SampleForm                              |

Application = Gap Analysis (JD) Edit Filter

Select Cancel

- Click on the  button to save your changes.

Designer Form Entity

NewGapAnalysis (New Gap Analysis)

Gantt\_Chart\_Items

Edit

Name: Gantt\_Chart\_Items

Title: Gantt\_Chart\_Items

☒ Complex Type ☒ Is a list

Simple Type

Data Type: # BigInt (64-bit integer)

Size: Precision:

Complex Type

Entity: BPMAPP\_Gantt\_Chart Select Clear

## 6. Configure Gantt Chart Control Binding Properties

- After creating the appropriate attribute, users can return to the Designer tab and define the binding properties for the Gantt chart control.



Once the control properties are configured, users must save the form. This will activate the Quick Forms feature, prompting users to confirm the generation of both the main

entity and the form behavior for the main form.



## 12.1.2.1.1.9.1.2. Control Properties – Gantt Chart Tab

In the Gantt Chart tab of the Control Properties window, users can build and customize the Gantt chart.

✿ In the General tab of this window, users can configure business rules for the Gantt chart control. For more information on this matter, please [click here](#) and navigate to the relevant section.

1. **Base Info:** This option allows users to populate the control with data from a specific table.
  - If this option is selected, it will enable the **Query (HQL Only)** field, allowing users to select or define a custom query for the Gantt chart data source.

! Users can also choose to populate the control by writing the necessary code either in a [pre-form action](#) or within the [code of the control](#) itself. To do so, users should select the **Call at Loading Page** option in the [General](#) tab and ensure that the **Base Info** checkbox in this tab is not selected.

2. **Scale:** This is a single-select dropdown field that allows users to choose the time scale for the control.
  - Time Scale Options: Hours, Days, Months, or Years.

◦ Time Scale Preview: Web View

3. **Attribute:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute to which the condition will be applied.
4. **Condition:** This is a text field that allows users to specify a condition that needs to be met for chart elements to be colored.

• Gantt Chart Conditions

For example, users can apply the following conditions to the **Progress** attribute:

- If value > 50, display the text color in white and the task color and progress color in different shades of green.

ScaleDays

Progress: value > [50]

AttributeProgress

Conditionvalue > [50]

Task ColorSeaGreen

Text Color255, 255, 255

Progress ColorDarkSeaGreen

+

Add

-

Remove

- If value < 50, display the text color in white and the task color and progress color in different shades of red.
- If value = 50, display the text color in white and the task color and progress color in different shades of orange.

When all three conditions are applied, the tasks in the chart will appear like this:

| Task name                                   | Priority | 2023 |    |    |    |    |    |    |    |    |    |    |
|---|----------|------|----|----|----|----|----|----|----|----|----|----|
|   |          | May  |    |    |    |    |    |    |    |    |    |    |
|   |          | 21   | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Risk Management & Risk-Based Thinking Trair | 0        |      |    |    |    |    |    |    |    |    |    |    |
| Employee Health & Safety Training           | 0        |      |    |    |    |    |    |    |    |    |    |    |
| Document Control Training                   | 0        |      |    |    |    |    |    |    |    |    |    |    |

! This condition (and its corresponding color styles) will apply only to the selected attribute.

5. **Task Color:** This is a color picker that allows users to choose the color that will be applied to the tasks displayed in the Gantt chart.
  - It contains three tabs: Custom, Web, and System.
    - In the Custom tab, users can choose a color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
    - In the Web tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
    - In the System tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
6. **Text Color:** This is a color picker that allows users to choose the color that will be applied to the text displayed in the Gantt chart.
  - It also has three tabs, similar to the **Task Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.
7. **Progress Color:** This is a color picker that allows users to choose the color that will be used to indicate progress in the Gantt chart.
  - It also has three tabs, similar to the **Task Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.



If the condition is not met, the specified task color, text color, and progress color will not be applied.

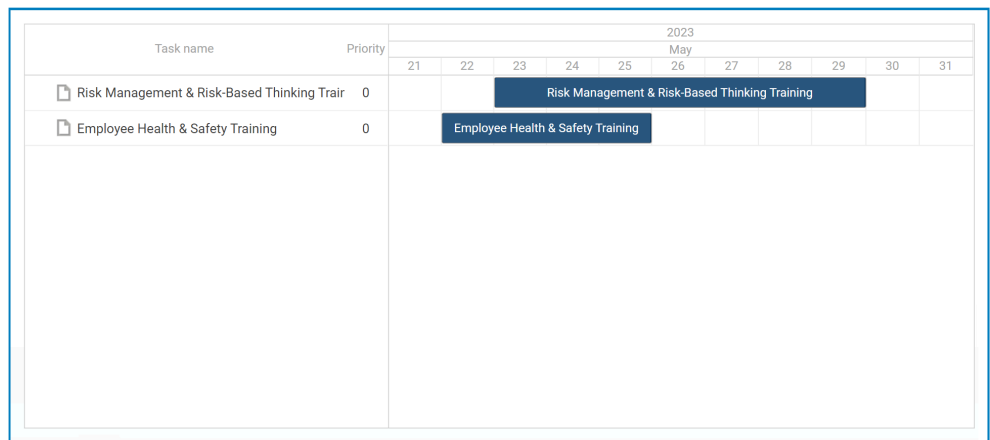
8. **Add:** This button allows users to add a new condition for the control.
  - If selected, it will add a new entry where users can define a new condition and specify the corresponding color styles to be applied.

9. **Remove:** This button allows users to remove selected conditions from the control.
10. **Nested:** In this section, users can select the nested attributes within the Gantt chart.
  - **Binding:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute that will be used for binding or establishing the hierarchy within the Gantt chart.
  - **Form:** This is a single-select dropdown field that retrieves the list of forms linked to the associated entity. It allows users to select the form from which data will be pulled to build the Gantt chart.
  - **Display Attribute:** This is a single-select dropdown field that retrieves the list of attributes

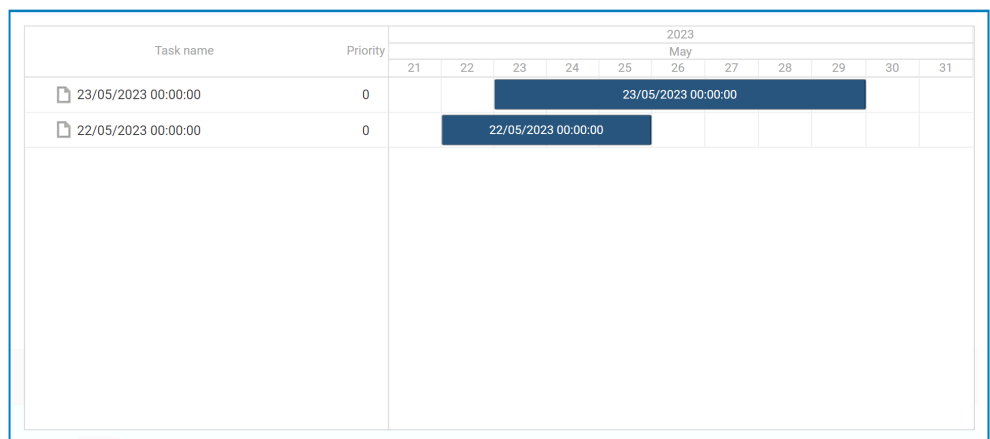
available in the associated entity. It allows users to select the attribute whose value will be displayed as text on individual items within the Gantt chart.

- **Display Attribute Preview: Web View**

- For example, if the selected attribute is **Name**, the control will update the display of each task item to showcase its corresponding name.



- Similarly, if the selected attribute is **Start**, the control will update the display of each task item to showcase its corresponding start date/time.



- **Start Date:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the start date for the tasks in the Gantt chart.
- **End Date:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the end date for the tasks in the Gantt chart.
- **Duration:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the duration of the tasks in the Gantt chart.
- **Progress:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the progress status of the tasks in the Gantt chart.
- **Weight:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the weight or priority of the tasks in the Gantt chart.

## Example: Nested Attributes Selection

The screenshot shows the 'Control Properties' dialog box for a 'GanttChart' control. The 'Gantt Chart' tab is selected, displaying configuration options for the 'BPMAPP\_Gantt\_ChartParent' entity. The 'Base Info' section is checked, and the 'Query (HQL only)' is set to 'QS\_Gantt\_ListGrid'. The 'Scale' is set to 'Days'. The 'Nested' section shows a list of attributes: Binding, Form, Display attribute, Start Date, End Date, Duration, Progress, and Weight. The 'Binding' attribute is set to 'BPMAPP\_Gantt\_Chartsf', 'Form' is 'Gantt\_Chart\_Form', 'Display attribute' is 'Name', 'Start Date' is 'Start', 'End Date' is 'End', 'Duration' is empty, 'Progress' is 'Progress', and 'Weight' is empty. The 'Add' and 'Delete' buttons are visible below the attribute list.

\* Please note that in this context, the term “relevant entity” or “associated entity” refers to the entity specifically created for the Gantt chart control (described [here](#)).

11. **Apply:** This button allows users to save their changes.
12. **OK:** This button allows users to save their changes and close the window.
13. **Cancel:** This button allows users to cancel their changes and close the window.




## 12.1.2.1.1.9.2. Gantt Chart – Control Governance

In the Grid tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of grid controls.

1. **Enable Edit:** This is a checkbox which, if selected, allows users to enable edit mode for the control.
  - If this checkbox is selected, it will enable the **Edit Form** field.
    - **Edit Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'edit' functionality.
      - To define a new form behavior, select the button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the button.
      - To remove a form behavior, select the button.

! To set up the 'Edit Form', users only need to provide a name and a title.

- \* With **Edit** enabled, users can edit the chart data in a modal window linked to the control. Users can access the 'Edit Form' by double-clicking on an item or task in the control.

2. **Enable Import from Excel:** This is a checkbox which, if selected, allows users to enable both the **Excel Template** and **Import from Excel** functions for the control.
  - If this checkbox is selected, it will enable the **Import Form** field.
    - **Import Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'inline add' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.
      - To remove a form behavior, select the  button.

! To set up the 'Import Form', users only need to provide a name, a title, and select the **Starter** checkbox.

- \* When this option is enabled, users can download an Excel template that contains the appropriate column headers for the control, and then populate the template with their data. Users can then upload the completed template into the control to import the data.

3. **Enable Export from Excel:** This is a checkbox which, if selected, allows users to enable the **Export to Excel** function for the control.

- \* When this option is enabled, users can export the data from the control into an Excel file.

4. **Enable Select:** This is a checkbox which, if selected, allows users to enable the selection of chart data from another form using a dropdown control.

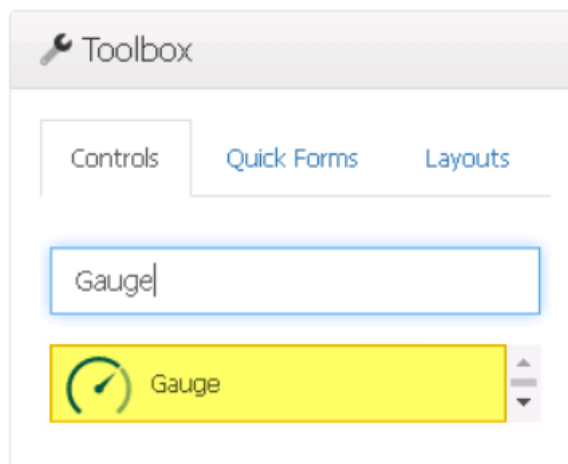
- \* When this option is enabled, users can interact with a Gantt chart displayed in a separate form and select items by double-clicking on them. The selected value will then be returned and populated in the associated dropdown control.

5. **Apply:** This button allows users to save their changes.
6. **OK:** This button allows users to save their changes and close the window.
7. **Cancel:** This button allows users to cancel their changes and close the window.

- \* In the Validation tab of this window, users can configure the required field validation properties for the Gantt chart control. For more information on this matter, please [click here](#) and navigate to the relevant section.

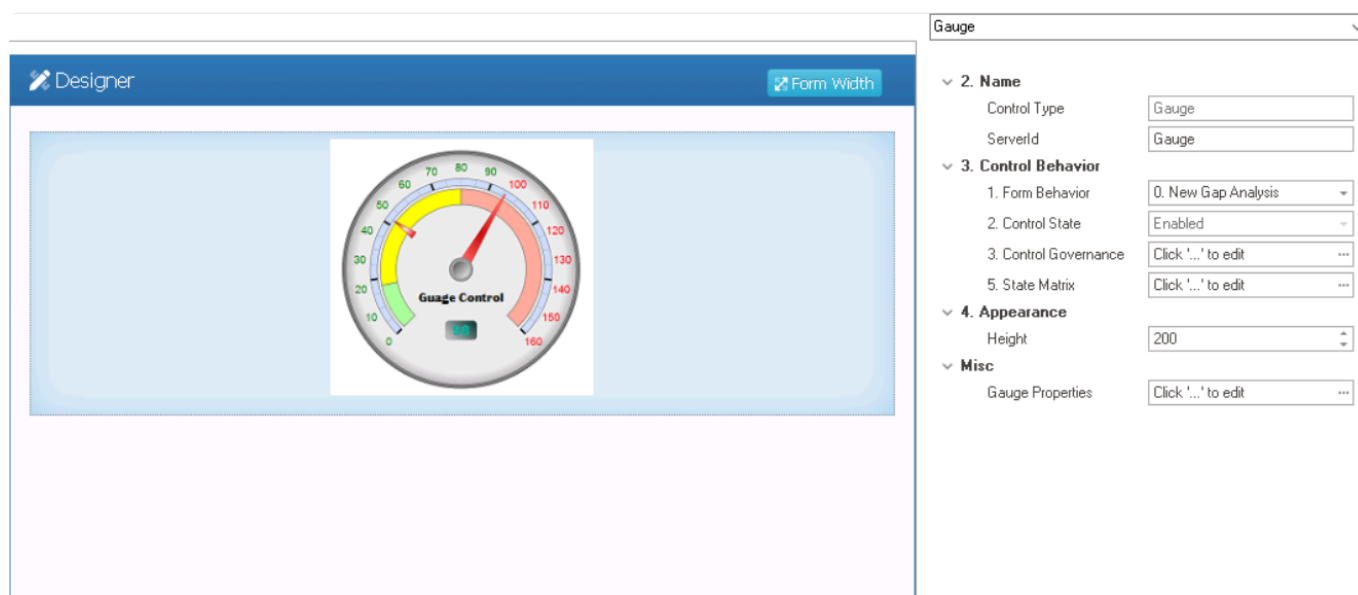
## 12.1.2.1.10. Gauge

The gauge control is a graphical component that allows users to display and track values within a certain range. It provides a visual representation of a specific parameter or measurement, enabling users to quickly assess and understand the presented data.



✿ This control presents information in the form of a graphical gauge or meter, resembling instruments like a speedometer or fuel gauge.

When a gauge control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.



- **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
2. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
- **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state properties set for the control will apply only to the selected form behavior.



Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **Control Governance:** This field allows users to configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Gauge]

Apply OK Cancel




Control Name **Chart**

Chart

☐ Enable View View Form Behaviour [Empty]

- **Enable View:** This is a checkbox which, if selected, allows users to enable view

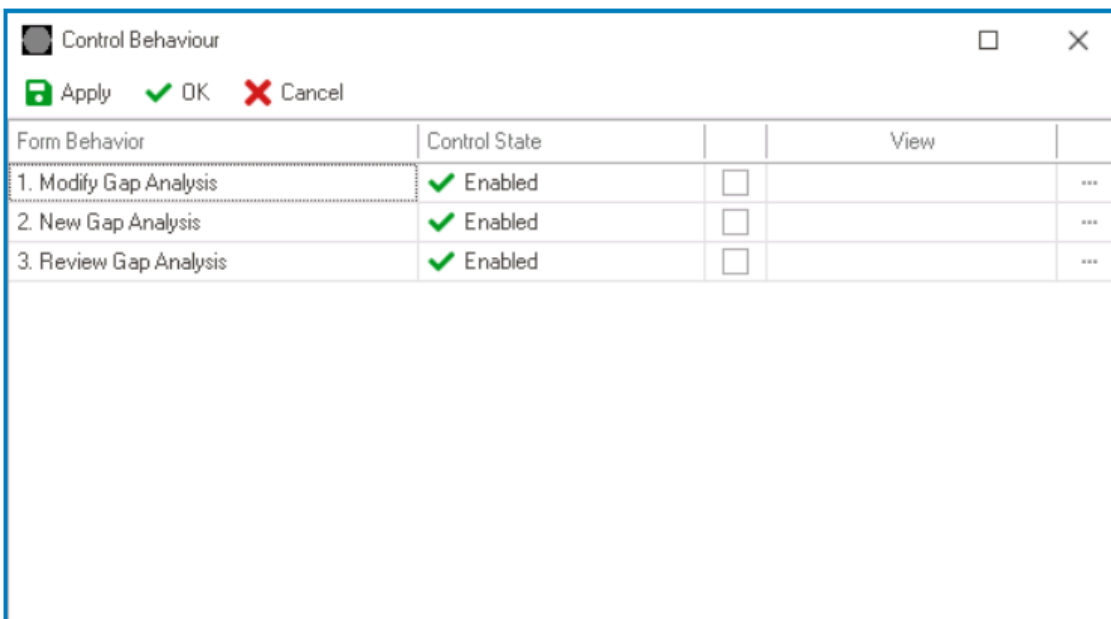
mode for the control.

- If this checkbox is selected, it will enable the **View Form Behavior** field.
- **View Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
  - To define a new form behavior, select the  button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing form behavior, select the  button.
  - To remove a form behavior, select the  button.



With **View** enabled, users can view the relevant data in a modal window linked to the control. Users can access the 'View Form' by double-clicking on an element in the control.

- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.



| Form Behavior          | Control State |                                     | View |
|------------------------|---------------|-------------------------------------|------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input checked="" type="checkbox"/> | ...  |
| 2. New Gap Analysis    | ✓ Enabled     | <input checked="" type="checkbox"/> | ...  |
| 3. Review Gap Analysis | ✓ Enabled     | <input checked="" type="checkbox"/> | ...  |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Control Governance:** This field allows users to configure the control governance properties.
  - If selected, it will display the [Form / Behavior / Control](#) window (described above).
- **Apply:** This button allows users to save their changes.

- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.

✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

! In this window, users can also configure the governance properties for the control across multiple form behaviors. To gain an understanding of how these properties can be configured, [click here](#).

3. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 200. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
4. **Misc:** In this category, users can define additional properties for the control.
  - **Gauge Properties:** This field allows users to configure the gauge properties.
    - To access the gauge properties, click on the ellipses (...) inside the field.
      - It will display the [Gauge Edit Form](#) window.

! Users do not need to create or link an attribute for gauge controls.

#### Gauge Control Preview: Web View



## 12.1.2.1.1.10.1. Gauge Edit Form

---

In the Gauge Edit Form window, users can configure the properties of the gauge control.

These properties are displayed in a set of tabs:

1. [Appearance](#)
2. [Label](#)
3. [Scale](#)
4. [Needle](#)
5. [Cap](#)
6. [Data Source](#)
7. [Data Binding](#)



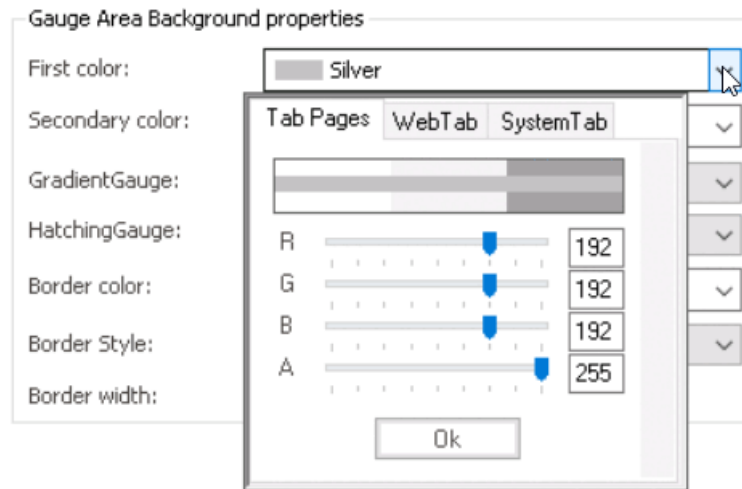
To save changes and exit the Gauge Edit Form window, click on the **OK** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 12.1.2.1.10.1.1. Appearance

In the Appearance tab, users can customize the visual appearance of the gauge control.

\* All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. For more information on these tabs, please refer to the description below for the **First Color** field.

1. **Gauge Area Background Properties:** In this section, users can configure the visual properties related to the gauge area background.
  - **First Color:** This is a color picker that allows users to choose the first color in the gradient that defines the background of the gauge area.
    - It contains three tabs: Tab Pages, Web, and System.

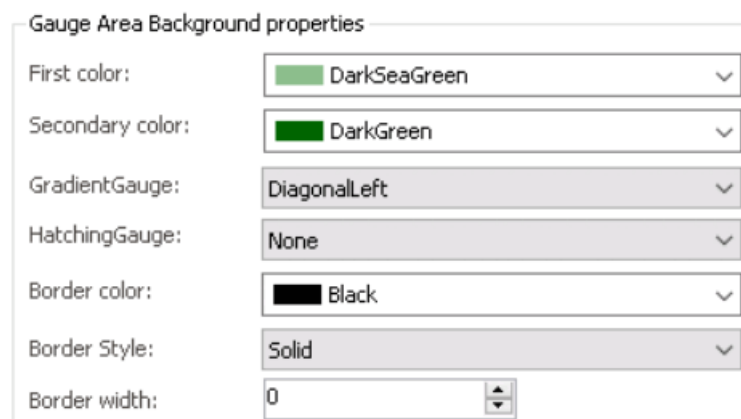


- In the Tab Pages tab, users can select a gradient color by manipulating sliders or input fields for each color channel in the RGBA color picker.



With an RGBA color picker, users can specify not only the red, green, and blue (RGB) values but also the alpha channel value (A), which controls the transparency or opacity of the color.

- In the Web tab, users can select a gradient color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
- In the System tab, users can select a gradient color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
- **Second Color:** This is a color picker that allows users to choose the second color in the gradient that defines the background of the gauge area.
  - **Gauge Gradient Colors Preview: Web View**



- **Gradient Gauge:** This field allows users to choose the gradient style of the gauge area background from a dropdown menu.

- **Gauge Gradient Style Preview: Web View**

Gauge Area Background properties

|                  |  |    |
|------------------|--|----|
| First color:     | <span style="display: inline-block; width: 20px; height: 10px; background-color: #2F8B42; border: 1px solid black;"></span> DarkSeaGreen | ▼  |
| Secondary color: | <span style="display: inline-block; width: 20px; height: 10px; background-color: #008000; border: 1px solid black;"></span> DarkGreen    | ▼  |
| GradientGauge:   | TopBottom  | ▼  |
| HatchingGauge:   | None   | ▼  |
| Border color:    | <span style="display: inline-block; width: 20px; height: 10px; background-color: black; border: 1px solid black;"></span> Black          | ▼  |
| Border Style:    | Solid  | ▼  |
| Border width:    | 0  | ▲▼ |

- **Hatching Gauge:** This field allows users to choose a specific hatching pattern for the gauge area background from a dropdown menu.

- **Gauge Hatching Pattern Preview: Web View**

Gauge Area Background properties

|                  |  |    |
|------------------|--|----|
| First color:     | <span style="display: inline-block; width: 20px; height: 10px; background-color: #C0C0C0; border: 1px solid black;"></span> Silver | ▼  |
| Secondary color: | <span style="display: inline-block; width: 20px; height: 10px; background-color: #808080; border: 1px solid black;"></span> Gray   | ▼  |
| GradientGauge:   | DiagonalLeft   | ▼  |
| HatchingGauge:   | LargeCheckerBoard  | ▼  |
| Border color:    | <span style="display: inline-block; width: 20px; height: 10px; background-color: black; border: 1px solid black;"></span> Black    | ▼  |
| Border Style:    | Solid  | ▼  |
| Border width:    | 0  | ▲▼ |

- **Border Color:** This is a color picker that allows users to choose the border color of the gauge control.

- **Gauge Border Color Preview: Web View**

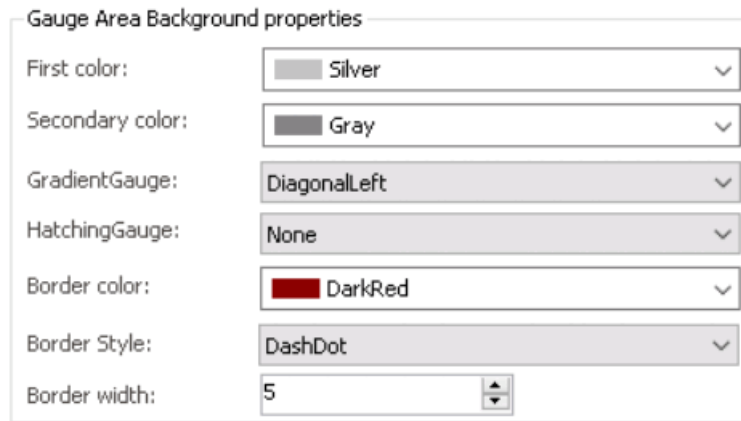
- **Border Style:** This field allows users to choose a border style for the control from a dropdown menu.

- **Gauge Border Style Preview: Web View**

- **Border Width:** This is a numeric updown field that allows users to set the width of the control's border.

- Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.

- **Gauge Border Width Preview: Web View**



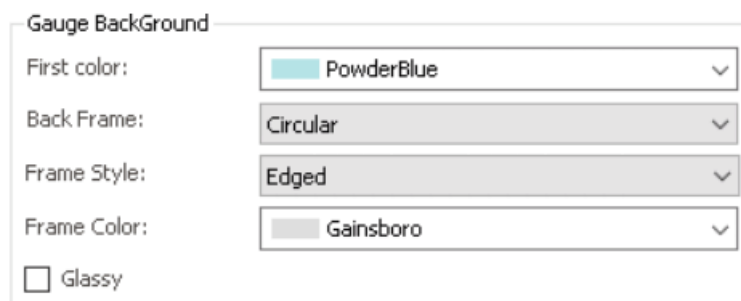
Gauge Area Background properties

|                  |   |
|------------------|---|
| First color:     | <span style="background-color: silver; border: 1px solid black; padding: 2px;">Silver</span> ▼        |
| Secondary color: | <span style="background-color: gray; border: 1px solid black; padding: 2px;">Gray</span> ▼            |
| GradientGauge:   | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">DiagonalLeft</span> ▼ |
| HatchingGauge:   | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">None</span> ▼         |
| Border color:    | <span style="background-color: darkred; border: 1px solid black; padding: 2px;">DarkRed</span> ▼      |
| Border Style:    | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">DashDot</span> ▼      |
| Border width:    | <input type="text" value="5"/> ▼  |

2. **Gauge Background:** In this section, users can customize the visual properties related to the gauge background.

- **First Color:** This is a color picker that allows users to choose the color of the gauge background.

- **Gauge Background Color Preview: Web View**



Gauge BackGround

|                                 |  |
|---------------------------------|--|
| First color:                    | <span style="background-color: powderblue; border: 1px solid black; padding: 2px;">PowderBlue</span> ▼ |
| Back Frame:                     | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">Circular</span> ▼      |
| Frame Style:                    | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">Edged</span> ▼         |
| Frame Color:                    | <span style="background-color: #d3d3d3; border: 1px solid black; padding: 2px;">Gainsboro</span> ▼     |
| <input type="checkbox"/> Glassy |  |

- **Back Frame:** This field allows users to choose the shape of the control's background frame from a dropdown menu.

- **Gauge Background Frame Preview: Web View**

- **Frame Style:** This field allows users to choose the style of the control's background frame from a dropdown menu.

- **Gauge Background Frame Style Preview: Web View**

- **Edged vs. Simple vs. None**

- **Frame Color:** This is a color picker that allows users to choose the color of the control's background frame.

- **Gauge Background Frame Color Preview: Web View**

- **Glassy:** This is a checkbox which, if selected, allows users to apply a glassy effect to the gauge background, adding a glossy or reflective texture to the overall appearance.



- **Gauge Glassy Effect Preview: Web View**



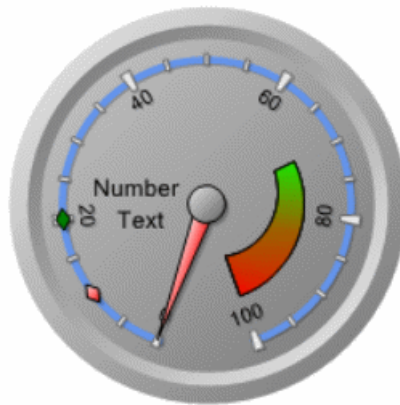
## 12.1.2.1.10.1.2. Label

In the Label tab, users can customize various properties related to the label of the gauge control.

\* All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. For more information on these tabs, please refer to the [description](#) for the **First Color** field.

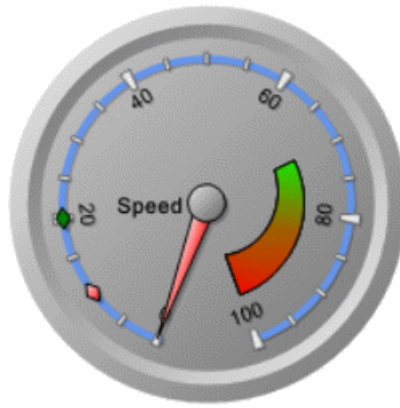
1. **Label Name:** This field allows users to choose the specific label they want to configure from a dropdown menu.



- Labels: Label Number or Label Text.
  - **Gauge Labels Preview: Web View**

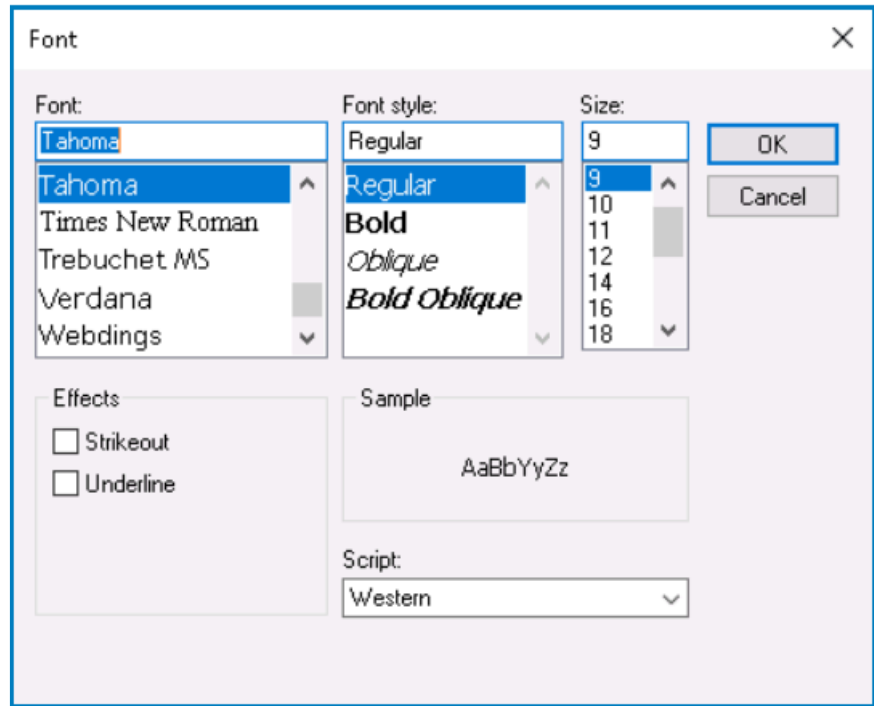


2. **General:** In this tab, users can configure general properties related to the selected label.
  - **Enable:** This is a checkbox which, if selected, allows users to enable the display of the label associated with the gauge control.
  - **X Position:** This is a numeric updown field that allows users to specify the horizontal position of the label.
    - Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.
  - **Y Position:** This is a numeric updown field that allows users to specify the vertical position of the label.
  - **Box Width:** This is a numeric updown field that allows users to set the width of the label box. The label box refers to the rectangular area that contains the label's content.
  - **Box Height:** This is a numeric updown field that allows users to specify the height of the label box.

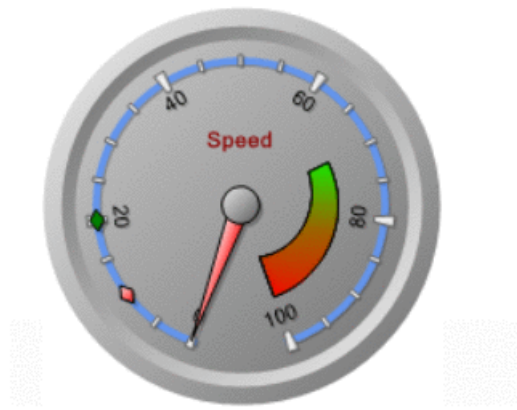
## Gauge Label Position Customization: Web View



3. **Appearance:** In this tab, users can customize the visual appearance of the selected label.
- **Back Color:** This is a color picker that allows users to choose the background color of the label.
    - This color defines the primary or fill color for the label's background.
  - **Secondary Color:** This is a color picker that allows users to choose the secondary color of the label.
    - This color is used for gradient effects or complementary visual elements within the label.
  - **Border Color:** This is a color picker that allows users to choose the color of the label's border.
  - **Border Width:** This is a numeric updown field that allows users to set the width of the label's border.
  - **Font:** This field allows users to configure the font properties for the text displayed in the label.
    - To configure the font properties for the text, click on the  button.
      - **Configuring Font Properties**  
If the  button is selected, the Font window will appear.



- **Font:** This field allows users to choose the font family from a scrollable list of options.
- **Font Style:** This field allows users to choose the font style from a scrollable list of options.
  - Depending on the font family selected, the options displayed in this list may vary.
- **Size:** This field allows users to choose the font size from a scrollable list of options.
  - Users can also simply type in a specific font size.
- **Effects:** In this section, users can modify the appearance of the text by applying common effects, such as strikeout and underline.
  - **Strikeout:** This is a checkbox which, if selected, allows users to apply a strikeout effect to the text.
  - **Underline:** This is a checkbox which, if selected, allows users to apply an underline effect to the text.
- **Sample:** This field displays a preview of how the text will look based on the chosen font family, style, size, and effect. It updates in real-time as the user makes changes to the font settings.
- **Script:** This field allows users to choose a specific script or writing system for the text from a dropdown menu.
  - By selecting the appropriate script, users can ensure that their text is correctly displayed and formatted for the desired language or writing system.
- **OK or Cancel:** To save your changes and close the window, click on the **OK** button. To cancel your changes and close the window, click on the **Cancel** button.
- **Font Color:** This is a color picker that allows users to choose the font color for the text displayed in the label.

**Gauge Label Appearance Customization: Web View**

## 12.1.2.1.10.1.3. Scale (Base)

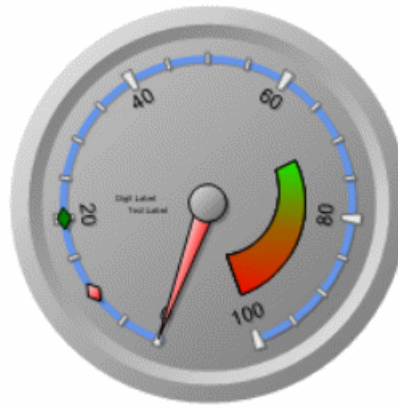
In the Scale tab, users can customize various properties related to the scale of the gauge control.

The screenshot shows the 'Gauge Edit Form' window with the 'Scale' tab selected. The 'Scale Name' dropdown is set to 'Base Scale' and the 'Enable' checkbox is checked. The 'Back color' is 'CornflowerBlue', 'Back color 2' is 'White', and 'Border color' is 'Black'. The 'Border width' is 0, 'Distance' is 37, and 'Start width' is 5. The 'Font' is 'Microsoft Sans Serif' and 'Fore color' is 'Black'. The 'Interval' is 10, 'StartAngle' is 20, and 'SweepAngle' is 320. The 'Cancel' and 'Ok' buttons are at the bottom right.

\* All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. For more information on these tabs, please refer to the [description](#) for the **First Color** field.

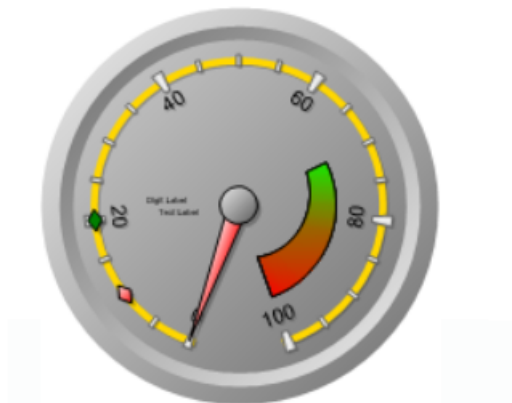
1. **Scale Name:** This field allows users to select the specific scale they want to configure from a dropdown menu.
  - Scales: Base Scale, Min Scale, Middle Scale, or Max Scale.

◦ **Gauge Scales Preview: Web View**



! Depending on the selected scale, the available fields in this window may vary. In this section of the manual, we will be focusing on the fields specifically applicable to the base scale. For information on the fields relevant to the minimum, middle, or maximum scales, [click here](#).

2. **Enable:** This is a checkbox which, if selected, allows users to enable the display of the selected scale.
3. **Back Color:** This is a color picker that allows users to choose the background color of the scale.
  - This color defines the fill color for the scale's background.
  - **Gauge Scale Background Color Preview: Web View**

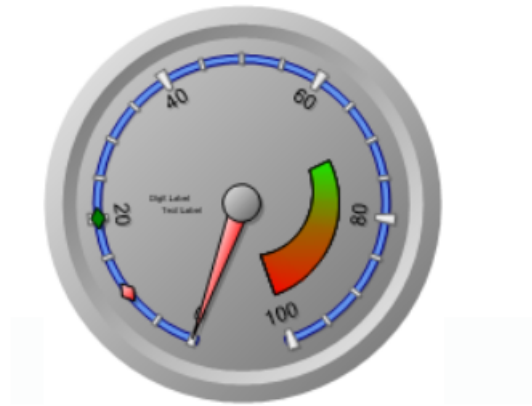


4. **Back Color 2:** This is a color picker that allows users to choose the secondary color of the scale.

! This field is only applicable to the minimum, middle, and maximum scales. It is disabled for the base scale.

5. **Border Color:** This is a color picker that allows users to choose the color of the scale's border.
6. **Border Width:** This is a numeric updown field that allows users to set the width of the scale's border.

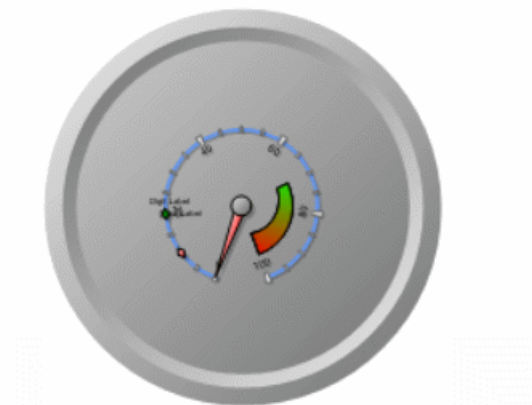
- **Gauge Scale Border Preview: Web View**



7. **Distance:** This is a numeric updown field that allows users to specify the distance between the scale and the gauge control.

- This value determines the spacing or gap between the scale and other elements of the gauge control.

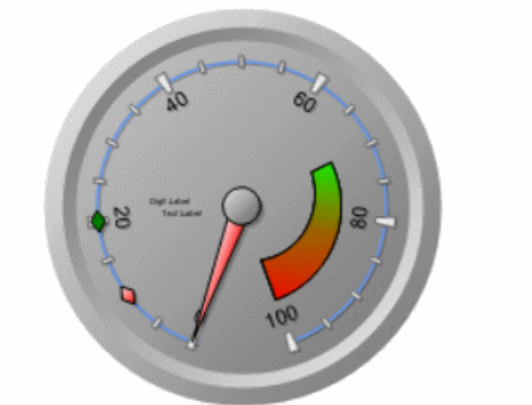
- **Gauge Scale Distance Customization: Web View**



8. **Start Width:** This is a numeric updown field that allows users to specify the starting width of the scale.


- This value determines the initial width of the scale at its starting point.

- **Gauge Scale Start Width Customization: Web View**




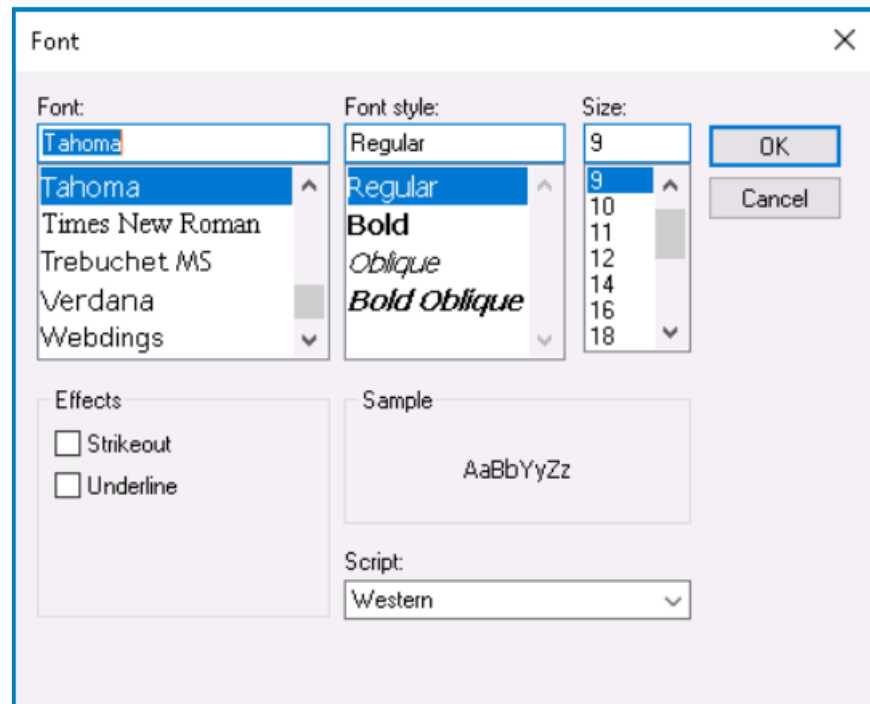
9. **Font:** This field allows users to configure the font properties for the text displayed in the label.



- To configure the font properties for the text, click on the  button.

- **Configuring Font Properties**

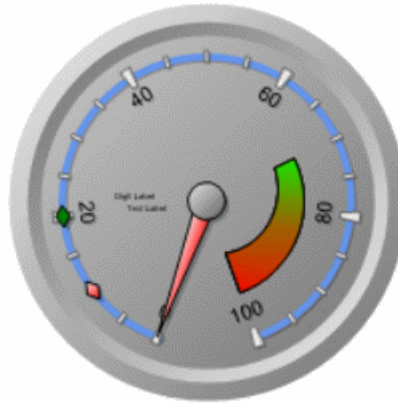
If the  button is selected, the Font window will appear.



- **Font:** This field allows users to choose the font family from a scrollable list of options.
- **Font Style:** This field allows users to choose the font style from a scrollable list of options.
  - Depending on the font family selected, the options displayed in this list may vary.
- **Size:** This field allows users to choose the font size from a scrollable list of options.
  - Users can also simply type in a specific font size.
- **Effects:** In this section, users can modify the appearance of the text by applying common effects, such as strikeout and underline.
  - **Strikeout:** This is a checkbox which, if selected, allows users to apply a strikeout effect to the text.
  - **Underline:** This is a checkbox which, if selected, allows users to apply an underline effect to the text.
- **Sample:** This field displays a preview of how the text will look based on the chosen font family, style, size, and effect. It updates in real-time as the user makes changes to the font settings.
- **Script:** This field allows users to choose a specific script or writing system for the text from a dropdown menu.
  - By selecting the appropriate script, users can ensure that their text is correctly displayed and formatted for the desired language or writing system.
- **OK or Cancel:** To save your changes and close the window, click on the **OK** button. To cancel your changes and close the window, click on the **Cancel** button.

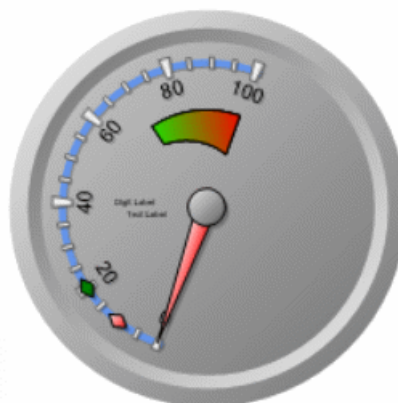
10. **Font Color:** This is a color picker that allows users to choose the font color for the text displayed within the scale.

- **Gauge Scale Font Customization: Web View**



11. **Interval:** This is a numeric updown field that allows users to specify the numerical interval between each scale mark or division.
- This value determines the spacing or distance between consecutive scale marks on the gauge.
12. **Start Angle:** This is a numeric updown field that allows users to specify the starting angle for the scale.
- This value determines the initial position or angle at which the scale starts within the gauge control.
  - **Gauge Scale Start Angle Customization: Web View**

13. **Sweep Angle:** This is a numeric updown field that allows users to specify the angular span or extent of the scale.
- This value determines the length or arc covered by the scale within the gauge control.
  - **Gauge Scale Sweep Angle Customization: Web View**



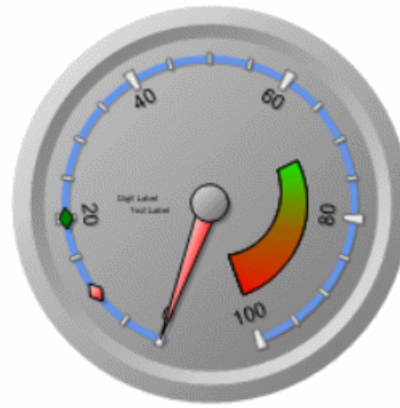
## 12.1.2.1.1.10.1.3.1. Scale (Minimum, Middle, Maximum)

In the Scale tab, users can customize various properties related to the scale of the gauge control.

\* All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. For more information on these tabs, please refer to the [description](#) for the **First Color** field.

1. **Scale Name:** This field allows users to select the specific scale they want to configure from a dropdown menu.
  - Scales: Base Scale, Min Scale, Middle Scale, or Max Scale.

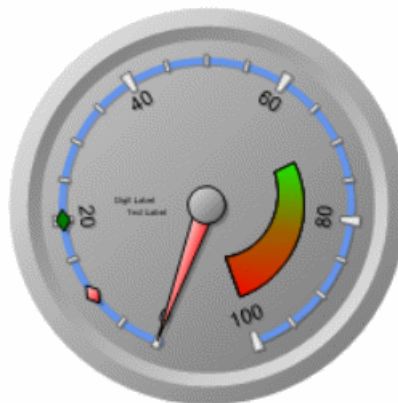
◦ **Gauge Scales Preview: Web View**



! Depending on the selected scale, the available fields in this window may vary. In this section of the manual, we will be focusing on the fields specifically applicable to the minimum, middle, or maximum scales. For information on the fields relevant to the base scale, [click here](#).

2. **Enable:** This is a checkbox which, if selected, allows users to enable the display of the selected scale.
3. **Back Color:** This is a color picker that allows users to choose the background color of the scale.
  - This color is used as the initial color or the starting point of the gradient that defines the scale's background.
4. **Back Color 2:** This is a color picker that allows users to choose the secondary color of the scale.
  - This color is used as the final color or the ending point of the gradient that defines the scale's background.
5. **Border Color:** This is a color picker that allows users to choose the color of the scale's border.
6. **Border Width:** This is a numeric updown field that allows users to set the width of the scale's border.

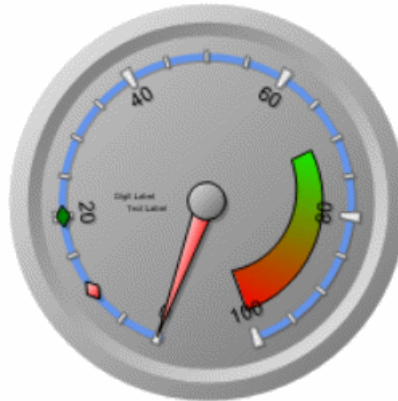
• **Gauge Scale Background Color Customization: Web View**



7. **Distance:** This is a numeric updown field that allows users to specify the distance between the

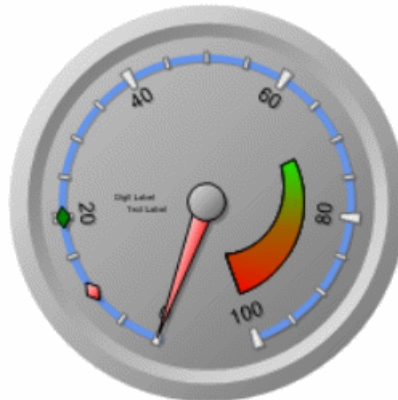
scale and the gauge control.

- This value determines the spacing or gap between the scale and other elements of the gauge control.
- **Gauge Scale Distance Customization: Web View**



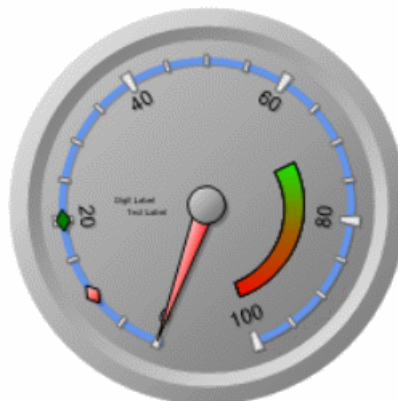
8. **Start Width:** This is a numeric updown field that allows users to specify the starting width of the scale.

- This value determines the initial width of the scale at its starting point.
- **Gauge Scale Start Width Customization: Web View**



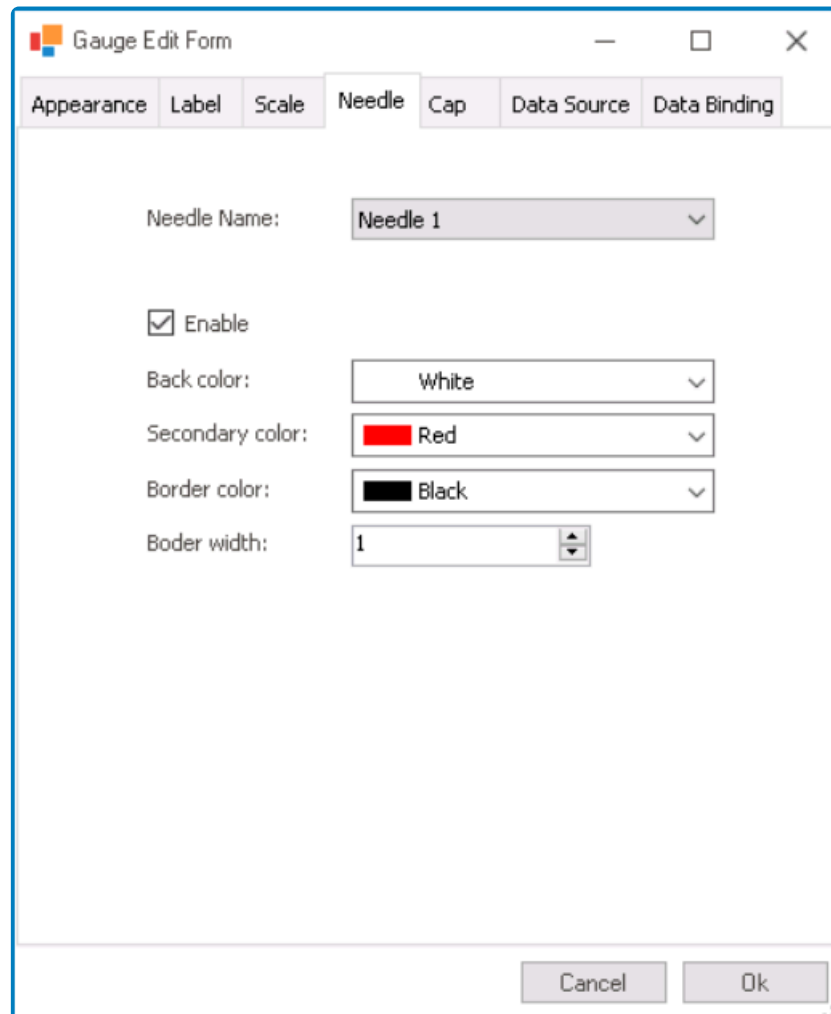
9. **End Width:** This is a numeric updown field that allows users to specify the ending width of the scale.

- This value determines the final width of the scale at its ending point.
- **Gauge Scale End Width Customization: Web View**



## 12.1.2.1.10.1.4. Needle

In the Needle tab, users can customize various properties related to the needle of the gauge control.



The screenshot shows the 'Gauge Edit Form' window with the 'Needle' tab selected. The window has a title bar with a close button and a maximize button. The tabs are 'Appearance', 'Label', 'Scale', 'Needle', 'Cap', 'Data Source', and 'Data Binding'. The 'Needle' tab contains the following controls:

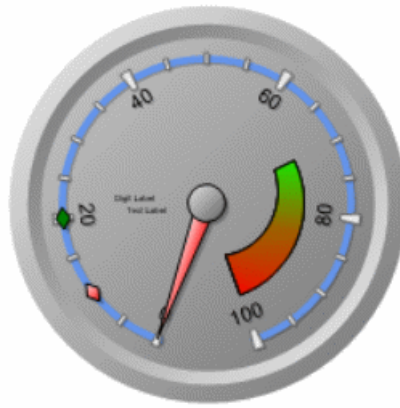
- Needle Name:** A dropdown menu showing 'Needle 1'.
- Enable:** A checked checkbox.
- Back color:** A color picker showing 'White'.
- Secondary color:** A color picker showing 'Red' with a red color swatch.
- Border color:** A color picker showing 'Black' with a black color swatch.
- Boder width:** A spinner box showing '1'.

At the bottom right of the window are 'Cancel' and 'Ok' buttons.

\* All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. For more information on these tabs, please refer to the [description](#) for the **First Color** field.

1. **Needle Name:** This field allows users to select the specific needle they want to configure from a dropdown menu.
  - Needles: Needle 1, Needle 2, or Needle 3.

- **Gauge Needles Preview: Web View**

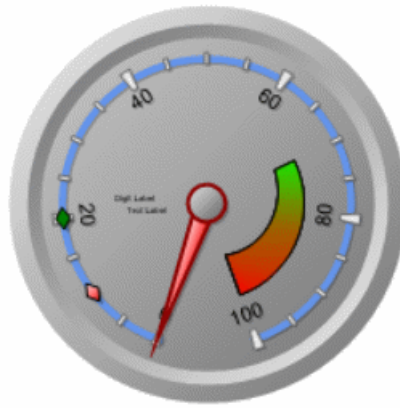


2. **Enable:** This is a checkbox which, if selected, allows users to enable the display of the selected needle.
3. **Back Color:** This is a color picker that allows users to choose the background color of the needle.
  - This color is used as the initial color or the starting point of the gradient that defines the needle's background.
4. **Back Color 2:** This is a color picker that allows users to choose the secondary color of the needle.
  - This color is used as the final color or the ending point of the gradient that defines the needle's background.
  - **Gauge Needle Background Color Preview: Web View**



5. **Border Color:** This is a color picker that allows users to choose the color of the needle's border.
6. **Border Width:** This is a numeric updown field that allows users to set the width of the needle's border.

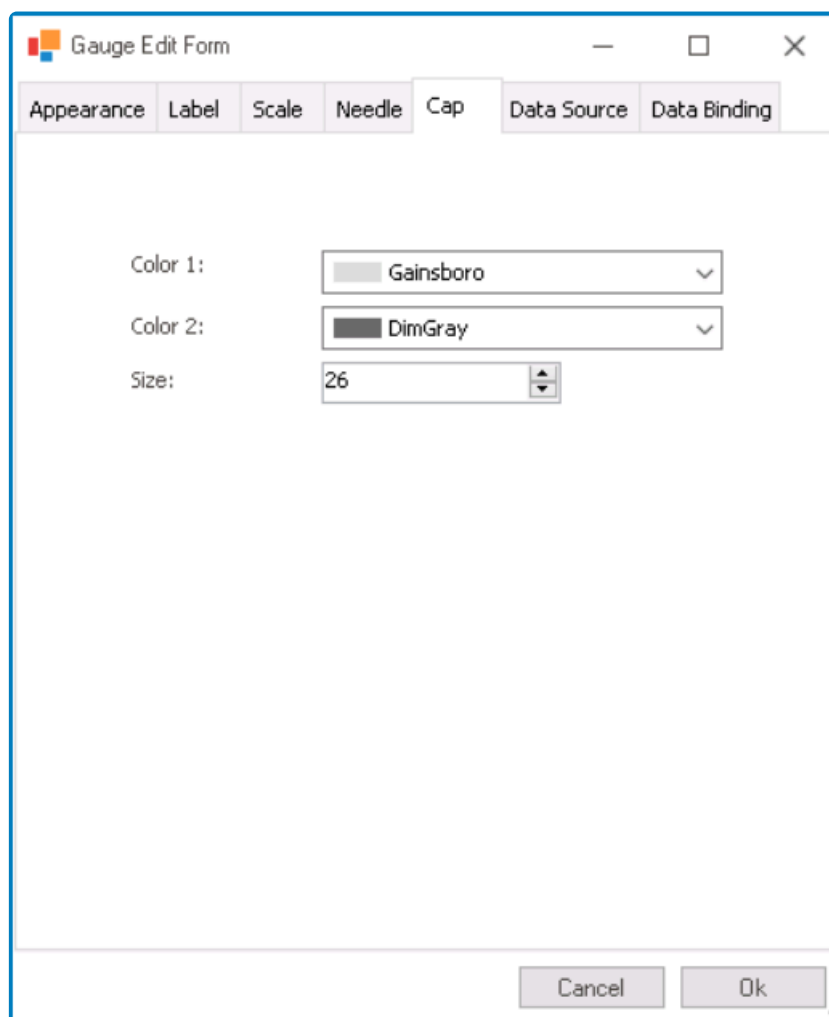
- **Gauge Needle Border Preview: Web View**





## 12.1.2.1.10.1.5. Cap

In the Cap tab, users can customize various properties related to the cap of the gauge control.

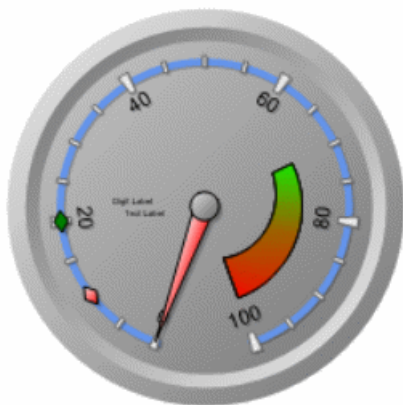


1. **Color 1:** This is a color picker that allows users to choose the color of the cap.
  - This color defines the primary or fill color of the cap.
2. **Color 2:** This is a color picker that allows users to choose the secondary color of the cap.
  - This color is used for gradient effects or complementary visual elements within the cap.
  - **Gauge Cap Color Preview: Web View**



3. **Size:** This is a numeric updown field that allows users to set the size or diameter of the cap.

- **Gauge Cap Size Customization: Web View**



## 12.1.2.1.10.1.6. Data Source

In the Data Source tab, users can select and configure the data source for the gauge control, including any necessary query parameters.

1. **Select Your Data Source:** In this section, users can select the data source for the gauge control.

- **Date Source:** This field retrieves the list of data sources (from both the [Dropdown Data Sources](#) section of the **Forms** module and the [KPIs](#) section of the **Entities** module). It allows users to select the data source for the gauge control.
  - To define a new data source, select the button.
    - It will display the [KPI Properties](#) window.
  - To modify an existing data source, select the button.
  - To remove a selected data source, select the button.

The selected data source will provide the data for the gauge control.

2. **Set Parameters & Run:** In this section, users can set parameters for the query and execute it with the selected data source.

- Parameters are values that are used to filter or modify the data returned by the query.
  - If the selected data source has a parametric query (i.e., a query that contains

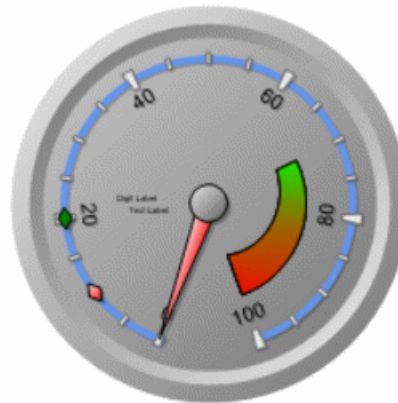
parameter placeholders), users can enter the parameter names in the Description column and the corresponding values for the parameters in the Value column.

- **Set Values:** This button allows users to run or execute the parametric query with the selected data source and the specified parameter values.
  - When the query is executed, it uses the provided parameter values to filter or modify the data from the selected data source.

## 12.1.2.1.10.1.7. Data Binding

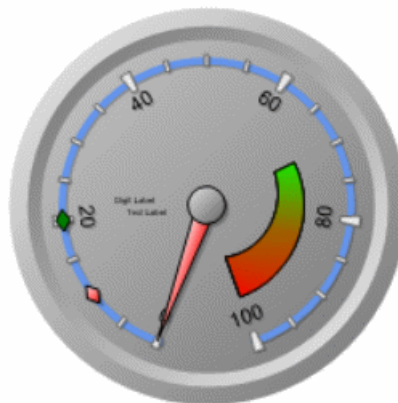
In the Data Binding tab, users can configure properties related to the needle, scale, and label of the gauge control. They can choose between static and dynamic values for these properties.

1. **Needle:** In this tab, users can configure properties related to the needle of the gauge control.
  - **Needle Name:** This field allows users to select the specific needle they want to configure from a dropdown menu.
    - Needles: Needle 1, Needle 2, or Needle 3.
    - **Gauge Needles Preview: Web View**



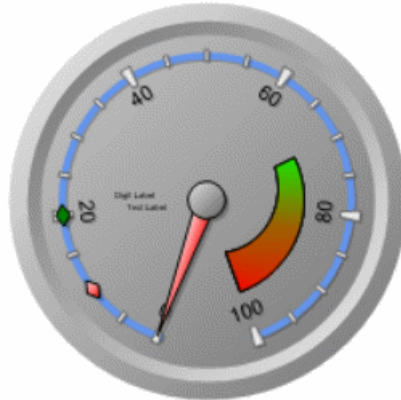
- **Static:** This option allows users to set a fixed value for the selected needle.
- **Dynamic (from DB):** This option allows users to set a dynamic value for the selected needle by choosing a column from the associated data source.
  - If this option is selected, the **Needle Value** field will display a dropdown of the available columns.
- **Needle Value:** This field allows users to input or select the value for the needle.
  - Depending on the option selected from the radio button, this value can be a fixed value or a dynamically bound value from the associated data source.

### Gauge Needle Value Customization: Web View



2. **Scale:** In this tab, users can configure properties related to the scale of the gauge control.

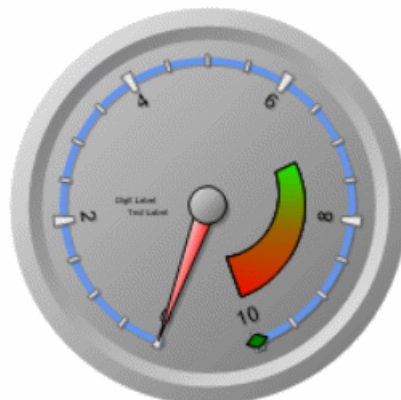
- **Scale Name:** This field allows users to select the specific scale they want to configure from a dropdown menu.
  - Scales: Base Scale, Min Scale, Middle Scale, or Max Scale.
    - **Gauge Scales Preview: Web View**



- **Static:** This option allows users to set fixed start and end values for the selected scale.
- **Dynamic (from DB):** This option allows users to set dynamic start and end values for the selected scale by choosing columns from the associated data source.
  - If this option is selected, the **Start Value** and **End Value** fields will display dropdowns of the available columns.
- **Start Value:** This field allows users to input or select the start value for the scale.
  - Depending on the option selected from the radio button, this value can be a fixed value or a dynamically bound value from the associated data source.
- **End Value:** This field allows users to input or select the end value for the scale.
  - Depending on the option selected from the radio button, this value can be a fixed value or a dynamically bound value from the associated data source.

#### Gauge Scale Values Customization: Web View

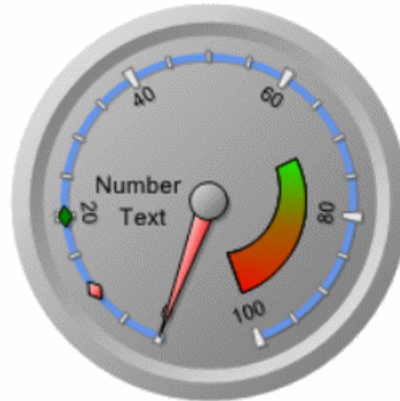
- **Base Scale Preview**



- **Min, Mid, & Max Scales Preview**

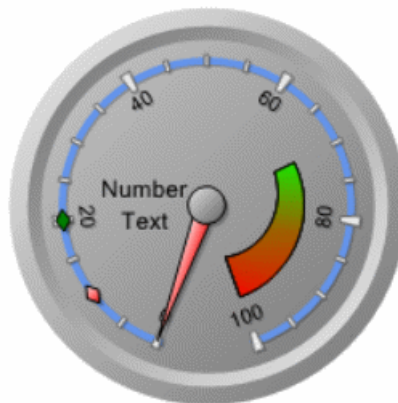
3. **Label:** In this tab, users can configure properties related to the label of the gauge control.

- **Label Name:** This field allows users to choose the specific label they want to configure from a dropdown menu.
  - Labels: Label Number or Label Text.
    - **Gauge Labels Preview: Web View**



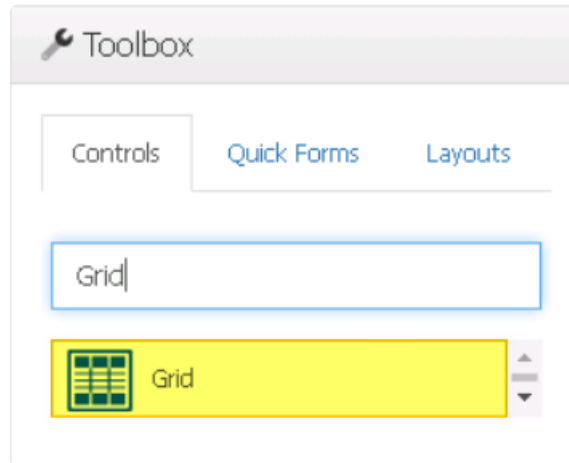
- **Static:** This option allows users to set a fixed value for the selected label.
- **Dynamic (from DB):** This option allows users to set a dynamic value for the selected label by choosing a column from the associated data source.
  - If this option is selected, the **Label Value** field will display a dropdown of the available columns.
- **Label Value:** This field allows users to input or select the value for the label.
  - Depending on the option selected from the radio button, this value can be a fixed value or a dynamically bound value from the associated data source.

#### Gauge Label Value Customization: Web View



## 12.1.2.1.11. Grid

Grid controls are useful tools for displaying and manipulating large sets of data in a structured format. They enable users to sort, filter, add, edit, and interact with data in a tabular view, making it easier to visualize and analyze complex information.



\* For tips and tricks on how to manage grids using business rules, [click here](#).

When a grid control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

\* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

\* The choice of attribute for a grid depends on the type of grid being created. For more information on selecting appropriate attributes for specific grid types, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.



- To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
  - This window consists of two tabs:
    - [Grid](#)
    - [Validation](#)

! Please note that users are limited to configuring only the required field validation properties for grid controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).

! Please note that the linked section contains information on setting an initial value, which is not applicable to grid controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Default Page Size:** This field allows users to set the number of items that are displayed in the grid at one time.
  - By default, the size is set to 20. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

\* In the web application, a pagination bar appears at the bottom of a grid when the number of items in the grid exceeds the specified threshold set by the **Default Page Size** field. The pagination bar includes controls such as buttons for moving to the next or previous page, buttons for navigating to the first or last page, and a dropdown field for changing the default number of items displayed per page.

- **Height:** This is a numeric updown field that allows users to set the height of the control.
  - By default, the height is set to 400. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.

\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

## Grid Control Preview: Web View

\* For more information on how to use this field in the web application, [click here](#).

### Quick Forms Application

To apply the **Quick Forms** feature to grid controls, take the following steps:

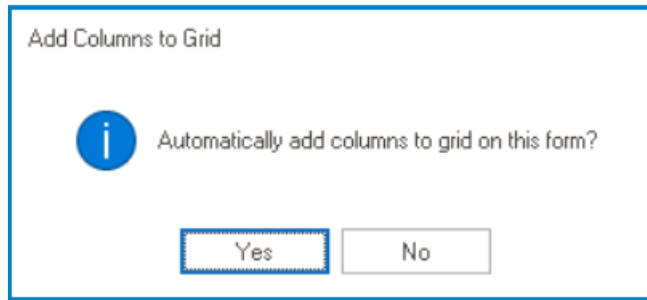
- Create a new form and leave the **Entity** field empty.

The screenshot shows the 'Form Properties' dialog box. It has three buttons at the top: 'Apply' (with a floppy disk icon), 'OK' (with a green checkmark icon), and 'Cancel' (with a red X icon). Below these are three input fields: 'Name' with the text 'GapAnalysisList', 'Form Entity' with the text 'GapAnalysisJDGapAnalysisListDS', and 'Entity' which is highlighted in yellow and contains the text '[Please Select...]' with a dropdown arrow.

- Confirm the prompt to design the form.
- In the Form Designer, drag a grid control from the toolbox and drop it onto the design environment.
- Go to the Entity Designer and copy the name of the main entity—that is, the entity associated with the form that the grid is referencing.

The screenshot shows the 'GapAnalysisJD / Entity Designer' window. On the left is a tree view of entities: 'NewGapAnalysis' (selected), 'AnalysisScope' (String / NVARCHAR (Max)), 'DataAnalysisMethods' (Data Analysis Methods Mid), 'GapAnalysisID' (String / NVARCHAR (Max)), 'GapAnalysisName' (String / NVARCHAR (Max)), and 'TargetDate' (Date/Time / Date). On the right is the 'Entity' tab for 'NewGapAnalysis'. It has fields for 'Name' (highlighted in yellow), 'Title' (NewGapAnalysis), and 'Description'. Below these are four buttons: 'Add new Entity' (with a plus icon), 'Copy this Entity' (with a copy icon), 'Delete Entity' (with a minus icon), 'Add new Attribute' (with a plus icon), and 'Paste Attribute' (with a paste icon).

- Return to the newly created form and paste the entity name into the **Server ID** field.
- Save the form.
- If you want to automatically add columns to the grid (based on the attributes defined in the main entity), confirm the following prompt:



- Once added, these columns can be viewed and customized in the [control properties](#) of the grid.

! When dealing with numerous attributes, the automatic addition of these columns can become confusing. In these situations, it is recommended to decline the aforementioned prompt and add the desired columns manually instead.

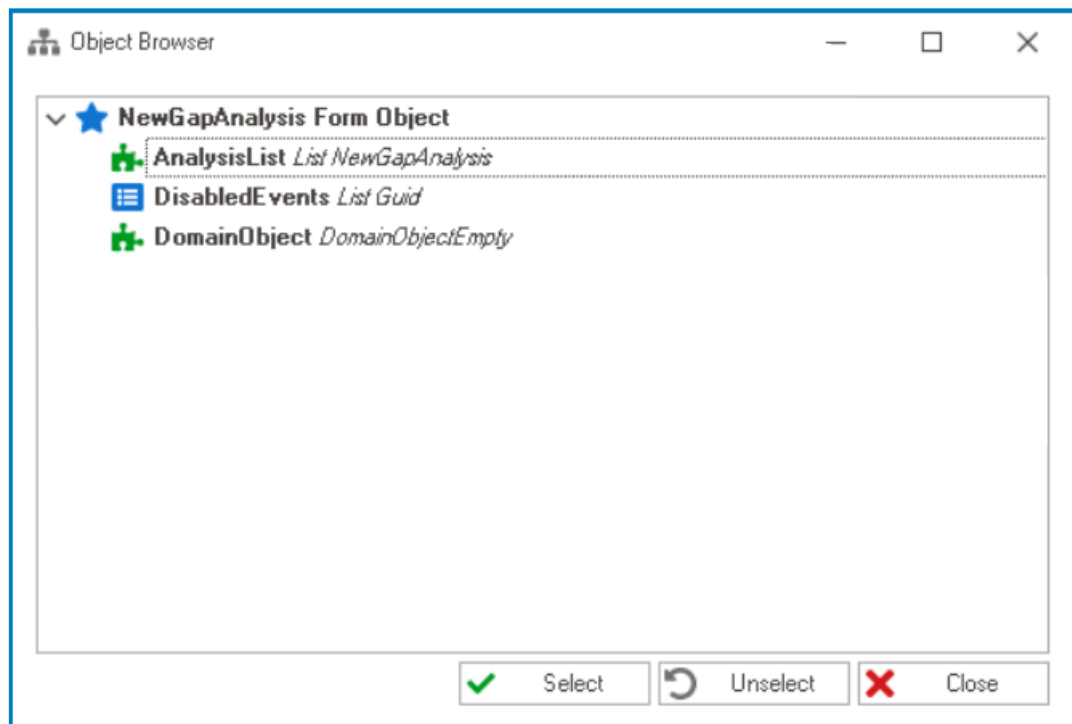
- Confirm the prompt to generate a form behavior.
- After users have answered the prompts, the form will close automatically. When the form is opened again, users will find that the grid is now linked to the appropriate attribute.

\* For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.11.1. Grid – Binding

In the **Binding** category, users can link a grid to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the grid control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated (form) entity's attributes—specifically, the attributes relevant to the grid control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

\* The choice of attribute for a grid depends on the type of grid being created. For more information on selecting appropriate attributes for specific grid types, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. It will display the Control Properties window.
  - This window consists of three tabs:
    1. [General](#)
    2. [Grid](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Control Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.



For tips and tricks on how to manage grids using business rules, [click here](#).

## 12.1.2.1.11.1.1. Grid – Attributes

The choice of attribute for a grid depends on the type of grid being created.

There are two types of grids:

### 1. Data Grid

A data grid is a type of grid that displays all created instances—all active or submitted instances—of another form. It can also be used to define options for dropdown menus.

- A data grid requires a custom entity. This entity is associated with the form that the grid references and contains the data that will be displayed in the grid.
- A data grid must be linked to a local attribute.
  - To learn more about how to create the necessary entity and attribute for this type of grid, [click here](#).



Users can also create the appropriate attribute for a data grid by applying the **Quick Forms** method. For more information on this method, [click here](#).

### 2. Subform Grid

A subform grid is a type of grid that is embedded within a form. It is typically used to supplement the form with additional details by enabling users to directly add and edit information.

- A subform grid requires at least two entities: a main entity, in which the grid will be used, and a custom entity, which will contain the attributes for the grid.



The main entity is the entity associated with the form where the grid control is being implemented. It represents the primary data object that the user is working with or collecting data for. A custom entity is an entity that is linked to another form from which data will be pulled. It is used to associate data with another form that is not related to the main entity.

- A foreign key linking the custom entity to the main entity must be created to establish a reference point between them.
- A subform grid must be linked to the foreign key associated with the main entity.
  - To learn more about how to create the necessary entities and foreign keys for this type of grid, [click here](#).



Users cannot apply the **Quick Forms** method to subform grids.

# 12.1.2.1.1.11.1.1.1. Data Grid – Attribute

As the data grid is not directly linked to any entity or attribute created in the [Entity Designer](#), users must instead create a form-specific local attribute for it.

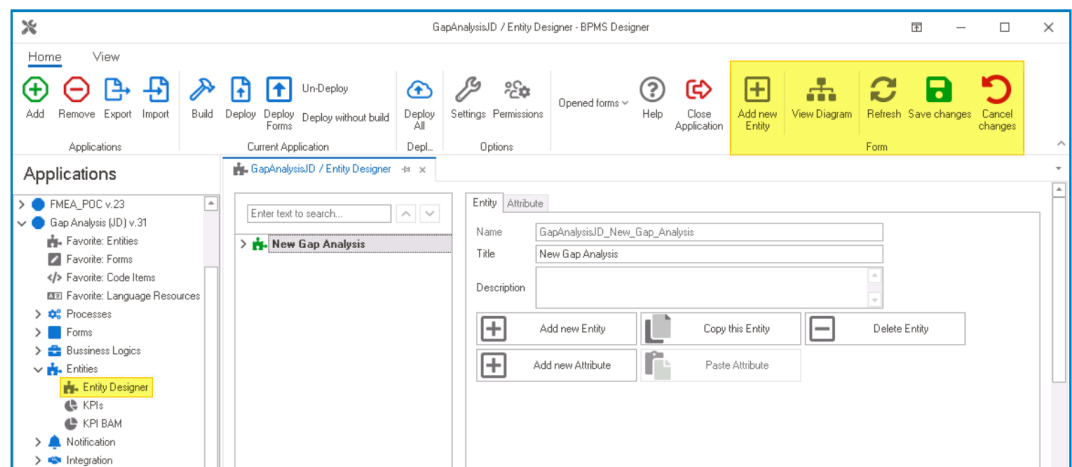
✿ Local attributes are specific to the form itself. They are used to store data that is relevant to the form, rather than to the entire application.

To create data grids and their associated attributes, follow these steps:

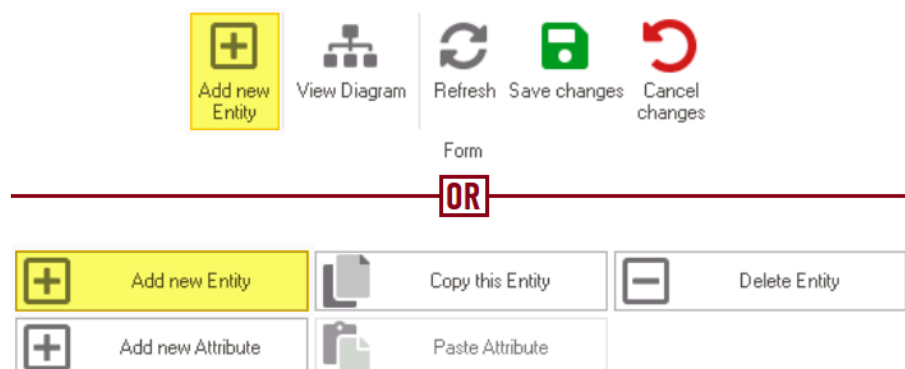
## 1. [Create Custom Entity](#)

### • Access Entity Properties

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the [Entity Designer](#) tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the [Entity](#) tab, users can define the properties of the newly created entity.

GapAnalysisJD / Entity Designer

Entity Attribute

Name: GapAnalysisJD\_NewEntity

Title:

Description:

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

## 2. [Create Custom Attributes](#)

### • Access Attribute Properties

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

GapAnalysisJD / Entity Designer

Entity Attribute

Name: NewAttribute

Title:

Description:

☒ Data field    ☐ Reference

Data field Reference

Data Type: Uniquelidentifier (Globally unique identifier)

Size:    Precision:   

Show as a column in the task list

☐ Hide    ☐ Hidden by default    ☒ Show by default

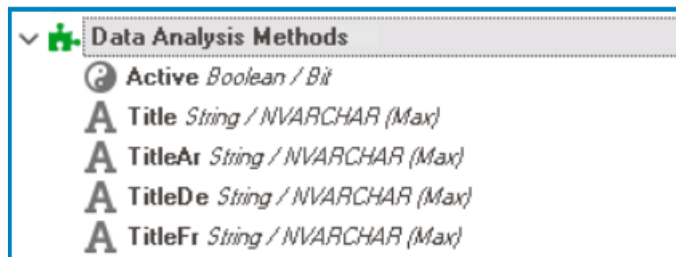
☐ Read Only    ☐ Defines field    ☐ Track changes

+ Add new Attribute    Copy Attribute    - Delete Attribute



- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose an attribute type— **Data Field** or **Reference**.
  - For the **Data Field** attribute, select the appropriate data type from the dropdown.
  - For the **Reference** attribute, select the appropriate reference from the System Reference window.

### • Custom Entity Attributes Example

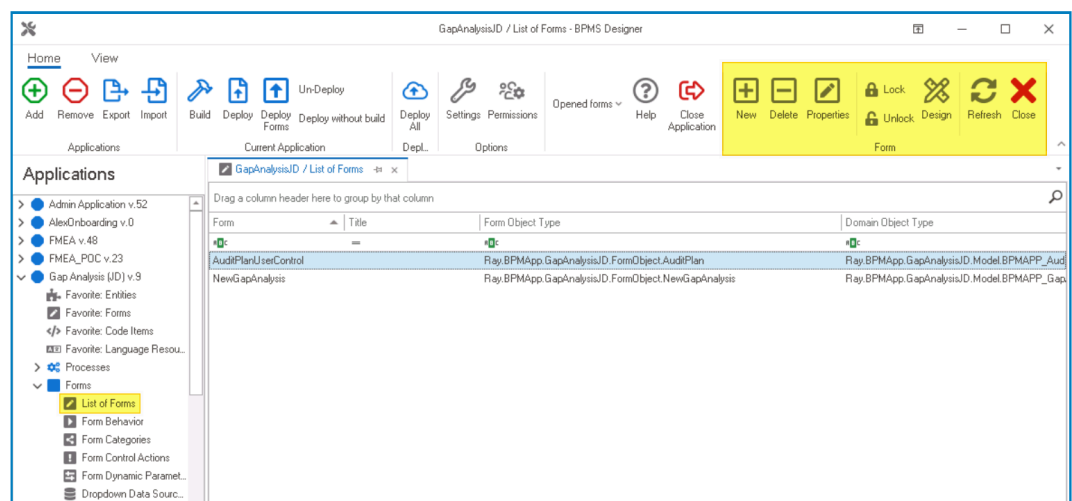


! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

## 3. Create & Design Custom Form

### • Access Form Properties

- From the Application Tree, navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



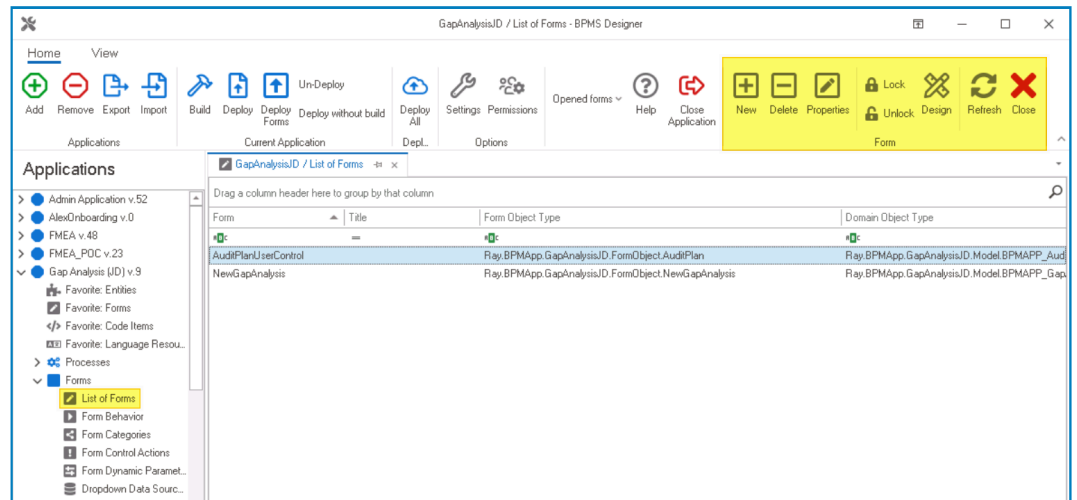
- In the Form Properties window, users can create a new form for the application.

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

**!** When using the grid to reference multiple instances or display modal forms, users should carefully design the layout of the custom form. When using the grid to define dropdown options, however, using **Inline Add** and **Inline Edit** functionalities, users can simply add the required controls (and their corresponding attributes) without a strict layout.

#### 4. Create & Design Main Form

- **Access Form Properties**
  - From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

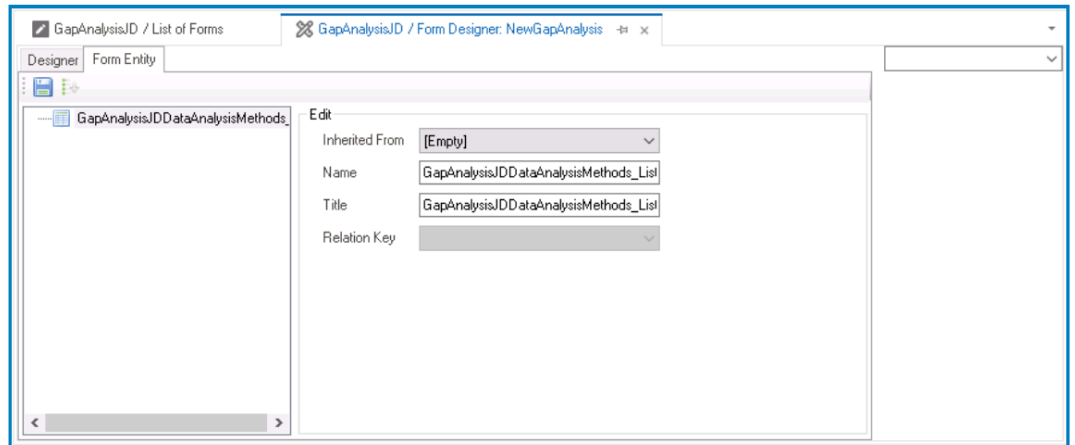
- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- Leave the **Entity** field empty.
- Click **OK** and confirm the prompt to start designing the form.

✿ Once they have launched the Form Designer, users should drag the grid control from the toolbox and drop it onto the design environment.

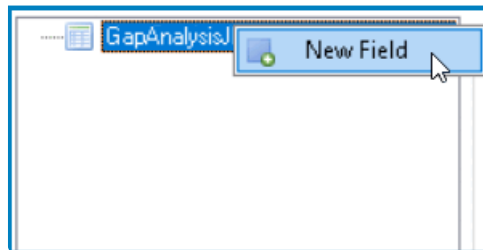
## 5. Create Local Attribute

- **Access Attribute Properties**

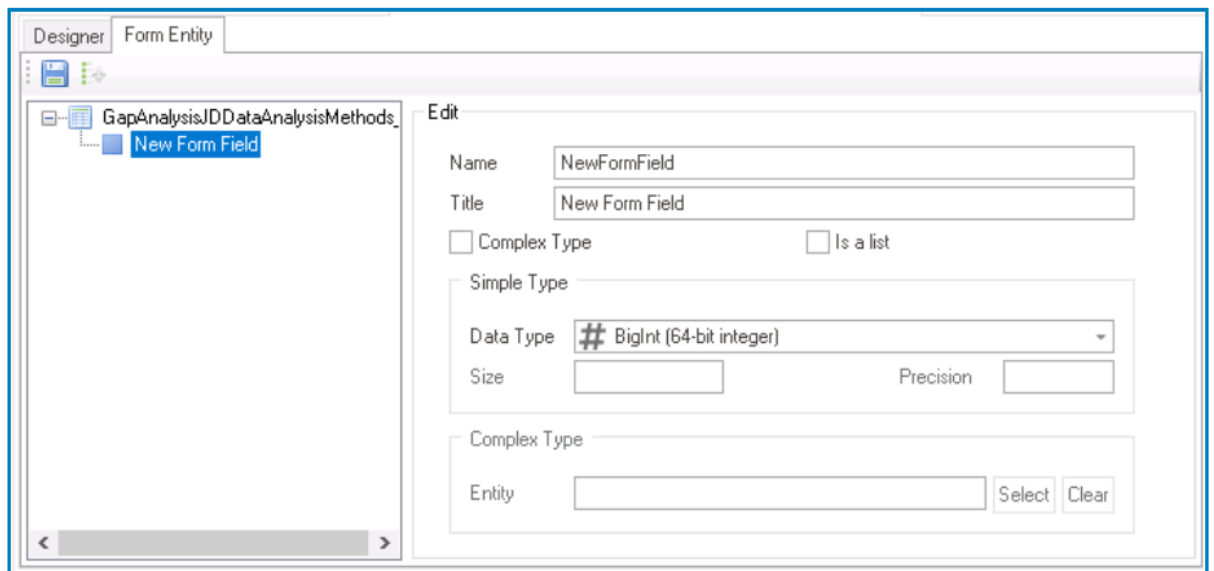
- From the Form Designer, navigate to the Form Entity tab.



- Right-click on the form entity and select **New Field** from the context menu. It will generate a new field below the form entity, which can then be customized and defined to suit the needs of the form.



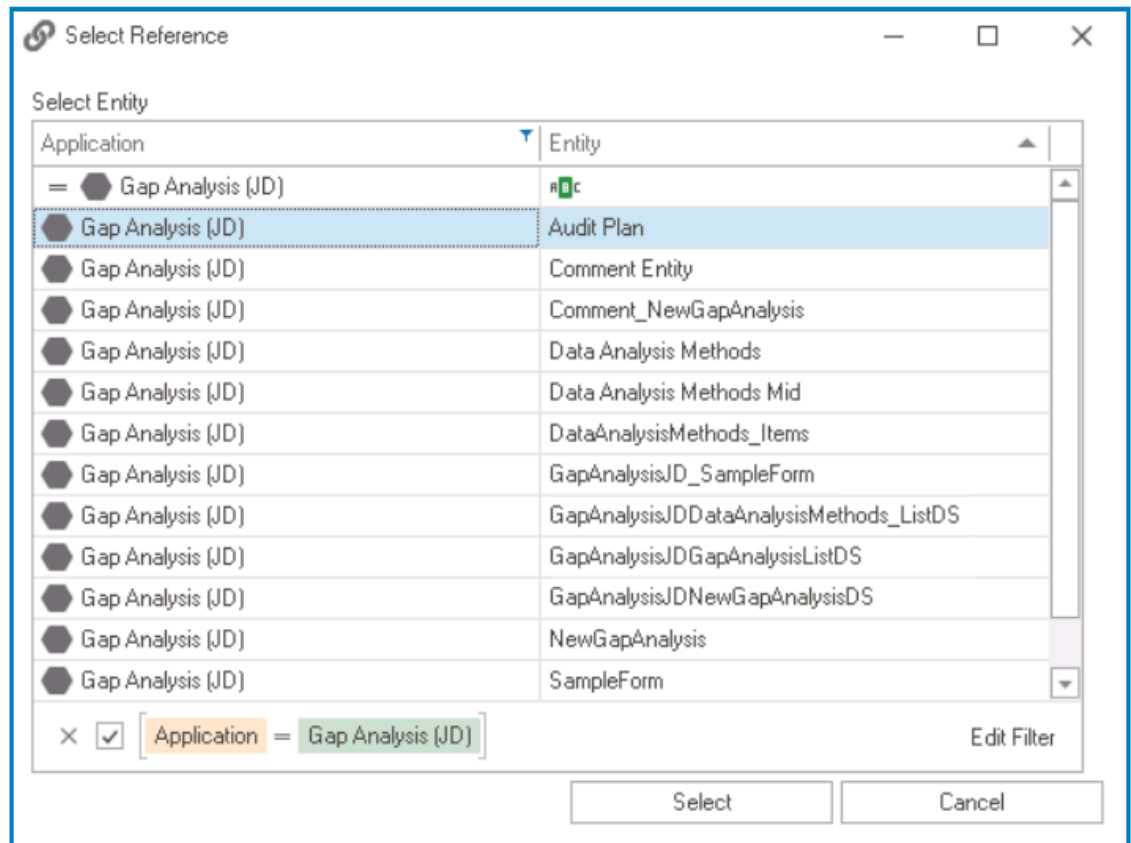
- In the Edit Properties pane, users can define the properties of the newly created attribute.



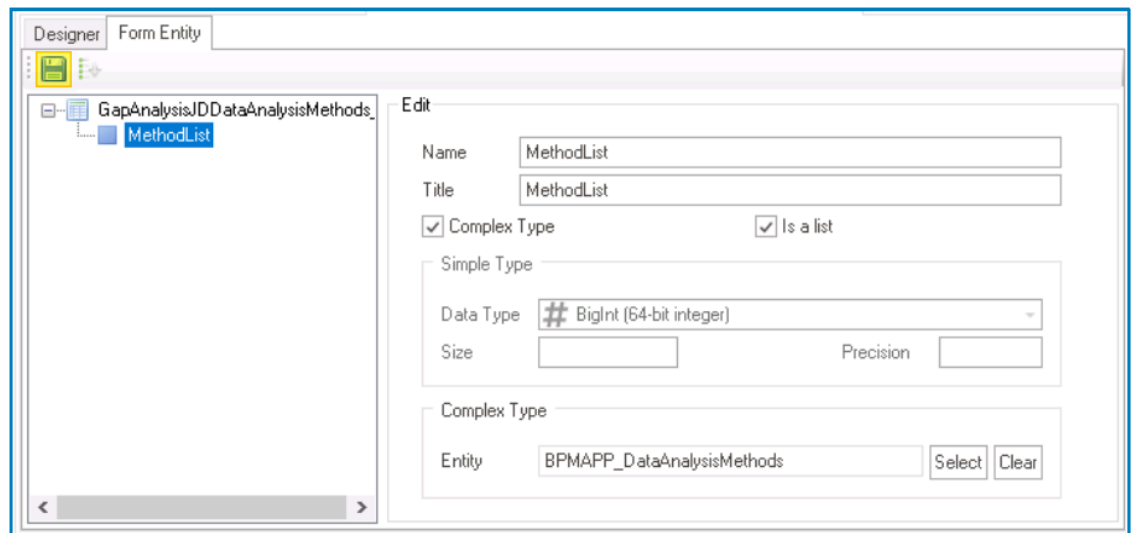
- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Select the **Complex Type** checkbox (because the control will reference data from another form).
- Select the **Is a List** checkbox (because the control will hold multiple instances of the

attribute's value).

- In the **Complex Type** section, select the custom entity from which the attribute will reference data.



- Click on the  button to save your changes.



## 6. Bind Attribute to Grid Control

- Once users have created the appropriate attribute, they can return to the Designer tab and bind the attribute to the grid control.



Once the control properties are configured, users need to save the form. This will activate the Quick Forms feature, prompting users to confirm the generation of both the main entity and the form behavior for the main form.

## 12.1.2.1.11.1.2. Subform Grid – Attribute

To create a subform grid, users need two entities: a main entity where the grid will be implemented and a custom entity that contains the attributes for the grid. A foreign key linking the custom entity to the main entity must be created to establish a reference point between them.

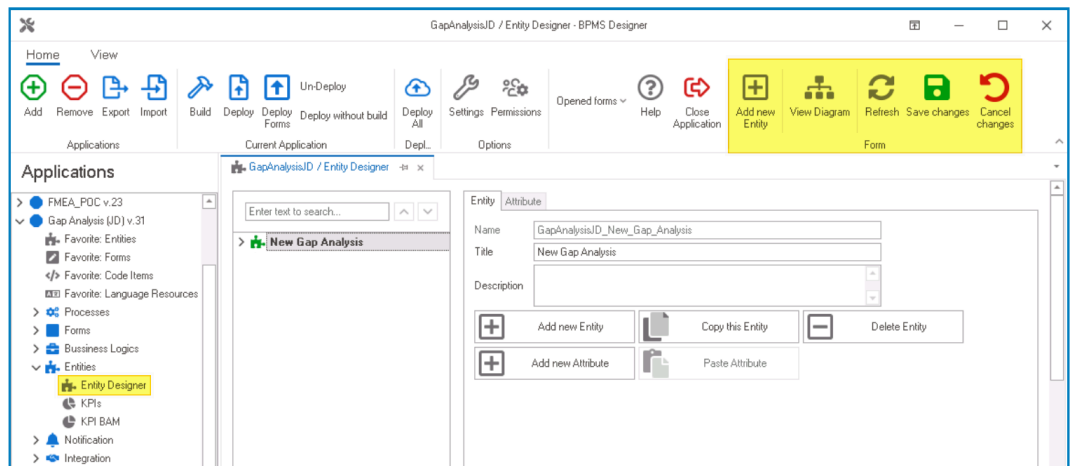
✿ Before proceeding with the following steps, please ensure you have created the main entity with its attributes, and linked the main entity to the main form.

To create subform grids and their associated attributes, follow these steps:

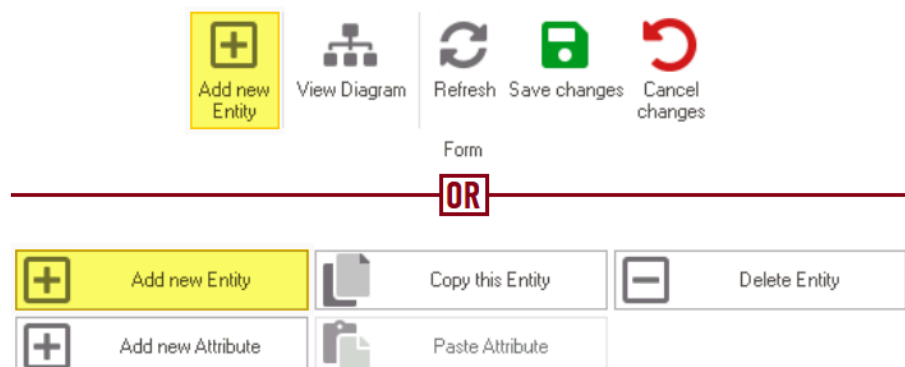
### 1. Create Custom Entity

#### • Access Entity Properties

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.

GapAnalysisJD / Entity Designer

Entity Attribute

Name: GapAnalysisJD\_NewEntity

Title:

Description:

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

## 2. [Create Custom Attributes](#)

### • Access Attribute Properties

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

GapAnalysisJD / Entity Designer

Entity Attribute

Name: NewAttribute

Title:

Description:

☒ Data field    ☐ Reference

Data field Reference

Data Type: Uniquelidentifier (Globally unique identifier)

Size:    Precision:

Show as a column in the task list

☐ Hide    ☐ Hidden by default    ☒ Show by default

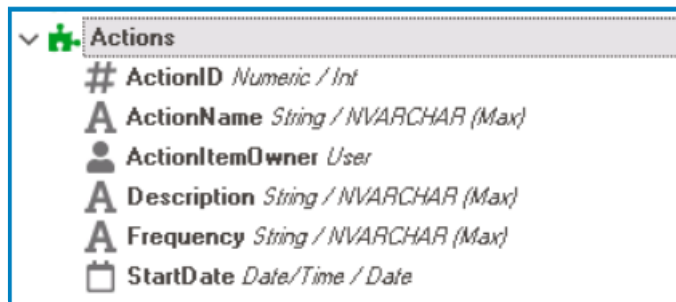
☐ Read Only    ☐ Defines field    ☐ Track changes

+ Add new Attribute    Copy Attribute    - Delete Attribute



- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose an attribute type— **Data Field** or **Reference**.
  - For the **Data Field** attribute, select the appropriate data type from the dropdown.
  - For the **Reference** attribute, select the appropriate reference from the System Reference window.

#### • Custom Entity Attributes Example



### 3. Create Foreign Key

#### • Access Attribute Properties

- Once you have created the custom entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.



- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

GapAnalysisJD / Entity Designer

Entity Attribute

Name:

Title:

Description:

☒ Data field ☐ Reference

Data field Reference

Data Type:

Size:  Precision:

Show as a column in the task list

☐ Hide ☐ Hidden by default ☒ Show by default

☐ Read Only ☐ Defines field ☐ Track changes

Add new Attribute   Delete Attribute

- In the **Name** field, provide a name for the foreign key.
- In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
- Choose the **Reference** option.
- Click on the button. It will display the Select Reference window. From the entity section, select **[AppName] – [MainEntity]**. From the field section, select **ID**.

Select Reference

Select Entity

Application Entity

|   |                   |                  |
|---|-------------------|------------------|
| = | Gap Analysis (JD) | Entity           |
|   | Gap Analysis (JD) | Actions          |
|   | Gap Analysis (JD) | New Gap Analysis |

Select field

|   |                 |                         |
|---|-----------------|-------------------------|
| A | AnalysisScope   | String / NVARCHAR (Max) |
| A | Category        | String / NVARCHAR (Max) |
| A | GapAnalysisID   | String / NVARCHAR (Max) |
| A | GapAnalysisName | String / NVARCHAR (Max) |
| K | Id              | Id / UniquelIdentifier  |
| A | Source          | String / NVARCHAR (Max) |
| T | TargetDate      | Date/Time / Date        |

☒ Application = Gap Analysis (JD)

- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[MainEntity]** and the field section will display **ID**.

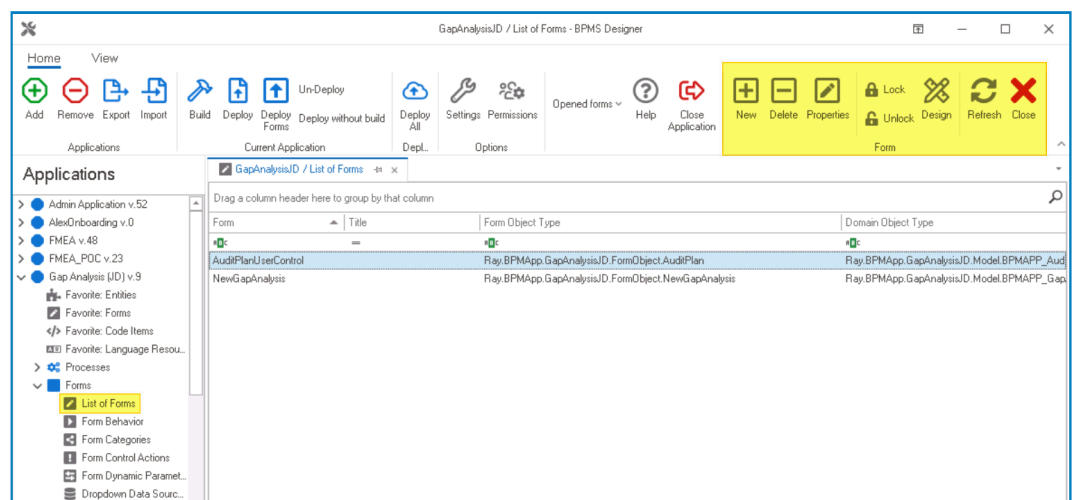


Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

#### 4. [Create & Design Custom Form](#)

- **Access Form Properties**

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

! When working with grids, users can choose between two sets of functions: **Add** and **Edit** functions that display modal forms, or **Inline Add** and **Inline Edit** functions that allow for direct information entry. If users opt for modal forms, they will need to properly design the layout of the custom form to ensure optimal usability. If they choose to use **Inline Add** and **Inline Edit** functions, however, they can simply add the necessary controls (and their corresponding attributes) without a strict layout.

## 5. Configure Grid Control Binding Properties

- After creating and configuring the custom form, users can return to their main form and define the binding properties for the grid control.
- **Bind Grid Control to Attribute**
  - A subform grid must be linked to the foreign key associated with the main entity.
- **Configure Binding Properties**
  - To access the control properties, click on the ellipses (...) inside the field. In the displayed window, navigate to the Grid tab.

Control Properties

Apply OK Cancel

Form Control: Grid

Entity and Attribute Object type: DomainObject.BPMAPP\_Actionseslk\_NewGapAnalysis, Ray.BPMApp.GapAnalysisID.Model.BPMAPP\_Actions

General Grid

Caption: Grid [Grid]

Form: [Empty]

Query (HQL only): [Empty]

Base info: ☐ Base info, ☒ Master-detail

Foreign key: [Empty]

Ascending sort: ☒ Ascending sort, ☐ Show aggregate, ☐ Show filter

Column width in %: ☐ Column width in %, ☐ Hide row number, Max row height (0=Off): 0

Add Column Remove Column Add Columns for Entity Fields

| Position | Field | Field Type | Header | Width | Align | Direction | Resizable | Sortable | Sort | Sort Type | Format | Visible | Edit |
|----------|-------|------------|--------|-------|-------|-----------|-----------|----------|------|-----------|--------|---------|------|
|----------|-------|------------|--------|-------|-------|-----------|-----------|----------|------|-----------|--------|---------|------|

Record 0 of 0

- From the **Form** dropdown, select the custom form.
  - The custom form is the form associated with the custom entity. It will be called when the **Add** function is selected in the grid.
- Choose the **Master-Detail** option.
- From the **Foreign Key** dropdown, select the foreign key linked to the main entity.
- Click on the **Add Columns for Entity Fields** button. It will automatically add all fields in the main and custom entity as columns in the grid.

Control Properties

Apply OK Cancel

Form Control: Grid

Entity and Attribute Object type: DomainObject.BPMAPP\_Actionseslk\_NewGapAnalysis, Ray.BPMApp.GapAnalysisID.Model.BPMAPP\_Actions

General Grid

Caption: Grid [Grid]

Form: Actions

Query (HQL only): [Empty]

Base info: ☐ Base info, ☒ Master-detail

Foreign key: BPMAPP\_NewGapAnalysisfk\_New

Ascending sort: ☒ Ascending sort, ☐ Show aggregate, ☐ Show filter

Column width in %: ☐ Column width in %, ☐ Hide row number, Max row height (0=Off): 0




Add Column Remove Column Add Columns for Entity Fields

| Position | Field                  | Field Type      | Header            | Width | Align | Direction | Resizable                           | Sortable                            | Sort | Sort Type | Format         | Visible                             | Edit |
|----------|------------------------|-----------------|-------------------|-------|-------|-----------|-------------------------------------|-------------------------------------|------|-----------|----------------|-------------------------------------|------|
| 0        | ActionName             | System.String   | Action Name       | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | TextBox        | <input checked="" type="checkbox"/> |      |
| 0        | Description            | System.String   | Description       | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | TextBox        | <input checked="" type="checkbox"/> |      |
| 0        | Frequency              | System.String   | Frequency         | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | TextBox        | <input checked="" type="checkbox"/> |      |
| 0        | StartDate              | System.DateTime | Start Date        | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | GregorianDa... | <input checked="" type="checkbox"/> |      |
| 0        | ActionID               | System.Int32    | Action ID         | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | Numeric        | <input checked="" type="checkbox"/> |      |
| 0        | UsesActionItemOwner... | System.String   | Action Item Owner | 300   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   | None           | <input checked="" type="checkbox"/> |      |

Record 1 of 12



























- Customize the columns of the grid as needed, adjusting properties such as their position order, width, and format.
  - Click on the **OK** button to save your changes and exit the window.
- Configure Control Governance Properties**
    - To access the control governance properties, click on the ellipses (\*\*\*) inside the field. In the displayed window, navigate to the Grid tab.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Grid]

 Apply  OK  Cancel

Control Name **Grid**


**Grid** Validation

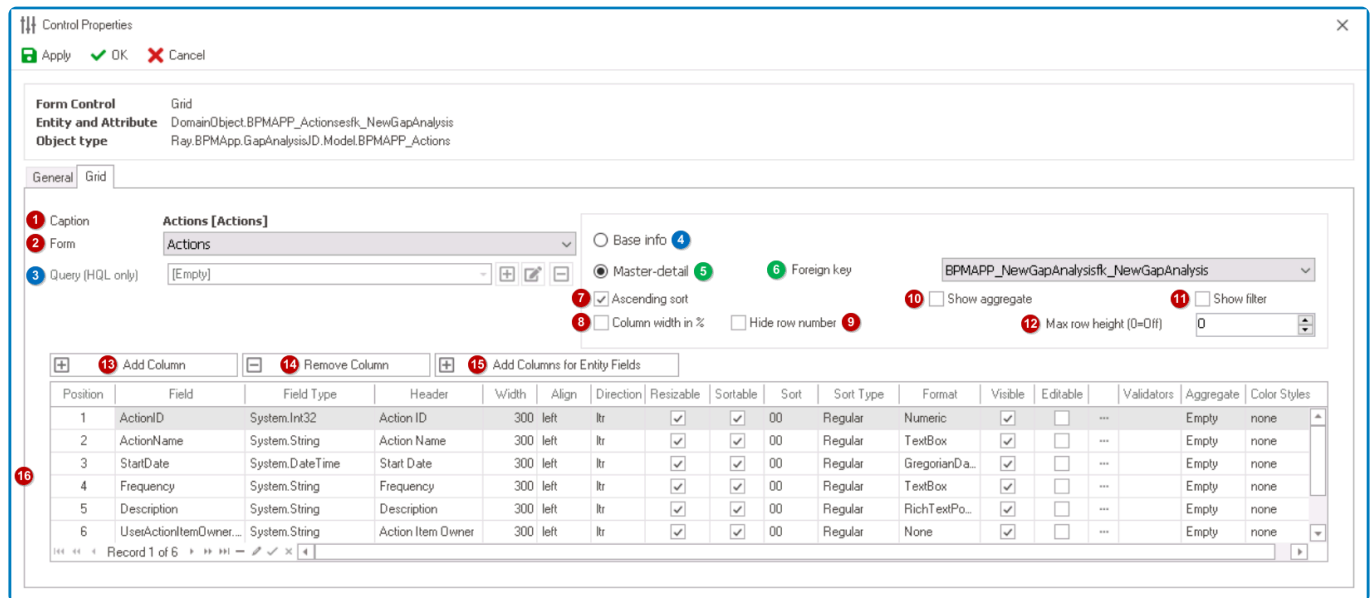
|   |                  |         |   |
|---|------------------|---------|---|
| <input type="checkbox"/>  Enable Edit                            | Edit Form        | [Empty] |    |
| <input type="checkbox"/>  Enable New                             | New Form         | [Empty] |    |
| <input type="checkbox"/>  Enable View                            | View Form        | [Empty] |    |
| <input type="checkbox"/>  Enable Delete                          | Delete Form      | [Empty] |    |
| <input type="checkbox"/> Inline New   | Import Form      | [Empty] |    |
| <input type="checkbox"/> Inline Edit  | Inline Edit Form | [Empty] |    |
| <input checked="" type="checkbox"/> Enable import from Excel  | Import Form      | [Empty] |    |
| <input checked="" type="checkbox"/> Enable export from Excel  |                  |         |   |
| <input type="checkbox"/>  Enable Select (only in selector forms) |                  |         |   |

- To enable a functionality, users must link a corresponding form behavior.
  - To set up a form behavior, users only need to provide a name and a title.
  - When creating the form behaviors for the 'New' functionalities, users must select the **Starter** checkbox.
  - The 'Delete' functionality does not require a form behavior.
- Click on the **OK** button to save your changes and exit the window.
- Click **Save Form** and deploy the application to the web.


## 12.1.2.1.11.1.2. Grid – Control Properties




In the Grid tab of the Control Properties window, users can build and customize the grid.

 In the General tab of this window, users can configure business rules for the grid control. For more information on this matter, please [click here](#) and navigate to the relevant section.



1. **Caption:** This field displays the (English) caption of the grid.
  - To modify the caption displayed in this field, refer to the [Name](#) category in the Control Properties panel.
2. **Form:** This is a single-select dropdown field that retrieves the list of forms (from the [List of Forms](#) section of the **Forms** module). It allows users to select which form they want to display the attributes of in the grid.

 Only forms associated with the selected attribute for the grid will be displayed in this dropdown. Depending on which attribute is selected, the list of available forms may vary. For more information on the binding properties for grids, [click here](#).

3. **Query (HQL Only):** This is a single-select dropdown field that retrieves the list of relevant queries (from the [KPI](#) section of the **Entities** module). It allows users to select a custom HQL query for the data source of a grid, providing greater flexibility and control over the displayed information.
  - To define a new query, select the  button.
    - It will display the [Query Editor](#) window.
  - To modify an existing query, select the  button.
  - To remove a query, select the  button.

 To enable this field, users must first select the **Base Info** option.

4. **Base Info:** This option allows users to display the fields of the custom entity associated with the grid, without showing related detail records. It is useful for presenting a simplified view of the data when only basic information is required.
  - If this option is selected, it will enable the **Query (HQL Only)** field, allowing users to select or define a custom query for the grid data source.

✿ Detail records are additional records related to the custom entity associated with the grid, often through a foreign key relationship.

5. **Master-Detail:** This option allows users to display related detail records in addition to the custom entity's fields in the grid, based on a defined relationship (such as a foreign key). This is helpful when a more detailed view of the data is required.
  - If this option is selected, it will enable the **Foreign Key** field, allowing users to select a relationship between the main entity and the custom entity's detail records, which will be displayed in the grid.
6. **Foreign Key:** This is a single-select dropdown field that retrieves the list of foreign keys (from the [Entity Designer](#) section of the **Entities** module. It allows users to specify a relationship between the main entity and the custom entity's detail records, which will be displayed in the grid.

! Only foreign keys under the entity associated with the grid will be displayed in this dropdown.

✿ To enable this field, users must first select the **Master-Detail** option.

7. **Ascending Sort:** This is a checkbox which, if selected, allows users to sort the grid data in ascending order based on the selected column header.
8. **Column Width in %:** This is a checkbox which, if selected, allows users to automatically adjust the width of the grid columns as a percentage of the grid width.
9. **Hide Row Number:** This is a checkbox which, if selected, allows users to hide the row numbers displayed in the first column of the grid.
10. **Show Aggregate:** This is a checkbox which, if selected, allows users to display aggregate values for selected columns in the grid footer, such as sum or average.



| Products     |            |               |                    |                   |          |                     |
|--------------|------------|---------------|--------------------|-------------------|----------|---------------------|
| + Inline Add |            | ✎ Inline Edit |                    | 📄 Export to Excel |          | 📄 Excel template    |
|              |            |               |                    |                   |          | 📄 Import from Excel |
|              | Product ID | Product Name  | Category           | Price             | Quantity | Total Sales         |
| 1            | P-00001    | ABC Product   | Clothing and Ac... | 15.00             | 1,000    | 15,000.00           |
| 2            | P-00002    | XYZ Product   | Clothing and Ac... | 5.00              | 500      | 2,500.00            |
|              |            |               |                    |                   |          | 17,500.00           |

11. **Show Filter:** This is a checkbox which, if selected, allows users to display a filter row at the top of the grid where they can filter the data based on specific criteria.
12. **Max Row Height:** This is a numeric updown field that allows users to set the maximum height of the rows in the grid.
  - By default, the maximum height is set to 0. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

✳ If this field is kept at 0 or left empty altogether, the row height will automatically adjust to display rich text content. If a value is inputted in the field, however, all rows in the grid will be set to the determined height.

13. **Add Column:** This button allows users to add a new column to the grid.
  - If selected, it will display the [Grid Column](#) window, in which users can define the properties of the column.
14. **Remove Column:** This button allows users to remove a selected column in the grid.
15. **Add Columns for Entity Fields:** This button allows users to add all fields in the related entity as columns in the grid.
16. **Column Customization:** In this table, users can access and customize the columns populating the grid.

| Position | Field                  | Field Type      | Header            | Width | Align  | Direction | Resizable                           | Sortable                            | Sort | Sort Type |
|----------|------------------------|-----------------|-------------------|-------|--------|-----------|-------------------------------------|-------------------------------------|------|-----------|
| 1        | ActionID               | System.Int32    | Action ID         | 300   | center | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |
| 2        | ActionName             | System.String   | Action Name       | 300   | unset  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |
| 3        | StartDate              | System.DateTime | Start Date        | 300   | left   | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |
| 4        | Frequency              | System.String   | Frequency         | 300   | left   | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |
| 5        | Description            | System.String   | Description       | 300   | left   | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |
| 6        | UserActionItemOwner... | System.String   | Action Item Owner | 300   | left   | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Regular   |

Record 1 of 6

- Position:** This field allows users to specify the order in which the column appears in the grid.
- Field:** This field allows users to select the corresponding attribute for the column.
  - To select a field or attribute, click on the ellipses (...) inside the box. It will display the [Object Browser](#) window, in which users can access a list of the associated entity's

attributes.

- **Field Type:** This field displays the data type of the attribute linked to the column.
- **Header:** This field allows users to specify the text that appears in the header of the column.
  - Users can input their preferred header directly into this field. Alternatively, they can click on the ellipses to display the Grid Column Header Edit window, in which they can input or modify the caption in all available languages.

- As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the column, regardless of other language options that may be available.



Users can determine which languages are available for use in the Settings of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.


- **Width:** This field allows users to set the width of the column in pixels.
- **Align:** This field allows users to specify the alignment of the text within the column. Users can choose their alignment preference from a dropdown menu.
  - **Alignment Options**
    1. Left: This option aligns the text to the left side of the column.
    2. Right: This option aligns the text to the right side of the column.
    3. Center: This option centers the text within the column.
    4. Default: This option aligns the text based on the language or culture setting in the web application.
- **Direction:** This field allows users to specify the direction of the text within the column. Users can choose their direction preference from a dropdown menu.
  - **Direction Options**
    1. Left-to-Right: This option specifies that the text in the column should be read from left to right, in the standard direction for languages that use a Latin-based alphabet (such as English).
    2. Right-to-Left: This option specifies that the text in the column should be read from right to left, in the opposite direction to the standard direction for languages that use a Latin-based alphabet (such as Arabic or Hebrew).

3. **Default:** This option specifies that the text direction should be based on the language or culture setting in the web application.

- **Resizable:** This is a checkbox which, if selected, allows users to resize the width of the column in the grid.
- **Sortable:** This is a checkbox which, if selected, allows users to sort the column by clicking on its header.
- **Sort:** This field allows users to set the initial sort order of the column. Users can choose the sort order from a dropdown menu.
  - By selecting a value from this dropdown, users are essentially assigning a priority level for the column.
- **Sort Type:** This field allows users to specify the type of sorting that should be used for the column. Users can choose their sorting preference from a dropdown menu.
  - **Sort Types**
    1. **Regular:** This option allows users to sort the data in the column based on their string representation. For example, if the column contains data in the format of “AA, BB, CC”, the sorting will be done alphabetically.
    2. **As Numbers:** This option allows users to sort the data in the column based on their numerical representation. For example, if the column contains numerical data, such as “10, 20, 30”, the sorting will be done based on their numerical value, not their string representation.
- **Format:** This field allows users to specify the display format of the column.
  - To select or modify the display format, click on the ellipses (...) inside the field.
    - It will display the [Edit Type](#) window.



If users choose to add a date field to the grid, they should select a format that matches the chosen display type for the corresponding date/time picker. This is because the selected format will override the display type. For more information on the display type options, [click here](#).

- **Visible:** This is a checkbox which, if selected, allows for the column to be visible in the grid.
- **Editable:** This is a checkbox which, if selected, allows users to edit data inside the column.
- **Column State:** This field allows users to set the column state per form behavior.
  - To access or modify the column state, click on the  button.
    - It will display the [Column Behavior Grid](#) window.
- **Validators** This field allows users to specify validation rules for the data within the column.
  - For more information on required field and range validation, [click here](#).
- **Aggregate:** This field allows users to specify how the data in the column should be aggregated. Users can choose their preferred aggregation option from a dropdown menu.
  - **Aggregation Options**
    1. **Average:** This option allows users to calculate the average of the values in the column.
    2. **Empty:** This option allows users to specify that the grid should not perform any aggregation on the column.
    3. **Sum:** This option allows users to calculate the sum of the values in the

column.

- **Color Styles:** This field allows users to apply color styles to the column in the grid.
  - To apply color styles to the column, click on the ellipses (⋮) inside the field.
    - It will display the [Column: Color Styles](#) window.



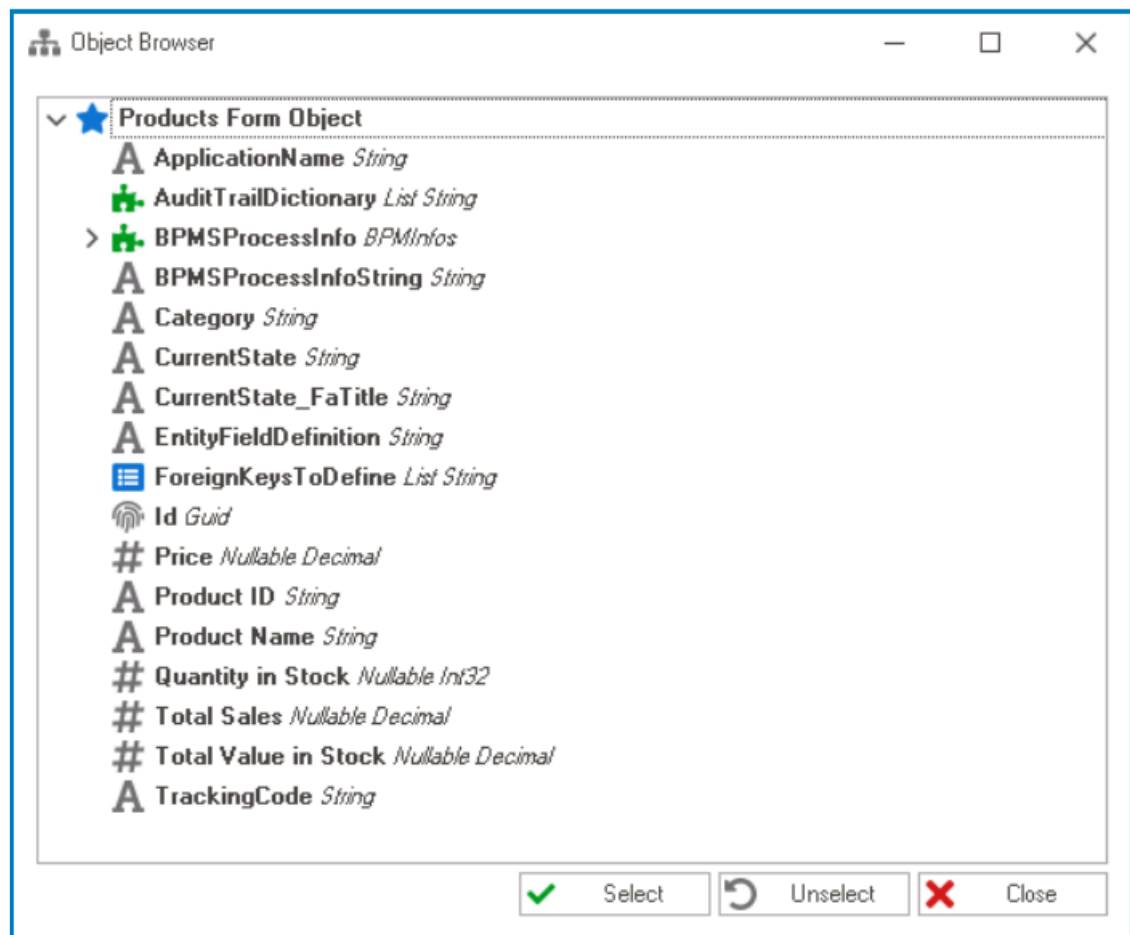
By using color styles, users can visually highlight important data or patterns in the column, making it easier to interpret and analyze the information. Users can specify conditional formatting rules for the data within the column, such as applying a specific background or text color to cells that meet certain criteria.

## 12.1.2.1.11.1.2.1. Grid Column

In the Grid Column window, users can define the properties of the column.

\* Once users have added a column to the grid, they can modify its properties by accessing the corresponding table in the Control Properties window.

1. **Select:** This button allows users to select the corresponding attribute for the column.
  - If selected, it will display the Object Browser window, in which users can access a list of the associated entity's attributes.



✿ To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, its name will populate the **Field** box and its data type will populate the **Field Type** box.

|            |               |        |
|------------|---------------|--------|
| Field      | ProductID     | Select |
| Field Type | System.String |        |

! It is mandatory for users to link each column in the grid to its corresponding attribute.

2. **Caption:** This field allows users to input the name or caption of the column.
  - As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the column, regardless of other language options that may be available.

✿ Users can determine which languages are available for use in the [Settings](#) of

the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.



It is mandatory for users to set an English caption for the column.

3. **Width (%):** This is a numeric updown field that allows users to set the width of the column.
  - By default, the width is set to 100. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
4. **Format:** This field allows users to choose the display format of the column from a dropdown menu.
  - For more information on some of the options in this dropdown, [click here](#).
5. **Alignment:** This field allows users to choose the text alignment of the column from a dropdown menu.
  - Alignment Options: Left, Right, Center, or Not Set.
    - Left: This option aligns the text to the left side of the column.
    - Right: This option aligns the text to the right side of the column.
    - Center: This option centers the text within the column.
    - Not Set: This option aligns the text based on the language or culture setting in the web application.
      - It will typically align left for left-to-right languages and right for right-to-left languages.
6. **Direction:** This field allows users to specify the text direction of the column from a dropdown menu.
  - Direction Options: Left-to-Right, Right-to-Left, or Not Set.
    - Left-to-Right: This option specifies that the text in the column should be read from left to right, in the standard direction for languages that use a Latin-based alphabet (such as English).
    - Right-to-Left: This option specifies that the text in the column should be read from right to left, in the opposite direction to the standard direction for languages that use a Latin-based alphabet (such as Arabic or Hebrew).
    - Not Set: This option specifies that the text direction should be based on the language or culture setting in the web application.
7. **Sortable:** This is a checkbox which, if selected, allows users to sort the column by clicking on its header.
8. **Visible:** This is a checkbox which, if selected, allows for the column to be visible in the grid.
9. **Resizable:** This is a checkbox which, if selected, allows users to resize the width of the column in the grid.
10. **Apply:** This button allows users to save their changes.
11. **OK:** This button allows users to save their changes and close the window.
12. **Cancel:** This button allows users to cancel their changes and close the window.

## 12.1.2.1.11.1.2.2. Edit Type

In the Edit Type window, users can select or modify the display format of the column.

In order to ensure accurate and appropriate display of data, users must select the display format option that corresponds to the attribute type linked to the column.

- If the **Server** option is selected, it will enable the column to display dates in the format used by the server.

✿ Users can set the date format for the server in the Settings of the Designer.

- For more information on the other date display options, please [click here](#) and navigate to the **Display Type** section.
- If the **Dropdown** option is selected, it will enable the adjacent field.

- **Dropdown Data Source:** This field retrieves the list of dropdown data sources available in either the application or system. Depending on the data source chosen in this field, the available options for selection in the dropdown control will vary.

✿ If sourced from within the application, the dropdown data sources in this field will be retrieved from the Dropdown Data Sources section of the **Forms** module. If sourced from the system itself, they will be retrieved from the database directly.

- For more information on the rich textbox display options, please [click here](#).
- For more information on the numeric textbox display options, please [click here](#) and navigate to the **Format** section.



## 12.1.2.1.11.1.2.3. Column Behavior Grid

In the Column Behavior Grid window, users can determine the column state per form behavior. In other words, users can make columns visible, invisible, or disabled at specific stages of a given workflow.

1. **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to main form will be listed in this window.
2. **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
  - **Control States**
    1. Enabled: This option allows for the column to be visible and editable in the web application.
    2. Disabled: This option allows for the column to be visible but not editable in the web application.
    3. Invisible: This option allows for the column be hidden in the web application.



In the above example, the **Frequency** column is enabled at the New Gap Analysis and Modify Gap Analysis stages of the workflow, but is disabled at the Review Gap Analysis stage.


3. **Apply:** This button allows users to save their changes.
4. **OK:** This button allows users to save their changes and close the window.
5. **Cancel:** This button allows users to cancel their changes and close the window.



For more details on this feature and how it can be applied to workflows in the web application, [click here](#).

# 12.1.2.1.11.1.2.4. Column: Color Styles

In the Column: Color Styles window, users can specify conditional formatting rules for the data within the column, such as applying a specific background or text color to cells that meet certain criteria.

 By using color styles, users can visually highlight important data or patterns in the column, making it easier to interpret and analyze the information.

Column Total Sales: color styles

Apply OK Cancel

Empty

Value

Condition

Color94, 94, 94

☐ Requires background color

Background0, 0, 0, 0

Icon

☐ Hide value itself

☐ Icon has own color

Icon color94, 94, 94

+

Add

-

Remove

1. **Value:** This field allows users to input a numeric or text value associated with the color style.

2. **Condition:** This field allows users to set conditions that must be met for the color style to be applied.

- **Comparing Data Between Two Columns**

• Total Sales vs. Price

◦ If Total Sales > Price, display the value in green.
- Products

+ Inline Add

✎ Inline Edit

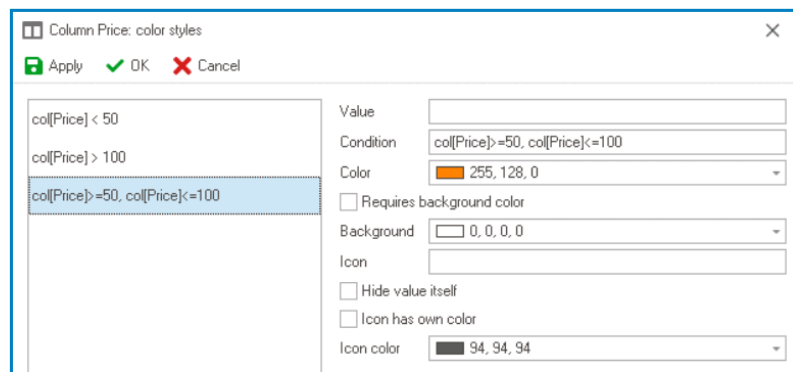
📄 Export to Excel

📄 Excel template

📄 Import from Excel

|   |   | Product ID | Product Name | Category     | Price    | Quantity | Total Sales |
|---|---|------------|--------------|--------------|----------|----------|-------------|
| 1 | 🗑 | P-00001    | Laptop       | Electronics  | 1,000.00 | 20       | 2,000.00    |
| 2 | 🗑 | P-00002    | Desk Lamp    | Home and ... | 50.00    | 10       | 400.00      |
- If Total Sales < Price, display the value in red.
- © 2023 Interfacing Technologies Corporation
- Page 524 of 1144

- If Total Sales = Price, display the value in **orange**.
- Completion Date vs Due Date
  - If Completion Date < / = Due Date, display the value in **green**.
  - If Completion Date > Due Date, display the value in **red**.
- **Comparing Data Within the Same Column**
  - Price
    - If Price < 50, display the value in **green**.
    - If Price > 100, display the value in **red**.
    - If 50 <= Price <= 100, display the value in **orange**.



3. **Color:** This field allows users to select the color they want to use for the color style.
  - It contains three tabs: Custom, Web, and System.
    - In the Custom tab, users can choose a color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
    - In the Web tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
    - In the System tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
4. **Requires Background Color:** This checkbox indicates whether the color style requires a background color to be applied.
  - If this checkbox is selected, users will need to specify a background color before the color style can be applied.
5. **Background:** This field allows users to select the background color they want to use for the color style.
  - It also has three tabs, similar to the **Color** field, where users can select a color from a color

picker or a scrollable list of pre-defined options.

- **Background Color Style Preview: Web View**

- Priority

- If Priority = High, display the value in **red**. If Priority = Medium, display the value in **orange**. If Priority = Low, display the value in **green**.

✿ To apply a background color style, users must first select the **Requires Background Color** checkbox.

6. **Icon:** This field allows users to set an icon to be associated with the color style.

- Users must choose icons from the [Font Awesome](#) library.
- **Icon Color Style Preview: Web View**

! When assigning icons to dropdown options, make sure that the color style value corresponds to the data value established in the [Query Editor](#).

7. **Hide Value Itself:** This checkbox determines whether the value associated with the color style should be displayed in the column.
- If this checkbox is selected, only the icon will be displayed.
  - **Hidden Value Preview: Web View**
8. **Icon Has Own Color:** This checkbox indicates whether the icon associated with the color style should have its own color.
- If this checkbox is selected, users will be able to select a separate color for the icon.
9. **Icon Color:** This field allows users to select the color for the icon associated with the color style.
- It also has three tabs, similar to the **Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.



To apply a background color style, users must first select the **Icon Has Own Color** checkbox.

10. **Add:** This button allows users to add a new color style.
11. **Remove:** This button allows users to remove a selected color style.

### Tips & Tricks for Creating Conditions in Grid Color Styles

- When creating a condition, use the “**value**” statement to reference the value of the column the condition is created on.
- To reference the value of a specific column for a field member, use the “**col[Field]**” statement.
  - This allows you to compare the value of two columns.
- To access the value of the selected option for dropdowns, use “**val[Field]**”.
  - **Note:** “**col[Field]**” or “**value**” will give you the text for the selected option instead.
- To compare values, use “**=**” instead of “**==**”, as it is more common and easier to read.
- To combine multiple conditions, use “**or**” or “**||**” and “**and**” or “**&&**”.
- To style the entire row according to the value of one column, repeat the same condition for each cell.

## 12.1.2.1.11.1.3. Grid – Business Rules for Data Management

In this section of the manual, we'll explore some useful tips and tricks for managing grids using business rules.

### 1. How To Add a New Row in a Grid Linked to a Mid-Entity (Many-to-Many) When the Item Does Not Already Exist

To learn how to add a new row in a grid linked to a mid-entity (or mid-table) when the item does not already exist, refer to the following set of instructions:

- Create the two entities that you want to link together (e.g. **Product** and **Supplier**)
  - For each entity, create the relevant set of attributes.
- Create a mid-entity with the two foreign keys (e.g. **Product\_Supplier**)
- Create a form associated with the mid-entity
  - Inside the form, add all required controls for the attributes of the child entity (e.g. **all attributes of Supplier**)
- Create a pre-form action for the form.
  - In the code, create a new domain object for the referenced child entity.

#### ▪ Code Example

```
if (domainObject.BPMAPP_BCM_Action_ItemFK_ActionItem  null)
{
    var actionItem = (Ray.BPMApp.BCM.Model.BPMAPP_BCM_Action_Item) (bpmAppService.UnitOfWork.CreateDomainObject(
        typeof(Ray.BPMApp.BCM.Model.BPMAPP_BCM_Action_Item)));
    domainObject.BPMAPP_BCM_Action_ItemFK_ActionItem = actionItem;
    if (actionItem.BPMAPP_BCM_BCP_ActionItem_Execute_MidsFK_ActionItem  null)
    {
        actionItem.BPMAPP_BCM_BCP_ActionItem_Execute_MidsFK_ActionItem = new List<Ray.BPMApp.BCM.Model.BPMAPP_BCM_BCP_ActionItem_Execute_Mid>();
    }
    actionItem.BPMAPP_BCM_BCP_ActionItem_Execute_MidsFK_ActionItem.Add(domainObject);
}
```

- Configure the governance properties of the grid control (to which the mid-entity is linked).
  - Enable the control's 'new' functionality and set up the corresponding form behavior ('New Form').
    - This will allow users to open the previously created form directly from the grid.

- \* A grid linked to a mid-entity is a common way to manage many-to-many relationships in a database. Essentially, it allows users to have a grid that displays the relationship between two entities, each represented by a separate table.

## 2. How To Prevent Duplicates in a Grid

To learn how to prevent duplicate entries in a grid, consider the following example:

- In the database, there are two custom entities, **Product** and **Supplier**, which have their own corresponding set of attributes.
  - These two entities are linked together by a mid-entity through foreign keys.
- In the Supplier form, there is a grid linked to the mid-entity, which displays the mid-objects, i.e., the supplier's products.
  - Using the grid, users can add products to the supplier, but each product should only be added once.

- \* If a user attempts to add the same product again, it should be filtered out in the grid's code.

- To achieve this, users must write code to implement the necessary logic.

### ▪ Code Example

```
if(domainObject.BPMAPP_Mid_Supplier_ProductsSupplier
!= null)
{
    var productIds = new List();
    var midToDelete = new List();
    foreach(var mid in domainObject.BPMAPP_Mid_Sup
plier_ProductsSupplier)
    {
        if(productIds.Contains(mid.BPMAPP_Prod
uctProduct.Id))
            midToDelete.Add(mid);
        else
            productIds.Add(mid.BPMAPP_Prod
uctProduct.Id);
    }
    foreach(var mid in midToDelete)
    {
        bpmAppService.BPMSServices.DomainObjec
tService.DeleteDomainObject(mid);
    }
    domainObject.BPMAPP_Mid_Supplier_ProductsSuppl
ier = domainObject.BPMAPP_Mid_Supplier_ProductsSupplie
r.Where(m => !midToDelete.Contains(m)).ToList();
}
```

```
}
```



To see examples of different grid functionalities in action, check out the following applications: [Grid Management – Create Copy on Edit](#) and [Grid Management – Multi-Column Conditions](#).



## 12.1.2.1.11.2. Grid – Control Governance

In the Grid tab of the [Form / Behavior / Control](#) window, users can configure properties that govern the functionality of grid controls.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Grid]

Apply OK Cancel

Control Name **Grid**

Grid Validation




|   |                  |         |  |
|---|------------------|---------|--|
| <input type="checkbox"/> Enable Edit                            | Edit Form        | [Empty] |  |
| <input type="checkbox"/> Enable New                             | New Form         | [Empty] |  |
| <input type="checkbox"/> Enable View                            | View Form        | [Empty] |  |
| <input type="checkbox"/> Enable Delete                          | Delete Form      | [Empty] |  |
| <input type="checkbox"/> Inline New                             | Import Form      | [Empty] |  |
| <input type="checkbox"/> Inline Edit                            | Inline Edit Form | [Empty] |  |
| <input checked="" type="checkbox"/> Enable import from Excel    | Import Form      | [Empty] |  |
| <input checked="" type="checkbox"/> Enable export from Excel    |                  |         |  |
| <input type="checkbox"/> Enable Select (only in selector forms) |                  |         |  |

1. **Enable Edit:** This is a checkbox which, if selected, allows users to enable the **Edit** function for the grid.
  - If this checkbox is selected, it will enable the **Edit Form** field.
    - **Edit Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'edit' functionality.
      - To define a new form behavior, select the button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the button.
      - To remove a form behavior, select the button.

**!** To set up the 'Edit Form', users only need to provide a name and a title.

✿ With **Edit** enabled, users can edit the grid data using a modal window linked to the grid.




2. **Enable New:** This is a checkbox which, if selected, allows users to enable the **Add** function for the grid.

- If this checkbox is selected, it will enable the **New Form** field.
  - **New Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'new' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'New Form', users only need to provide a name, a title, and select the **Starter** checkbox.

✿ With **Add** enabled, users can add data to the grid using a modal window linked to the grid.

3. **Enable View:** This is a checkbox which, if selected, allows users to enable the **View** function for the grid.

- If this checkbox is selected, it will enable the **View Form** field.
  - **View Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.




! To set up the 'View Form', users only need to provide a name and a title.

✿ With **View** enabled, users can view an item in the grid using a modal window linked to the grid.

4. **Enable Delete:** This is a checkbox which, if selected, allows users to enable items to be deleted in the grid.




- If this checkbox is selected, it will enable the **Delete Form** field.
  - **Delete Form:** This is a single-select dropdown field that retrieves the list of form

behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'delete' functionality.

- To define a new form behavior, select the  button.
  - It will display the [Form Behavior Properties](#) window.
- To modify an existing form behavior, select the  button.
- To remove a form behavior, select the  button.

! Unless users need to add custom functionality when deleting, it is not necessary for them to set up the 'Delete Form'. If users do need to add custom functionality when deleting, however, they will need to create the 'Delete Form' and then create a [post-form action](#) and attach it to the selected form behavior. It's important to note that the pre-form actions of the delete form behavior will not be called.




5. **Inline Add:** This is a checkbox which, if selected, allows users to enable the **Inline Add** function for the grid.

- If this checkbox is selected, it will enable the **Import Form** field.
  - **Import Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'inline add' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'Import Form', users only need to provide a name, a title, and select the **Starter** checkbox.




\* With **Inline Add** enabled, users can directly add new information inside the grid without opening a separate modal window.

6. **Inline Edit:** This is a checkbox which, if selected, allows users to enable the **Inline Edit** function for the grid.

- If this checkbox is selected, it will enable the **Inline Edit Form** field.
  - **Inline Edit Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'inline edit' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'Inline Edit Form', users only need to provide a name and a title.

\* With **Inline Edit** enabled, users can directly edit existing information inside the grid without opening a separate modal window.

7. **Enable Import from Excel:** This is a checkbox which, if selected, allows users to enable both the **Excel Template** and **Import from Excel** functions for the grid.
- If this checkbox is selected, it will enable the **Import Form** field.
    - **Import Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'inline add' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.
      - To remove a form behavior, select the  button.

! To set up the 'Import Form', users only need to provide a name, a title, and select the **Starter** checkbox.

\* When this option is enabled, users can download an Excel template that contains the appropriate column headers for the grid, and then populate the template with their data. Users can then upload the completed template into the grid to import the data.

8. **Enable Export from Excel:** This is a checkbox which, if selected, allows users to enable the **Export to Excel** function for the grid.

\* When this option is enabled, users can export the data from the grid into an Excel file.

9. **Enable Select:** This is a checkbox which, if selected, allows users to enable the selection of grid data from another form using a dropdown control.

\* When this option is enabled, users can interact with a grid displayed in a separate form and select items by double-clicking on them. The selected value will then be returned and populated in the associated dropdown control.

10. **Apply:** This button allows users to save their changes.
11. **OK:** This button allows users to save their changes and close the window.
12. **Cancel:** This button allows users to cancel their changes and close the window.



In the Validation tab of this window, users can configure the required field validation properties for the grid control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.11.3. Grid – Web Display

For those unfamiliar with using the grids in the **DBP** web application, the following instructions will guide you through the process:

Users can create grids with various functions that can be combined in any arrangement. There are, however, two commonly used types of grids: a grid with **Add**, **Edit**, and **View** functions, and a grid with **Inline Add** and **Inline Edit** functions.



Depending on the specific needs of the user, these grids can be customized further by enabling additional functions such as **Delete**, **Import from Excel**, **Excel Template**, and **Export from Excel** functions for either of the two grids. For more information on how to enable grid functions, [click here](#).

### 1. Grid with Add / Edit / View Functions

This type of grid enables users to create, modify, and view data by way of the modal forms linked to the **Add**, **Edit**, and **View** buttons.

| Actions |   |           |                |              |          |                   |            |
|---------|---|-----------|----------------|--------------|----------|-------------------|------------|
| + Add   |   | ✎ Edit    |                | 👁 View       |          | 📄 Export to Excel |            |
|         |   | Action ID | Action Name    | Action Owner | Priority | Frequency         | Due Date   |
| 1       | 🗑 | 1         | Data Backup    | John Doe     | High     | Monthly           | 31/03/2023 |
| 2       | 🗑 | 2         | Data Cleanse   | Jane Doe     | Medium   | Quarterly         | 14/04/2023 |
| 3       | 🗑 | 3         | Security Audit | Richard Roe  | Low      | Yearly            | 01/06/2023 |

1. **Add:** This button allows users to add a new item to the grid.
  - If selected, it will open a pop-up window with the 'New Form'.
  - In this form, users can define the required details for the item.
  - At the bottom of the form, users will be given the following options:

Save and Return

Save and New

- **Save and New:** This button allows users to save the item and open a new form to create another item.
- **Save and Return:** This button allows users to save the item and return to the main form.
  - Once they have saved their work and returned to the main form, users will find any new items they have created now added to the grid.

! Users can configure the 'New Form' in the Control Governance tab. The 'New Form' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the grid attributes.

2. **Edit:** This button allows users to edit an existing item in the grid.

- To edit, select the item that you want to edit from the grid and click on the **Edit** button. (Users can also just double-click on the desired item).
  - It will open a pop-up window with the 'Edit Form'. This form is typically a duplicate of the 'New Form', with the exception being that the fields are all populated with previously inputted information.
- To save your changes, click on the **Apply** button at the bottom of the form.
  - Users will find the item in the grid now updated with the latest modifications.

! Users can configure the 'Edit Form' in the Control Governance tab. The 'Edit Form' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the grid attributes.







3. **View:** This button allows users to view an item in the grid.

- To edit, select the item that you want to edit from the grid and click on the **View** button. It will open a pop-up window with the 'View Form'.
  - In viewing mode, users cannot make any modifications.




! Users can configure the 'View Form' in the Control Governance tab. The 'View Form' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the grid attributes.

## 2. Grid with Inline Add / Inline Edit Functions

This type of grid enables users to add and edit data directly in the grid, without opening a modal form.

| Products     |   |               |              |  |                  |                     |             |
|--------------|---|---------------|--------------|--|------------------|---------------------|-------------|
| + Inline Add |   | ✎ Inline Edit |              | ✕ Export to Excel  | ✕ Excel template | ✕ Import from Excel |             |
|              |   | Product ID    | Product Name | Category   | Price            | Quantity            | Total Sales |
| 1            |  | P-00001       | Laptop       |  Electronics    | 1,000.00         | 20                  | 2,000.00    |
| 2            |  | P-00002       | Desk Lamp    |  Home and G...  | 50.00            | 10                  | 400.00      |
| 3            |  | P-00003       | T-Shirt      |  Clothing an... | 25.00            | 50                  | 1,250.00    |


1. **Inline Add:** This button allows users to add a new item into the grid.
  - If selected, it will create a new row in the grid into which users can input data.

| Products |  |            |              |  |                      |                      |                      |
|----------|--|------------|--------------|--|----------------------|----------------------|----------------------|
| ✓ Save   |  | ✗ Cancel   |              | ✕ Export to Excel  | ✕ Excel template     | ✕ Import from Excel  |                      |
|          |  | Product ID | Product Name | Category   | Price                | Quantity             | Total Sales          |
| 1        |   |            |              | <input type="text"/>   | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| 2        |  | P-00001    | Laptop       |  Electronics | 1,000.00             | 20                   | 2,000.00             |

- To save the item, either click on the **Save** button or click anywhere outside the grid.
  - To cancel the item, click on the **Cancel** button.
2. **Inline Edit:** This button allows users to edit an existing item in the grid.
    - To edit inline, select the item that you want to edit from the grid and click on the **Inline Edit** button. (Users can also just double-click on the desired item).
    - All the editable fields for that item will then reappear and users can make, save, or cancel changes as desired.

## Additional Grid Functions

Users can choose to enable **Delete**, **Import from Excel**, **Excel Template**, and **Export from Excel** functions for either of the two grids.

1. **Delete:** This option is represented in the grid by the  button.
  - If this button is selected, users will be alerted of the action with the following confirmation message:






## Confirm

Remove selected row?

**Submit** **Cancel**

- Click on either the **Submit** button to confirm the deletion or the **Cancel** button to cancel it.

2. **Data Extraction / Integration:** These options are represented in the grid by the following buttons:

 Export to Excel    Excel template    Import from Excel

- **Export to Excel:** This button allows users to export the grid data to Excel.
  - If selected, an Excel file containing the grid's contents will be automatically downloaded to the user's device.
- **Excel Template:** This button allows users to generate an Excel template of the grid—that is, an Excel sheet with the grid's column headers.



The primary purpose of this template is to facilitate data entry and streamline the import process. Once it has been downloaded, users can fill the template with the required data and import it into the grid.

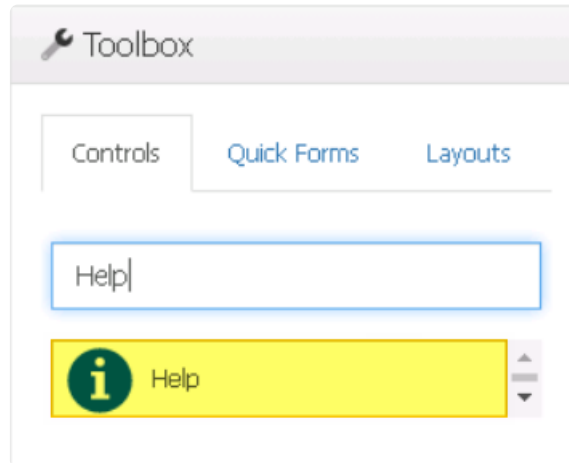
- **Import from Excel:** This button allows users to import data from an Excel sheet into the grid.
  - If selected, a file dialog will appear, enabling users to select the appropriate file from their local system.



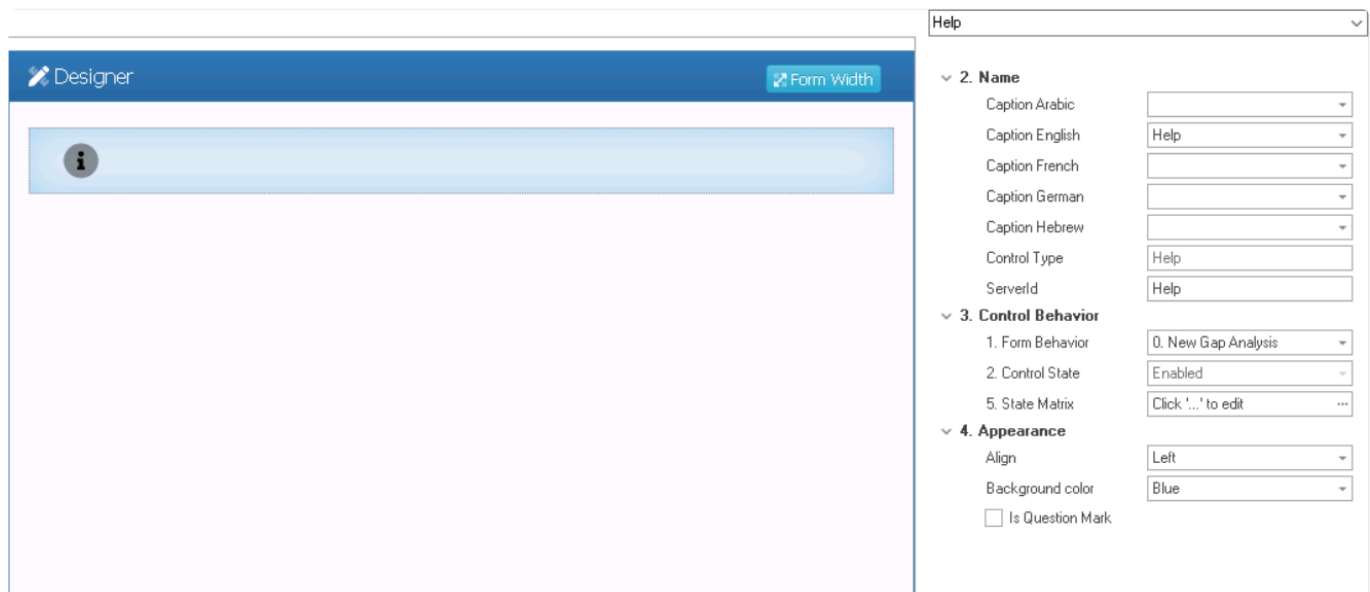
Only the data inside the columns of the worksheet will be imported. If a control is not listed as a column, it cannot be imported into the grid.

## 12.1.2.1.12. Help

The help control is a useful tool for providing contextual guidance to users as they navigate the interface. When users hover their mouse over the control, a message will appear to give them additional information or assistance.



When a help control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

1. **Name:** In this category, users can set the caption that will be displayed when hovering over the control in the web application. They can also access basic properties required for the control's identification and proper referencing in the code.
  - **Caption:** This field allows users to input the text that will be displayed when hovering over the control in the web application.
    - As English is set as the default language in the system, any caption inputted in the

English caption field will be used as the display caption for the control, regardless of other language options that may be available.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Control Type:** This field displays the control type. It cannot be modified.
- **Server ID:** This is a text field that allows users to input the Server ID for the control.
  - Server ID refers to the name used to call the control in the code.
- 2. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state or validation properties set for the control will apply only to the selected form behavior.

✿ Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses ( **...** ) inside the field.



| Form Behavior          | Control State |
|------------------------|---------------|
| 1. Modify Gap Analysis | ✓ Enabled     |
| 2. New Gap Analysis    | ✓ Enabled     |
| 4. Review Gap Analysis | ✓ Enabled     |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.



By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

### 3. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Align:** This is a single-select dropdown field that allows users to choose the alignment of the control.
  - Alignment Options: Right, Center, or Left.
    - Right: This option aligns the control to the right side of the screen.
    - Center: This option aligns the control to the center of the screen.
    - Left: This option aligns the control to the left side of the screen.
- **Background Color:** This is a single-select dropdown field that allows users to choose the background color of the control.
  - Background Colors: Blue, Light Green, Dark Pink, Violet, Yellow, Grass Green, Amethystine, or Grey.
- **Is Question Mark:** This is a checkbox which, if selected, allows the help control's icon to be changed from the default information (  ) symbol to the question mark (  ) symbol.



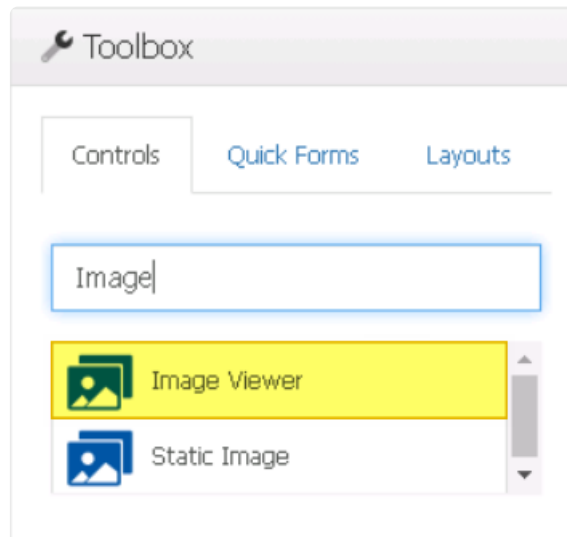
Users do not need to create or link an attribute for help controls.

**Help Control Preview: Web View**



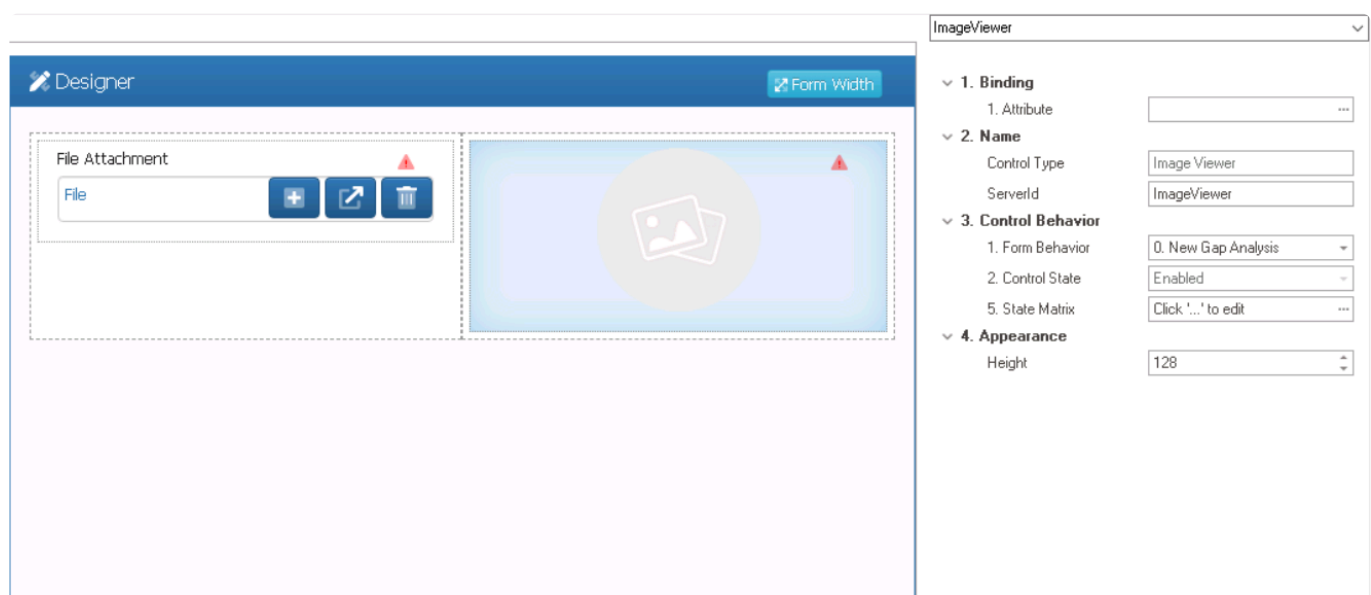
## 12.1.2.1.13. Image Viewer

The image viewer control works in conjunction with the file attachment control to enable users to easily view uploaded images. Once an image is uploaded in the file attachment field, it will appear in the image viewer for display.



\* For more information on the file attachment control, [click here](#).

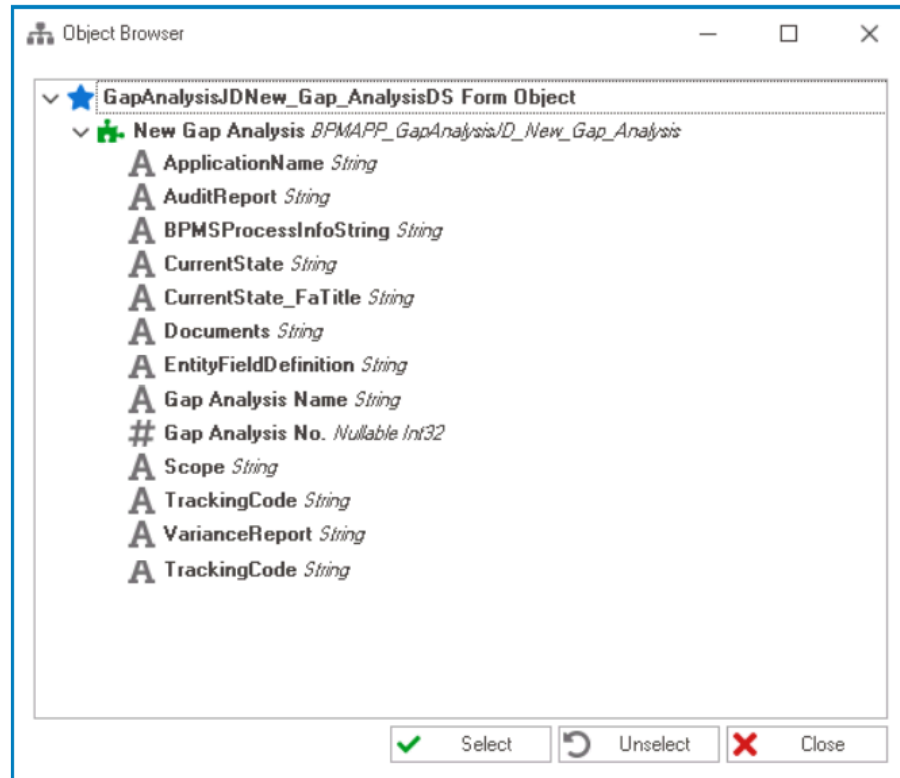
When an image viewer control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



\* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding**: In this category, users can link a control to its corresponding attribute.

- **Attribute:** This field allows users to select the attribute associated to the control.
  - To access the list of relevant attributes, click on the ellipses (...) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

DomainObject.Documents ...



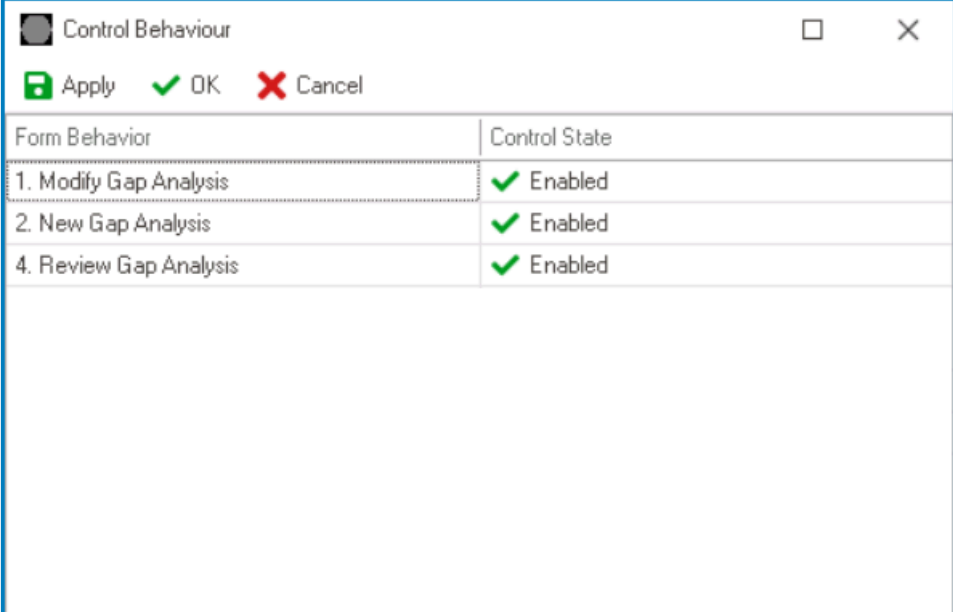
Users must link the image viewer control to the same attribute as the corresponding file attachment control.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.

- Any state or validation properties set for the control will apply only to the selected form behavior.

✿ Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.



| Form Behavior          | Control State |
|------------------------|---------------|
| 1. Modify Gap Analysis | ✓ Enabled     |
| 2. New Gap Analysis    | ✓ Enabled     |
| 4. Review Gap Analysis | ✓ Enabled     |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.

✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.



4. **Appearance:** In this category, users can customize the visual appearance of the control.
- **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 128. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.



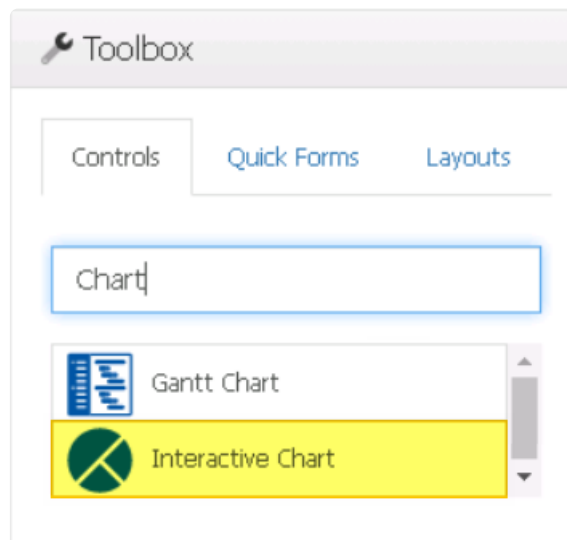
For the image viewer feature to work, users must also select the **Autopostback** checkbox in the control properties of the file attachment control.

## File Attachment Control Configuration

### Image Viewer Control Preview: Web View

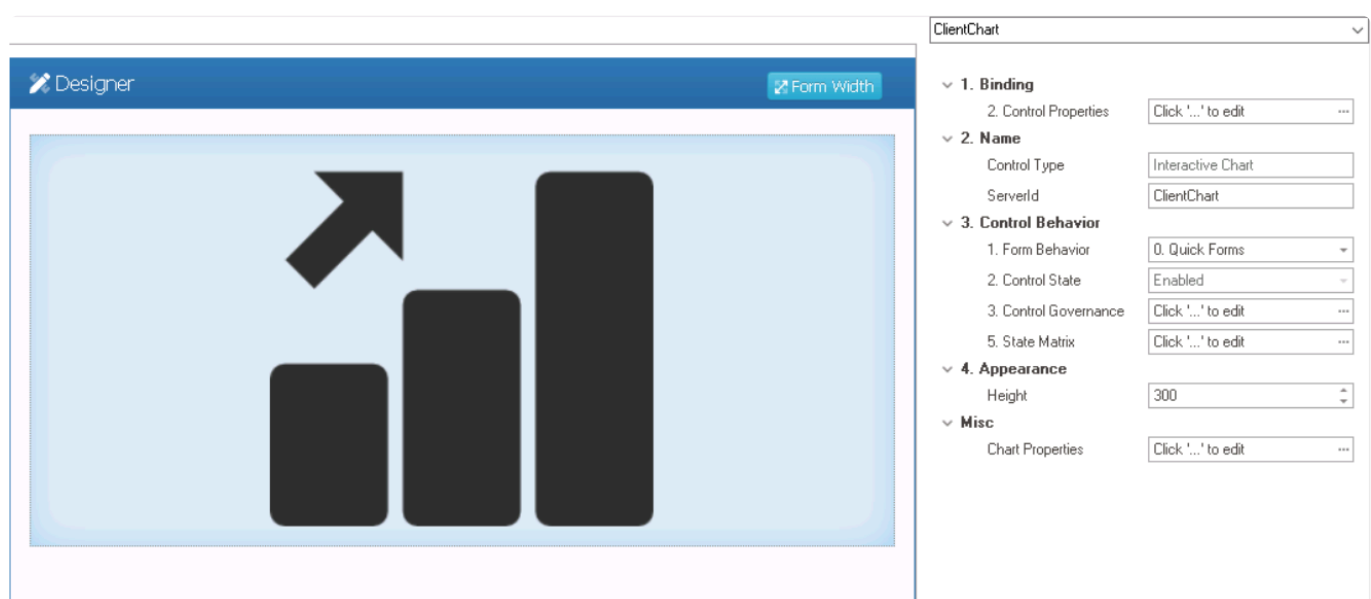
## 12.1.2.1.14. Interactive Chart

Charts, like tables and grids, can be powerful tools for visualizing data in a form. In this section, we will explore the capabilities of interactive charts, one of the two chart controls available in the program. Interactive charts allow users to display and explore data in a dynamic and engaging way. They can be used to create dashboards, which visually display key performance indicators (KPIs) for easy data interpretation and analysis.



\* In the Designer, there are two types of chart controls available: interactive charts and Gantt charts. For more information on Gantt charts, [click here](#).

When an interactive chart control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can define the binding properties of the control.
  - **Control Properties:** This field allows users to configure the control properties.
    - For more information on these properties and how to configure them, [click here](#) and navigate to the relevant section.
2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state properties set for the control will apply only to the selected form behavior.

✿ Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **Control Governance:** This field allows users to configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Chart]

Apply OK Cancel

Control Name **ClientChart**

Chart

☐ + Enable View View Form Behaviour [Empty] + edit -

- **Enable View:** This is a checkbox which, if selected, allows users to enable view mode for the control.
  - If this checkbox is selected, it will enable the **View Form Behavior** field.
- **View Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
  - To define a new form behavior, select the button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing form behavior, select the button.
  - To remove a form behavior, select the button.

✿ By selecting a form behavior in this field, users are essentially enabling the chart control's drilldown functionality. For more information on this feature, [click here](#).

- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.

| Form Behavior          | Control State |                          | View |
|------------------------|---------------|--------------------------|------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | ...  |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> | ...  |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | ...  |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Control Governance:** This field allows users to configure the control governance properties.
  - If selected, it will display the Form / Behavior / Control window (described above).
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.

✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

! In this window, users can also configure the governance properties for the control across multiple form behaviors. To gain an understanding of how these properties can be configured, [click here](#).

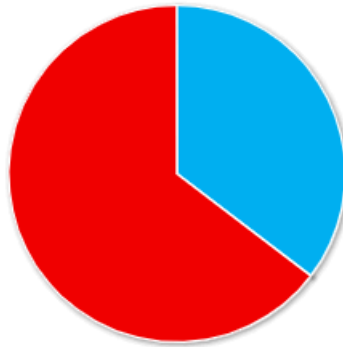
4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 300. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Misc:** In this category, users can define additional properties for the control.
  - **Chart Properties:** This field allows users to configure the chart properties.
    - To access the chart properties, click on the ellipses (...) inside the field.

- It will display the [Chart Editor](#) window.

\* With interactive charts, users can interact with the data on the chart in real-time, such as hovering over data points to view detailed information, selecting different subsets of data to compare, and drilling down to view data at a more detailed level.

#### Interactive Chart Control Preview: Web View

##### NO. OF CAPA ACTION ITEMS ON TIME VS. PAST DUE



Deadline Passed ● Deadline Not Passed ●

\* For more information on how to use this field in the web application, [click here](#).

# 12.1.2.1.14.1. Dashboard with Drilldown Functionality

Charts are a useful tool for creating dashboards, which provide a visual representation of data to help users quickly understand trends and patterns. To create a dashboard with drilldown functionality, users must create three forms: a custom form, a drilldown form, and a main form.

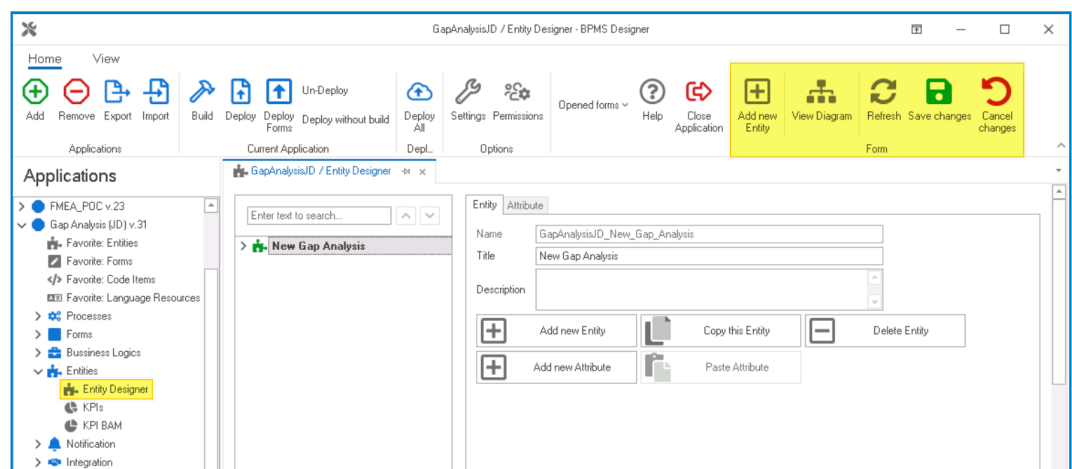
- ✿ The custom form is used as the basis for generating a detailed report in the drilldown form. The drilldown form is used as the basis for generating a summary report in the chart. The main form is where the chart control is implemented and the dashboard is created. Each form has its own set of attributes, which define the data fields and properties associated with it.

By following these steps, users can create a dashboard with drilldown functionality that allows for a more nuanced understanding of their data.

## 1. Create Custom Entity

### • Access Entity Properties

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



Form

**OR**



- In the fields of the **Entity** tab, users can define the properties of the newly created entity.

The screenshot shows the 'GapAnalysisJD / Entity Designer' window. On the left is a sidebar with a search bar 'Enter text to search...' and a tree view containing 'New Gap Analysis'. The main area has two tabs: 'Entity' (selected) and 'Attribute'. Under the 'Entity' tab, there are three text input fields: 'Name' (containing 'GapAnalysisJD - NewEntity'), 'Title', and 'Description'. Below these fields are five buttons arranged in two rows: 'Add new Entity', 'Copy this Entity', and 'Delete Entity' in the top row; 'Add new Attribute' and 'Paste Attribute' in the bottom row.

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

## 2. **Create Custom Attributes**

- **Access Attribute Properties**

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

This image is a close-up of the buttons from the previous screenshot. The 'Add new Attribute' button, which features a plus icon and a document icon, is highlighted with a yellow background. The other buttons are 'Add new Entity', 'Copy this Entity', 'Delete Entity', and 'Paste Attribute'.

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.



The screenshot shows the 'GapAnalysisJD / Entity Designer' application. On the left is a tree view with 'Data Analysis Method', 'NewAttribute Id / Uniquel identifier', and 'New Gap Analysis'. The main area has two tabs: 'Entity' and 'Attribute'. The 'Attribute' tab is active, showing fields for 'Name' (NewAttribute), 'Title', and 'Description'. Below these are radio buttons for 'Data field' (selected) and 'Reference'. Under 'Data field', there's a 'Data Type' dropdown set to 'Uniquel identifier (Globally unique identifier)', and 'Size' and 'Precision' input fields. A section 'Show as a column in the task list' has radio buttons for 'Hide', 'Hidden by default', and 'Show by default' (selected). At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', and three buttons: '+ Add new Attribute', 'Copy Attribute', and '- Delete Attribute'.

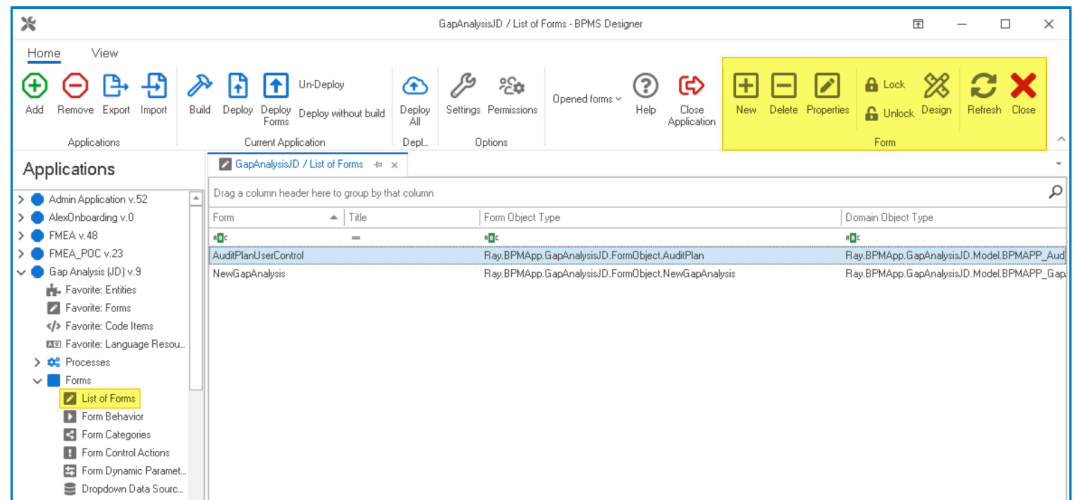
- In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose an attribute type— **Data Field** or **Reference**.
    - For the **Data Field** attribute, select the appropriate data type from the dropdown.
    - For the **Reference** attribute, select the appropriate reference from the System Reference window.
- **Custom Entity Attributes Example**

The screenshot shows a list of actions under a 'v' icon. The list includes: '# ActionID Numeric / Int', 'A ActionName String / NVARCHAR (Max)', 'A ActionItemOwner User', 'A Description String / NVARCHAR (Max)', 'A Frequency String / NVARCHAR (Max)', and 'A StartDate Date/Time / Date'.

! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

### 3. Create & Design Custom Form

- **Access Form Properties**
  - From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

**Form Properties**


Apply OK Cancel

Name:

Form Entity:

Entity:

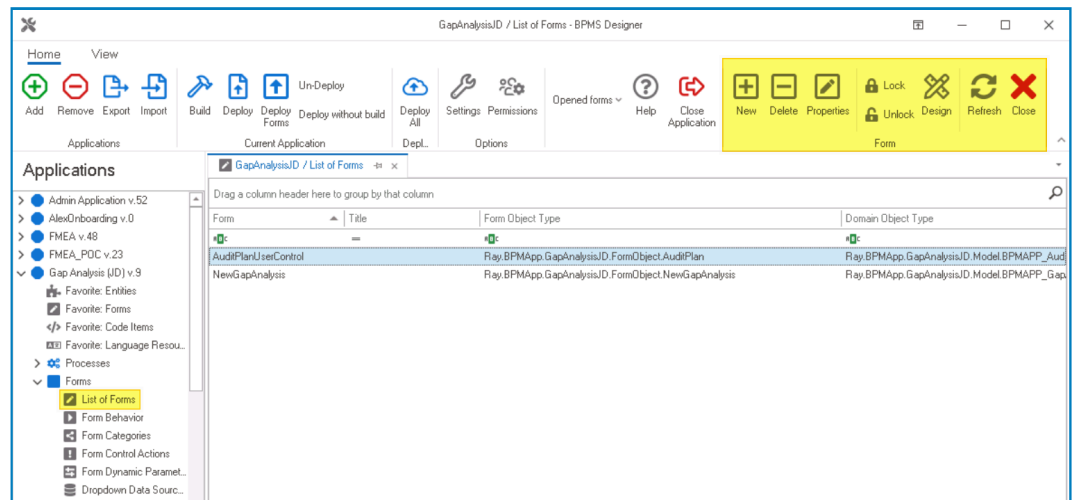
- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

 Design the form by adding the necessary controls (and their corresponding attributes) in a clear and organized layout.

#### 4. Create & Design Drilldown Form

- **Access Form Properties**

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the [Form Properties](#) window.



- In the [Form Properties](#) window, users can create a new form for the application.

**Form Properties**

Apply OK Cancel

Name:

Form Entity:

Entity:

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

✿ A drilldown form typically contains a data grid, which displays all instances of the custom form. **For more information on how to create a data grid and its required attribute, click here.** (Please note, however, that if you have already created the custom form and its attributes, you can skip to the fifth step in the instructions.)

## 5. Configure Grid / Table Data Source

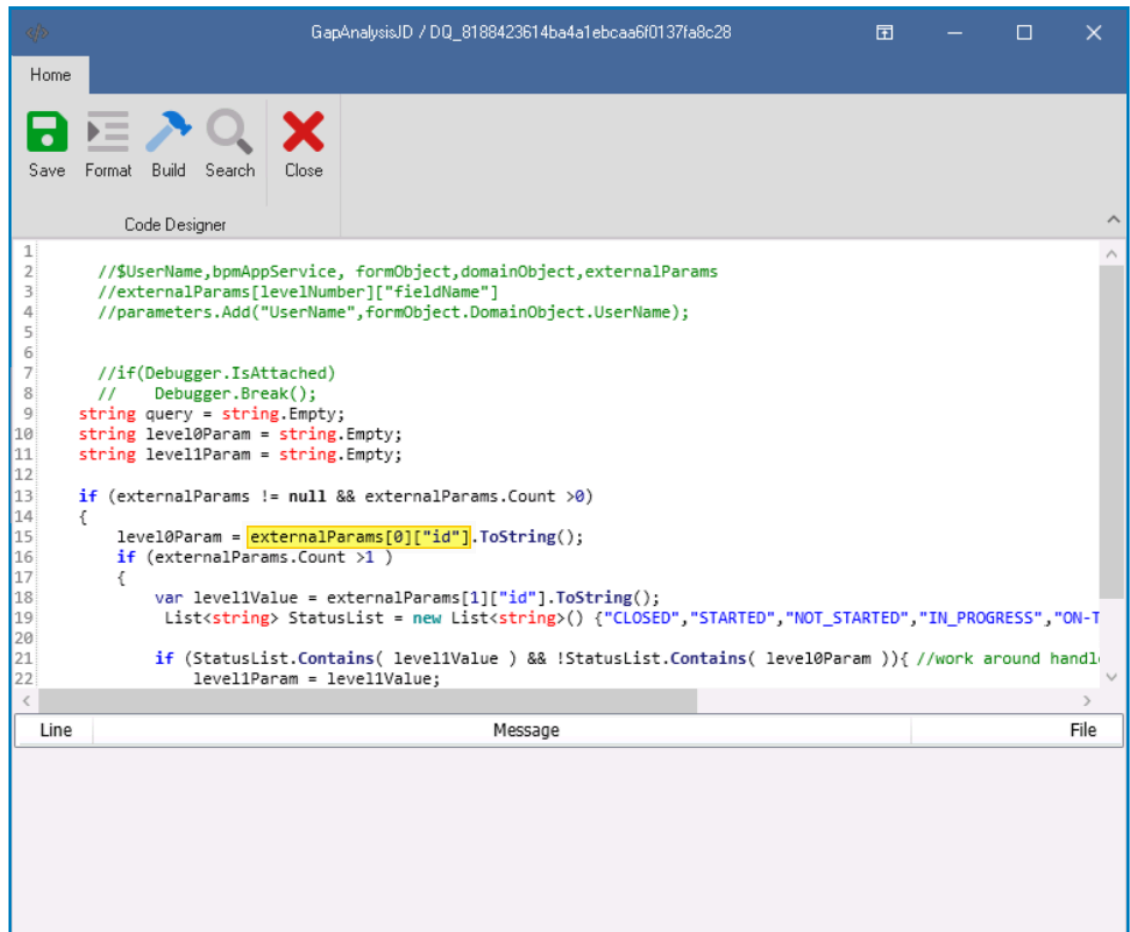
- In the [Query Editor](#) window associated with the grid, perform the following modifications to the grid's data source:
  - To filter the grid's data, add a filter to the data source.

The screenshot shows the 'Query Editor' window with the following configuration:

- Title:** BPMAPP\_QMSDashboard\_QS\_ActionItemsListGrid
- Static:** ☐ (unchecked)
- Dynamic:** ☒ (checked)
- Connection:** BPMS
- DataBase Table:**
  - Select Table:** (empty dropdown)
  - SQL Table:** (empty text box)
  - SQL:** ☐ (unchecked). The text box contains: `FROM BPMAPP_QMS_ActionItem ai $q`
- Buttons:** Execute, Dynamic, </> DQ\_ActionItemsListGrid, </> Edit Code, Designer, Remove
- Indicate if your control is ComboBox:**
  - Value Member:** Id
  - Display Member:** Name

✿ To determine which clause to use for filtering, consider the type of filtration required: use the \$q clause to search for specific text-based data, or use the \$filter clause to filter based on specific values or conditions.

- To add the parameter to the query, launch the [Code Designer](#) and write the corresponding code.

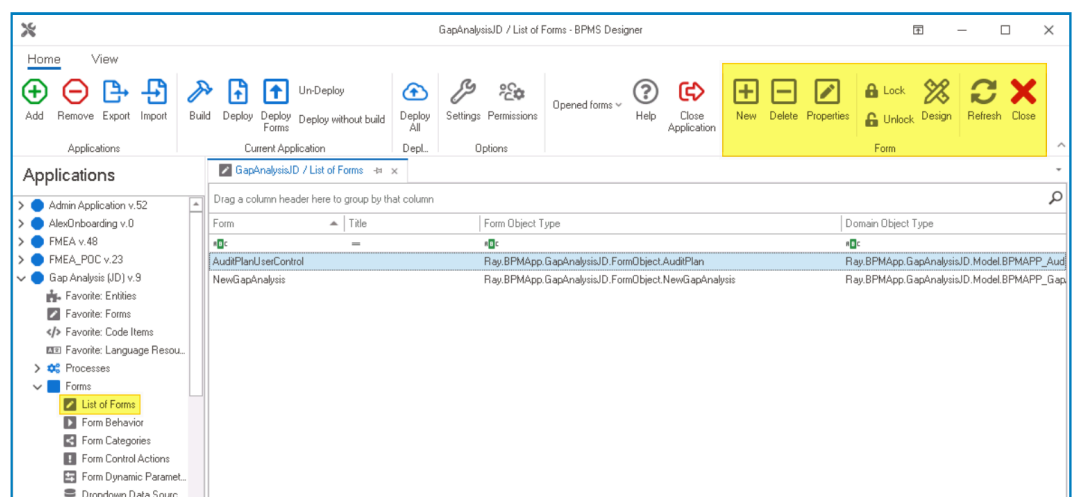


✿ Use the variable “externalParams [0][“id”]” to retrieve the value passed from the chart. Note that you must use “id” as the name of the attribute passed by the chart; this part of the code is invariable.

## 6. [Create Dashboard Form](#)

### • Access Form Properties

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.


The image is a screenshot of a 'Form Properties' dialog box. It has a title bar with a pencil icon and the text 'Form Properties' and a close button (X). Below the title bar are three buttons: 'Apply' (with a floppy disk icon), 'OK' (with a green checkmark), and 'Cancel' (with a red X). The main area contains three fields: 'Name' with a text input field; 'Form Entity' with a text input field; and 'Entity' with a dropdown menu showing '[Please Select...]' and a downward arrow.

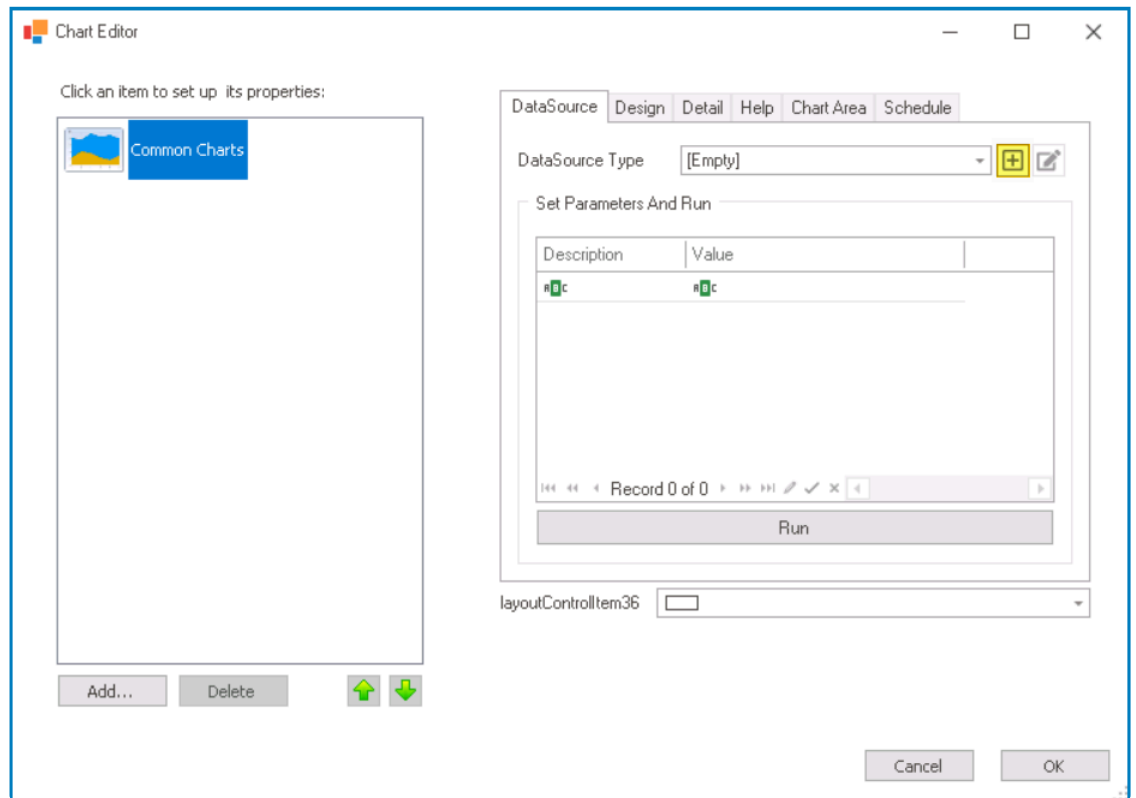
- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

✿ Once they have launched the Form Designer, users should drag the interactive chart control from the toolbox and drop it onto the design environment.

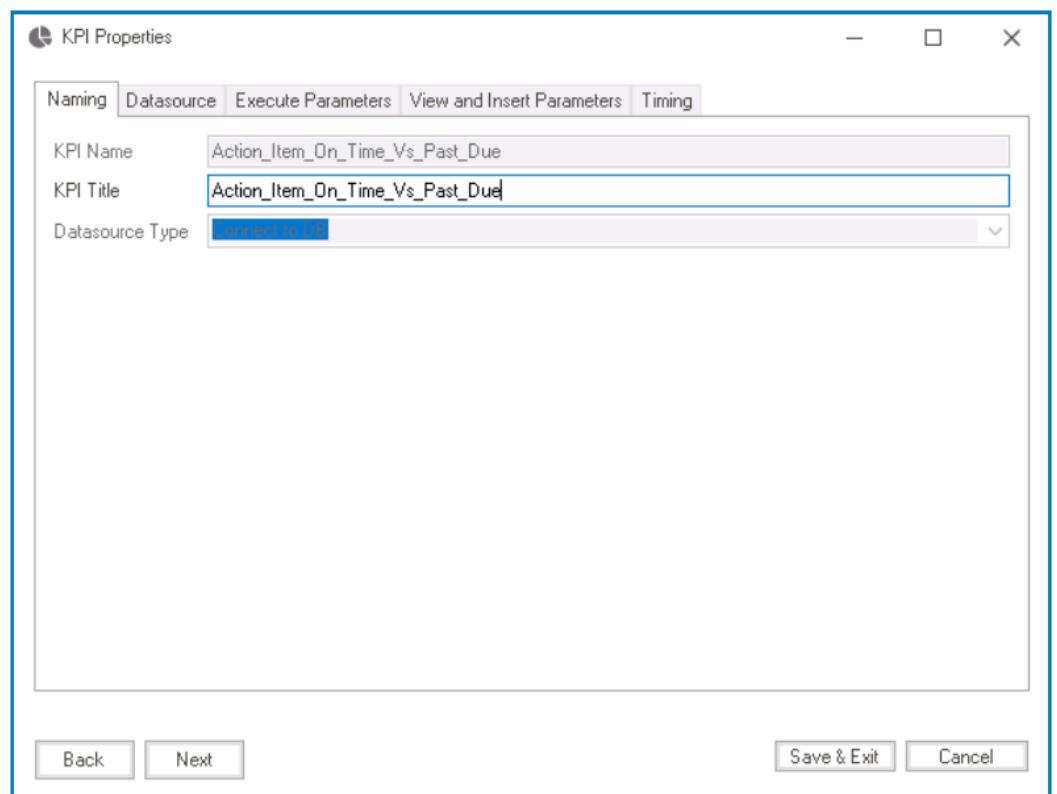
## 7. Configure Chart Properties

### a. **Create Data Source**

- To create a data source for the chart, click on the  button. It will display the KPI Properties window.



- In the Naming tab, provide the KPI name, title, and data source type.



- In the DataSource tab, customize the query using C# code.

KPI Properties

Naming Datasource Execute Parameters View and Insert Parameters Timing

Database WebService

Connection BPMS

Select Table

Select Command Count Command

```
WITH ActionItemData(ActionItemStatusValue, ActionItemStatusLabel) AS ( SELECT
ActionItemStatusValue = (CASE WHEN DateCompleted <= DueDate THEN 'ON-TIME' ELSE 'PAST-DUE'
END), ActionItemStatusLabel = (CASE WHEN DateCompleted <= DueDate THEN 'On Time' ELSE 'Past
Due Date' END) FROM ray.BPMAPP_QMS_ActionItem where IsCompleted = 1 $q ) SELECT
ActionItemStatusValue, ActionItemStatusLabel, count(*) AS ActionItemCount FROM ActionItemData
GROUP BY ActionItemStatusValue, ActionItemStatusLabel
```

DQ\_fc69a2dc19554f2cb9c7a71af7d3cf79 Edit Code Designer Remove

Back Next Save & Exit Cancel

1. In the command box, write the query.
  2. In the Code Designer, write the corresponding code.
- In the Execute Parameters tab, execute the query.

KPI Properties

Naming Datasource Execute Parameters View and Insert Parameters Timing

Enter Parameter Name

```
SELECT * FROM ${DB}.ray.Students where Id = '${Id}'
1- ${DB}
2- $Id
```

Drag a column header here to group by that column

| Description | Value |
|-------------|-------|
| \$q         | \$q   |

Record 1 of 1

Execute

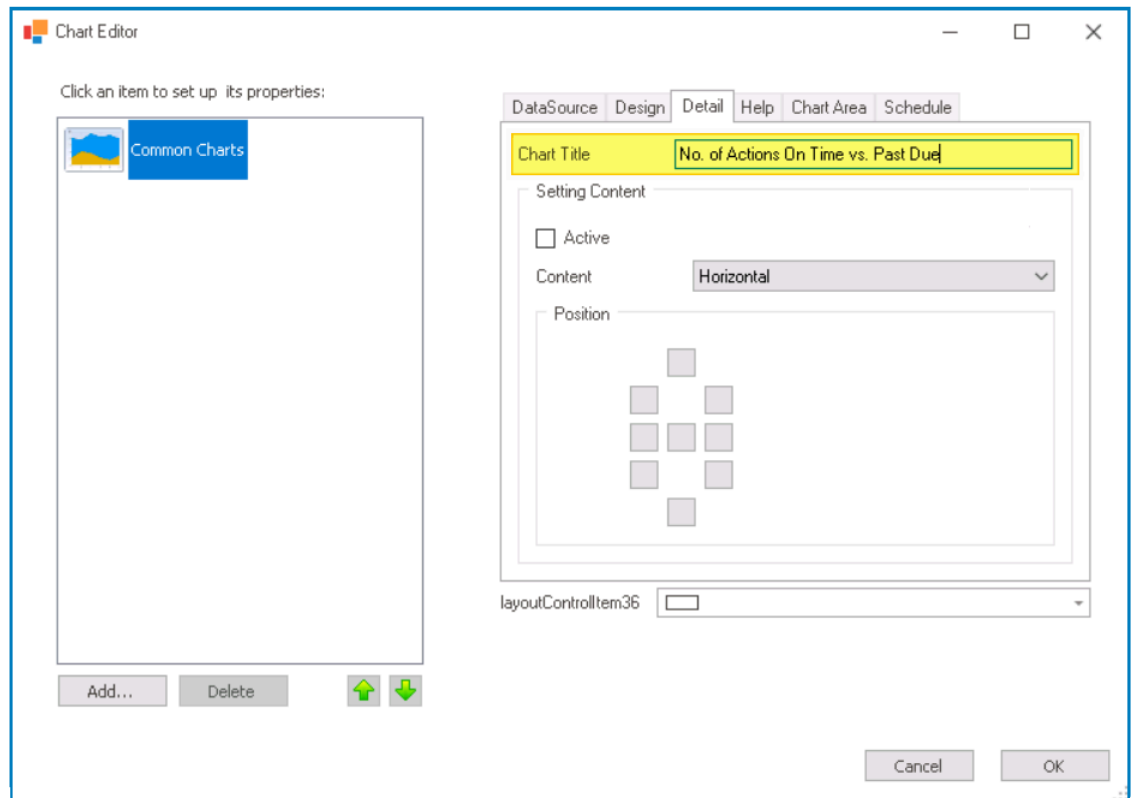
Back Next Save & Exit Cancel

- Return to the Chart Editor window by clicking on the **Save & Exit** button.

#### b. Add Chart Title

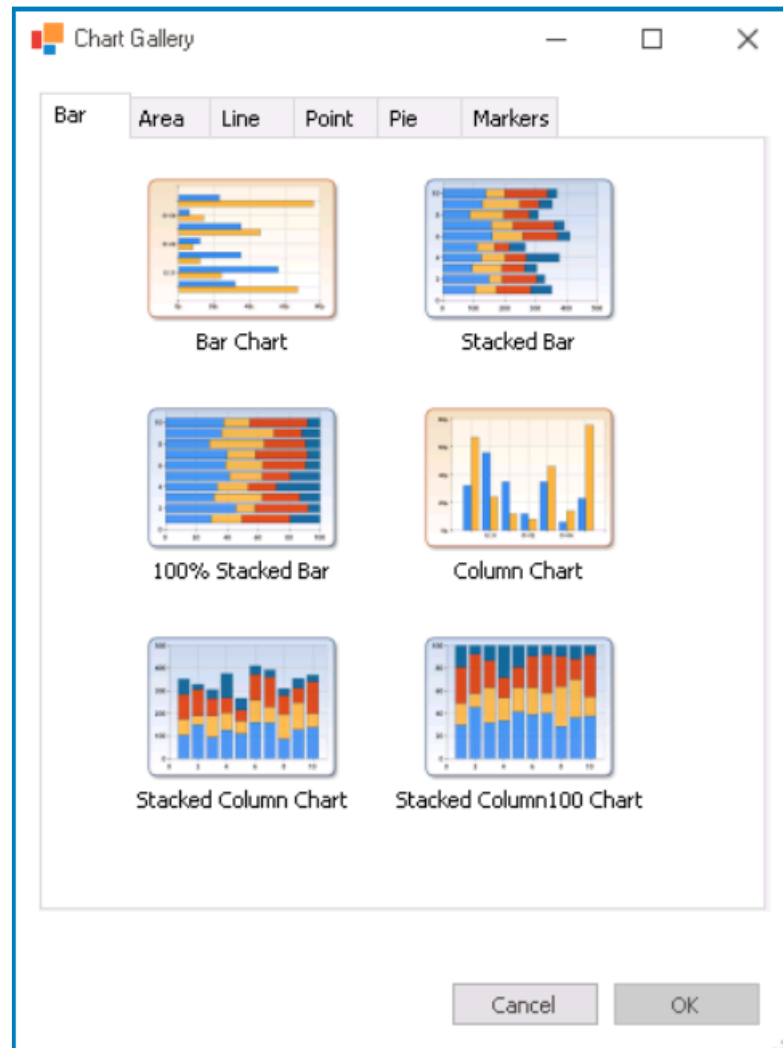
- In the Detail tab, add a title for the chart.





c. **Add New Chart Type / Data Series**

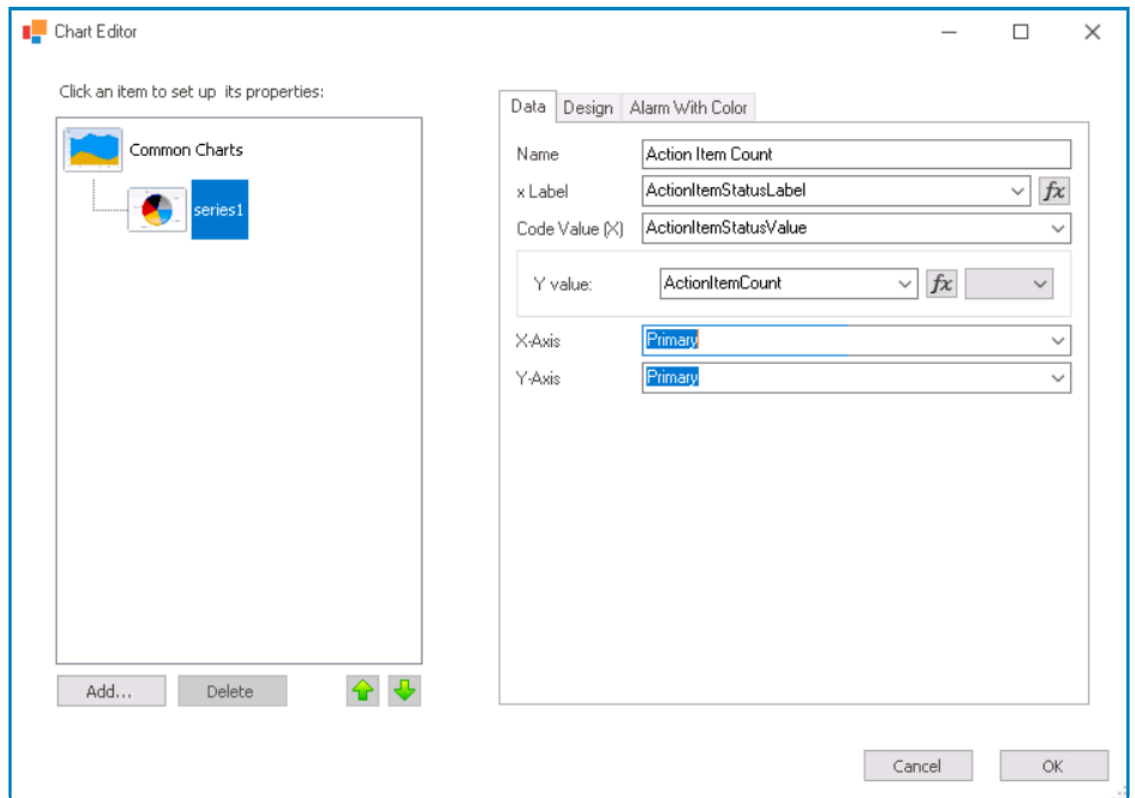
- To add a new chart type, click on the **Add** button. It will display the Chart Gallery window, from users can select the desired chart.



- Select the chart type from the window and click on the **OK** button. It will be listed as a new series under **Common Charts**.

d. **Configure Series Properties**

- In the Data tab, provide the labels and values for the chart elements.

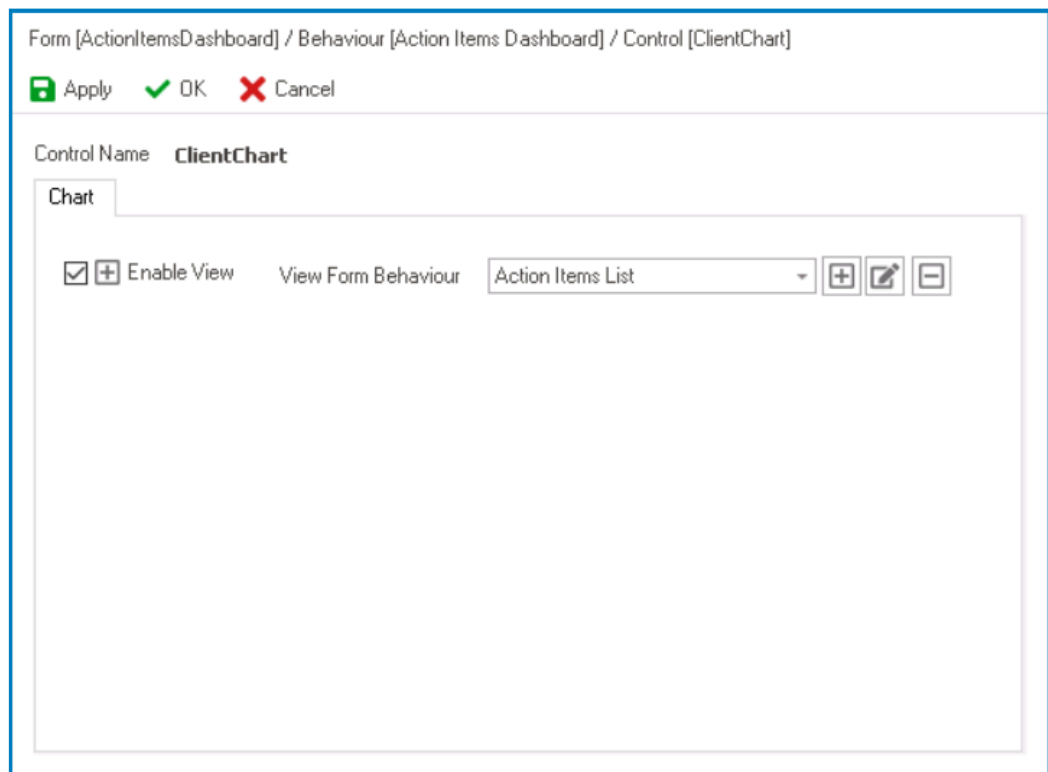


e. **Save Chart Properties**

- To save changes and exit the Chart Editor window, click on the **OK** button.

f. **Configure Control Governance Properties**

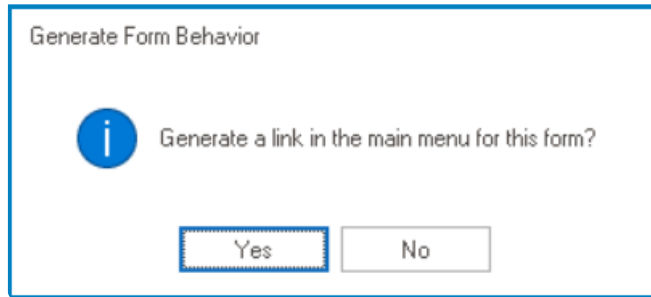
- To access the control governance properties, click on the ellipses (...) inside the field.




- Select the **Enable View** checkbox.
- Select and/or create the appropriate form behavior.

## 8. **Save Dashboard Form & Deploy Application**

- Save the form and select **Yes** in response to the following prompt:



Generate Form Behavior

 Generate a link in the main menu for this form?

- Deploy the application to the web.



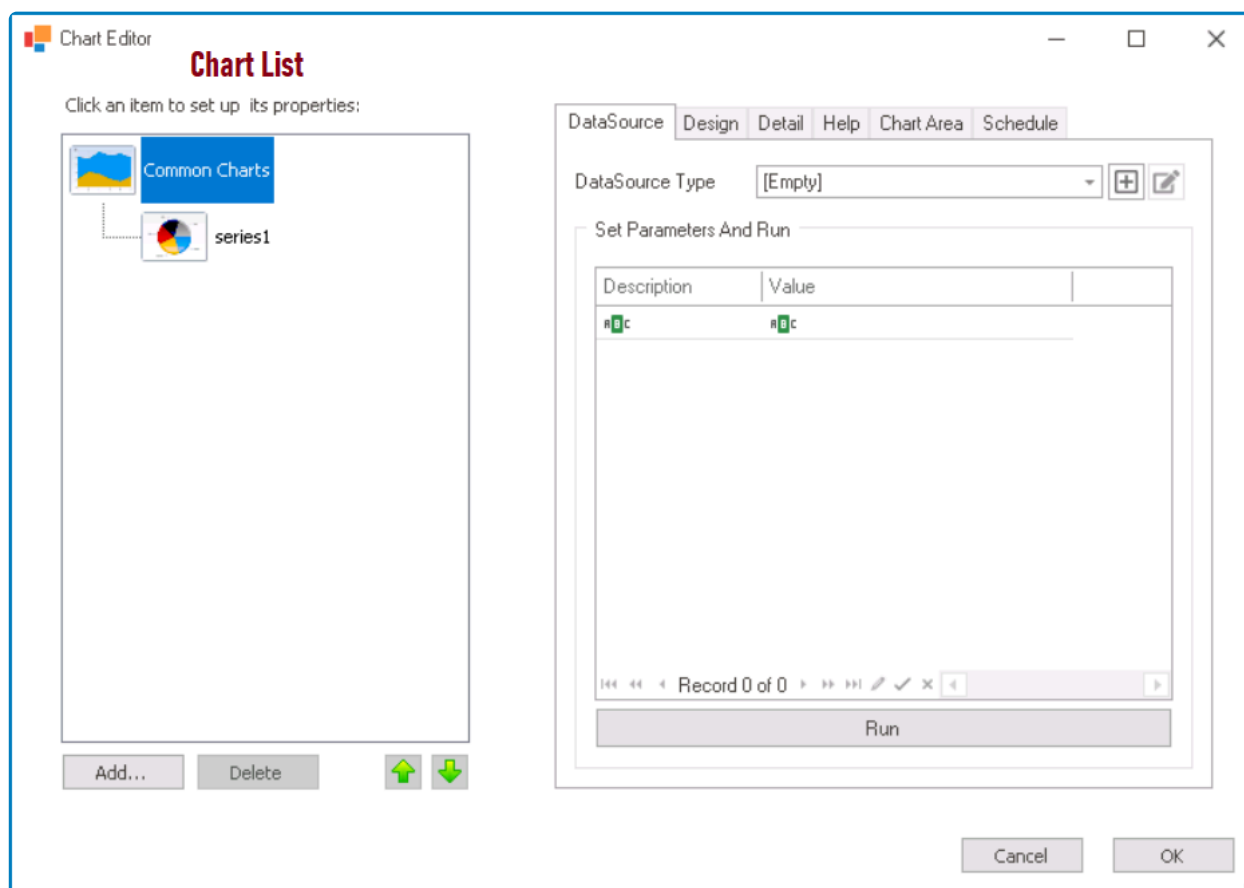
For more information on the functionality of interactive charts with drilldown capability in the web application, [click here](#).

## 12.1.2.1.14.2. Chart Editor

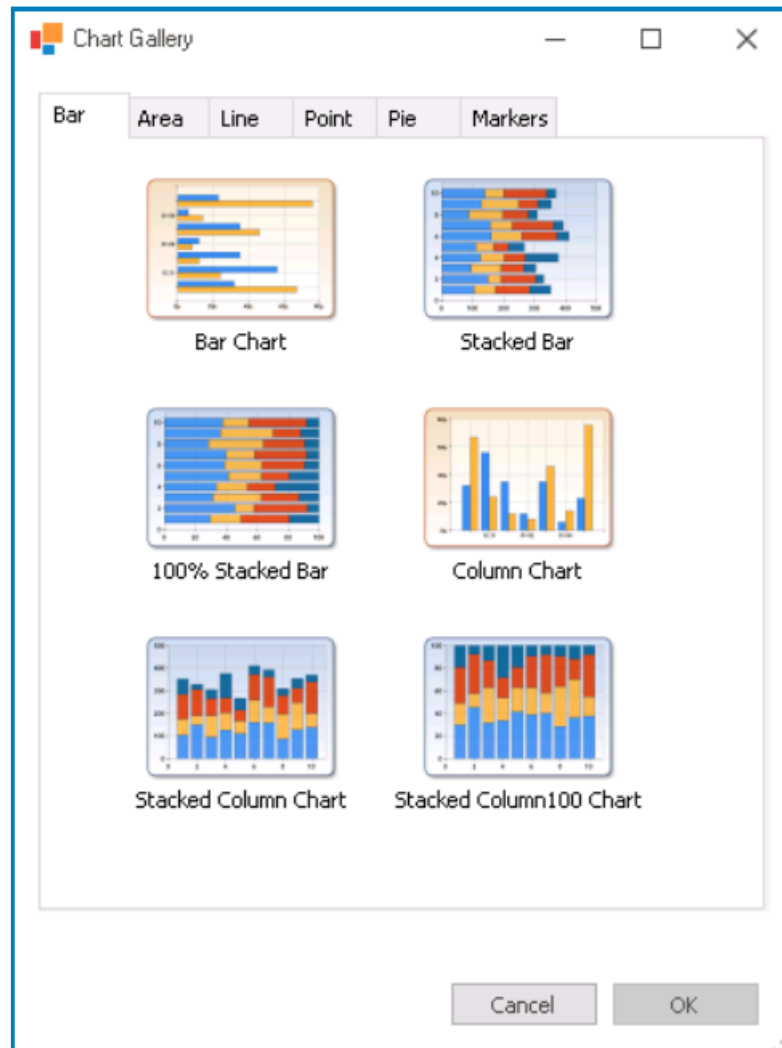
In the Chart Editor, users can configure the chart properties. This window is divided into two sections: Chart List and Chart & Series Properties.



### Chart List

In this section of the window, users can access options for adding and deleting charts.



1. **Add:** This button allows users to add a new chart type (or data series).
  - It will display the Chart Gallery window, in which users will find from a variety of pre-configured chart types to choose from.



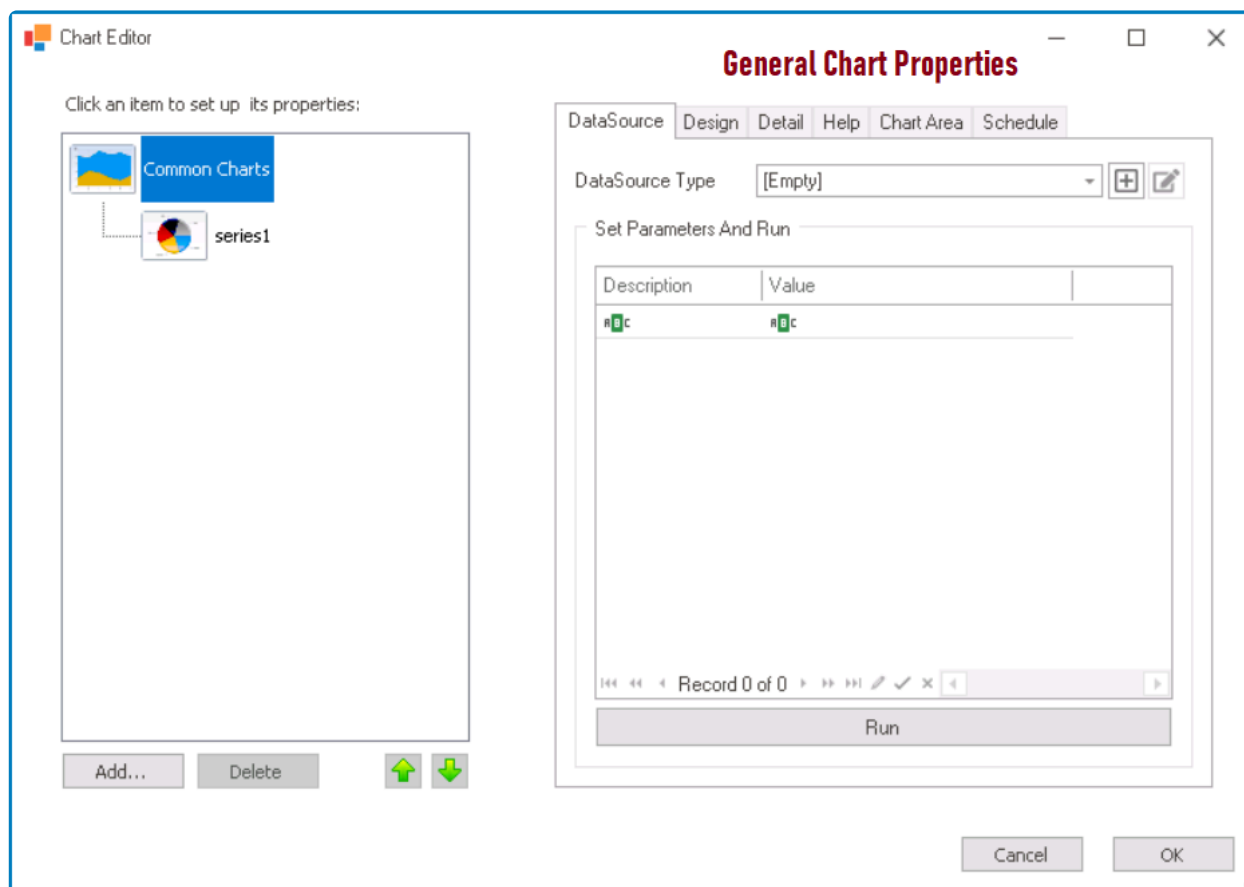
- To add a chart type, select it from the window and click on the **OK** button.
- Once a chart type is added, it will be listed as a new series under **Common Charts**.
- 2. **Delete:** This button allows user to delete a selected chart type (or data series).
- 3. **Move Up / Move Down:** These buttons allow users to reorder the charts in the list.
  - To move a chart up or down, select it from the list and click on the  or  button. This will move the selected chart one position up or down in the list, respectively.

## Chart & Series Properties

In this section of the window, users can access two sets of properties:

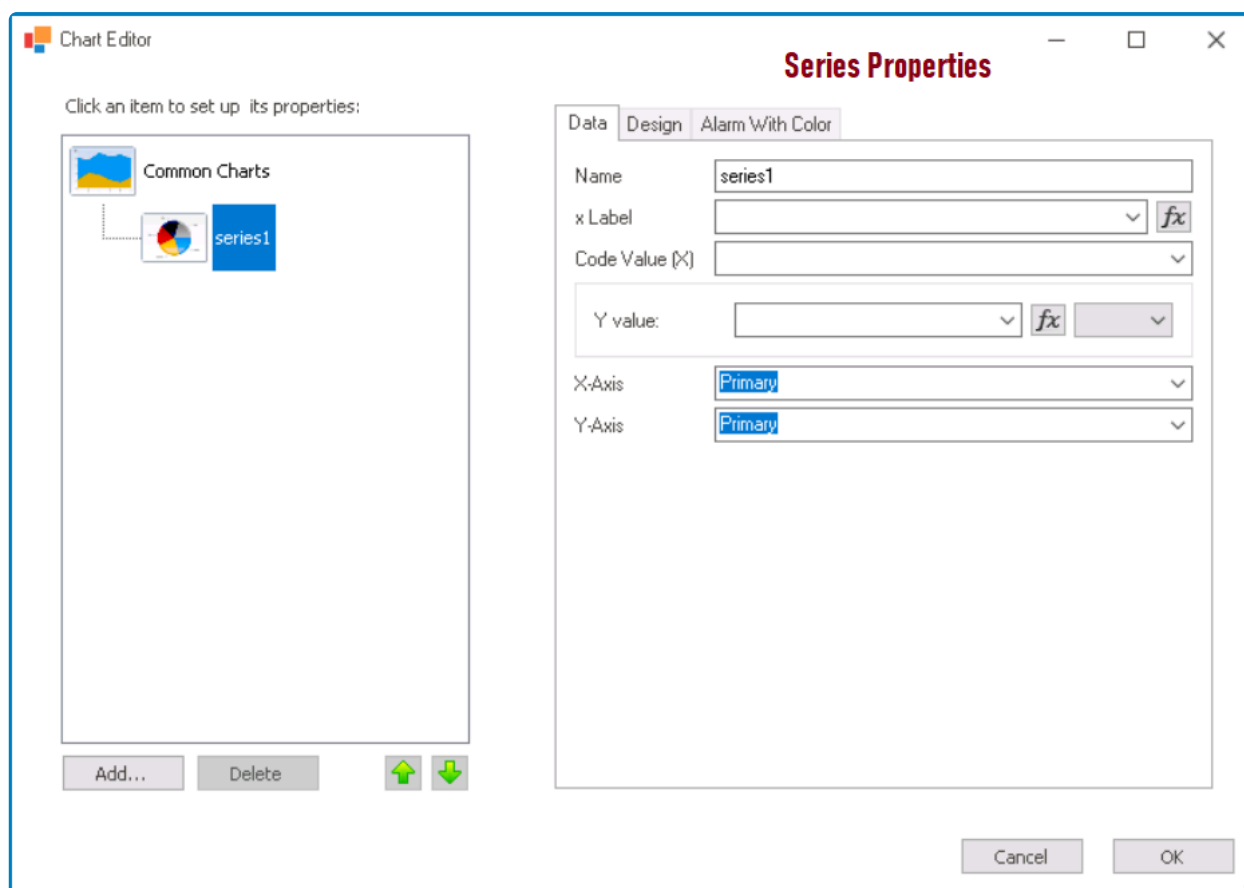
1. [General Chart Properties](#)

To access the general chart properties, select **Common Charts** from the list.



## 2. [Series Properties](#)

To access the data series properties, select a series from the list.



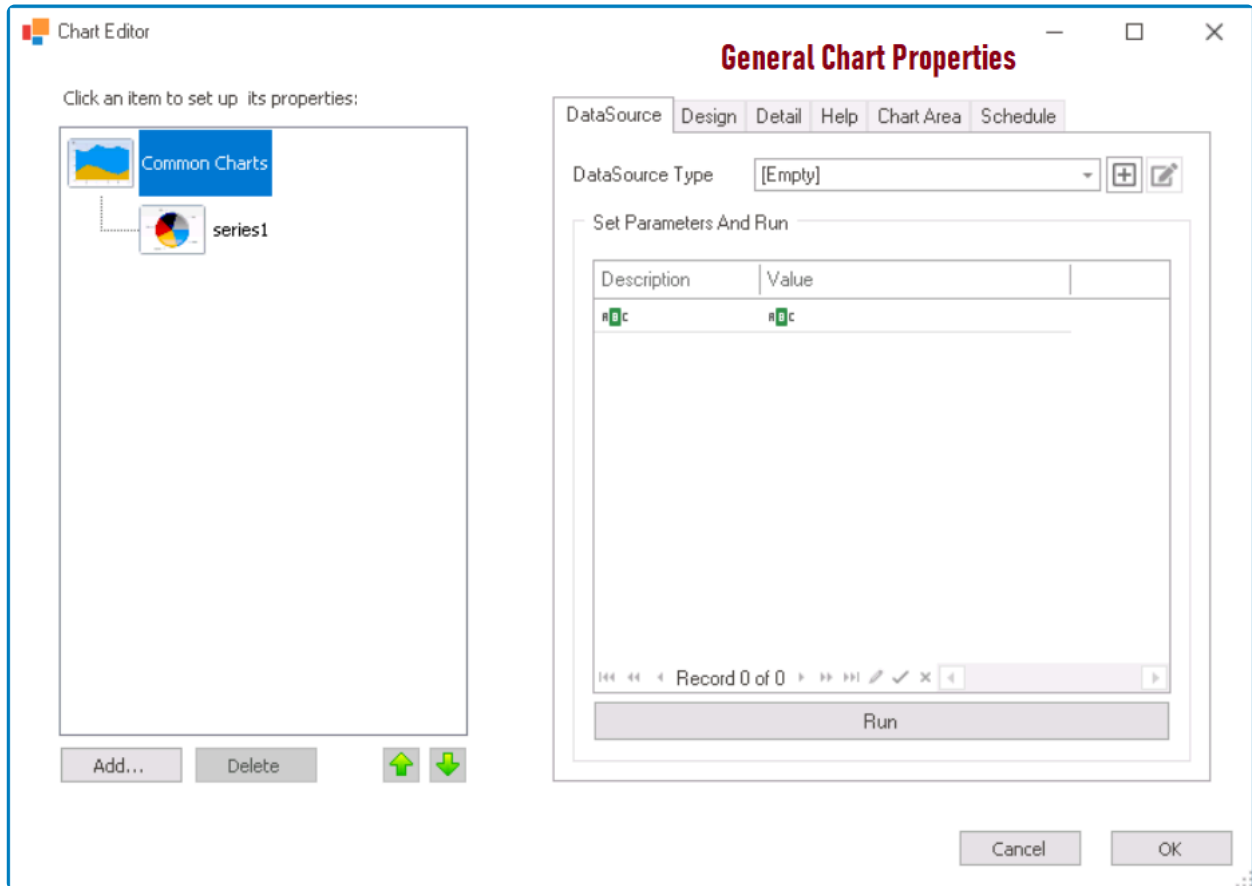


To save changes and exit the Chart Editor window, click on the **OK** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button.



## 12.1.2.1.14.2.1. General Chart Properties

Selecting **Common Charts** will display the general chart properties on the left side of the window.



These properties are displayed in a set of tabs:

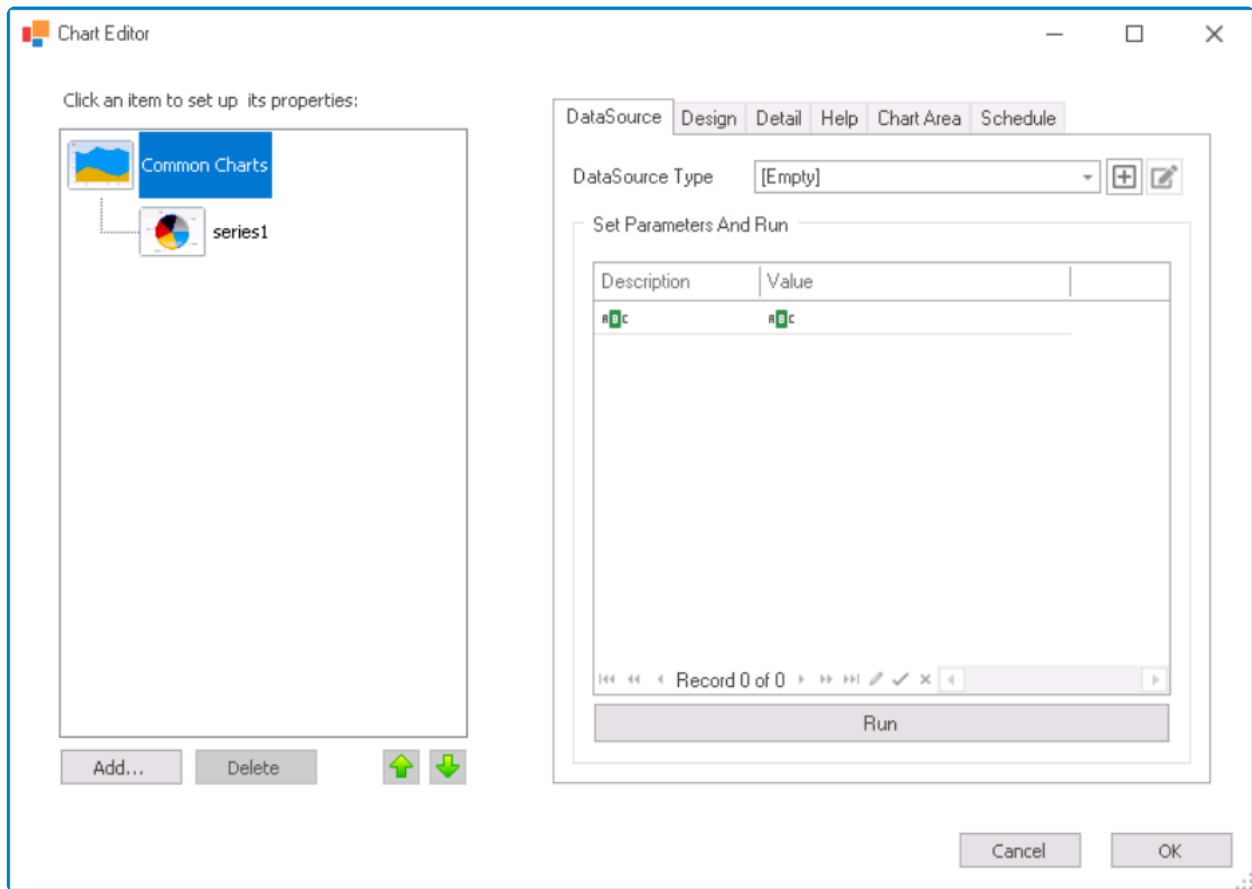
1. [DataSource](#)
2. [Design](#)
3. [Detail](#)
4. [Help](#)
5. [Chart Area](#)
6. [Schedule](#)





To save changes and exit the [Chart Editor](#) window, click on the **OK** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 12.1.2.1.14.2.1.1. DataSource

In the DataSource tab, users can select and configure the data source for the chart, including any necessary query parameters.



1. **Data Source:** This field retrieves the list of data sources (from both the [Dropdown Data Sources](#) section of the **Forms** module and the [KPIs](#) section of the **Entities** module). It allows users to select the data source for the chart.
  - To define a new data source, select the  button.
    - It will display the [KPI Properties](#) window.
  - To modify an existing data source, select the  button.



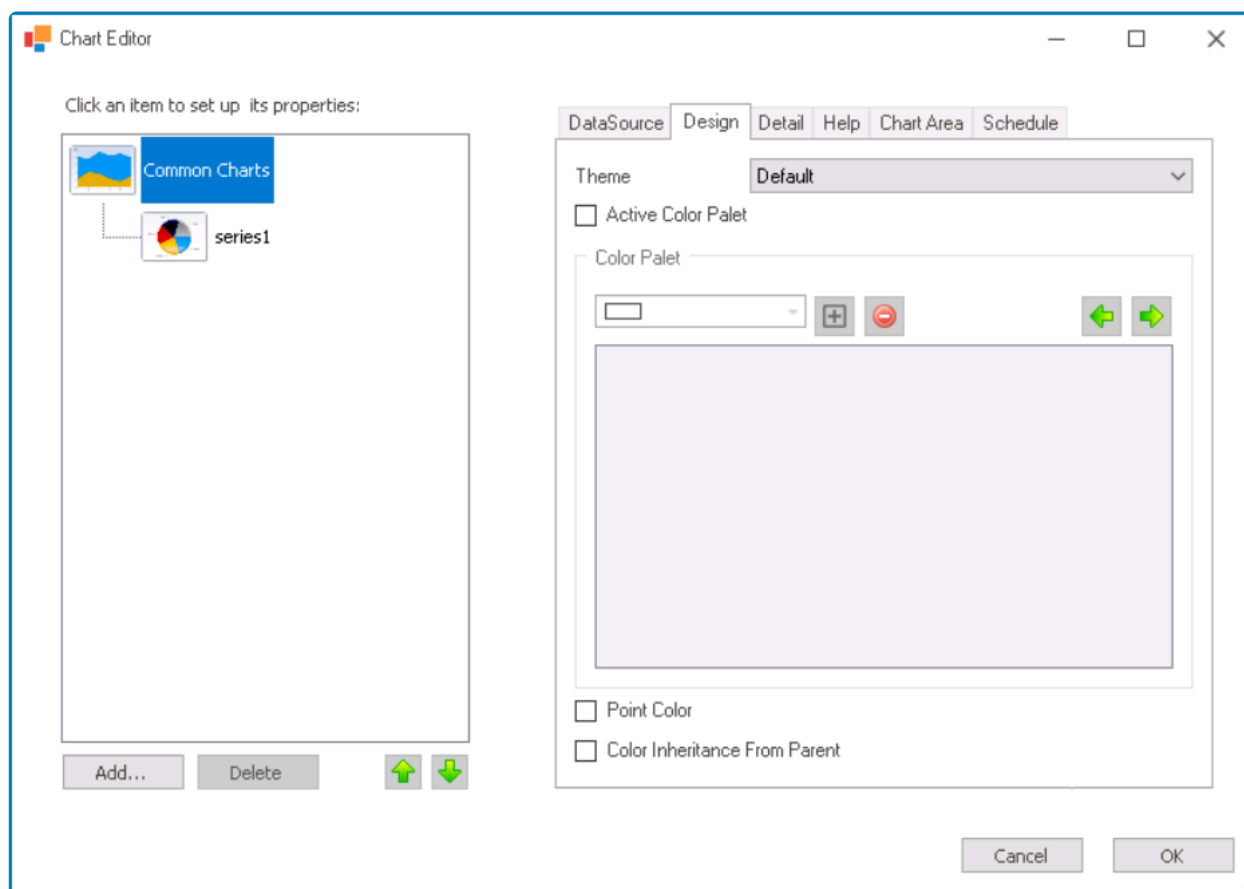
The selected data source will provide the data for the chart.


2. **Set Parameters and Run:** In this section, users can set parameters for the query and execute it with the selected data source.
  - Parameters are values that are used to filter or modify the data returned by the query.
    - If the selected data source has a parametric query (i.e., a query that contains parameter placeholders), users can enter the parameter names in the [Description](#) column and the corresponding values for the parameters in the [Value](#) column.
  - **Run:** This button allows users to run or execute the parametric query with the selected data source and the specified parameter values.
    - When the query is executed, it uses the provided parameter values to filter or modify

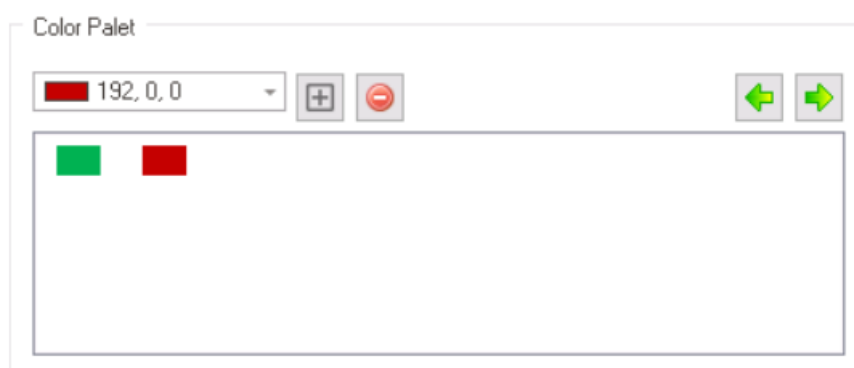
the data from the selected data source.


## 12.1.2.1.14.2.1.2. Design



In the Design tab, users can customize the visual appearance of their chart.



1. **Theme:** This is single-select dropdown field that displays a list of pre-defined color schemes. It allows users to choose a color scheme for the chart.
2. **Active Color Palette:** This is a checkbox which, if selected, allows users to apply a color palette to the chart.
3. **Color Palette:** In this section, users can manage and customize the color palette for the chart.
  - To add a new color to the palette, first select it from the color picker and then click on the  button.
  - Once selected and added, colors are displayed in a box inside the section, which represents the current color palette being used by the chart.



- To remove a color from the palette, select it from the box and click on the  button.

- To rearrange the order of a color, select it from the box and click on the  or  button.  
This will move the selected color one position to the right or the left, respectively.
- 4. **Point Color:** This is a checkbox which, if selected, allows users to apply the selected color palette to the individual data points on the chart.
- 5. **Color Inheritance from Parent:** This is a checkbox which, if selected, allows the child elements of the chart (such as data points, legends, and labels) to inherit their color from the parent element (the chart).
  - If this checkbox is selected, it will disable the **Active Color Palette** checkbox.

#### Interactive Chart Color Palette Preview: Web View

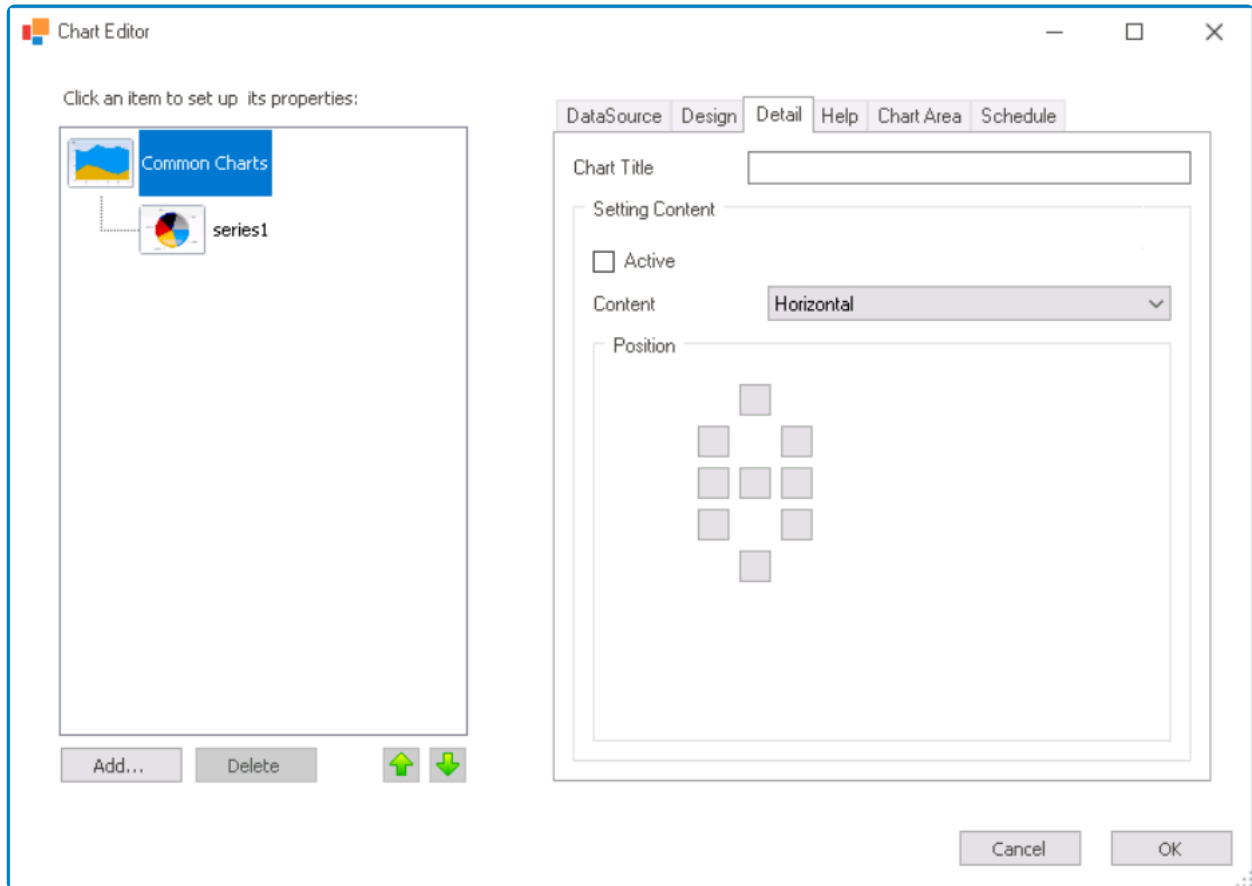
##### NO. OF AUDITS ON TIME VS. PAST DUE



PAST-DUE  ON-TIME 

## 12.1.2.1.14.2.1.3. Detail

In the Detail tab, users can customize various aspects of the chart's appearance and layout.

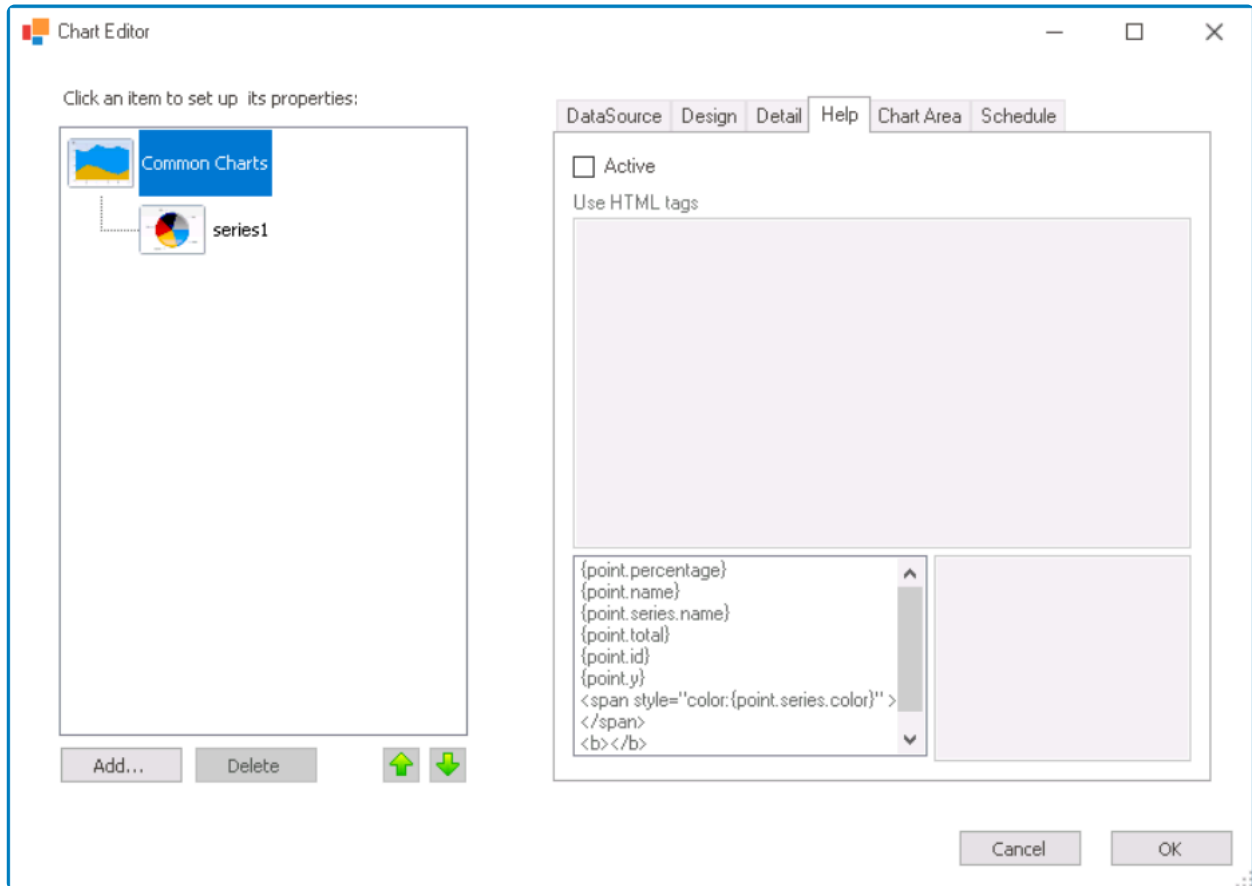


1. **Chart Title:** This is a text field that allows users to input the title of the chart.
2. **Setting Content:** In this section, users can customize various aspects of the chart's legend, including its visibility, orientation, and position relative to the chart.
  - **Active:** This is a checkbox which, if selected, allows users to display the legend on the chart.
  - **Content:** This is a single-select dropdown that displays the list of content orientations. It allows users to select the orientation of the legend items.
    - Orientation Options: Horizontal or Vertical.
      - Horizontal: This option allows users to display the legend items in a single line.
      - Vertical: This option allows users to display the legend items in a column.
  - **Position:** This field allows users to select the position of the legend relative to the chart.
    - The options include above chart, below chart, left of chart, right of chart, and center of chart.

### Interactive Chart Legend Customization: Web View

## 12.1.2.1.14.2.1.4. Help

In the Help tab, users can customize the chart's labels and tooltips using HTML tags.



1. **Active:** This is a checkbox which, if selected, allows users to enable the use of HTML tags in the chart's labels and tooltips.
  - If this checkbox is selected, it will enable the **Use HTML Tags** field.
2. **Use HTML Tags:** This field allows users to customize the text of the chart's labels and tooltips using HTML tags.
  - Users can quickly add formatting and content to their chart labels and tooltips by choosing a tag from the scrollable list box displayed below this field.
    - When a tag is selected from this list, it will automatically populate the **Use HTML Tags** field with the appropriate syntax. Users can then customize the tag by modifying the text within the tags or adding additional tags to the text field.

Use HTML tags

{point.percentage}

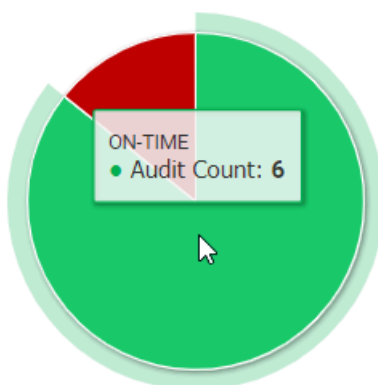
{point.percentage}  
{point.name}  
{point.series.name}  
{point.total}  
{point.id}  
{point.y}  
<span style="color:{point.series.color}">  
</span>  
<b></b>

Percentage at each point. This value is for  
Pie and Stack charts.

- The selected tag's information is displayed in the adjacent box field, providing guidance on how to use it.

### Interactive Chart Labels and Tooltips Preview: Web View

#### NO. OF AUDITS ON TIME VS. PAST DUE

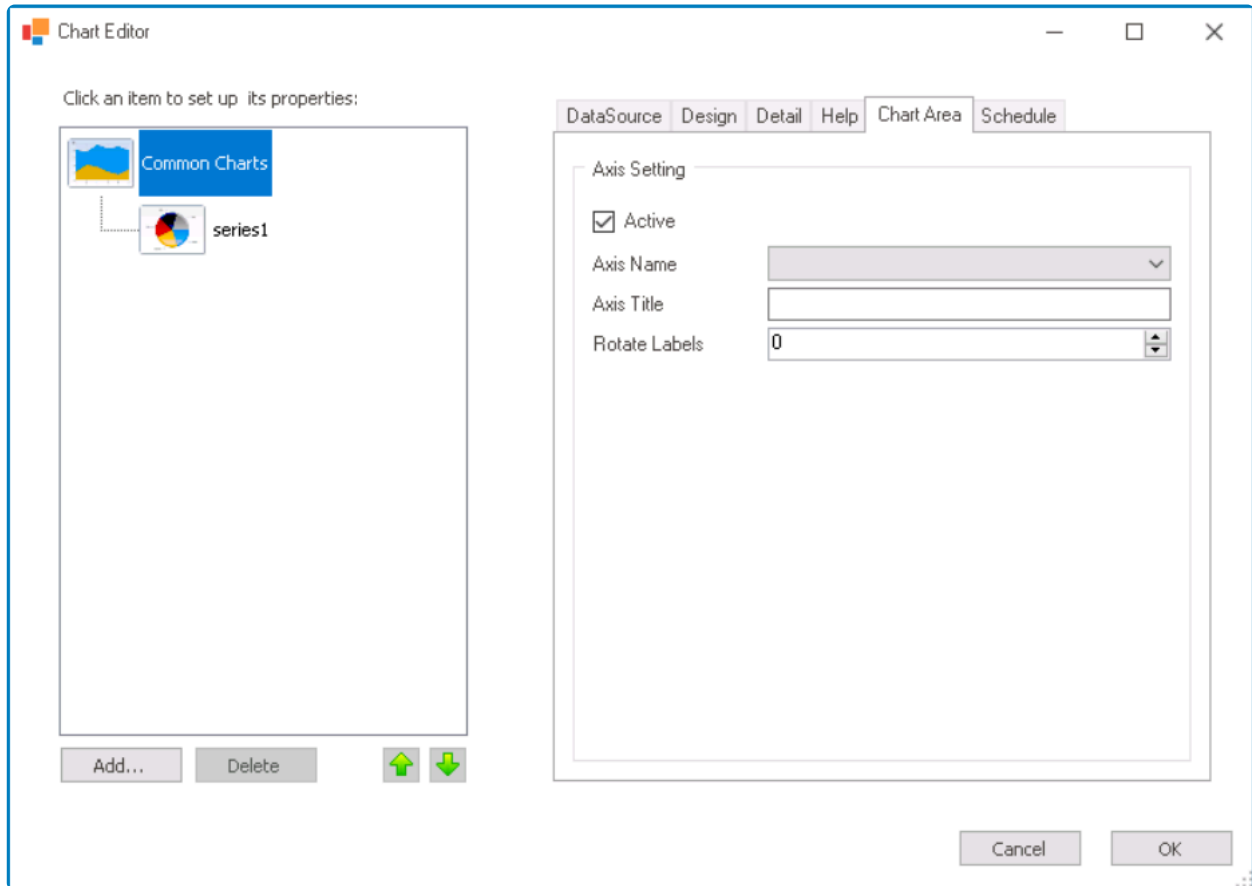


PAST-DUE ● ON-TIME ●



## 12.1.2.1.14.2.1.5. Chart Area

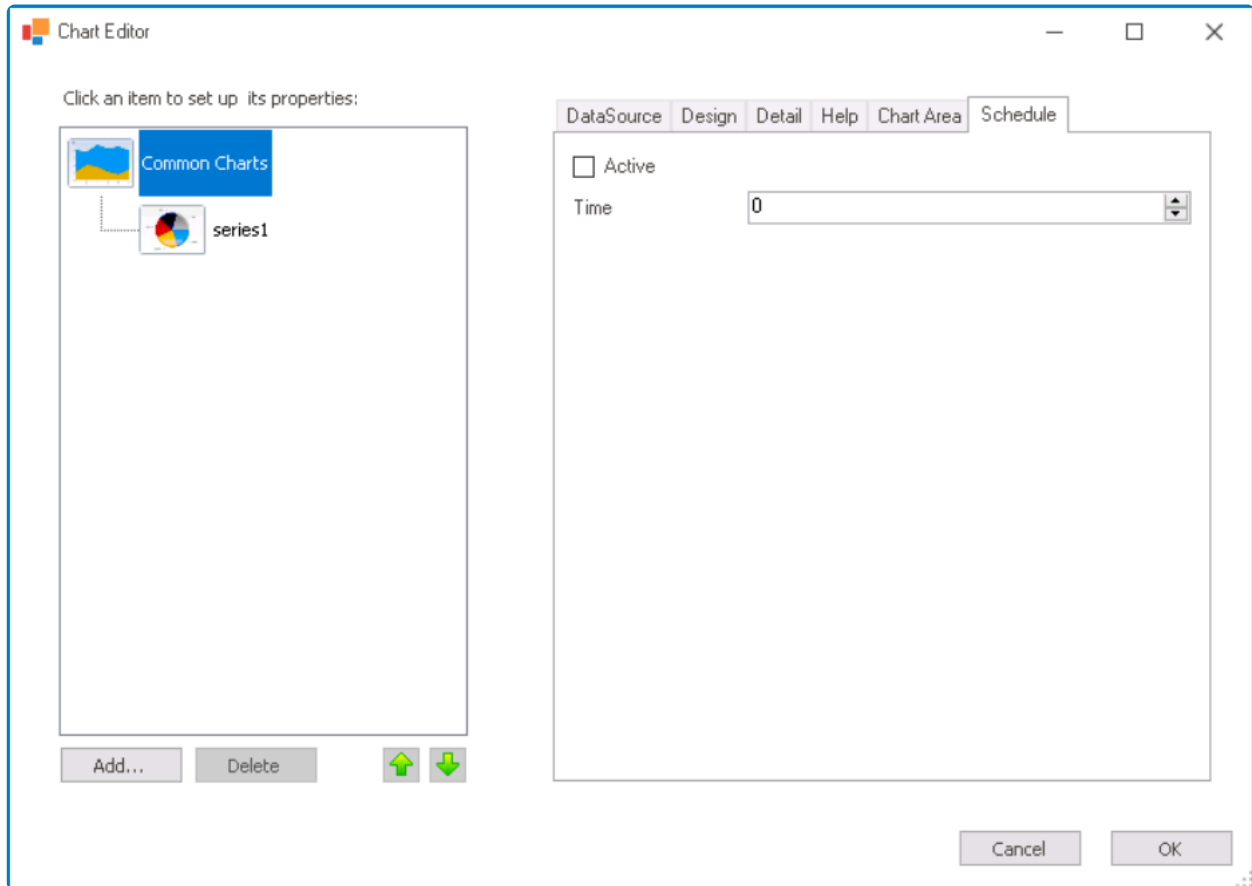
In the Chart Area tab, users can configure the axis settings.



1. **Axis Settings:** In this section, users can customize the appearance and behavior of the selected axis.
  - **Active:** This is a checkbox which, if selected, allows users to make changes to the selected axis.
  - **Axis Name:** This field allows users to select which axis they want to modify.
    - Axis Options: First X-Axis, First Y-Axis, Second X-Axis, or Second Y-Axis.
      - Depending on which option is selected, users can customize the corresponding axis in the chart.
  - **Axis Title:** This is a text field that allows users to input the title of the selected axis.
  - **Rotate Labels:** This is a numeric updown field that allows users to rotate the labels on the selected axis. Users can enter a number between 0 and 360 to specify the degree of rotation.
    - Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.

## 12.1.2.1.14.2.1.6. Schedule

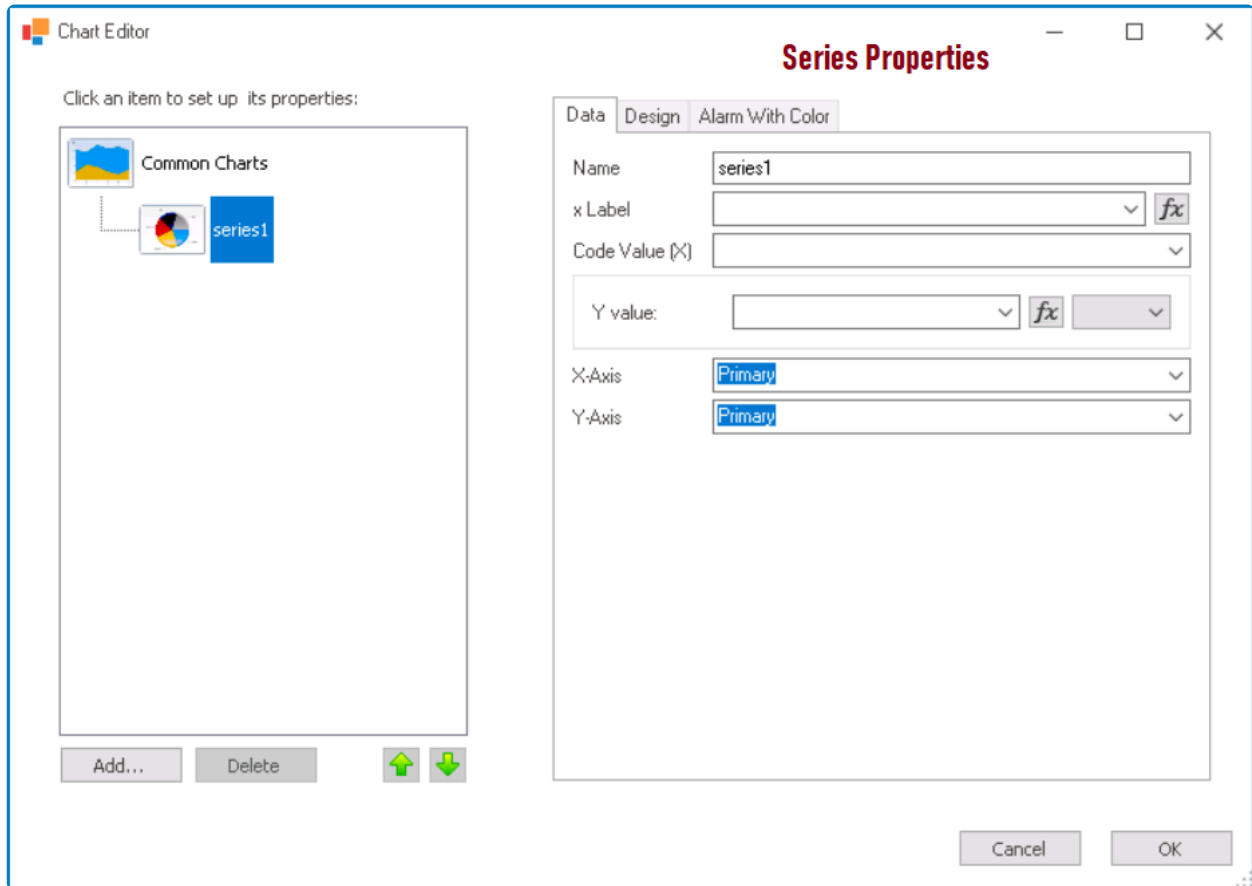
In the Schedule tab, users can set up a schedule for automatic chart refreshing.



1. **Active:** This checkbox enables or disables the schedule for automatic chart refreshing based on a specified time interval.
2. **Time:** This field allows users to specify the time interval for automatic chart refreshing in seconds.
  - Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.
    - Clicking these arrows automatically adjusts the time value by increments of 1000, which represents one second.

## 12.1.2.1.14.2.2. Series Properties

Selecting a series will display its associated properties on the left side of the window.



These properties are displayed in a set of tabs:

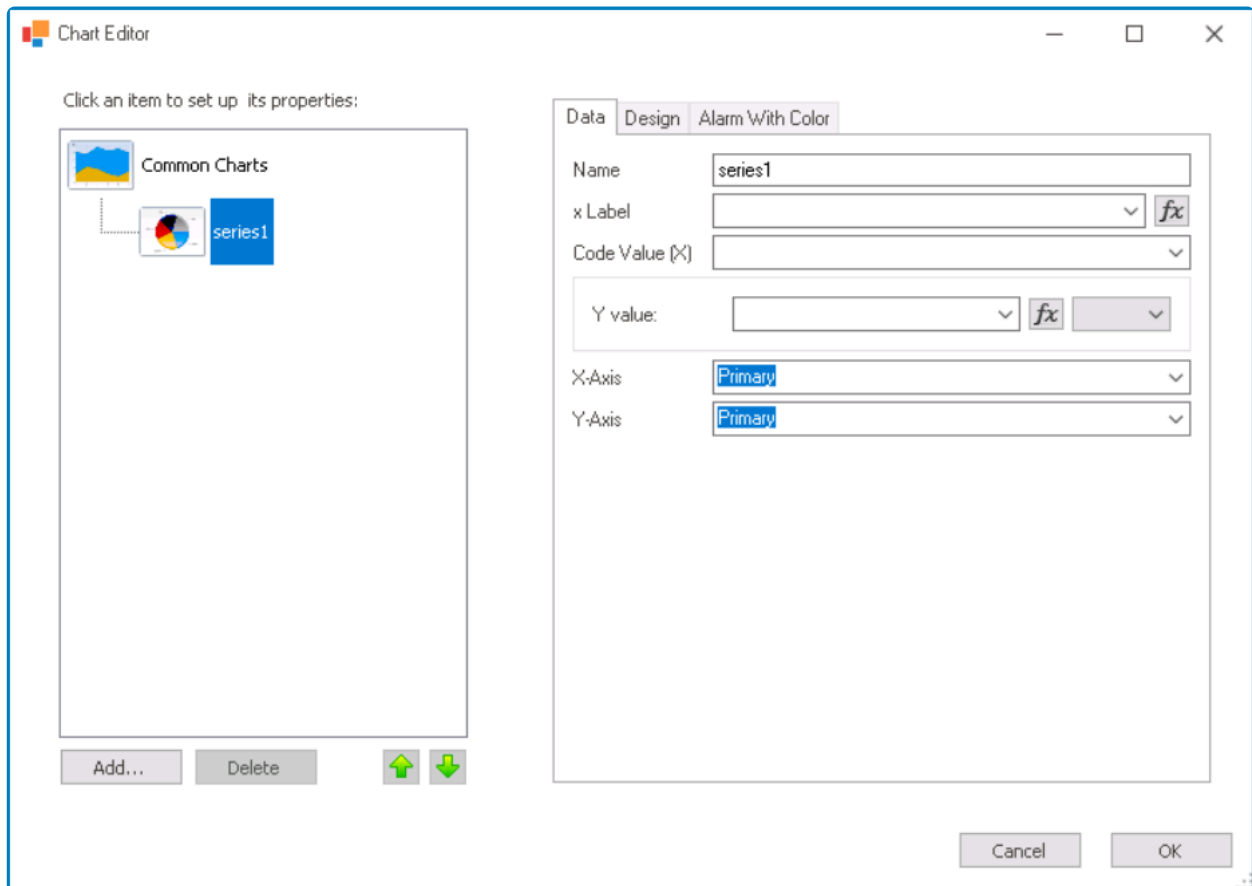
1. [Data](#)
2. [Design](#)
3. [Alarm with Color](#)



To save changes and exit the [Chart Editor](#) window, click on the **OK** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 12.1.2.1.14.2.2.1. Data

In the Data tab, users can configure and customize the data series that will be displayed on the chart.



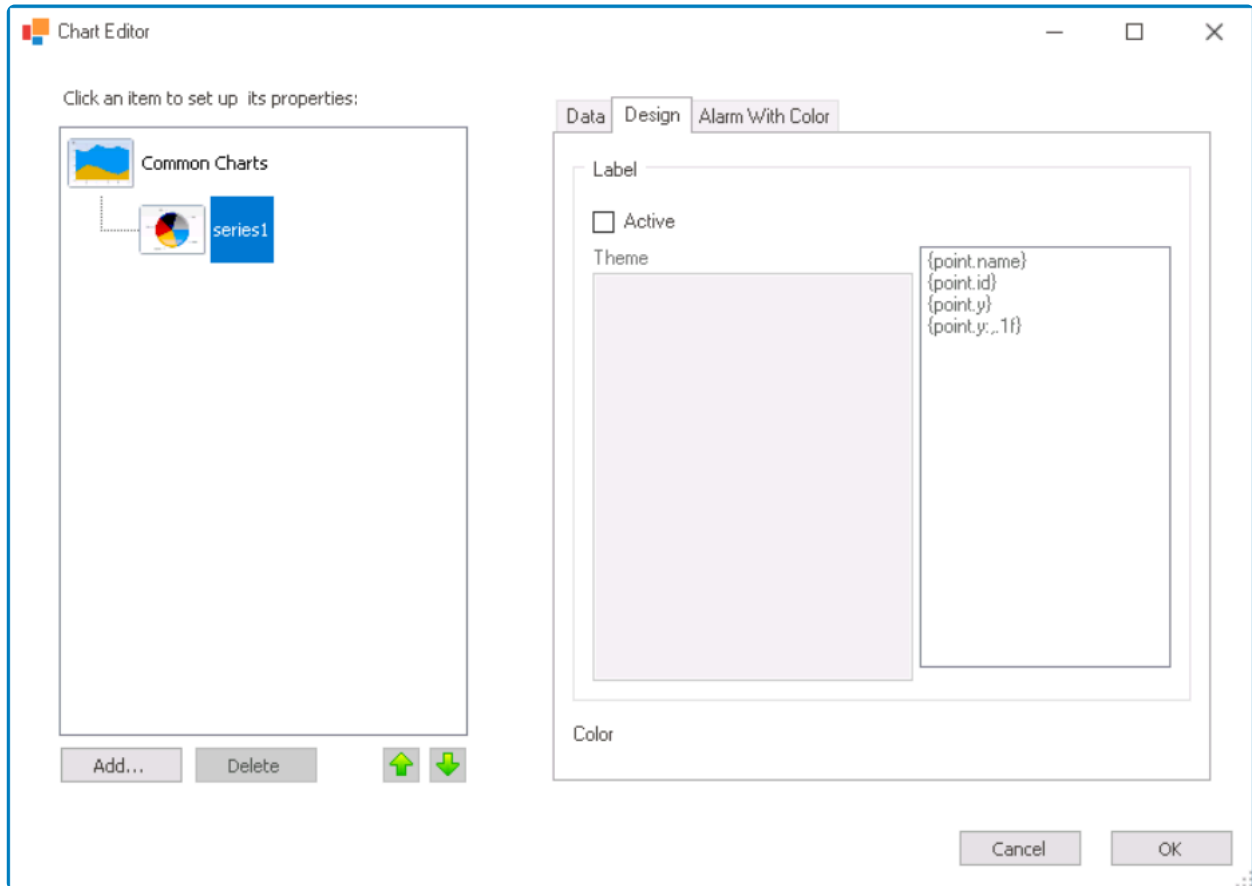
1. **Name:** This is a text field that allows users to input the name of the data series.

✿ This name is used as a tooltip label when hovering over a symbol in the chart. It provides a quick and easy way to identify and understand the data being represented by that symbol.

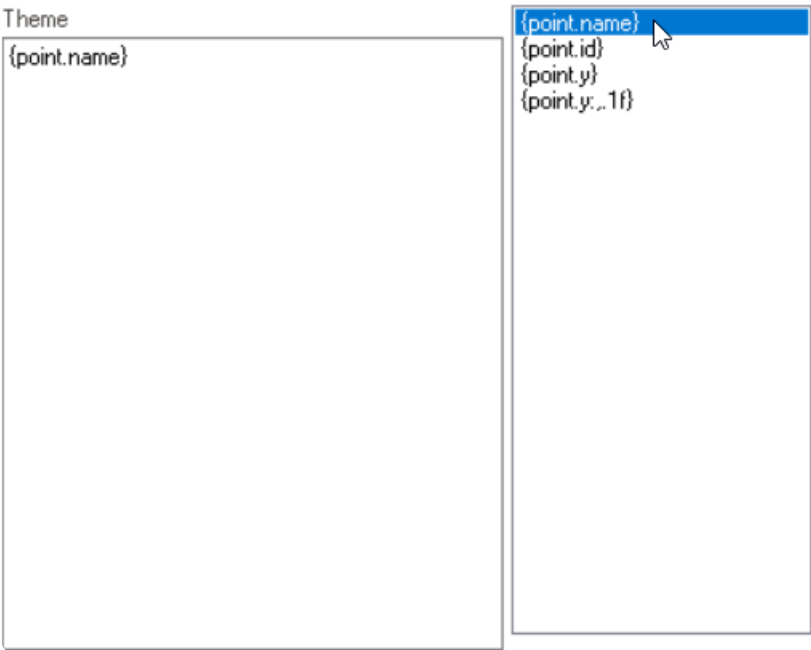
2. **X Label:** This is a single-select dropdown field that retrieves the list of columns associated with the selected data source. It allows users to select a column to use as the x-axis label for the chart.
3. **Code Value (X):** This is a single-select dropdown field that retrieves the list of columns associated with the selected data source. It allows users to select a column to use as the x-axis data for the chart.
4. **Y Value:** This is a single-select dropdown field that retrieves the list of columns associated with the selected data source. It allows users to select a column to use as the y-axis data for the chart.
5. **X-Axis:** This is a single-select dropdown field that displays the list of axis positions or placement options. It allows users to select whether the x-axis is the primary or secondary axis for the chart.
6. **Y-Axis:** This is a single-select dropdown field that displays the list of axis positions or placement options. It allows users to select whether the y-axis is the primary or secondary axis for the chart.

## 12.1.2.1.14.2.2.2. Design

In the Design tab, users can customize the labels of the data series using HTML tags.



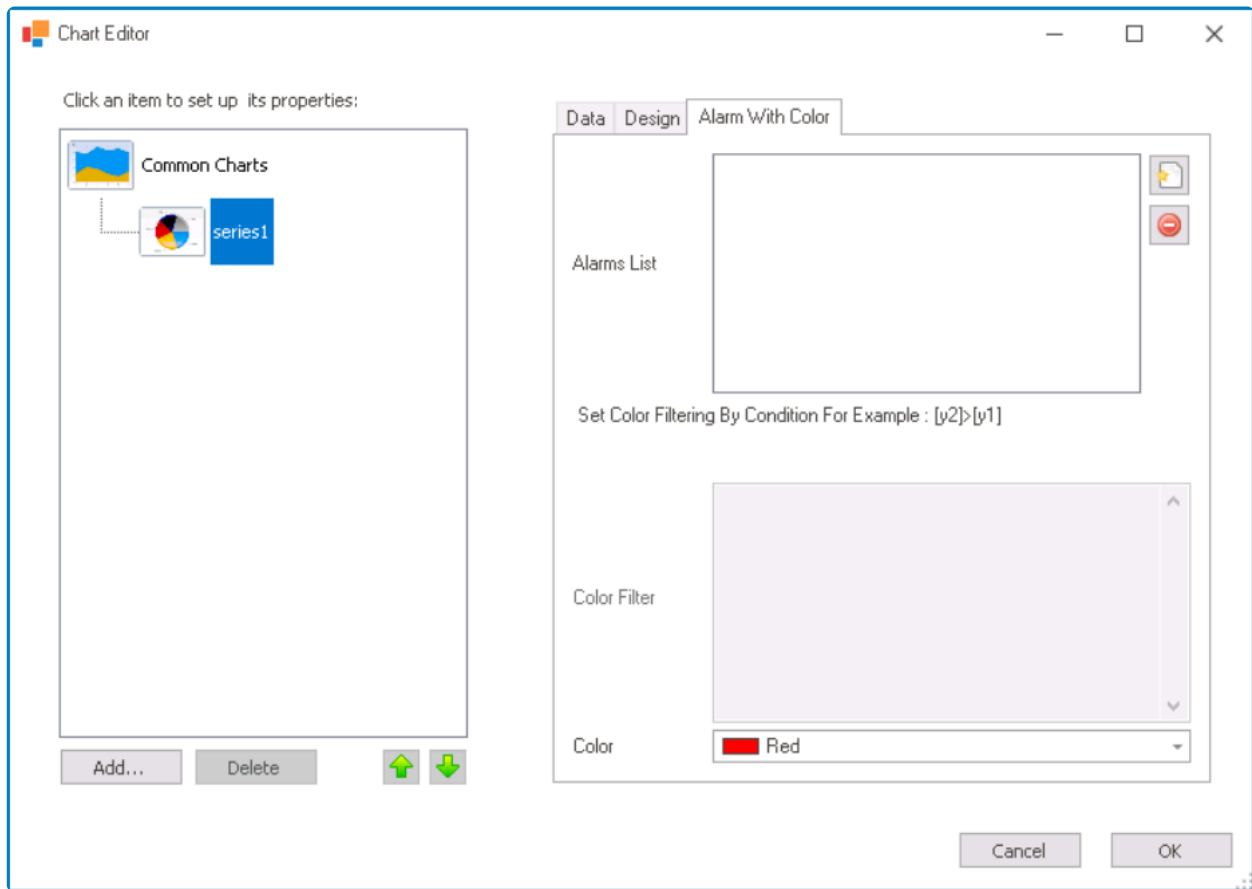
1. **Label:** In this section, users can customize the labels of the data series using HTML tags.
  - **Active:** This is a checkbox which, if selected, allows users to enable the use of HTML tags for the data series labels.
    - If this checkbox is selected, it will enable the **Theme** field.
  - **Theme:** This field allows users to apply HTML tags to customize the text of the data series labels.
    - Users can quickly add formatting and content to their data series labels by choosing a tag from the list box displayed next to this field.
      - When a tag is selected from this list, it will automatically populate the **Theme** field with the appropriate syntax. Users can then customize the tag by modifying the text within the tags or adding additional tags to the text field.




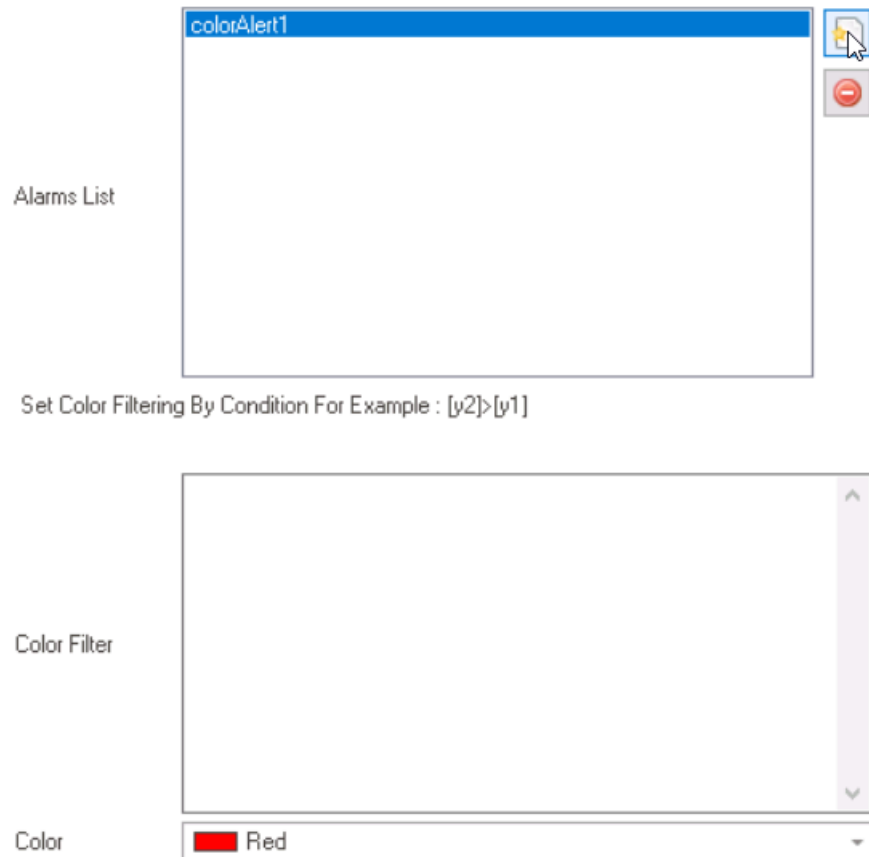
**Interactive Data Series Labels Preview: Web View**


## 12.1.2.1.14.2.2.3. Alarm with Color

In the Alarm with Color tab, users can apply color alarms or alerts to the chart, which can highlight specific chart elements based on certain criteria.



1. **Alarms List:** This field allows users to manage the color alarms for the chart.
  - To add a new color alarm, click on the  button.
    - If selected, it will both add a new entry to the **Alarms List** box and enable the **Color Filter** field.



- To remove a selected color alarm, click on the  button.
2. **Color Filter:** This field allows users to specify a condition that must be met for the chart element to be colored.
- The criteria can include typical operators and entity fields from the data source.

\* For example, users could specify the condition  $[y2] > [y1]$  to color a data series if the values in the second y-axis are greater than those in the first y-axis.

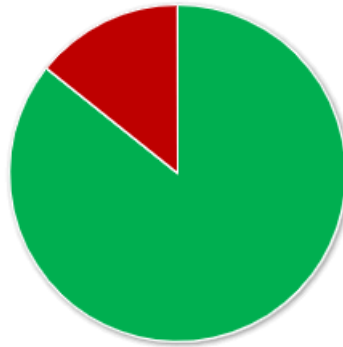
3. **Color:** This is a color picker that allows users to choose the color that will be applied to the chart element when the specified condition is met.
- It contains three tabs: Custom, Web, and System.
    - In the Custom tab, users can choose a font color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
    - In the Web tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
    - In the System tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.



## 12.1.2.1.14.3. Interactive Chart – Web Display

For those unfamiliar with using the interactive charts in the **DBP** web application, the following instructions will guide you through the process:

**NO. OF AUDITS ON TIME VS. PAST DUE**



PAST-DUE ● ON-TIME ●

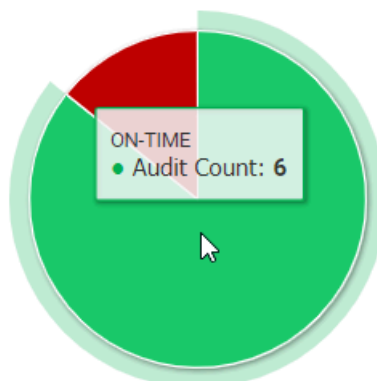
1. **Chart Legend:** This is a graphical element that provides a key for the data series displayed in the chart.

PAST-DUE ● ON-TIME ●

- It includes color-coded labels or symbols that correspond to the data being displayed. Users can refer to the legend to better understand the data represented in the chart.

2. **Chart Tooltip:** This is a graphical element that appears when the user hovers over a data point or marker in the chart.

**NO. OF AUDITS ON TIME VS. PAST DUE**



PAST-DUE ● ON-TIME ●

- It displays additional information about the data point, such as the label, value, or other

contextual details. Users can use the tooltip to gain more insight into the data represented in the chart.

3. **Drilldown Reports:** These are data reports that users can access by double-clicking on the different data points in the chart. They allow users to navigate from the summary report visualized by the chart to the detailed report that created it.
- For example, if users were to double-click on the sector of the pie chart representing audits that have been completed on time, a modal window would appear with the following form:

List of Completed Audit Plans

Search Panel

Audit Plan ID

Audit Plan Name

Filter

Clear Filter


View

Export to Excel

| Audit Plan ID | Audit Plan Name      | Completion Date | No. of Audit Definitions | No. of Requirements |
|---------------|----------------------|-----------------|--------------------------|---------------------|
| <div></div>   | <div></div>          |                 | <div></div>              | <div></div>         |
| AP-00001      | Information Sec...   | 11/23/2022      | 1                        | 1                   |
| AP-00002      | Occupational He...   | 11/28/2022      | 1                        | 2                   |
| AP-00003      | ISO 9001 Certific... | 11/28/2022      | 1                        | 2                   |
| AP-00004      | Laboratory Quali...  | 11/29/2022      | 1                        | 4                   |
| AP-00005      | Financial Foreca...  | 12/05/2022      | 1                        | 1                   |
| AP-00007      | Compliance Audit     | 12/01/2022      | 1                        | 5                   |

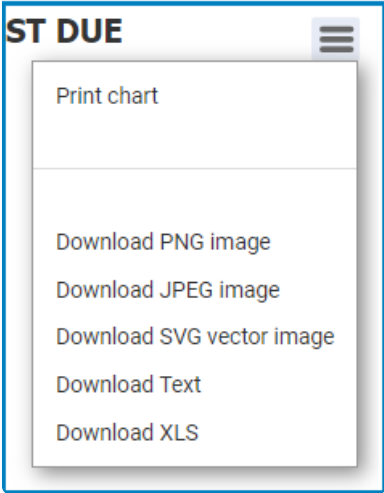
✿ Users can determine the contents of the modal window by creating the form and linking the corresponding form behavior to the control.

4. **Chart Context Menus:** These are hamburger buttons which, when collapsed, display a list of options that allow users to either print or download the chart.

- Icon: 
- If selected, it will display the following menu:

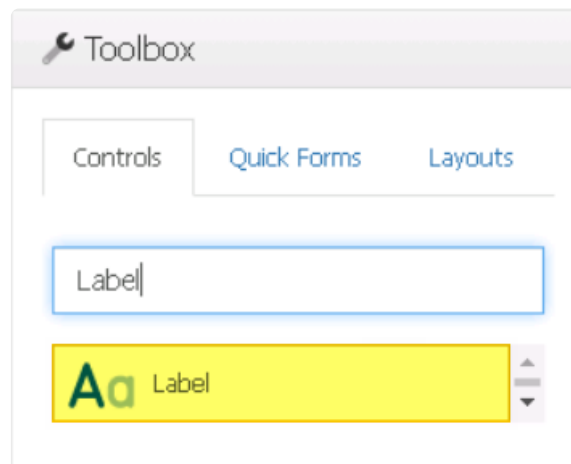
© 2023 Interfacing Technologies Corporation

Page 588 of 1144

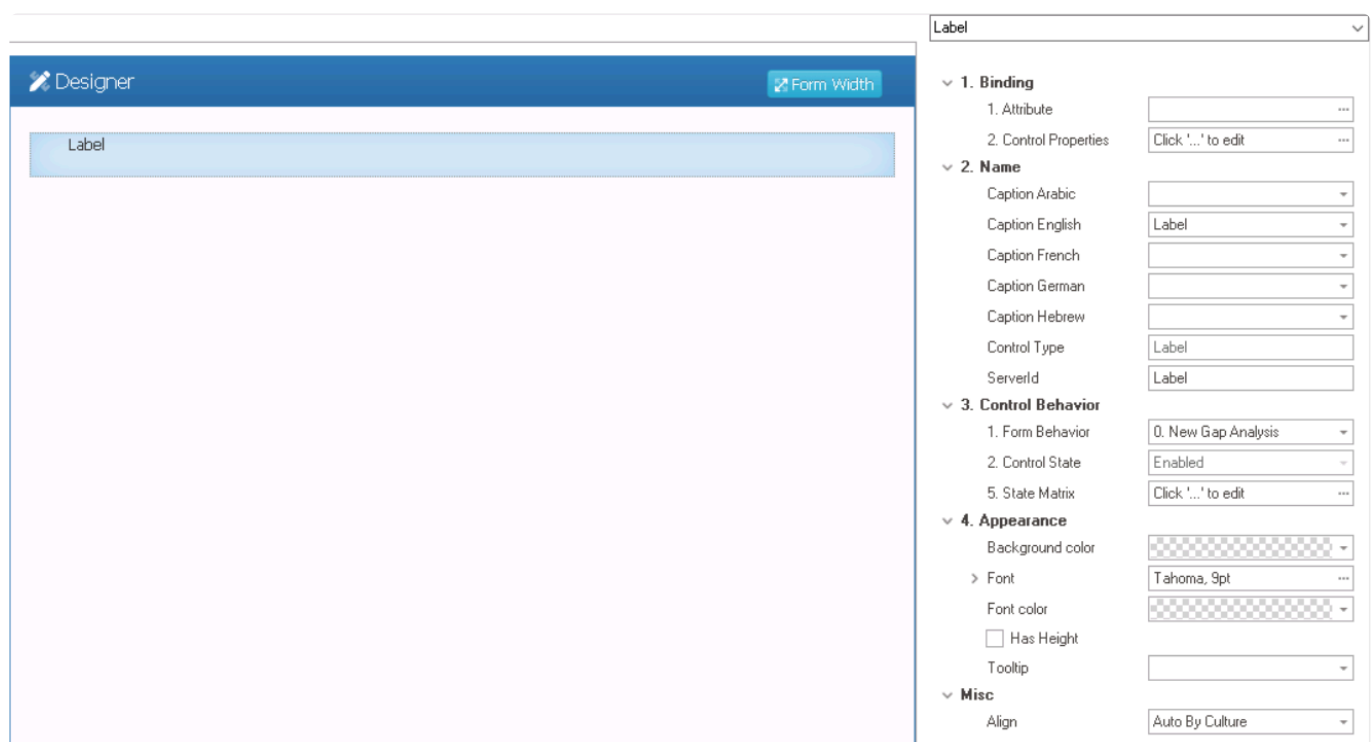


## 12.1.2.1.15. Label

Label controls are a simple yet effective way to display descriptive text or titles for various form elements, providing users with clear guidance and context for their input.



When a label control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

**!** Users can bind label controls to any attribute in the main entity. Generally, however, users do not need to create or link an attribute for label controls.

2. **Name:** In this category, users can set the caption that will be displayed as the label in the web application. They can also access basic properties required for the control's identification and proper referencing in the code.

- **Caption:** This field allows users to input the text that will be displayed as the label in the web application.
  - As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the control, regardless of other language options that may be available.

**\*** Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
- **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state or validation properties set for the control will apply only to the selected form behavior.

**\*** Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses ( **...** ) inside the field.

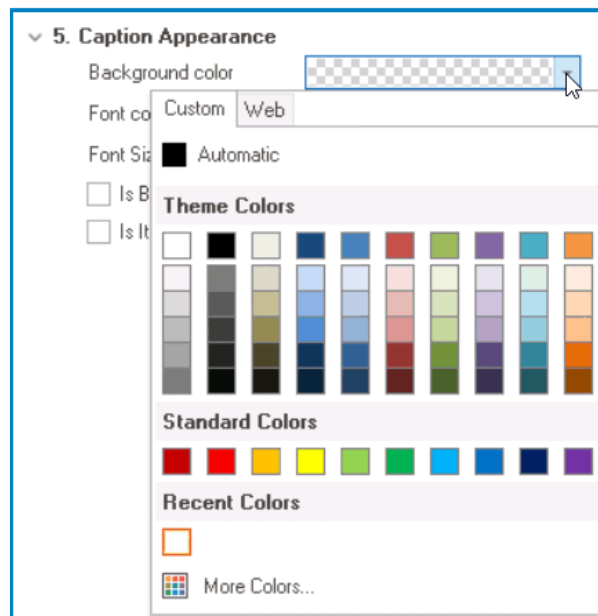
| Form Behavior          | Control State |
|------------------------|---------------|
| 1. Modify Gap Analysis | ✓ Enabled     |
| 2. New Gap Analysis    | ✓ Enabled     |
| 4. Review Gap Analysis | ✓ Enabled     |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.



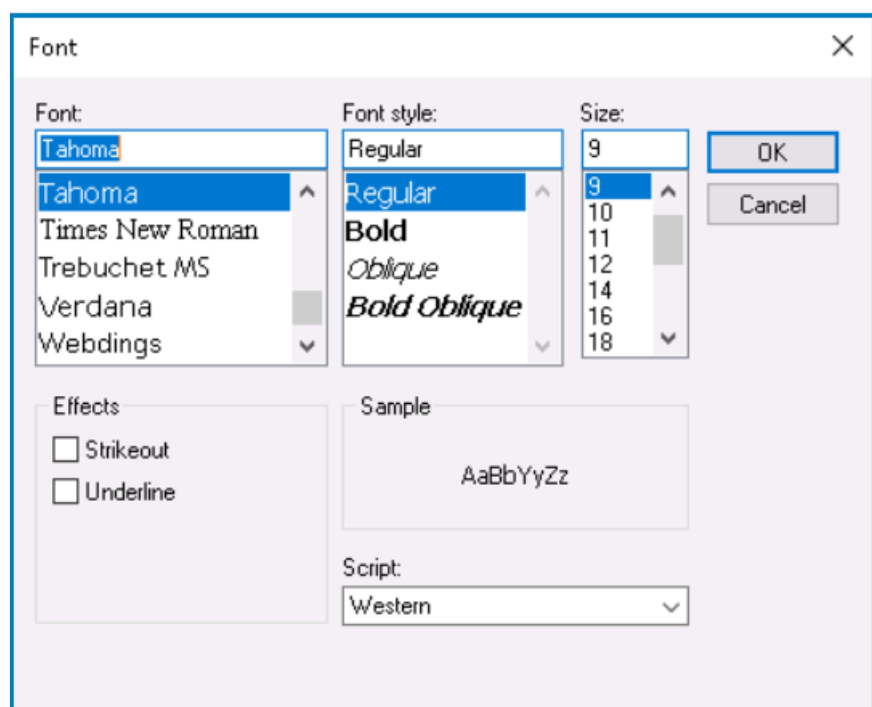
By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

4. **Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - **Background Color:** This is a single-select dropdown field that allows users to choose the background color of the caption text.
    - It contains two tabs: Custom and Web.



- In the Custom tab, users can choose a background color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
  - In the Web tab, users can select a background color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
- After choosing a background color, the control will update to reflect the new color scheme.
- **Font:** This field allows users to configure the font properties for the caption text.
  - To configure all font properties at once, click on the ellipses (...) inside the field.
  - **Configuring All Font Properties**

If the ellipses button is selected, the Font window will appear.



- **Font:** This field allows users to choose the font family from a

scrollable list of options.

- **Font Style:** This field allows users to choose the font style from a scrollable list of options.
  - Depending on the font family selected, the options displayed in this list may vary.
- **Size:** This field allows users to choose the font size from a scrollable list of options.
  - Users can also simply type in a specific font size.
- **Effects:** In this section, users can modify the appearance of the text by applying common effects, such as strikeout and underline.
  - **Strikeout:** This is a checkbox which, if selected, allows users to apply a strikeout effect to the text.
  - **Underline:** This is a checkbox which, if selected, allows users to apply an underline effect to the text.
- **Sample:** This field displays a preview of how the text will look based on the chosen font family, style, size, and effect. It updates in real-time as the user makes changes to the font settings.
- **Script:** This field allows users to choose a specific script or writing system for the text from a dropdown menu.
  - By selecting the appropriate script, users can ensure that their text is correctly displayed and formatted for the desired language or writing system.
- **OK or Cancel:** To save your changes and close the window, click on the **OK** button. To cancel your changes and close the window, click on the **Cancel** button.

- To instead configure individual font properties, toggle the arrow (▼) icon.

#### ▪ **Configuring Individual Font Properties**

If the arrow icon is toggled, the following fields will appear:

Font

Name: Tahoma

Size: 9

Unit: Point

☐ Bold

GdiCharSet: 1

☐ GdiVerticalFont

☐ Italic

☐ Strikeout

☐ Underline

- **Name:** This field allows users to choose the font family from a dropdown menu.
- **Size:** This field allows users to set the font size for the text.
  - Users can enter a font size into this field either manually or by clicking the increment/decrement arrows.
- **Unit:** This field allows users to choose a specific unit of



measurement for the font size from a dropdown menu

- **Bold:** This is a checkbox which, if selected, allows users to apply a bold effect to the text.




Please note that certain font families do not support bold formatting.

- **GdiCharSet:** This field allows users to specify the character set or encoding for the text.
  - Users can enter a number into this field either manually or by clicking the increment/decrement arrows.




The numeric value entered represents a specific character set or encoding as defined by the Windows API, such as the ANSI character set (0), the default system character set (1), or a specific language character set like the Hebrew (177) or Arabic (178) character sets.

- **GdiVerticalFont:** This is a checkbox which, if selected, indicates that the selected font supports vertical writing.
  - Vertical writing is commonly used in East Asian scripts such as Chinese, Japanese, and Korean.
- **Italic:** This is a checkbox which, if selected, allows users to apply an italic or slanted effect to the text.
- **Strikeout:** This is a checkbox which, if selected, allows users to apply a strikeout effect to the text.
- **Underline:** This is a checkbox which, if selected, allows users to apply an underline effect to the text.
- **Font Color:** This is a single-select dropdown field that allows users to choose the font color of the caption text.
  - It also has two tabs, similar to the **Background Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.
- **Has Height:** This is a checkbox which, if selected, allows for automatic adjustment of the vertical spacing or line-height of the caption text, improving its readability or aesthetic appeal.
  - If this checkbox is selected, a bottom margin of 12 px is automatically added to the label.
- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon (  ) next to the control.

- **Label Tooltip Preview: Web View**

What are your focus areas? What are your goals?

**Identify the scope for the gap analysis** 

 A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Misc:** In this category, users can define additional properties for the control.

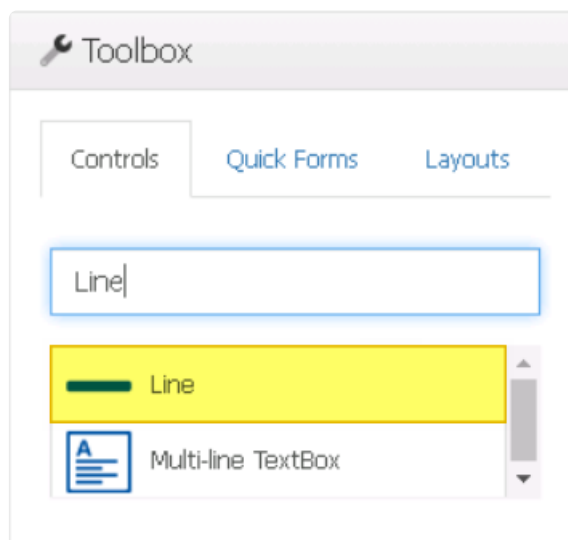
- **Align:** This is a single-select dropdown field that allows users to choose the alignment of the control.
  - Alignment Options: Right, Center, Left, or Auto By Culture.
    - Auto By Culture: This option aligns the control based on the language or culture setting in the web application.
    - Right: This option aligns the control to the right side of the screen.
    - Center: This option aligns the control to the center of the screen.
    - Left: This option aligns the control to the left side of the screen.

**Label Control Preview: Web View**

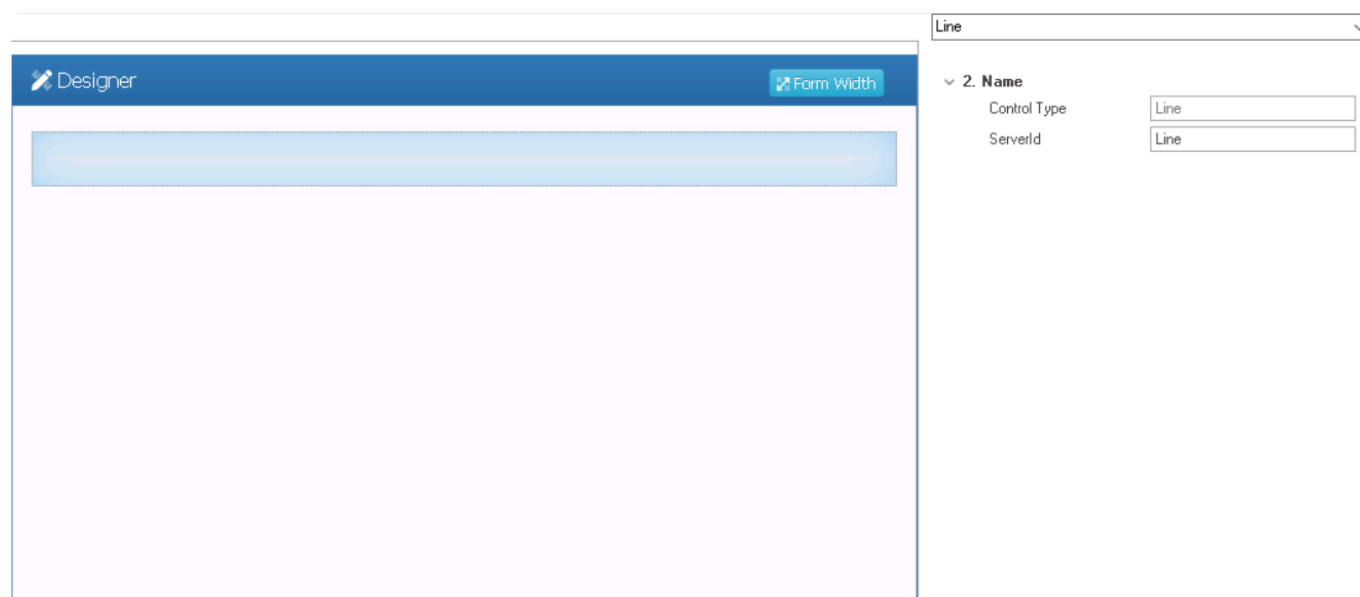
Identify the scope for the gap analysis

## 12.1.2.1.16. Line

The line control allows users to insert horizontal lines into a form, creating clear visual sections that improve the organization and readability of the form.



When a line control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

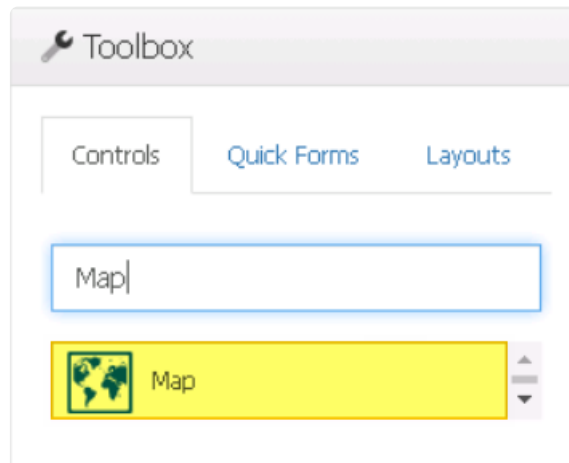


1. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.

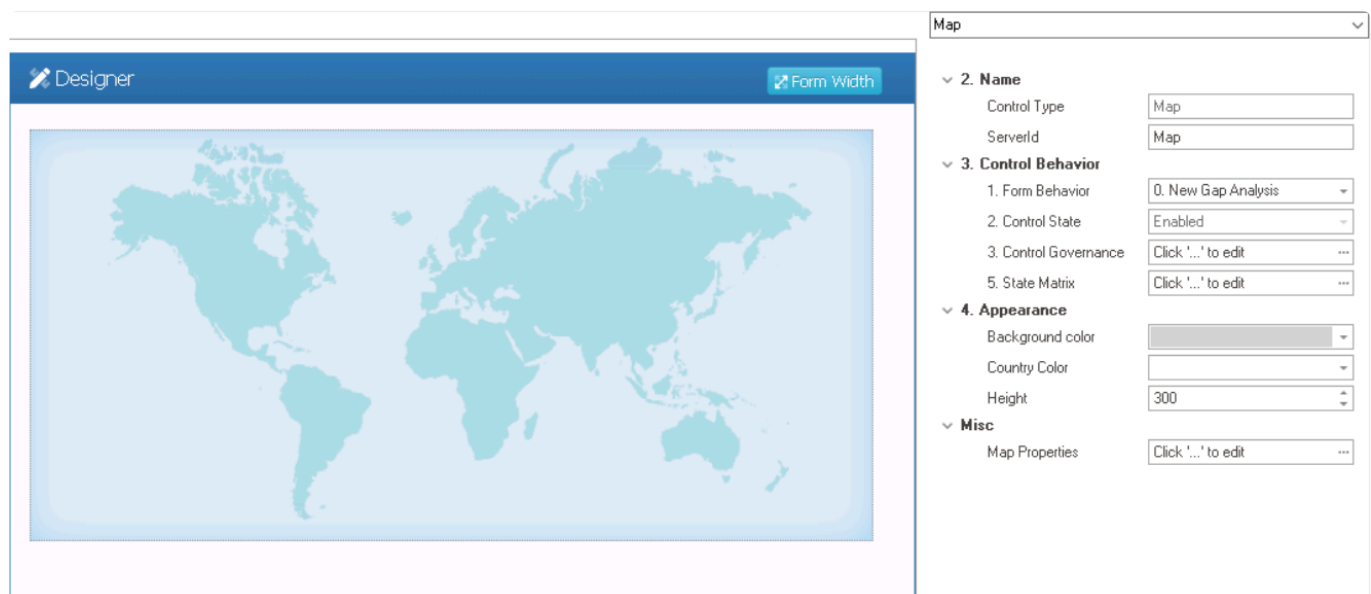
### Line Control Preview: Web View

## 12.1.2.1.17. Map

Map controls are useful tools that enable users to navigate, explore, and interact with geographic data. With these controls, users can manipulate maps, zoom in and out, pan across different regions, and extract valuable insights from spatial information.



When a map control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
2. **Control Behavior:** In this category, users can configure the behavior of the control and define

how it interacts with the form or application.

- **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
  - Any state properties set for the control will apply only to the selected form behavior.






Only form behaviors linked to the associated form will be displayed in this dropdown.


- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **Control Governance:** This field allows users to configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field.

The screenshot shows a dialog box titled "Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [Map]". At the top, there are three buttons: "Apply" (with a green square icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon). Below the buttons, the "Control Name" is set to "Chart". Underneath, there is a text input field containing the word "Chart". Below the text field, there is a section with two main items: "Enable View" and "View Form Behaviour". The "Enable View" item consists of a checkbox and a plus icon. The "View Form Behaviour" item consists of a dropdown menu currently showing "[Empty]", followed by three icons: a plus icon, a pencil icon, and a minus icon.

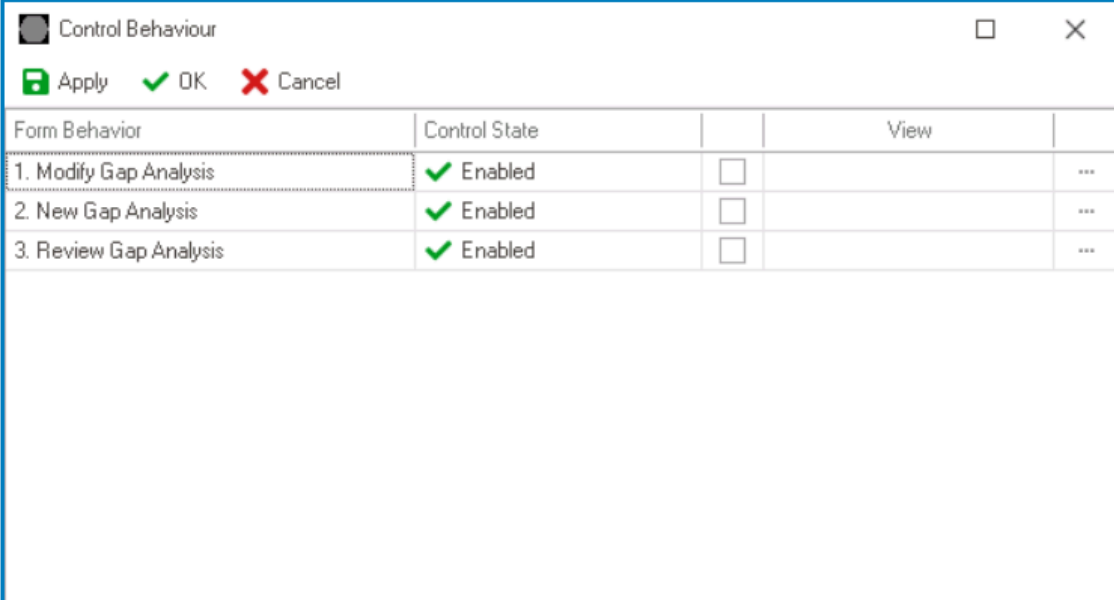
- **Enable View:** This is a checkbox which, if selected, allows users to enable view mode for the control.
  - If this checkbox is selected, it will enable the **View Form Behavior** field.
- **View Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It

allows users to select the form behavior linked to the control's 'view' functionality.

- To define a new form behavior, select the  button.
  - It will display the [Form Behavior Properties](#) window.
- To modify an existing form behavior, select the  button.
- To remove a form behavior, select the  button.

 With **View** enabled, users can view the relevant data in a modal window linked to the control. Users can access the 'View Form' by double-clicking on an element in the control.

- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.



| Form Behavior          | Control State |                          | View |     |
|------------------------|---------------|--------------------------|------|-----|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |      | ... |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> |      | ... |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> |      | ... |

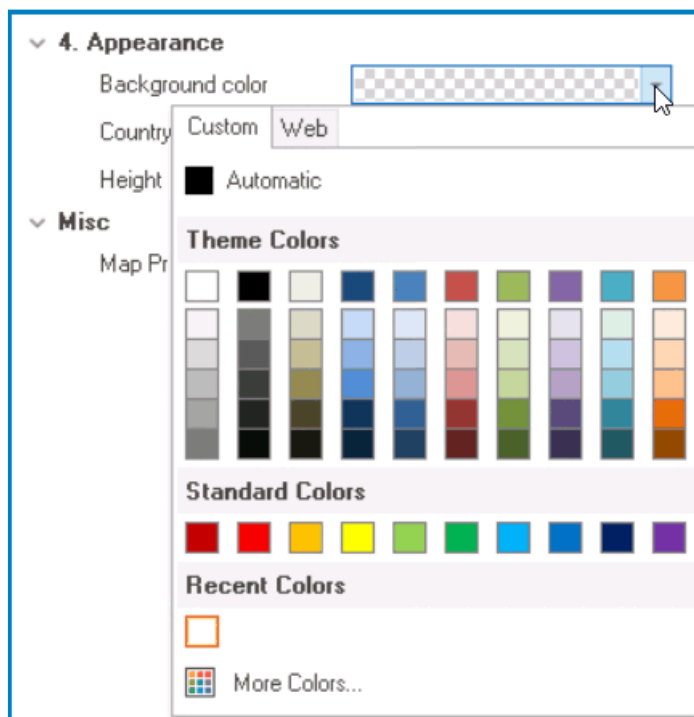
- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Control Governance:** This field allows users to configure the control governance properties.
  - If selected, it will display the [Form / Behavior / Control](#) window (described above).
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.

✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

! In this window, users can also configure the governance properties for the control across multiple form behaviors. To gain an understanding of how these properties can be configured, [click here](#).

3. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Background Color:** This is a single-select dropdown field that allows users to choose the background color of the map control.
  - It contains two tabs: Custom and Web.



- In the Custom tab, users can choose a background color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
- In the Web tab, users can select a background color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.

- **Map Background Color Preview: Web View**



- **Country Color:** This is a single-select dropdown field that allows users to choose the color of the countries in the map control.
  - It also has two tabs, similar to the **Background Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.
  - **Map Country Color Preview: Web View**



- **Height:** This is a numeric updown field that allows users to set the height of the control.
  - By default, the height is set to 300. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- 4. **Misc:** In this category, users can define additional properties for the control.
  - **Map Properties:** This field allows users to configure the gauge properties.
    - To access the map properties, click on the ellipses (...) inside the field.
      - It will display the [Map Editor Form](#) window.



Users do not need to create or link an attribute for map controls.



**Map Control Preview: Web View**

- ✱ As shown in the graphic above, to zoom in and out of the map, users can select the + or – buttons located in the upper left corner of the control. Alternatively, users can adjust the zoom level by using the mouse wheel or touchpad on their device.

## 12.1.2.1.17.1. Map Editor Form

---

In the Map Editor Form window, users can configure the properties of the map control.

These properties are displayed in a set of tabs:

1. [Data](#)
2. [Bubble Size](#)
3. [Coloring](#)
4. [Tooltip](#)



To save changes and exit the Map Editor Form window, click on the **Save** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 12.1.2.1.17.1.1. Data

In the Data tab, users can customize the map display by choosing the desired map and configuring fields to tailor the visualization of geographic data. Users can also select and configure the data source for the control, including any necessary query parameters.

The screenshot shows the 'Map Editor Form' with the 'Data' tab selected. The form is divided into several sections:

- Select Map:** A dropdown menu for 'Map name' is set to 'canada'.
- Select DataSource:** A dropdown menu for 'DataSource name' is set to '[Empty]'. There are '+' and '-' icons to the right.
- Bind Parameters, then click execute button:** A table with two columns: 'Description' and 'Value'. The table contains one row with a small icon in both columns. Below the table is a status bar showing 'Record 0 of 0' and navigation icons. An 'Execute' button is located below the table.
- Map Fields:** Four dropdown menus for 'Country or City', 'Map coloring field', 'Bubble coloring field', and 'Bubble size field'.

At the bottom right, there are 'cancel' and 'save' buttons.

1. **Select Map:** In this section, users can choose the map to be displayed in the map control, defining the scope and geographic context for their data visualization.
  - **Map Name:** This field allows users to select the specific map they want to display from a dropdown menu.
    - By choosing a particular map, users can visualize data at different geographical levels, whether country-specific, continent-specific, or global.







Users can further configure the control to visualize data in the following ways:

- Entire world, by country
- Entire world, by continent

- Specific continent, by country
- USA, by state
- Canada, by province

2. **Select DataSource:** In this section, users can select the data source for the map control.

- **DataSource Name:** This dropdown retrieves the list of data sources (from both the [Dropdown Data Sources](#) section of the **Forms** module and the [KPIs](#) section of the **Entities** module). It allows users to select the data source for the map control.
  - To define a new data source, select the  button.
    - It will display the [KPI Properties](#) window.
  - To modify an existing data source, select the  button.
  - To remove a selected data source, select the  button.

 The selected data source will provide the data for the map control.

3. **Bind Parameters & Execute:** In this section, users can set parameters for the query and execute it with the selected data source.

- Parameters are values that are used to filter or modify the data returned by the query.
  - If the selected data source has a parametric query (i.e., a query that contains parameter placeholders), users can enter the parameter names in the [Description](#) column and the corresponding values for the parameters in the [Value](#) column.
- **Execute:** This button allows users to run or execute the parametric query with the selected data source and the specified parameter values.
  - When the query is executed, it uses the provided parameter values to filter or modify the data from the selected data source.

4. **Map Fields:** In this section, users can map specific fields to columns in the selected data source, thereby configuring the map display and data visualization accordingly.

- **Country or City:** This dropdown retrieves the list of columns associated with the selected data source. It allows users to select the column that contains codes representing continents, countries, or cities (depending on the chosen map).
  - By mapping this field, users can define how the map visually associates data points with specific continents, countries, or cities.



Please note that the codes in this column must adhere to the format provided on the following page: <https://jvectormap.com/maps/world/>

- **Map Coloring Field:** This dropdown retrieves the list of columns associated with the selected data source. It allows users to select the column that determines the coloring of map elements (continents, countries, or cities).
  - By mapping this field, users can visually represent variations in data through colors or shading, emphasizing different aspects of their data.

✿ To set the color range for the map, please refer to the [Coloring](#) tab in the window.

- **Bubble Coloring Field:** This dropdown retrieves the list of columns associated with the selected data source. It allows users to select the column that determines the coloring of bubbles for each map element (continent, country, or city).
  - By mapping this field, users can highlight specific attributes or categories within the data, allowing for clearer differentiation and visualization.

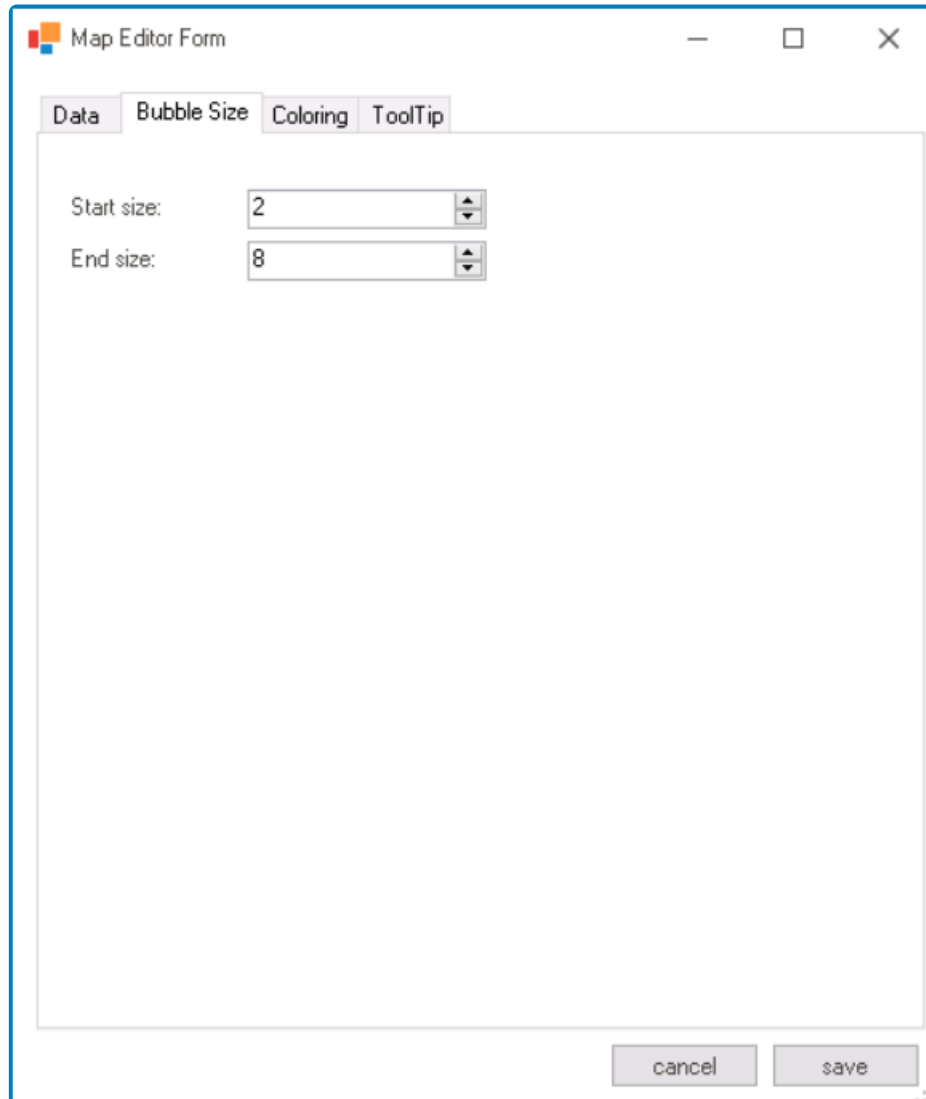
✿ To set the color range for the data bubbles, please refer to the [Coloring](#) tab in the window.

- **Bubble Size Field:** This dropdown retrieves the list of columns associated with the selected data source. It allows users to select the column that determines the size of the bubbles for each map element (continent, country, or city).
  - By mapping this field, users can visually convey varying magnitudes or quantities within the data, providing a clear representation of relative values.

✿ To set the size range for the data bubbles, please refer to the [Bubble Size](#) tab in the window.

## 12.1.2.1.17.1.2. Bubble Size

In the Bubble Size tab, users can determine the size range of the data bubbles displayed on the map.

The image shows a screenshot of a software window titled "Map Editor Form". It has four tabs: "Data", "Bubble Size", "Coloring", and "ToolTip". The "Bubble Size" tab is currently selected. Inside this tab, there are two numeric updown fields. The first is labeled "Start size:" and has the value "2". The second is labeled "End size:" and has the value "8". At the bottom right of the window, there are two buttons: "cancel" and "save".

1. **Start Size:** This is a numeric updown field that allows users to set the start or initial size of the bubble.
  - Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.
2. **End Size:** This is a numeric updown field that allows users to set the end or final size of the bubble.
  - Users can enter numbers into this field either manually or by clicking the increment/decrement arrows.

## 12.1.2.1.17.1.3. Coloring

In the Coloring tab, users can configure the color scheme for the map and data bubbles displayed on the map.



All color pickers in this window feature three tabs, providing users with the ability to select colors either by manipulating sliders or input fields in the RGBA color picker or by choosing from a scrollable list of pre-defined options. [For more information on these tabs, please refer to the description below for the Map Color field.](#)

1. **Map Color:** In this section, users can define the color range for the map.
  - **Start Color:** This is a color picker that allows users to choose the start or initial color of the map.
    - It contains three tabs: Tab Pages, Web, and System.
      - In the Tab Pages tab, users can select a color by manipulating sliders or input fields for each color channel in the RGBA color picker.



With an RGBA color picker, users can specify not only the red, green, and blue (RGB) values but also the alpha channel value (A), which controls the transparency or opacity of the color.

- In the Web tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
  - In the System tab, users can select a color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
  - **End Color:** This is a color picker that allows users to choose the end or final color of the map.
2. **Bubble Color:** In this section, users can define the color range for the data bubbles displayed on the map.
- **Start Color:** This is a color picker that allows users to choose the start or initial color of the bubbles.
  - **End Color:** This is a color picker that allows users to choose the end or final color of the bubbles.



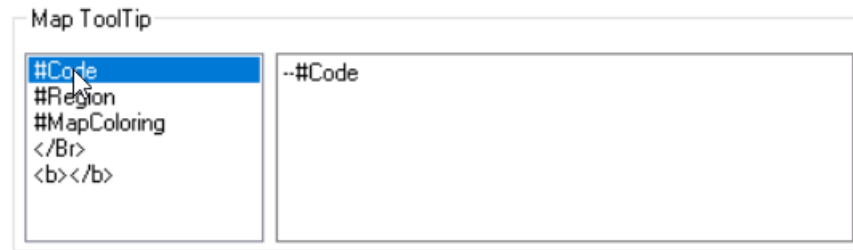
## 12.1.2.1.17.1.4. Tooltip

In the Tooltip tab, users can customize the tooltip content for both the data bubbles and the map elements.

1. **Bubble Tooltip:** This field allows users to apply HTML tags to customize the tooltip content for the data bubbles on the map.
  - Users can select a tag from the list box below this field, which will automatically populate the adjacent empty box with the appropriate syntax.

- Users can further customize the tooltip by modifying the text within the tags or adding additional tags to the text field.
2. **Map Tooltip:** This field allows users to apply HTML tags to customize the tooltip content for the map elements.

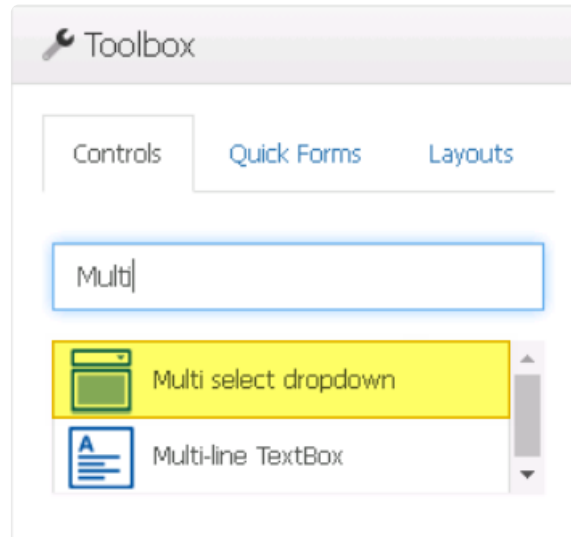
- Users can select a tag from the list box below this field, which will automatically populate the adjacent empty box with the appropriate syntax.



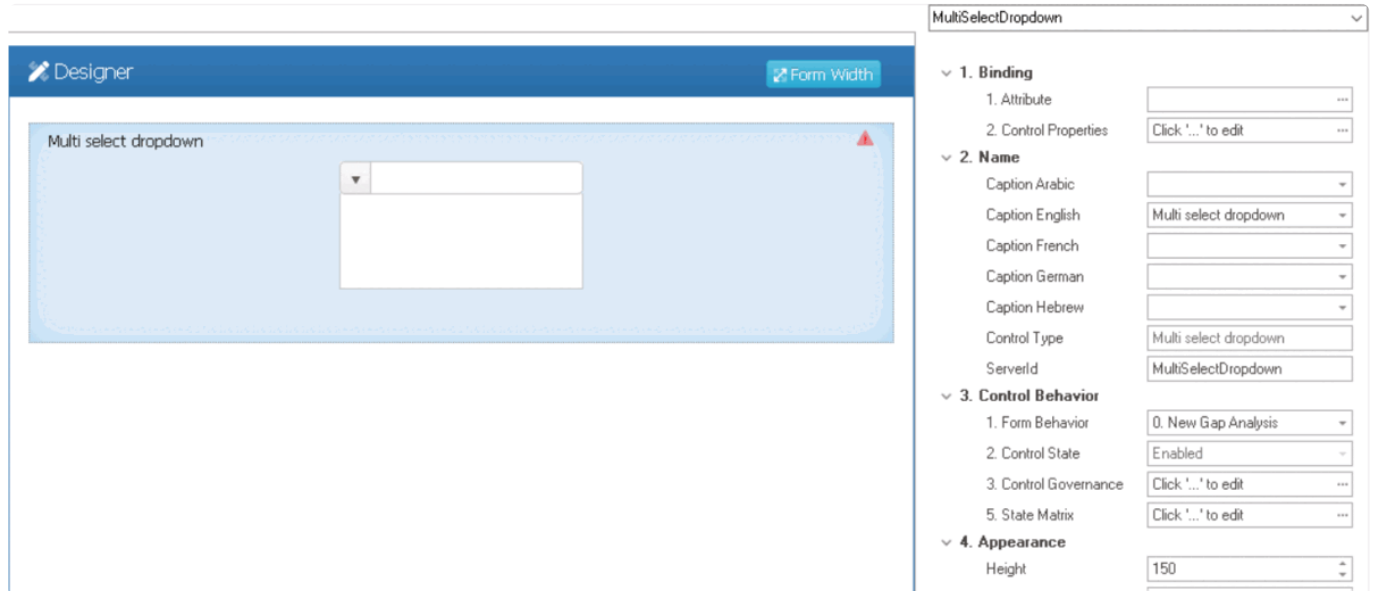
- Users can further customize the tooltip by modifying the text within the tags or adding additional tags to the text field.

## 12.1.2.1.18. Multi-Select Dropdown

A multi-select dropdown control is a user interface element that allows users to select multiple options from a list of predetermined choices. It can reference various system tables such as users, roles, and groups, as well as display a list created in a different form.



When a multi-select dropdown control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).



For more information on the required attributes for multi-select dropdown controls, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
      - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
        - This window consists of two tabs:
          - [Selector](#)
          - [Validation](#)



Please note that users are limited to configuring only the required field validation properties for multi-select dropdown controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).



Please note that the linked section contains information on setting an initial value, which is not applicable to multi-select dropdown controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Multi-Select Dropdown Height**

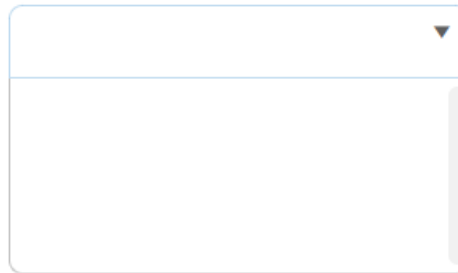
In the **Height** field, users can set the height of the control. By default, the height is set to 150. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows

    - For more information on these properties and how to configure them, [click here](#).
5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - For more information on the fields in this category, [click here](#).

**Multi-Select Dropdown Control Preview: Web View****Data Analysis Methods**

Select All

Deselect All



✿ For more information on how to use this field in the web application, [click here](#).

**Quick Forms Application**

Applying the **Quick Forms** feature to a multi-select dropdown control depends on whether it references a system table or another form.

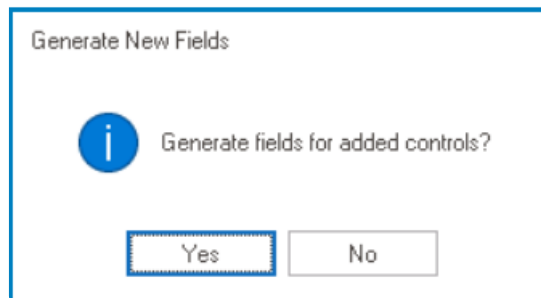
✿ To learn more about applying the **Quick Forms** feature in either scenario, [click here](#).

## 12.1.2.1.18.1. Multi-Select Dropdown – Quick Forms

Applying the **Quick Forms** feature to a multi-select dropdown control depends on whether it references a system table or another form. Follow the steps below for each scenario:

### A. Referencing a System Table

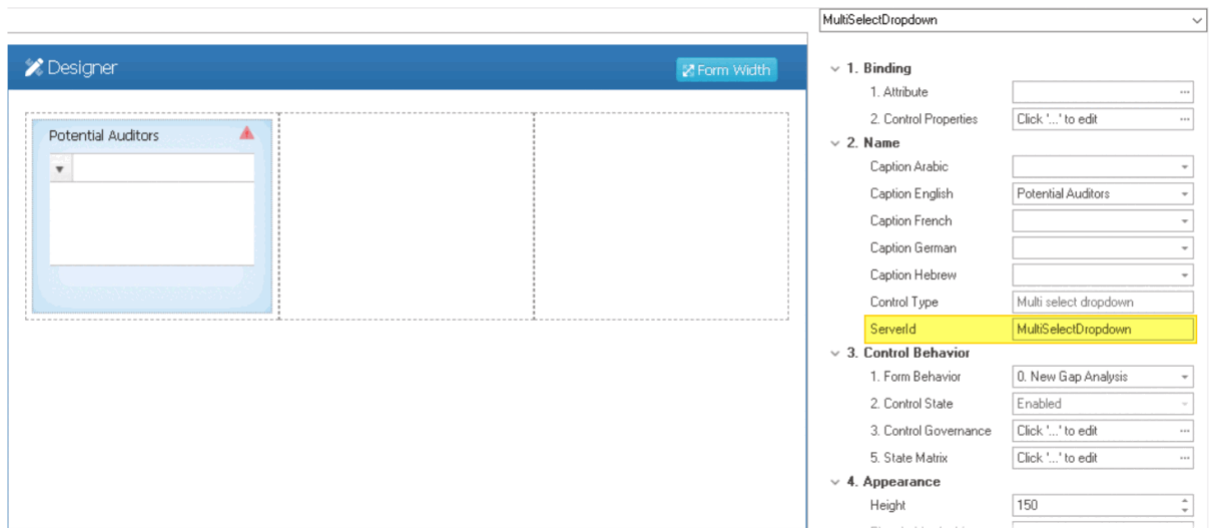
1. Drag a multi-select dropdown control from the toolbox and drop it onto the design environment.
  2. Provide a caption for the control.
  3. Specify the corresponding system entity in the **Server ID** field to associate the control with an existing table in the database.
    - Enter a **Server ID** consisting of the name of the attribute followed by the name of the system entity.
    - For example, “AuditorUser” would indicate that “Auditor” is the attribute and “User” is the reference to the User table.
- a. Save the form and select **Yes** in response to the following prompt:



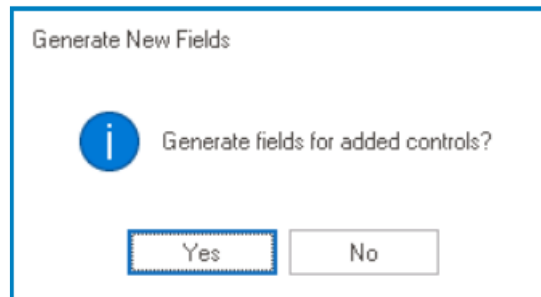
! In the Entity Designer, users will find the mid-entity created with its two foreign keys linking it to the main entity and specified system entity, respectively. Note that the caption (name of attribute) before the system entity is used primarily to keep track of which attribute is being referred to in the database.


### B. Referencing Data from Another Form

1. Drag a multi-select dropdown control from the toolbox and drop it onto the design environment.
2. Provide a caption for the control.
3. Specify the corresponding custom entity in the **Server ID** field to associate the control with a custom table in the database.
  - Enter a **Server ID** consisting of the name of the attribute followed by the name of the custom entity.
  - For example, “NameProduct” would indicate that “Name” is the attribute and “Product” is the reference to the custom entity.



4. Save the form and select **Yes** in response to the following prompt:



5. Reopen the form and access the Control Properties panel.
6. In the Control Properties panel, navigate to the **Binding** category.
7. From the **Binding** category, access the Control Properties window.
8. In the Control Properties window, navigate to the Single-Select tab.
9. Define a new dropdown data source by clicking on the  button. It will display the [Query Editor](#) window.
10. To call the correct information, create an SQL query.
  - Construct your query using the appropriate syntax and logic.
    - Sample Query: `select id, [Attribute] from ray.BPMAPP_[Entity]` (that is, the entity from which data is being pulled).
      - Example: `select id, Name from ray.BPMAPP_Product`



In the Entity Designer, users will find the custom entity now added to the database. They will also find the mid-entity created with its two foreign keys linking it to the main entity and custom entity, respectively.

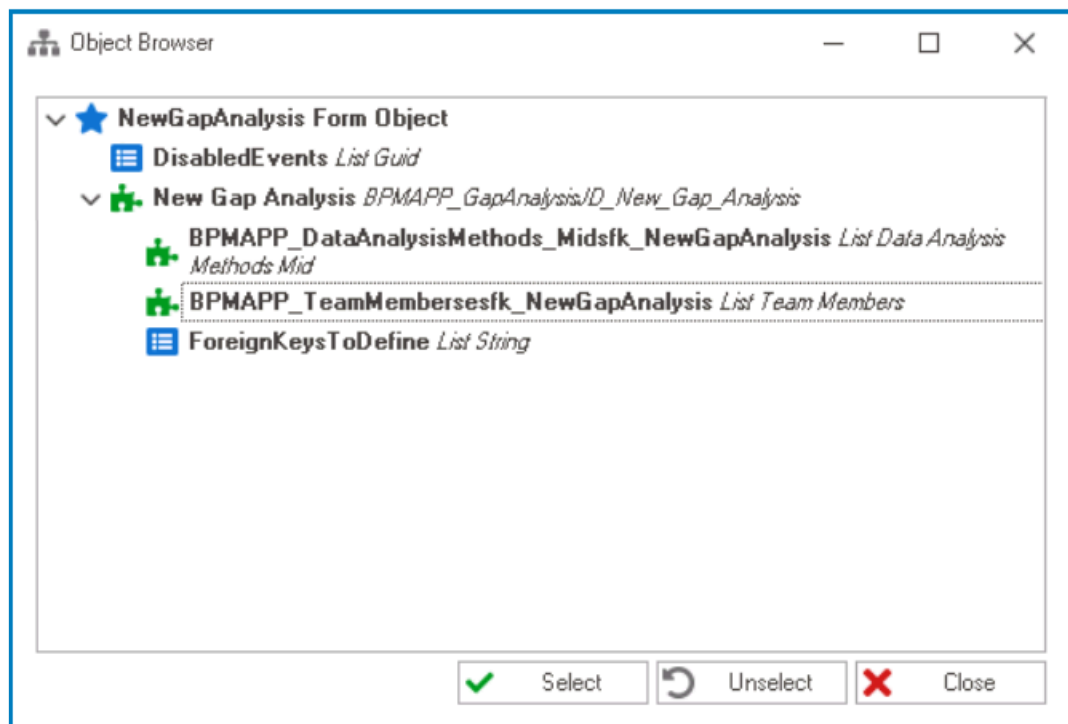


For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.18.2. Multi-Select Dropdown – Binding

In the **Binding** category, users can link a multi-select dropdown to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the multi-select dropdown control.
  - To access the list of relevant attributes, click on the ellipses (...) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the multi-select dropdown control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute



Users must bind the multi-select dropdown control to the foreign key linked to the main



entity. For more information on this matter, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. It will display the Control Properties window.
  - This window consists of three tabs:
    1. [General](#)
    2. [Single-Select](#)
    3. [Multi-Select](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Control Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

# 12.1.2.1.18.2.1. Multi-Select Dropdown – Attributes

---

To create a multi-select dropdown control, users need to have a main entity, a mid-entity (table), and either a system or custom entity. The mid-entity should have two foreign-key fields, one connected to the main entity and one connected to the system/custom entity, with a one-to-many relationship.

## 1. Main Entity

The main entity is the entity associated with the form where the multi-select dropdown control is being used. It represents the primary data object that the user is working with or collecting data for.

## 2. System Entity

A system entity is a pre-existing entity that references system tables such as users, roles, or departments. It allows data to be associated with these pre-existing entities, which are not specific to the main entity and do not require the user to create them.

## 3. Custom Entity

A custom entity is an entity that is linked to another form from which data will be pulled. It is used to associate data with another form that is not related to the main entity.

## 4. Mid-Entity

A mid-entity (table), by way of its foreign keys, serves as an intermediary between the main entity and the system or custom entity.

- To link the mid-entity to a system entity, please refer to the [corresponding section](#) in the manual.
- To link the mid-entity to a custom entity, please refer to the [corresponding section](#) in the manual.

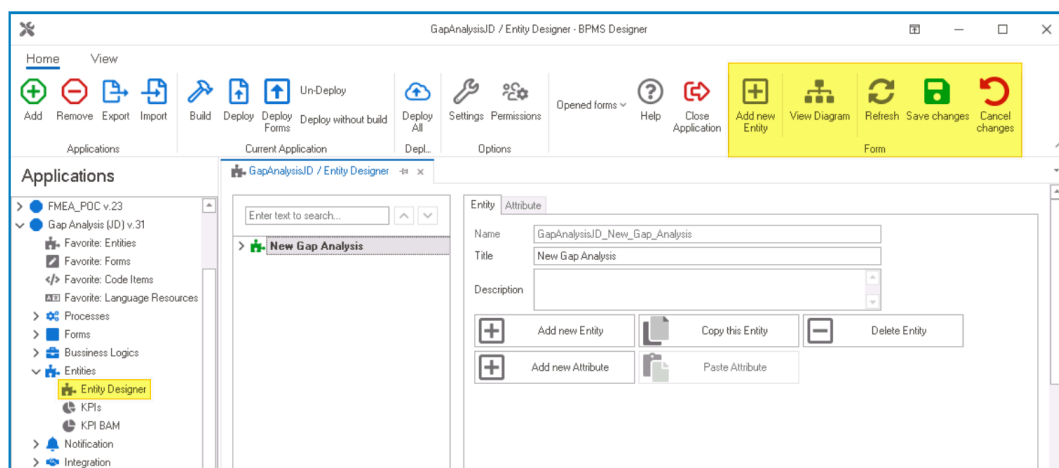
## 12.1.2.1.1.18.2.1.1. Link to System Entity

To link the mid-entity to a system entity, take the following steps:

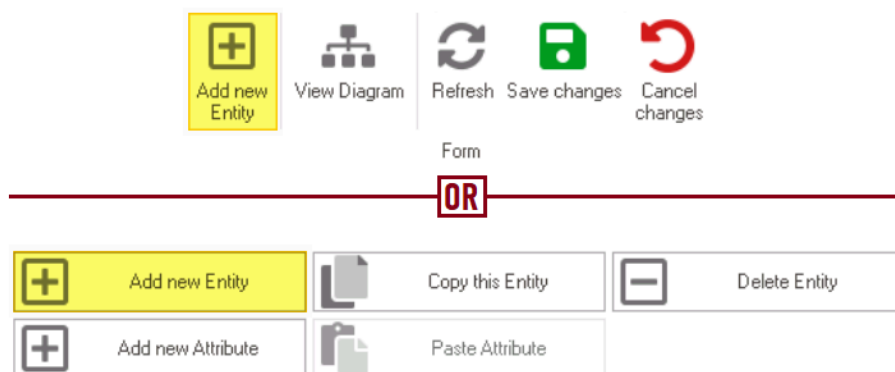
### 1. [Create Mid-Entity \(Table\)](#)

#### • Access Entity Properties

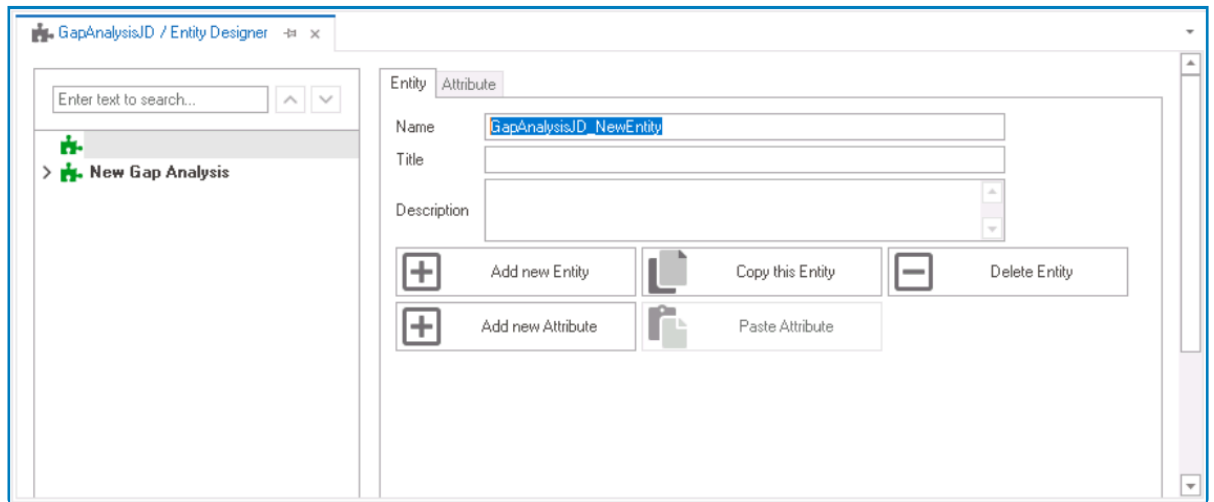
- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.



- In the **Name** field, provide a name for the mid-entity.
- In the **Title** field, either specify a custom title for the mid-entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the mid-entity, if needed.

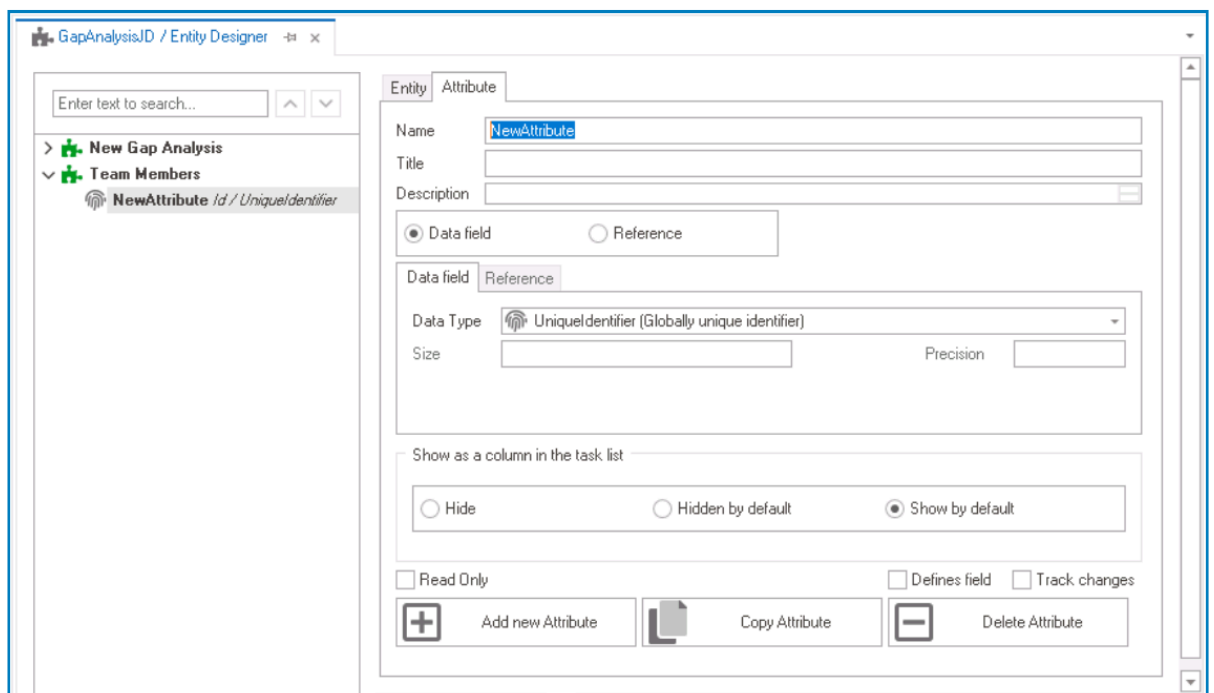
## 2. Create Foreign Keys

### • Access Attribute Properties


- Once you have created the mid-entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

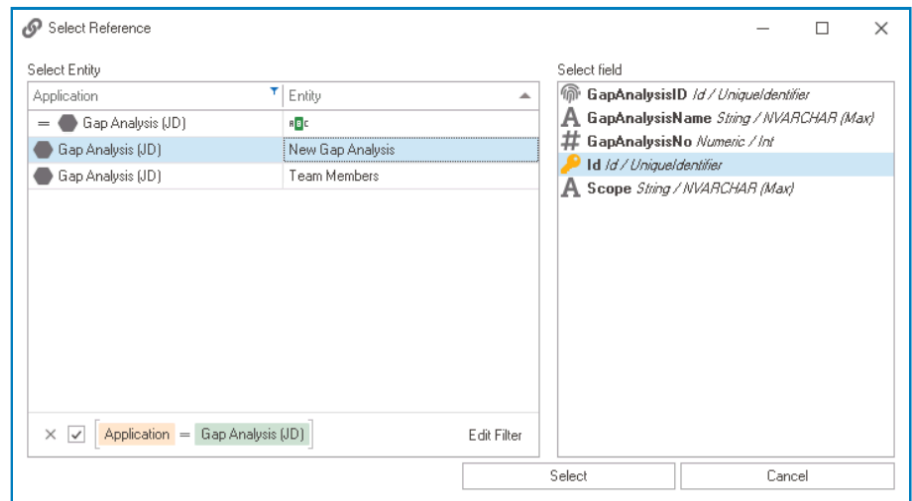


- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

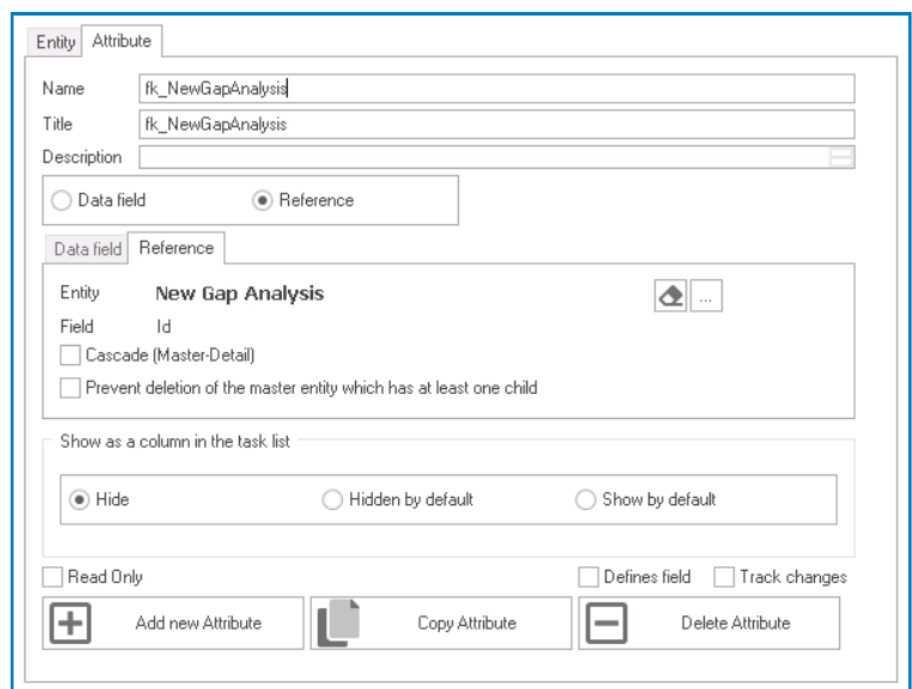


- **Foreign Key Towards Main Entity**

- This foreign key is linked to the entity associated with the form where the multi-select dropdown control is being implemented.
  - In the **Name** field, provide a name for the foreign key.
  - In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [MainEntity]**. From the field section, select **ID**.




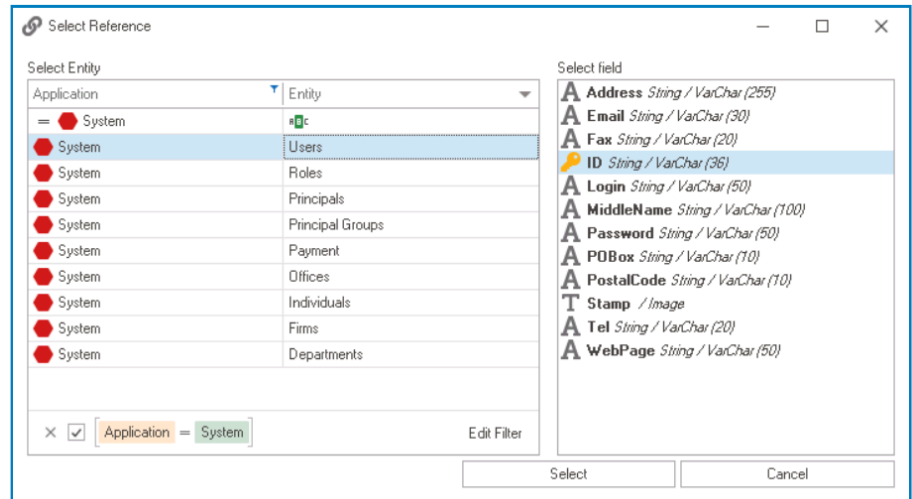
- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[MainEntity]** and the field section will display **ID**.



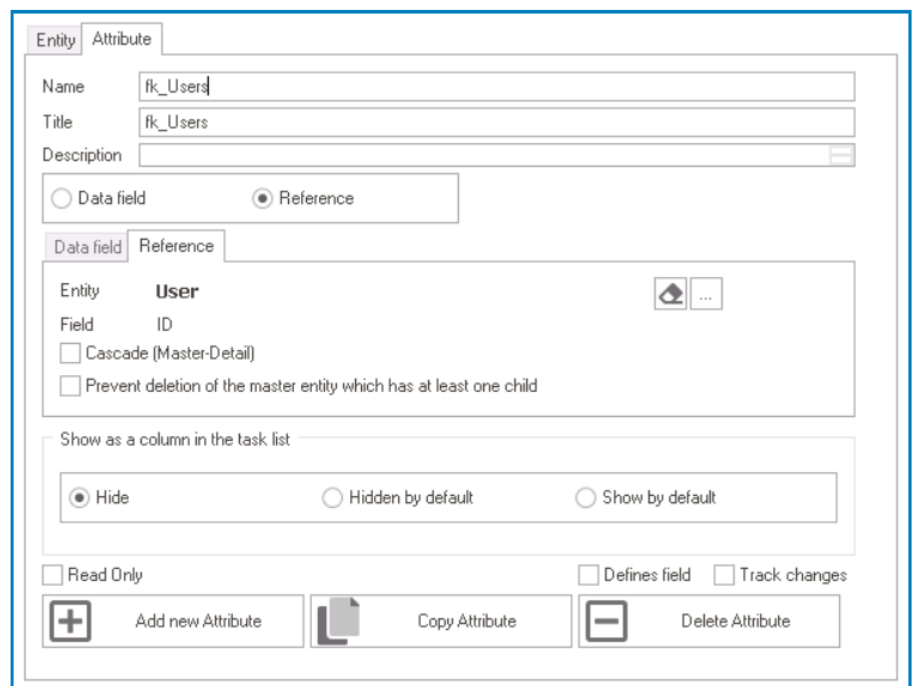
- **Foreign Key Towards System Entity**

- This foreign key is linked to the system entity—that is, the system table existing in the database.
  - In the **Name** field, provide a name for the foreign key.

- In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
- Choose the **Reference** option.
- Click on the  button. It will display the Select Reference window. From the entity section, select **System – [SystemReference]**. From the field section, select **ID**.



- Selecting the system reference will redirect to the Attribute tab, where the entity section will display **[SystemReference]** and the field section will display **ID**.



- Once users have created the necessary foreign keys for the mid-entity, the end-result should resemble the following list:



! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

\* After setting up the mid-entity, users can return to their main form and [define the binding properties for the multi-select dropdown control](#).

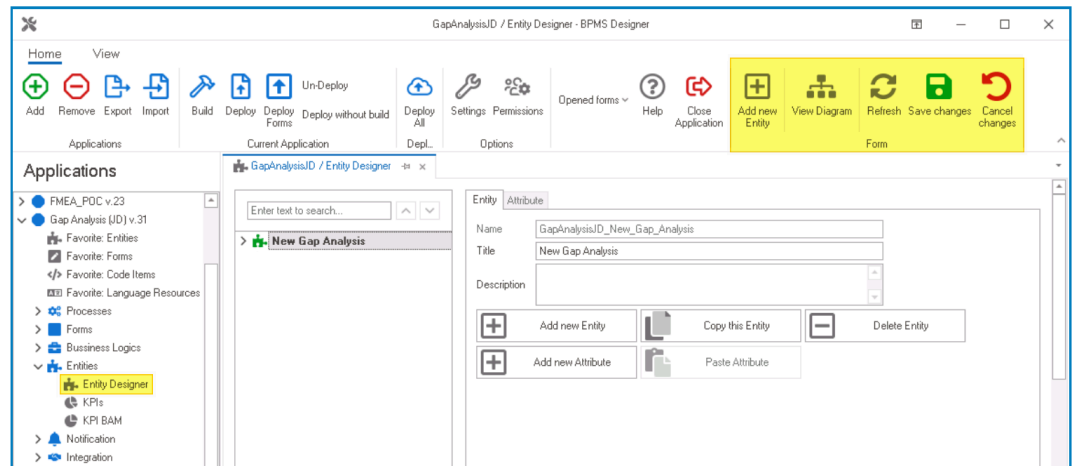
## 12.1.2.1.18.2.1.2. Link to Custom Entity

To link the mid-entity to a custom entity and set up the corresponding form(s) for the custom entity, take the following steps:

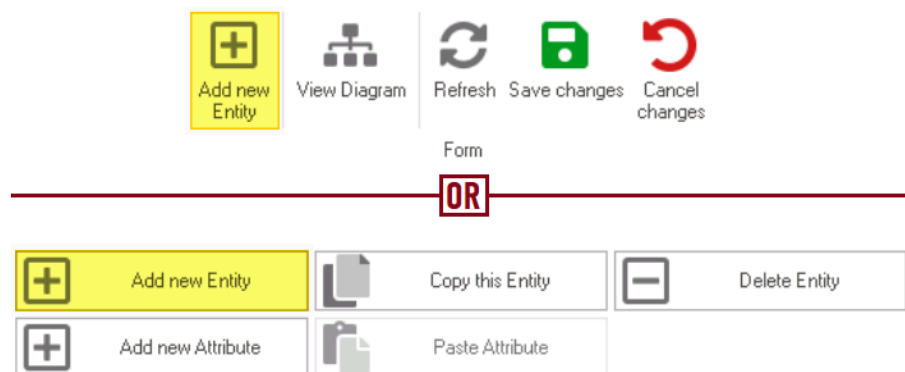
### 1. [Create Custom Entity](#)

- **Access Entity Properties**

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.



GapAnalysisJD / Entity Designer

Entity Attribute

Name: GapAnalysisJD\_NewEntity

Title:

Description:

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

## 2. [Create Attributes](#)

### • Access Attribute Properties

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

+ Add new Entity    Copy this Entity    - Delete Entity

+ Add new Attribute    Paste Attribute

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

GapAnalysisJD / Entity Designer

Entity Attribute

Name: NewAttribute

Title:

Description:

☒ Data field    ☐ Reference

Data field Reference

Data Type: Uniquelidentifier (Globally unique identifier)

Size:    Precision:   

Show as a column in the task list

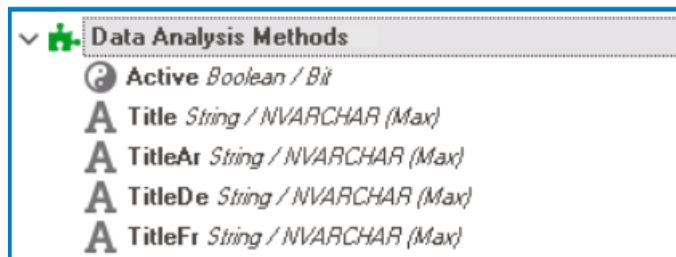
☐ Hide    ☐ Hidden by default    ☒ Show by default

☐ Read Only    ☐ Defines field    ☐ Track changes

+ Add new Attribute    Copy Attribute    - Delete Attribute

- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose an attribute type— **Data Field** or **Reference**.
  - For the **Data Field** attribute, select the appropriate data type from the dropdown.
  - For the **Reference** attribute, select the appropriate reference from the System Reference window.

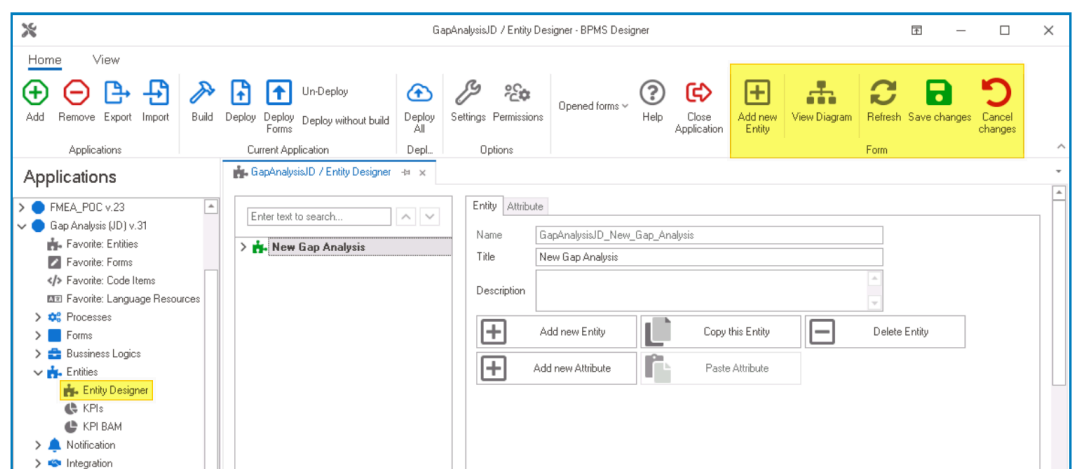
### • Custom Entity Attributes Example



## 3. Create Mid-Entity (Table)

### • Access Entity Properties

- From the Application Tree, navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.

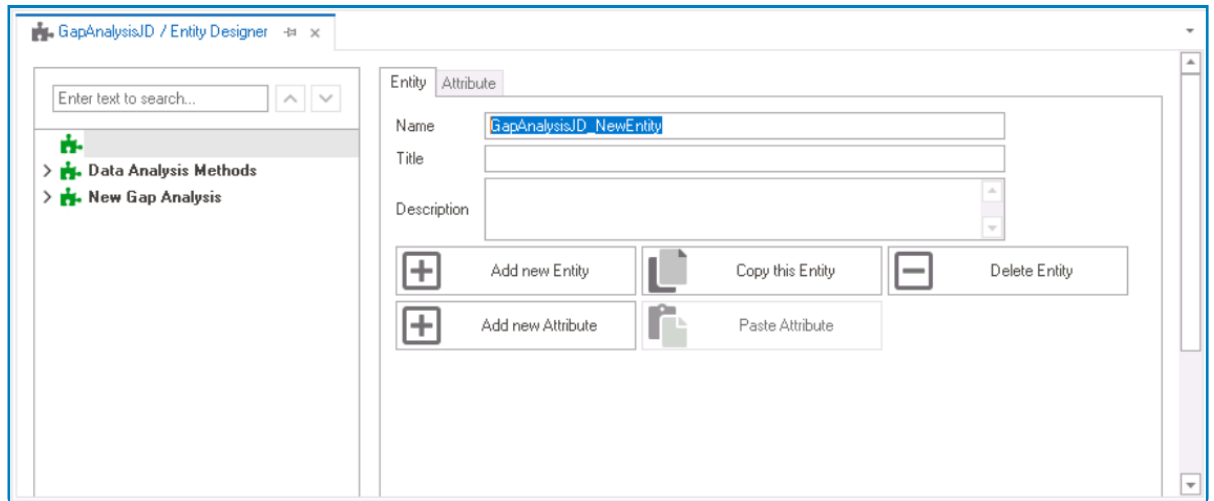


Form

**OR**



- In the fields of the **Entity** tab, users can define the properties of the newly created entity.



- In the **Name** field, provide a name for the mid-entity.
- In the **Title** field, either specify a custom title for the mid-entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the mid-entity, if needed.

#### 4. Create Foreign Keys

- **Access Attribute Properties**

- Once you have created the mid-entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.



- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' window. On the left is a tree view with 'New Gap Analysis' and 'Team Members' expanded, showing 'NewAttribute Id / UniqueIdentifier'. The main area has two tabs: 'Entity' and 'Attribute'. The 'Attribute' tab is active, showing fields for 'Name' (NewAttribute), 'Title', and 'Description'. Below these are radio buttons for 'Data field' (selected) and 'Reference'. Under 'Data field', there are sub-tabs for 'Data field' and 'Reference'. The 'Data field' sub-tab is active, showing 'Data Type' as 'UniqueIdentifier (Globally unique identifier)', 'Size', and 'Precision'. Below this is a section 'Show as a column in the task list' with radio buttons for 'Hide', 'Hidden by default', and 'Show by default' (selected). At the bottom are checkboxes for 'Read Only', 'Defines field', and 'Track changes', and three buttons: '+ Add new Attribute', 'Copy Attribute', and '- Delete Attribute'.

- **Foreign Key Towards Main Entity**


- This foreign key is linked to the entity associated with the form where the multi-select dropdown control is being implemented.
  - In the **Name** field, provide a name for the foreign key.
  - In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the button. It will display the Select Reference window. From the entity section, select **[AppName] – [MainEntity]**. From the field section, select **ID**.

The screenshot shows the 'Select Reference' dialog box. It has two main sections: 'Select Entity' and 'Select field'. In the 'Select Entity' section, there is a table with columns 'Application' and 'Entity'. The 'Application' column has a dropdown set to 'Gap Analysis (JD)'. The 'Entity' column has a dropdown set to 'New Gap Analysis'. Below the table is a filter bar with 'Application = Gap Analysis (JD)' and an 'Edit Filter' button. In the 'Select field' section, there is a list of fields: 'GapAnalysisID Id / UniqueIdentifier', 'GapAnalysisName String / NVARCHAR (Max)', 'GapAnalysisNo Numeric / Int', 'Id Id / UniqueIdentifier' (highlighted), and 'Scope String / NVARCHAR (Max)'. At the bottom are 'Select' and 'Cancel' buttons.

- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[MainEntity]** and the field section will display **ID**.

The screenshot shows the 'Attribute' configuration window. The 'Name' and 'Title' fields both contain 'fk\_NewGapAnalysis'. The 'Description' field is empty. The 'Data field' and 'Reference' radio buttons are present, with 'Reference' being the selected option. Below this, the 'Data field' and 'Reference' tabs are visible. Under the 'Reference' tab, the 'Entity' is set to 'New Gap Analysis' and the 'Field' is 'Id'. There are checkboxes for 'Cascade (Master-Detail)' and 'Prevent deletion of the master entity which has at least one child', both of which are currently unchecked. A section titled 'Show as a column in the task list' contains three radio buttons: 'Hide' (selected), 'Hidden by default', and 'Show by default'. At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', all of which are unchecked. Three buttons are at the bottom: 'Add new Attribute' (with a plus icon), 'Copy Attribute' (with a document icon), and 'Delete Attribute' (with a minus icon).

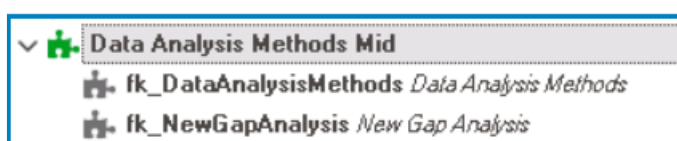
- **Foreign Key Towards Custom Entity**

- This foreign key is linked to the custom entity—that is, the entity linked to the form from which data will be pulled.
  - In the **Name** field, provide a name for the foreign key.
  - In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the  button. It will display the Select Reference window. From the entity section, select [AppName] – [CustomEntity]. From the field section, select ID.

The screenshot shows the 'Select Reference' dialog box. It has two main sections: 'Select Entity' and 'Select field'. In the 'Select Entity' section, there is a table with columns 'Application' and 'Entity'. The 'Application' column has a dropdown menu set to 'Gap Analysis (JD)'. The 'Entity' column lists several entities: 'Data Analysis Methods', 'Data Analysis Methods Mid', and 'New Gap Analysis'. The 'New Gap Analysis' entity is selected. In the 'Select field' section, there is a list of fields: 'Active Boolean / Bit', 'Id Id / UniqueIdentifier', 'Title String / NVARCHAR (Max)', 'TitleAr String / NVARCHAR (Max)', 'TitleDe String / NVARCHAR (Max)', and 'TitleFr String / NVARCHAR (Max)'. The 'Id Id / UniqueIdentifier' field is selected. At the bottom, there is a filter bar with a checkbox and the text 'Application = Gap Analysis (JD)'. There are 'Select' and 'Cancel' buttons at the bottom right.

- Selecting the entity will redirect to the Attribute tab, where the entity section will display [CustomEntity] and the field section will display ID.

- Once users have created the necessary foreign keys for the mid-entity, the end-result should resemble the following list:

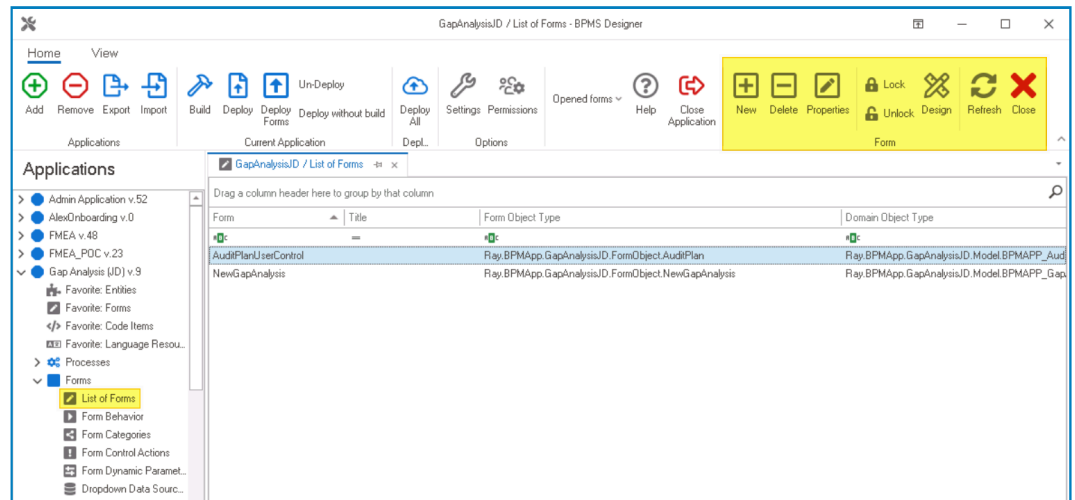


**!** Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

## 5. [Create and Design Custom Form\(s\)](#)

### • Access Form Properties

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

**Form Properties**

Apply OK Cancel

Name:

Form Entity:

Entity: [Please Select...] ▼

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

✿ To define and customize the options (for a multi-select dropdown control) in a grid, users will need to create two separate forms. One form should contain the necessary attributes, while the other should contain the grid itself. **For more**

information on this matter, [click here](#).

- \* Once users have created and designed the custom form(s), they can return to their main form and [define the binding properties for the multi-select dropdown control](#).



## 12.1.2.1.18.2.2. Control Properties – Single-Select Tab

In the Single-Select tab, users can configure and customize various aspects of the dropdown control, including its data source, loading behavior, display options, search functionality, and search type.

1. **Dropdown Data Source:** This field retrieves the list of dropdown data sources available in either the application or system. Depending on the data source chosen in this field, the available options for selection in the single-select dropdown control will vary.
  - To define a new dropdown data source, select the button.
    - It will display the [Query Editor](#) window, in which users can configure static or dynamic items.
  - To modify an existing dropdown data source, select the button.
  - To remove a dropdown data source, select the button.
    - This button will only be visible once users have added/selected a dropdown data source.

\* If sourced from within the application, the dropdown data sources in this field will be retrieved from the [Dropdown Data Sources](#) section of the **Forms** module. If sourced from

the system itself, they will be retrieved from the database directly.

2. **Load Data Asynchronously:** This is a checkbox which, if selected, allows the dropdown data to be retrieved in the background while the user interacts with other parts of the application.
  - Selecting this checkbox will speed up the display of dropdown options, improving performance.
3. **Sort By Display Member:** This is a checkbox which, if selected, allows users to sort the dropdown options by their display values, instead of their original order in the data source.
  - Users can specify the display member by editing the query in the [Query Editor](#) window
4. **Page Size:** This field allows users to set the number of items that are displayed in the dropdown at one time.
  - By default, the size is set to 1000. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.



When the 'More Items' option is selected, the dropdown will update to display the number of items according to the number specified in this field, and will continue to update by that amount for subsequent selections until all items are displayed.

5. **Load Delay (ms):** This field allows users to set the amount of time, in milliseconds, that the multi-select dropdown control will wait before loading or displaying its options.
  - By default, this value is set to 100. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
6. **Min Characters for Search:** This field allows users to specify the minimum number of characters a user must input into the search field before the search function is activated.
  - By default, the minimum value is set to 0. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.



By setting a minimum value in this field, users can not only help prevent unnecessary searches, but they can also ensure that a search query is specific enough to return relevant results.

7. **Search Type:** This field displays the list of search types. It allows users to choose how the search function will match their search query with the available options.
  - Search Types: Contains, Starts With, or Ends With.
    - Contains: This option will return results that include the search query anywhere in the text.
    - Starts With: This option will return results that begin with the search query.
    - End With: This option will return results that end with the search query.
8. **Apply:** This button allows users to save their changes.
9. **OK:** This button allows users to save their changes and exit the [Control Properties](#) window.
10. **Cancel** This button allows users to cancel their changes and exit the [Control Properties](#) window.

## 12.1.2.1.18.2.3. Control Properties – Multi-Select Tab




In the Multi-Select tab, users can select the relation property.


The screenshot shows the 'Control Properties' dialog box with the 'Multi select' tab selected. The dialog has a title bar with a maximize icon, the text 'Control Properties', and a close button. Below the title bar are three buttons: 'Apply' (with a green square icon), 'OK' (with a green checkmark icon), and 'Cancel' (with a red X icon). The main content area is divided into two sections. The top section contains three labels and their corresponding values: 'Form Control' is 'MultiSelectDropdown', 'Entity and Attribute' is 'DomainObject.BPMAPP\_TeamMembersesfk\_NewGapAnalysis', and 'Object type' is 'Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_TeamMembers'. The bottom section has three tabs: 'General', 'Single select', and 'Multi select'. The 'Multi select' tab is active, showing a 'Relation Property:' label followed by a dropdown menu with a downward arrow.

1. **Relation Property:** This field allows users to choose the relation property from a dropdown menu.
  - The relation property for a multi-select dropdown control usually corresponds to a field in the mid-entity.
    - To enable the multi-select dropdown control to reference a system table, select the foreign key linked to the system entity.
    - To enable the multi-select dropdown control to display data from another form, select the foreign key linked to the custom entity.
2. **Apply:** This button allows users to save their changes.
3. **OK:** This button allows users to save their changes and exit the Control Properties window.
4. **Cancel** This button allows users to cancel their changes and exit the Control Properties window.

## 12.1.2.1.18.3. Multi-Select Dropdown – Control Governance

In the Selector tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of multi-select dropdown controls.

1. **Browse Form Behavior:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'browse' functionality.
  - To define a new form behavior, select the  button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing form behavior, select the  button.
  - To remove a form behavior, select the  button.
    - This button will only be visible once users have added/selected a form behavior.

 By selecting a form behavior in this field, users are essentially linking the control to a modal window. **For more information on this feature, [click here](#).**

2. **Apply:** This button allows users to save their changes.
3. **OK:** This button allows users to save their changes and close the window.
4. **Cancel:** This button allows users to cancel their changes and close the window.



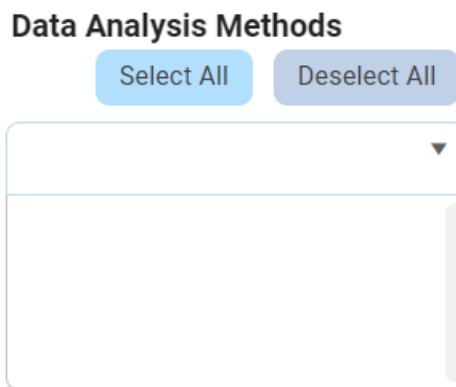
In the Validation tab of this window, users can configure the required field validation properties for the multi-select dropdown control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.18.4. Multi-Select Dropdown – Web Display

For those unfamiliar with using the multi-select dropdown field in the **DBP** web application, the following instructions will guide you through the process:

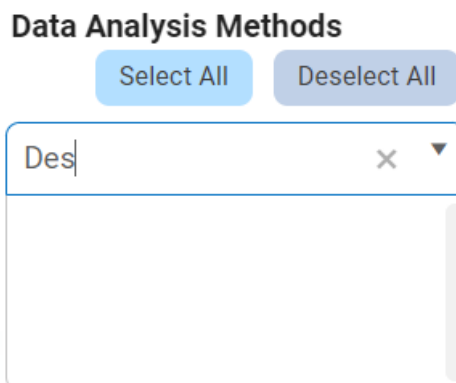
### Multi-Select Dropdown Field

A multi-select dropdown field allows users to select multiple values at a time from a predetermined list of options.



There are two ways to select items from the dropdown:

1. Click on the ▼ icon. It will release the list of values. Users can then scroll through the options and make their selection.
2. Search for the value in the dropdown by inputting its name or keywords relating to it. Any values that match the search will be retrieved. Users can then click on the desired value to finalize their selection.



After selecting items, users can remove them from the dropdown by clicking on the **X** button.

### Data Analysis Methods

Select All

Deselect All



To automatically select all items in the list, click on the **Select All** button. To automatically deselect all items in the list, click on the **Deselect All** button.

### Multi-Select Dropdown Field with Modal Window

If users have enabled the [browsing functionality](#) of the control, a modal window will be presented instead of the standard dropdown menu.

### Data Analysis Methods


Select All

Deselect All



A modal window is a pop-up window that disables the main form, requiring users to interact with it before they can regain access. Users can interact with a modal typically by either completing the set task or exiting the window.

To use a multi-select dropdown field linked to a modal window, follow these steps:

1. Click on the  icon. A modal window will appear, which typically contains a list of available options for selection.
2. Users can choose one or more options by clicking on them, or define new items for the dropdown as needed.
  - Once users have made their selections or defined the items, they will be returned to the main form where they will either find the field now populated with their selection(s) or find the new item listed in the dropdown menu.
3. Otherwise, to exit the modal window without making any changes, click on the **Exit (X)** button.

In the example below, the modal window contains a grid control which allows users to add to the items in the dropdown list.

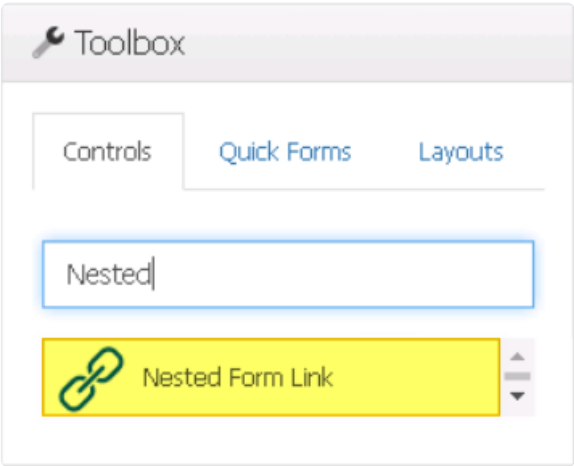


Users can determine the contents of the modal window by creating the form and linking the corresponding form behavior to the control.

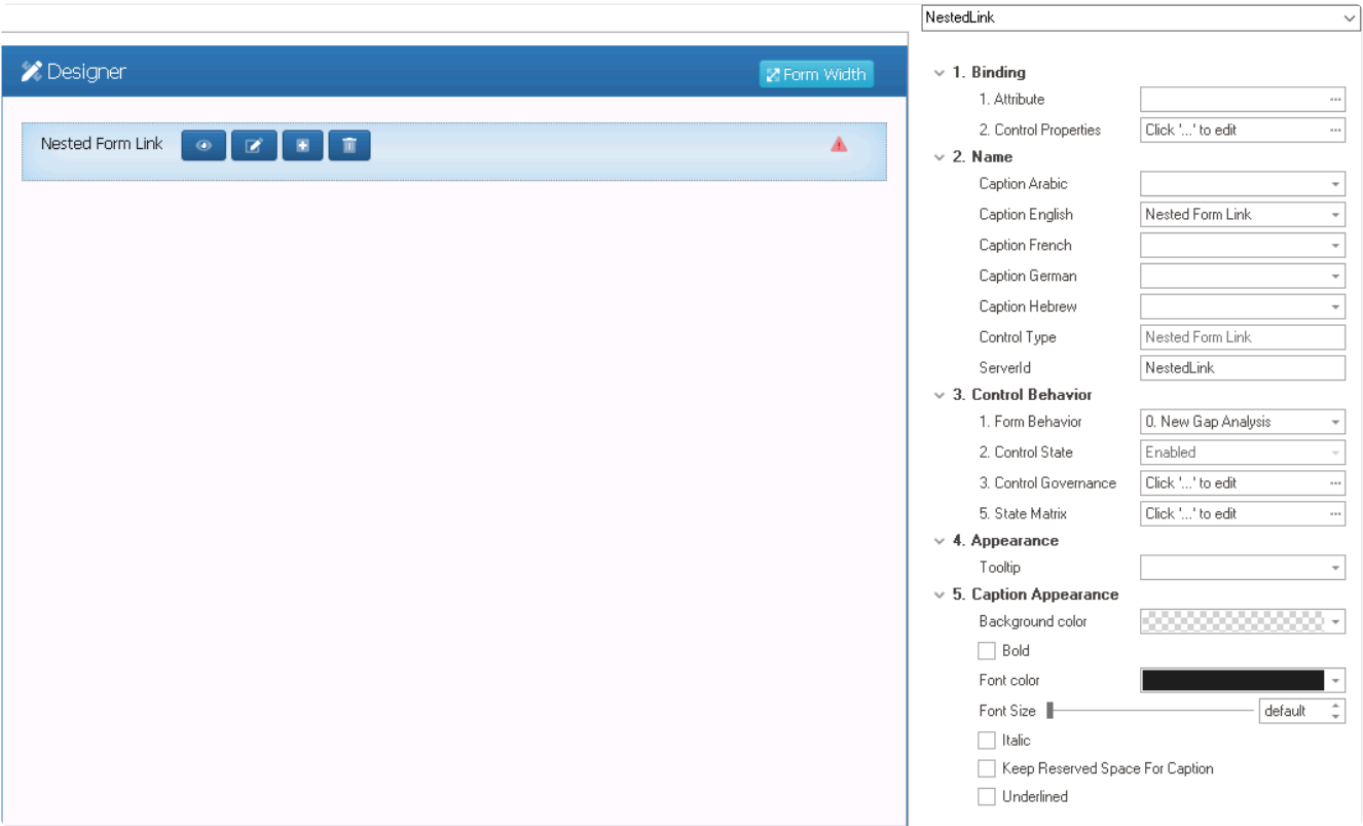



# 12.1.2.1.19. Nested Form Link

The nested form link control is a user interface element that allows users to link or associate one form with another form. It consists of a link or button that, when clicked, opens a nested form within the main form.



When a nested form link control is selected in the design environment, its associated properties are displayed in the Control Properties panel.




 To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).




 For more information on the required attributes for nested form link controls, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state properties set for the control will apply only to the selected form behavior.

 Only form behaviors linked to the associated form will be displayed in this dropdown.














- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **Control Governance:** This field allows users to configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field.




Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [NestedLink]


 Apply  OK  Cancel



Control Name **NestedLink**

**NestedLink**

|  |                     |         |   |
|--|---------------------|---------|---|
| <input type="checkbox"/>  Enable View   | View Form Behaviour | [Empty] |    |
| <input type="checkbox"/>  Enable Edit   | Edit Form Behaviour | [Empty] |    |
| <input type="checkbox"/>  Enable New    | New Form Behaviour  | [Empty] |    |
| <input type="checkbox"/>  Enable Delete |                     |         |   |

- **Enable View:** This is a checkbox which, if selected, allows users to enable the **View** function for the control.
  - If this checkbox is selected, it will enable the **View Form Behavior** field.
    - **View Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.
      - To remove a form behavior, select the  button.




 To set up the 'View Form', users only need to provide a name and a title.

- **Enable Edit:** This is a checkbox which, if selected, allows users to enable the **Edit** function for the control.
  - If this checkbox is selected, it will enable the **Edit Form Behavior** field.
    - **Edit Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'edit' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.

- To remove a form behavior, select the  button.






To set up the 'Edit Form', users only need to provide a name and a title.

- **Enable New:** This is a checkbox which, if selected, allows users to enable the **Add** function for the control.
  - If this checkbox is selected, it will enable the **New Form Behavior** field.
    - **New Form Behavior:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'new' functionality.
      - To define a new form behavior, select the  button.
        - It will display the [Form Behavior Properties](#) window.
      - To modify an existing form behavior, select the  button.
      - To remove a form behavior, select the  button.



To set up the 'New Form', users only need to provide a name, a title, and select the **Starter** checkbox.

- **Enable Delete:** This is a checkbox which, if selected, allows users to delete the added instance of the nested form.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.

| Control Behaviour <span style="float: right;">□ ×</span>   |               |                          |                          |                          |                          |                          |                          |                          |      |
|--|---------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|------|
| <div>  Apply            OK            Cancel         </div> |               |                          |                          |                          |                          |                          |                          |                          |      |
| Form Behavior  | Control State |                          | Add                      |                          | Edit                     |                          | Delete                   |                          | View |
| 1. Modify Gap Analysis   | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ...  |
| 2. New Gap Analysis  | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ...  |
| 3. Review Gap Analysis   | ✓ Enabled     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ...  |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.

- **Control Governance:** This field allows users to configure the control governance properties.
  - If selected, it will display the Form / Behavior / Control window (described above).
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.




By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.



In this window, users can also configure the governance properties for the control across multiple form behaviors. To gain an understanding of how these properties can be configured, [click here](#).

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon (  ) next to the control.
  - **Nested Form Link Tooltip Preview: Web View**

Click to open the Action form, where you can input additional information related to the current form.

Action



A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

**Nested Form Link Control Preview: Web View**

Action



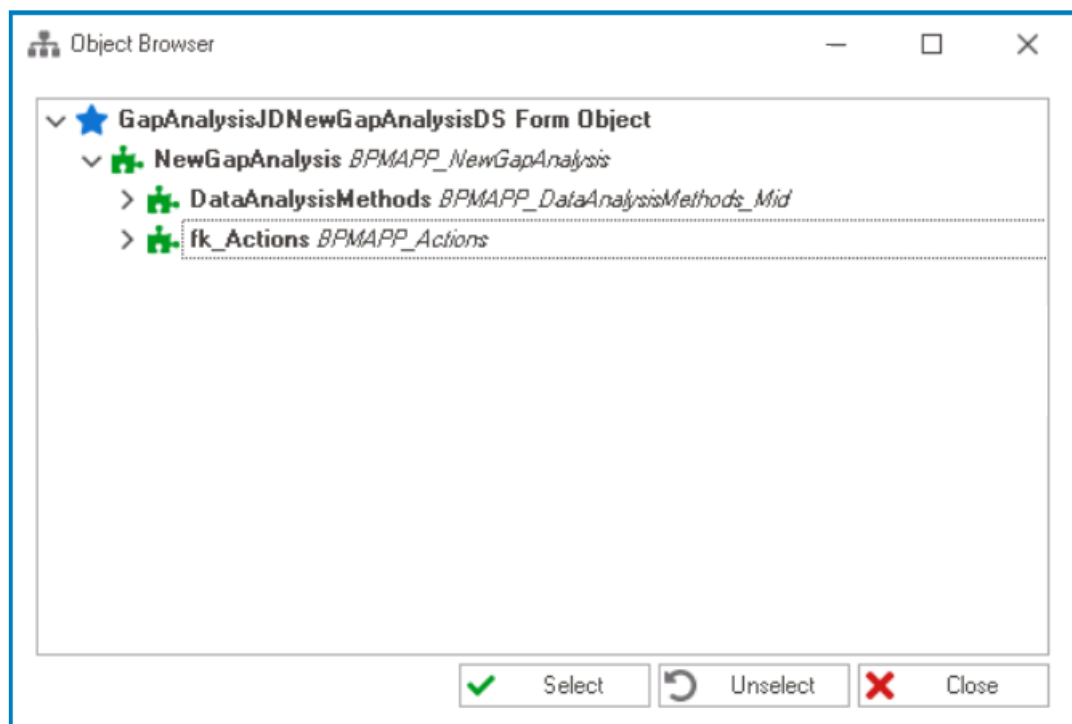


For more information on how to use this field in the web application, [click here](#).

## 12.1.2.1.19.1. Nested Form Link – Binding

In the **Binding** category, users can link a nested form link control to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the nested form link control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the nested form link control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute



Users must bind the nested form link control to the foreign key linked to the nested form. For more information on this matter, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. In the displayed window, navigate to the Nested Link tab.

The screenshot shows the 'Control Properties' dialog box with the 'Nested Link' tab selected. The dialog has a title bar with a close button (X) and three buttons: 'Apply' (green square), 'OK' (green checkmark), and 'Cancel' (red X). The main content area is divided into two sections. The top section contains three labels and their corresponding values: 'Form Control' is 'NestedLink', 'Entity and Attribute' is 'DomainObject.BPMAPP\_Actionsfk\_Actions', and 'Object type' is 'Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_Actions'. The bottom section has two tabs: 'General' and 'Nested Link', with 'Nested Link' being the active tab. Below the tabs is a 'Physical form' label followed by a dropdown menu showing 'Actions' with a downward arrow.

- From **Physical Form** dropdown, select the nested (or custom) form.
  - This form will be displayed within the main form when the nested form link control is activated or clicked by the user.



In the General tab of this window, users can configure business rules for the nested form link control. For more information on this matter, please [click here](#) and navigate to the relevant section.



## 12.1.2.1.1.19.1.1. Nested Form Link – Attributes

A nested form link control requires at least two entities: a main entity, in which the nested form link will be used, and a custom entity, which will contain the attributes for the nested (or custom) form.

- \* The main entity is the entity associated with the form where the nested form link control is being implemented. It represents the primary data object that the user is working with or collecting data for. A custom entity is an entity that is linked to another form from which data will be pulled. It is used to associate data with another form that is not related to the main entity.

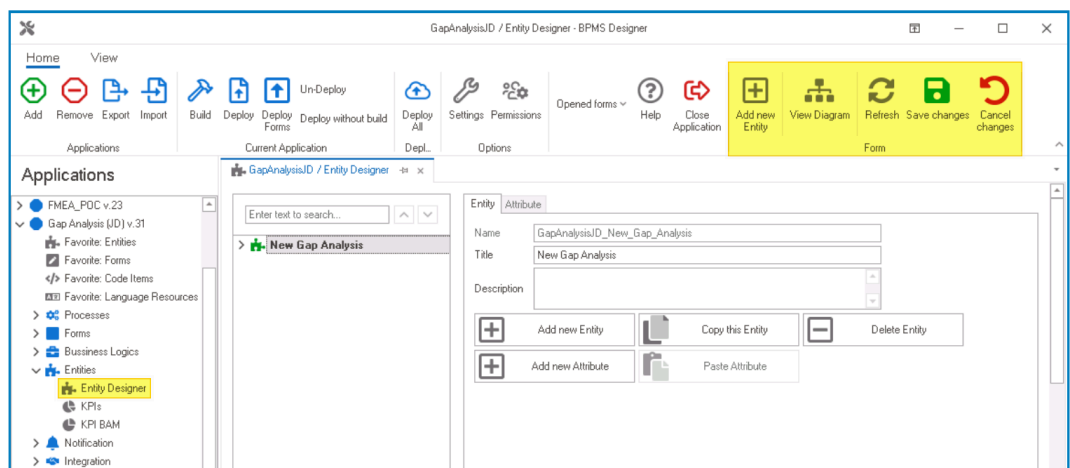
To create nested form link controls and their associated attributes, follow these steps:

- \* Before following the set of steps listed below, please create the main entity and its attributes.

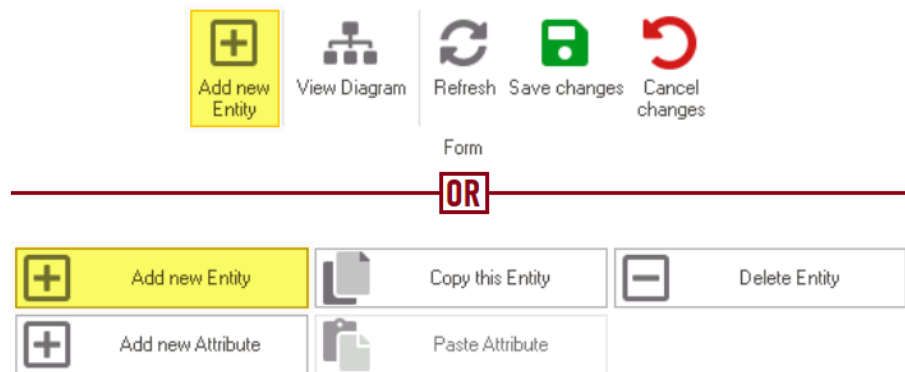
### 1. Create Custom Entity

- **Access Entity Properties**

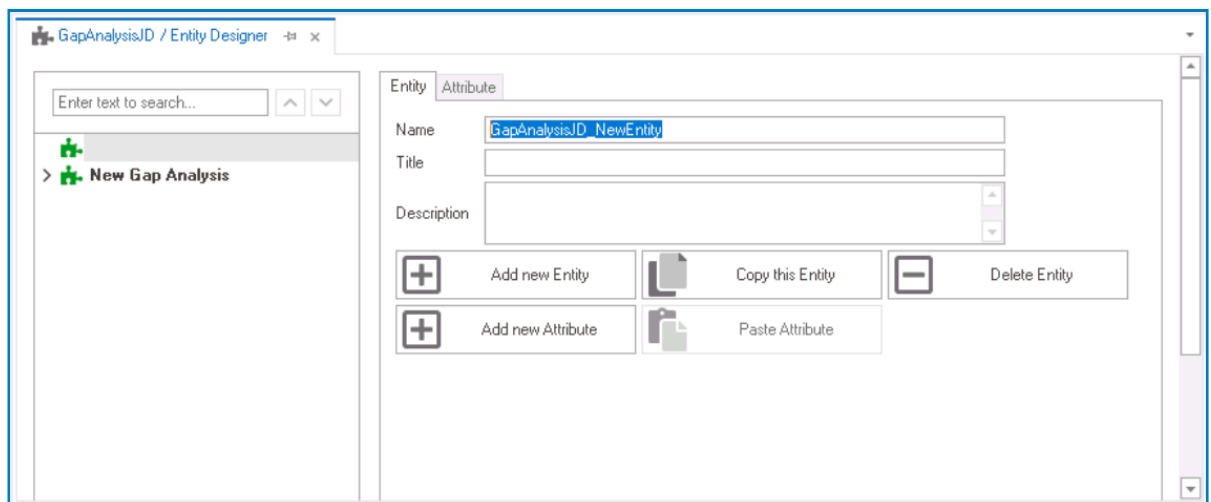
- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.



- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

## 2. [Create Custom Attributes](#)

### • Access Attribute Properties

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.



- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' window. On the left is a sidebar with a search bar and a tree view containing 'Actions', 'NewAttribute Id / UniqueIdentifier', and 'New Gap Analysis'. The main area has two tabs: 'Entity' and 'Attribute'. The 'Attribute' tab is active, showing a form for 'NewAttribute'. The form includes fields for 'Name' (filled with 'NewAttribute'), 'Title', and 'Description'. Below these are radio buttons for 'Data field' (selected) and 'Reference'. Under the 'Data field' section, there's a 'Data Type' dropdown set to 'Uniquel identifier (Globally unique identifier)', and 'Size' and 'Precision' input fields. A section 'Show as a column in the task list' has radio buttons for 'Hide', 'Hidden by default', and 'Show by default' (selected). At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', and three buttons: '+ Add new Attribute', 'Copy Attribute', and '- Delete Attribute'.

- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose an attribute type— **Data Field** or **Reference**.
  - For the **Data Field** attribute, select the appropriate data type from the dropdown.
  - For the **Reference** attribute, select the appropriate reference from the System Reference window.

### • Custom Entity Attributes Example

The screenshot shows a list of attributes for an entity named 'Actions'. The attributes are: '# ActionID Numeric / Int', 'A ActionName String / NVARCHAR (Max)', 'A ActionItemOwner User', 'A Description String / NVARCHAR (Max)', 'A Frequency String / NVARCHAR (Max)', and 'A StartDate Date/Time / Date'. Each attribute is preceded by a small icon representing its data type.

## 3. Create Foreign Key (Under Main Entity)

### • Access Attribute Properties


- Once you have created the custom entity and its attributes, return to the main entity. Select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

The screenshot shows the 'Entity' tab with a row of buttons: '+ Add new Entity', 'Copy this Entity', and '- Delete Entity'. Below this row are two more buttons: '+ Add new Attribute' (highlighted in yellow) and 'Paste Attribute'.

- In the fields of the Attribute tab, users can define the properties of the newly created

attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' window with the 'Attribute' tab selected. The 'Name' field is set to 'NewAttribute'. The 'Data field' radio button is selected. The 'Data Type' is set to 'Uniquelidentifier (Globally unique identifier)'. The 'Show as a column in the task list' section has 'Show by default' selected. At the bottom, there are buttons for 'Add new Attribute', 'Copy Attribute', and 'Delete Attribute'.

- In the **Name** field, provide a name for the foreign key.
- In the **Title** field, either specify a custom title for the foreign key or let the system automatically generate it based on the provided name.
- Choose the **Reference** option.
- Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [CustomEntity]**. From the field section, select **ID**.

The screenshot shows the 'Select Reference' window. In the 'Select Entity' section, 'Gap Analysis (JD)' is selected under the 'Entity' column. In the 'Select field' section, 'Id / Uniquelidentifier' is selected. At the bottom, there is a filter bar showing 'Application = Gap Analysis (JD)' and buttons for 'Select' and 'Cancel'.

- Selecting the entity will redirect to the Attribute tab, where the entity section will display **[CustomEntity]** and the field section will display **ID**.

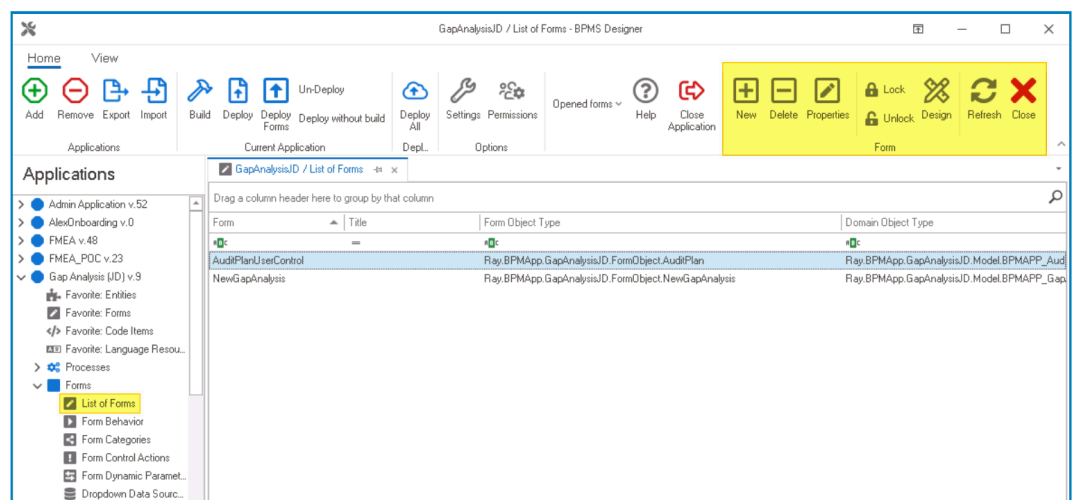


Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

#### 4. [Create & Design Custom Form](#)

- **Access Form Properties**

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the custom entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.

✿ Design the form by adding the necessary controls (and their corresponding attributes) in a clear and organized layout.

##### 5. Configure Nested Form Link Control Binding Properties

- After creating and configuring the custom form, users can return to their main form and define the binding properties for the nested form link control.
- **Bind Nested Form Link Control to Attribute**
  - A nested form link must be connected to the foreign key associated with the custom entity.
- **Configure Binding Properties**
  - To access the control properties, click on the ellipses (...) inside the field. In the displayed window, navigate to the Nested Link tab.




The screenshot shows the 'Control Properties' dialog box with the 'Nested Link' tab selected. The dialog has a title bar with a close button (X) and a toolbar with 'Apply', 'OK', and 'Cancel' buttons. The main content area is divided into two sections. The top section displays the following properties:

|                             |   |
|-----------------------------|---|
| <b>Form Control</b>         | NestedLink                                    |
| <b>Entity and Attribute</b> | DomainObject.BPMAPP_Actionsfk_Actions         |
| <b>Object type</b>          | Ray.BPMApp.GapAnalysisJD.Model.BPMAPP_Actions |

Below this, there are two tabs: 'General' and 'Nested Link'. The 'Nested Link' tab is active, showing a 'Physical form' dropdown menu with 'Actions' selected. The rest of the dialog box is empty.














- From the **Physical Form** dropdown, select the custom form.
  - Click on the **OK** button to save your changes and exit the window.
- **Configure Control Governance Properties**
    - To access the control governance properties, click on the ellipses (...) inside the field.

Form [NewGapAnalysis] / Behaviour [New Gap Analysis] / Control [NestedLink]

 Apply  OK  Cancel

Control Name **NestedLink**

**NestedLink**

|  |                     |         |   |
|--|---------------------|---------|---|
| <input type="checkbox"/>  Enable View   | View Form Behaviour | [Empty] |    |
| <input type="checkbox"/>  Enable Edit   | Edit Form Behaviour | [Empty] |    |
| <input type="checkbox"/>  Enable New    | New Form Behaviour  | [Empty] |    |
| <input type="checkbox"/>  Enable Delete |                     |         |   |


- To enable a functionality, users must link a corresponding form behavior.
  - To set up a form behavior, users only need to provide a name and a title.
  - When creating the form behaviors for the 'New' functionality, users must select the **Starter** checkbox.
- Click on the **OK** button to save your changes and exit the window.
- Click **Save Form** and deploy the application to the web.




## 12.1.2.1.19.2. Nested Form Link – Web Display

For those unfamiliar with using the nested form link control in the **DBP** web application, the following instructions will guide you through the process:


### Action

- : This button allows users to create an instance of the nested form.
  - If selected, it will open a modal window with the nested form or '**New Form**'.
    - To save the instance and return to the main form, click on the **Save and Return** button at the bottom of the form.
  - Once users have saved their work and returned to the main form, they will be able to access the control's **View**, **Edit**, and **Delete** functions.

! Users can configure the '**New Form**' in the Control Governance tab. The '**New Form**' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the nested form attributes.


- : This button allows users to view the instance of the nested form.
  - If selected, it will open a modal window with the '**View Form**'.
    - In viewing mode, users cannot make any modifications.

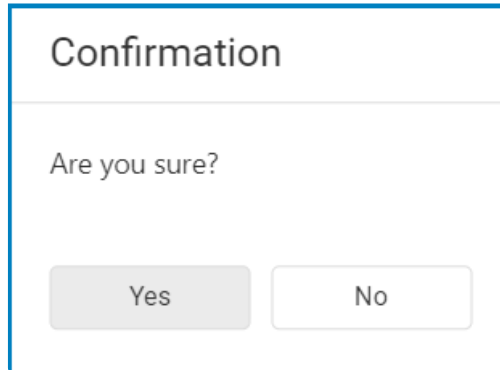
! Users can configure the '**View Form**' in the Control Governance tab. The '**View Form**' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the nested form attributes.

- : This button allows users to edit the instance of the nested form.
  - If selected, it will open a modal window with the '**Edit Form**'. This form is typically a duplicate of the '**New Form**', with the exception being that the fields are all populated with previously inputted information.
    - To save your changes, click on the **Apply** button at the bottom of the form.

! Users can configure the '**Edit Form**' in the Control Governance tab. The '**Edit Form**' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note,

however, that the actual contents and design of the form are pulled from the custom form that contains the nested form attributes.

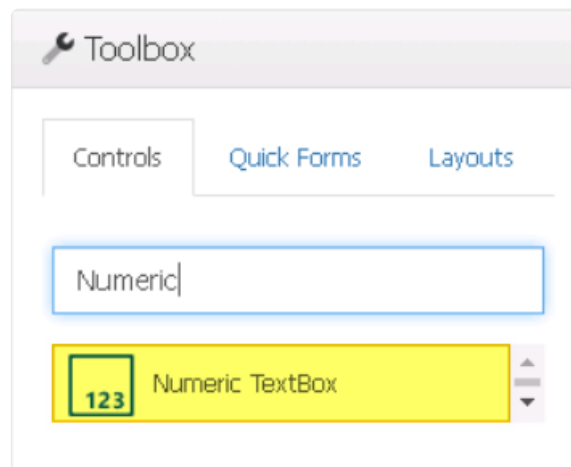
4.  : This button allows users to delete the instance of the nested form.
- If selected, users will be alerted of the action with the following confirmation message:

A confirmation dialog box with a blue border. The title bar is light gray and contains the text "Confirmation". Below the title bar is a horizontal line. The main area is white and contains the text "Are you sure?". At the bottom, there are two buttons: "Yes" (gray) and "No" (white with a gray border).

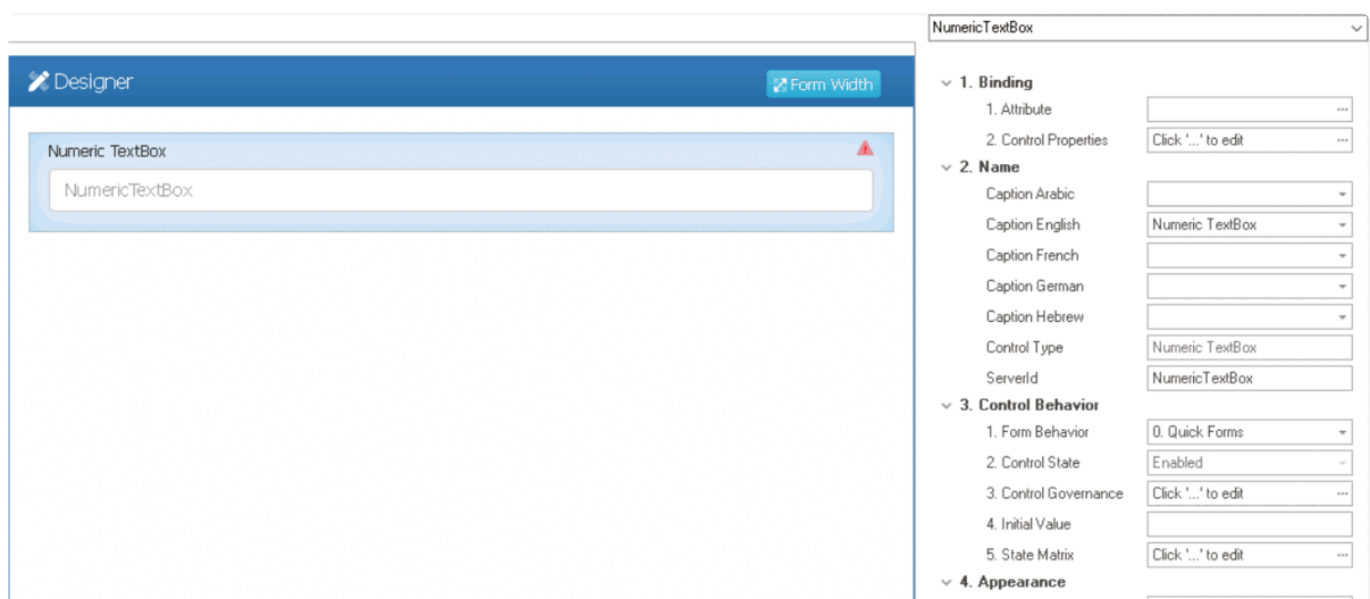
- Click on either the **Yes** button to confirm the deletion or the **No** button to cancel it.

## 12.1.2.1.1.20. Numeric Textbox

Numeric textbox controls are interactive elements that allow users to input, modify, and display numerical data in forms.



When a numeric textbox control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

✿ To create attributes for numeric textbox controls, users must choose numeric data types such as **integers** or **decimals**. Depending on the data type selected, users may have to

provide a size. For example, for decimals, users will need to provide the precision and scale.



Please note that decimal values cannot be entered into the numeric textbox control when it is configured to use the **integer** data type.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - For more information on these properties and how to configure them, [click here](#).
4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Align:** This is a single-select dropdown field that allows users to choose the alignment of the control.
    - Alignment Options: Right, Center, Left, or Auto By Culture.
      - Auto By Culture: This option aligns the control based on the language or culture setting in the web application.
      - Right: This option aligns the control to the right side of the screen.
      - Center: This option aligns the control to the center of the screen.
      - Left: This option aligns the control to the left side of the screen.
  - **Floating:** This is a checkbox which, if selected, allows the control to accept and handle floating-point numbers, including decimal values.



If the **Floating** checkbox is left unselected, the control will only accept integer values.

- **Format:** This field allows users to specify a format for displaying the numeric data in the control. Users can either enter the format manually or select from a dropdown menu that includes four available formats:
  - #,###: This format displays the number with thousands separators and no decimal places.
    - For example, the number 1234567 would be displayed as 1,234,567.
  - #,###.0: This format displays the number with thousands separators and one decimal place.
    - For example, the number 1234.567 would be displayed as 1,234.6.
  - #,###.00: This format displays the number with thousands separators and two decimal places.
    - For example, the number 1234.567 would be displayed as 1,234.57.
  - #,###.000: This format displays the number with thousands separators and three decimal places.
    - For example, the number 1234.567 would be displayed as 1,234.567.

✿ When using formats that include decimal values (such as #,###.0, #,###.00, and #,###.000), the **Floating** checkbox will be automatically enabled by the system.

- **Mask:** This field allows users to specify a format for the input data in the numeric textbox control. Users can either enter the format manually or select from a dropdown menu that includes two available formats:
  - 9999: This format is used for integers and specifies that the input should have a maximum of four digits, with leading zeros added if necessary.
  - 9999.99: This format is used for decimals and specifies that the input should have a maximum of four digits before the decimal point, and a maximum of two digits after the decimal point. The decimal point itself is added automatically.
  - **Numeric Textbox Mask Pattern Preview: Web View**

| Whole Number Mask Pattern | Decimal Mask Pattern | Custom Mask Pattern |
|---------------------------|----------------------|---------------------|
| 9999                      | 9999.99              | (999) 999-9999      |

- **Max Length:** This field allows users to specify the maximum number of characters that can be entered into the control.
  - By default, this field is left blank. To specify a maximum length of input, however, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- **Placeholder:** This field allows users to input the placeholder text for the control.
  - Users can also directly add a placeholder text to the control simply by clicking on the inside of the control in the design environment.

|                                 |  |
|---------------------------------|--|
| <div>Category</div> <div></div> | <div>5. State Matrix</div> <div>Click '...' to edit</div> <div>4. Appearance</div> <div>Placeholder Arabic</div> <div>Placeholder English</div> <div>Placeholder French</div> <div>Placeholder German</div> <div>Placeholder Hebrew</div> <div>Tooltip</div> |
|---------------------------------|--|

- As English is set as the default language in the system, any placeholder inputted in the English placeholder field will be used as the display placeholder for the control, regardless of other language options that may be available.
  - Similarly, any placeholder text entered in the design environment will automatically populate the English placeholder field.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Control Placeholder Text Preview: Web View**

### Height



A placeholder text is the default value or text that is displayed inside the control until the user enters their own value.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
  - **Numeric Textbox Tooltip Preview: Web View**



A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

### Quick Forms Application

To apply the **Quick Forms** feature to numeric textbox controls, take the following steps:

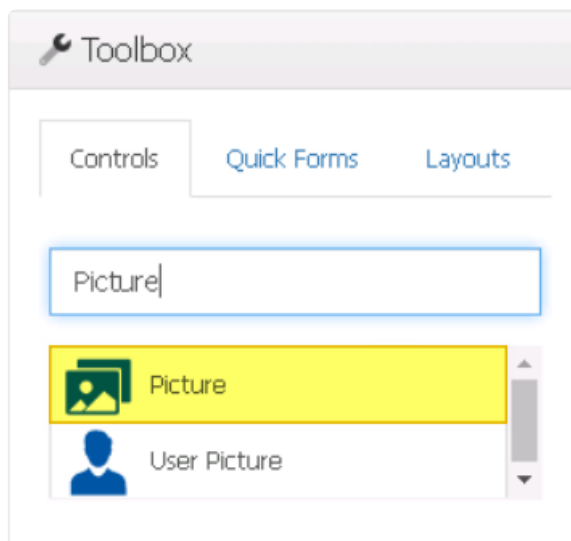
- Drag a numeric textbox control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:



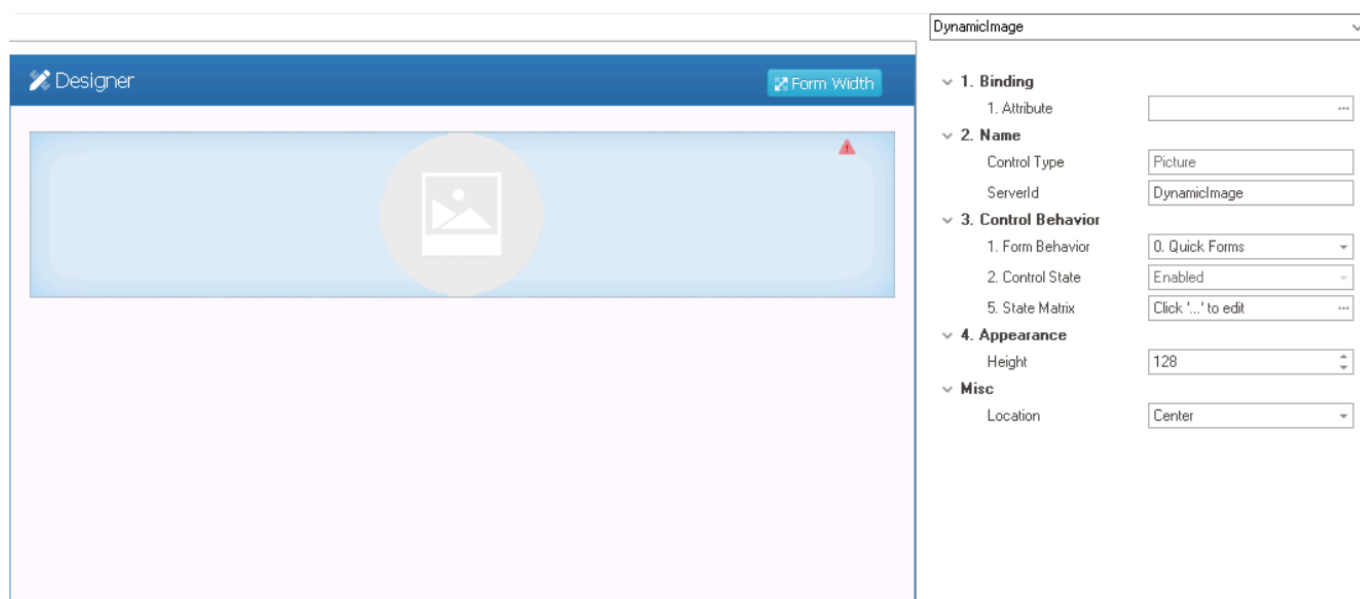
For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.21. Picture

With the use of picture controls, user images can be uploaded and dynamically displayed on forms.



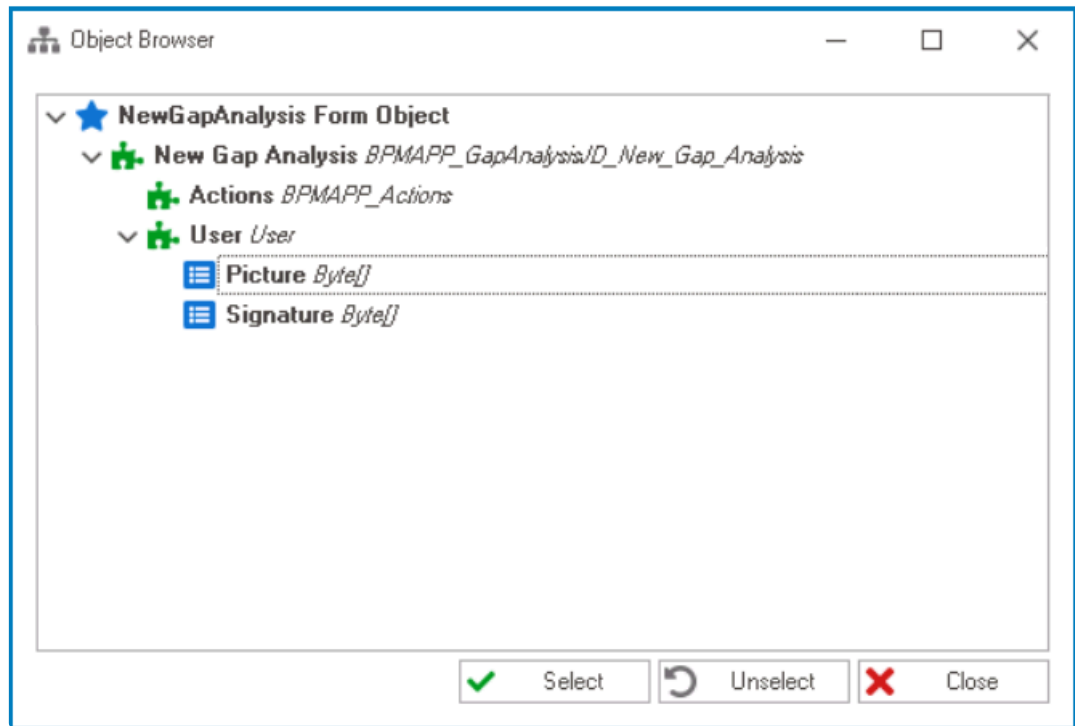
When a picture control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute.
  - **Attribute:** This field allows users to select the attribute associated to the control.
    - To access the list of relevant attributes, click on the ellipses ( ... ) inside the field.





✿ In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

DomainObject.UserUser.Pic... ...

! To enable picture controls to retrieve and display the relevant images in the system, a reference attribute must be created linking the control to all user accounts in the database. For more information on creating this type of attribute, [click here](#).

- Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - Control Type:** This field displays the control type. It cannot be modified.
  - Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
- Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state or validation properties set for the control will apply only to the selected form behavior.

✿ Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.

| Form Behavior          | Control State |     |
|------------------------|---------------|-----|
| 1. Modify Gap Analysis | ✓ Enabled     | ... |
| 2. New Gap Analysis    | ✓ Enabled     | ... |
| 4. Review Gap Analysis | ✓ Enabled     | ... |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.

✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.

- By default, the height is set to 128. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Misc:** In this category, users can define additional properties for the control.
- **Location:** This is a single-select dropdown field that allows users to choose the location of the control.
    - Location Options: Right, Center, or Left.
      - Right: This option places the control on the right-hand side of the screen.
      - Center: This option centers the control in the middle of the screen.
      - Left: This option places the control on the left-hand side of the screen.

## 12.1.2.1.1.21.1. User Reference Attribute

In order to easily manage and display user-specific images and signatures within the system, a user reference attribute must be created linking the relevant control to all user accounts in the database. The purpose of this attribute, essentially, is to connect the user picture or signature control in the form to the user in the system. By establishing this connection, the system can retrieve the correct image or signature for each user and display it accordingly.

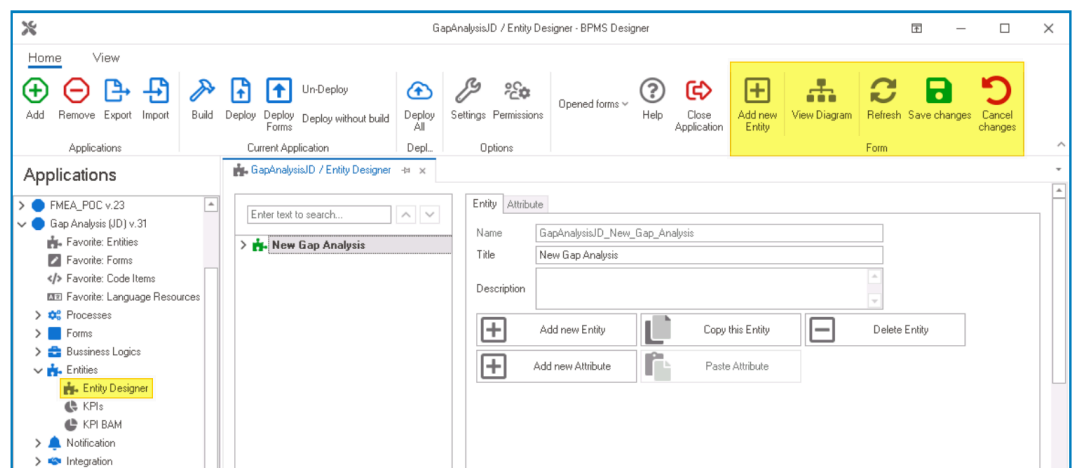
✿ This reference attribute applies to picture controls, user picture controls, and user signature controls.

To create and configure this type of attribute, take the following steps:

### 1. Create Attribute

#### • Access Entity Designer

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From the Entity tab, select the **Add New Attribute** option. It will direct you to the Attribute tab where you can define the properties of the new attribute.



### 2. Configure Attribute Properties

- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

GapAnalysisJD / Entity Designer

Enter text to search...

New Gap Analysis

- NewAttribute Id / UniquelIdentifier
- Actions Actions
- Active Boolean / Bit
- DueDate Date/Time / DateTime
- Scope String / NVARCHAR (Max)

Entity Attribute

Name: NewAttribute

Title:

Description:

☒ Data field ☐ Reference

Data field Reference

Data Type: UniquelIdentifier (Globally unique identifier)

Size: Precision:

Show as a column in the task list

☐ Hide ☐ Hidden by default ☒ Show by default

☐ Read Only ☐ Defines field ☐ Track changes

+ Add new Attribute Copy Attribute - Delete Attribute

- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Reference** option.
- Click on the button. It will display the Select Reference window. From the entity section, select **System – Users**. From the field section, select **ID**.

Select Reference

Select Entity

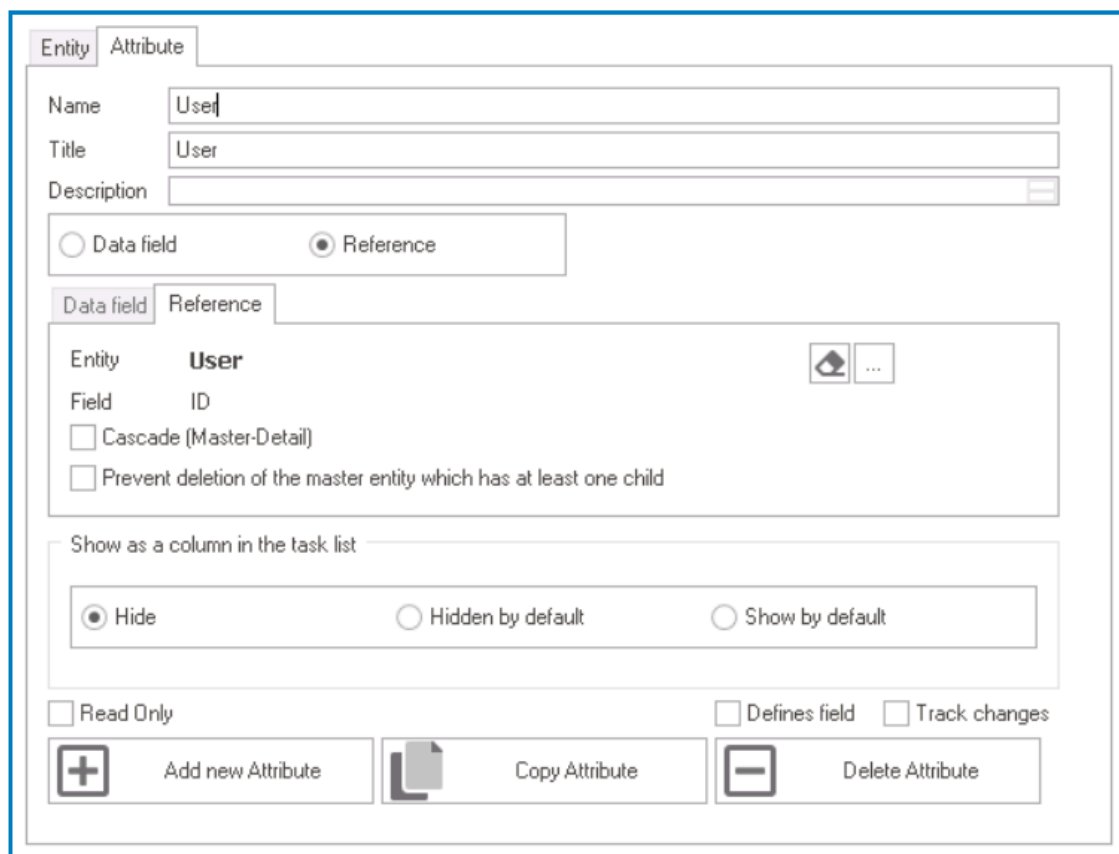
Application: System Entity: Users

| Entity | Field                             |
|--------|-----------------------------------|
| System | Address String / VarChar (255)    |
| System | Email String / VarChar (30)       |
| System | Fax String / VarChar (20)         |
| System | ID String / VarChar (36)          |
| System | Login String / VarChar (50)       |
| System | MiddleName String / VarChar (100) |
| System | Password String / VarChar (50)    |
| System | POBox String / VarChar (10)       |
| System | PostalCode String / VarChar (10)  |
| System | Stamp / Image                     |
| System | Tel String / VarChar (20)         |
| System | WebPage String / VarChar (50)     |

Application = System Edit Filter

Select Cancel

- Selecting the system reference will redirect to the Attribute tab, where the entity section will display **User** and the field section will display **ID**.



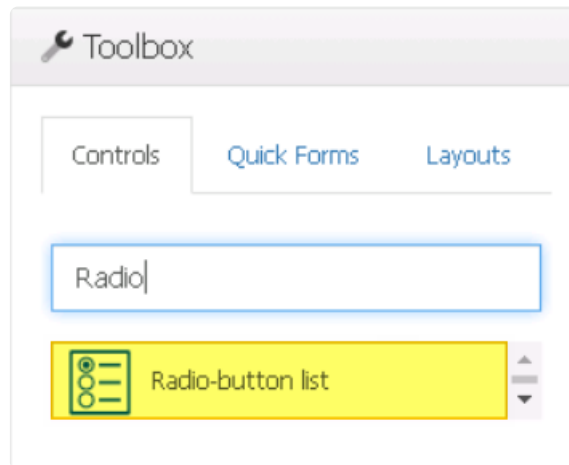
### 3. Save Attribute

- To save the attribute configuration, select the **Save Changes** option from the menu bar.



## 12.1.2.1.1.22. Radio-Button List

Unlike [checkbox controls](#), which allow users to select multiple options at once, radio button controls allow users to choose only one option from a set of mutually exclusive options.



- \* Radio buttons are particularly effective in implementing business rules, where specific actions can be triggered based on the user's selection. For example, if the user select Option A, then the XYZ fields will be displayed, whereas selecting Option B will result in a different set of fields being shown. [For more information on business logic and rules, click here.](#)

When a radio button control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

- \* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - [For more information on the fields in this category, click here.](#)

- \* When creating attributes for radio button controls, there are various data types to choose from, including string data types (such as **Char**, **NChar**, **NText**, **NVarChar**, or **Text**), numeric data types (such as **Decimal**, **Float**, **Int**, **Numeric**, or **Real**), and the Bit (Boolean) data type. Please note, however, that if you choose the **Bit (Boolean)** data type, you can only have two options available. If you require more than two options, then using a different data type is necessary to ensure proper functionality of the workflow.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.

- For more information on these properties and how to configure them, [click here](#).

3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.

- **Radio Button Initial Value**

Both the **Initial Value** field and the **Selected** checkbox (described below) are used to pre-select radio button options.

! The **Initial Value** field defines the pre-selected value per form behavior, while the **Selected** checkbox defines the pre-selected value for the entire workflow.

- If both properties are set, the **Initial Value** overrides the **Selected** checkbox.
- If the **Initial Value** matches a radio button option, that option will be pre-selected.
- If the **Initial Value** doesn't match any option, nothing will be pre-selected.
- If the Initial Value is not set, the option with the **Selected** checkbox checked will be pre-selected.

- For more information on these properties and how to configure them, [click here](#).

! Please note that when it comes to control governance properties, users are limited to configuring only the required field validation properties for radio button controls. It is not possible to set up required range validations.

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Tooltip:** This field allows users to input a tooltip for the control.

- In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
- **Radio Button Tooltip Preview: Web View**

Select the appropriate option to indicate the outcome of the inspection

**Quality Inspection Result**

☐ Pass

☐ Fail

☐ Requires Further Inspection

☐ Pending

\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.



5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - For more information on the fields in this category, [click here](#).
6. **Misc:** In this category, users can define additional properties for the control.
  - **Items:** This field allows users to create and customize items or options for the radio button.
    - To access the List Items Editor, click on the ellipses (...) inside **Items** field.

- **Add:** This button allows users to add a new item.
  - Once added, the new item will be displayed in the **Items** box and its properties will be shown in the **Item** section.
    - **Value:** This is a text field that allows users to input the data value for the item.
      - This value corresponds to a specific item in the radio button. When users select the item, the corresponding value is sent to the server, allowing it to be processed in the backend.



It's important to make sure that the value entered matches the item selected and is in the correct format for the data type you are working with (e.g., "true" or "false" for boolean values, or "0" or "1" for integer values). It is recommended to keep the value simple, as it will be used to write business logic if necessary.

- **Caption:** This is a text field that allows users to input the displayed name or caption of the item.
  - As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the item, regardless of other language options that may be available.

\* Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

! Please note that for the translation to be effective on the radio button control, the form must be saved and closed beforehand.

- **Selected:** This checkbox determines whether the radio button option is pre-selected or not.
  - To pre-select an option, users must check the **Selected** checkbox for that option.

\* When a user accesses a radio button control, the option that is pre-selected will be chosen by default. Only one option can be pre-selected at a time.

- **Enabled:** This checkbox determines whether the radio button option is available for selection or not.
  - To disable an option, users must uncheck the **Enabled** checkbox for that option.

! If an option is not enabled, it cannot be selected by the user in the web application.

- **Remove:** This button allows users to remove a selected item.
- **Up / Down:** These buttons allow users to reorder the items in the list.
  - To move an item up or down, select it from the list and click on the **Up** or **Down** button. This will move the selected item one position up or down in the list, respectively.
- **OK:** This button allows users to save their changes and exit the [List Items Editor](#) window.
- **Repeat Columns:** This field allows users to specify the number of columns in which the radio button options will be arranged.
  - By default, the number of columns is set to 1. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- **Repeat Directions:** This field allows users to specify the direction in which the radio button options will be arranged.
  - [Directions:](#) Vertical or Horizontal.
    - [Horizontal:](#) This option arranges the radio button options in rows from left to

right.

- Vertical: This option arranges the radio button options in columns from top to bottom.

#### Radio Button Control Preview: Web View

##### Action Type

☐ Internal ☐ External

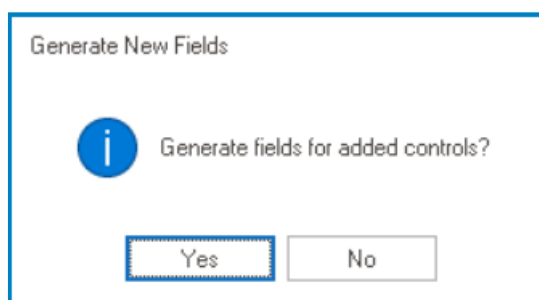
#### Customizing Radio Button Control: Web View

*Design A – Pre-Selected Option (Active); Design B – Direction: Vertical; Design C – Direction: Horizontal; Design D – Columns: 3, Direction: Vertical*

#### Quick Forms Application

To apply the **Quick Forms** feature to radio button controls, take the following steps:

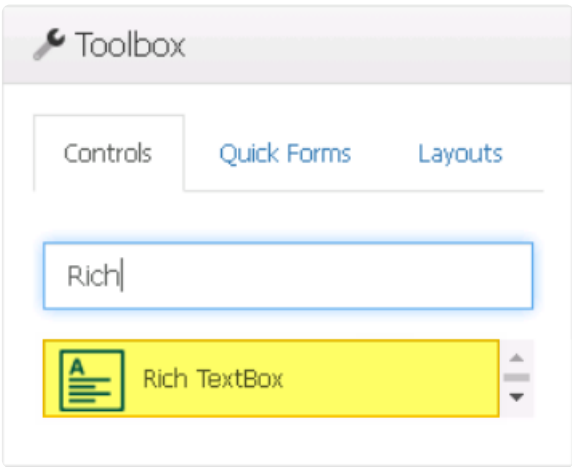
- Drag a radio button control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:



For more information on the **Quick Forms** feature, [click here](#).

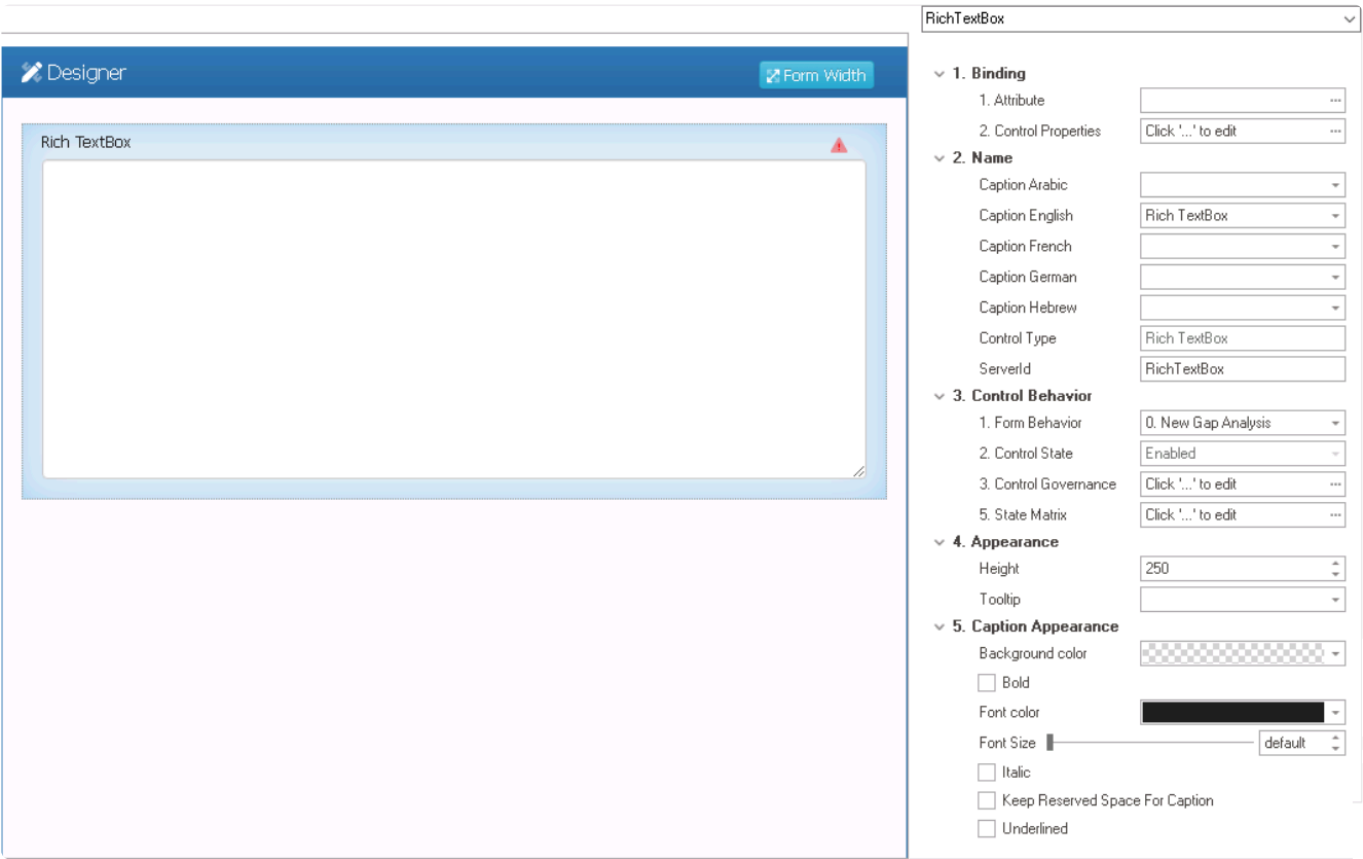
# 12.1.2.1.1.23. Rich Textbox

A rich textbox control is a form element that allows users to input and format text with a wide range of styling options, such as bold and italic text, bullet points, and hyperlinks.



✿ Unlike a multi-line textbox, which only allows plain text input, rich textbox controls provide a more flexible and expressive way to create and edit text-based content.

When a rich textbox control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

✿ To create attributes for rich textbox controls, users need to choose data types such as **VarChar** (VarChar or VarChar\_Max) or **NVarChar** (NVarChar or NVarChar\_Max).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - For more information on these properties and how to configure them, [click here](#).

! Please note that the linked section contains information on setting an initial value, which is not applicable to rich textbox controls and can be ignored. In addition, regarding control governance properties, users are limited to configuring only the required field validation properties for rich textbox controls and cannot set up required range validations.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 250. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
  - **Tooltip:** This field allows users to input a tooltip for the control.
    - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
    - **Rich Textbox Tooltip Preview: Web View**

Enter the specific parameters or criteria that define the scope of your analysis, such as the time period, geographic location, or target audience.

**Scope**

Paragraph
▼
¶
▼
AI
▼
A
▼
A
▼
B
I
U
↺
≡
▼
:=
:=
≡
≡
:


\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

#### Rich Textbox Control Preview: Web View

##### Scope

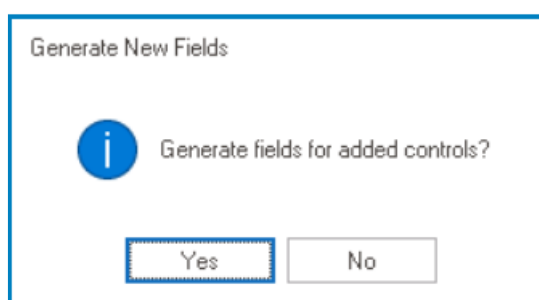
A screenshot of a rich text editor interface. At the top, there is a toolbar with various formatting options: a dropdown menu currently showing 'Paragraph', followed by icons for text color, background color, text alignment (left, center, right, justified), bold, italic, underline, strikethrough, bulleted list, numbered list, indent, and outdent. Below the toolbar is a large, empty text area for editing.

\* For more information on how to use this field in the web application, [click here](#).

#### Quick Forms Application

To apply the **Quick Forms** feature to rich textbox controls, take the following steps:

- Drag a rich textbox control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:

A dialog box titled 'Generate New Fields'. It contains an information icon (a blue circle with a white 'i') and the text 'Generate fields for added controls?'. At the bottom, there are two buttons: 'Yes' and 'No'. The 'Yes' button is highlighted with a blue border.

\* For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.23.1. Rich Textbox – Web Display

For those unfamiliar with using the rich text field in the **DBP** web application, here are some key features to consider:

### 1. Text Formatting

Users have access to various formatting options in the rich textbox control, including:

- **Heading Style:** This field allows users to choose a heading style from a dropdown menu.
- **Font Family:** This field allows users to choose the font family from a dropdown menu.
- **Font Size:** This field allows users to choose the font size from a dropdown menu.
- **Font Color:** This field allows users to choose the font color from a dropdown menu.
- **Font Background Color:** This field allows users to choose the font background color from a dropdown menu.
- **Bold:** This button allows users to apply a bold effect to the selected text.
- **Italic:** This button allows users to apply an italic or slanted effect to the selected text.
- **Underline:** This button allows users to apply an underline effect to the selected text.
- **Strikethrough:** This button allows users to apply a strikethrough effect to the selected text.
- **Text Alignment:** This field allows users to choose the alignment of the text from a dropdown menu.
- **Numbered List:** This button allows users to apply a numbered list format to selected text.
  - Users can create nested lists by using the **Indent** button, either in the rich text editor or on their keyboard.
- **Bulleted List:** This button allows users to apply a bulleted list format to selected text.
  - Users can create nested lists by using the **Indent** button, either in the rich text editor or on their keyboard.
- **Increase / Decrease Indent:** These buttons allow users to adjust the indentation level of the selected text.
- **Block Quote:** This button allows users to apply a block quote format to the selected text.



To undo any text formatting, users can either click on the **Undo** button or unselect the formatting option.

### 2. Hyperlinks

Users can insert hyperlinks to external web pages within the rich textbox control.

### 3. Media Support

Users can embed various types of media into the rich textbox control, either by uploading them from their computer or by providing a URL to online media content.

- ✿ Once an image has been inserted into the control, users can adjust its alignment, resize it, and add alternative text if needed.

#### 4. **Table Support**

Users can add, edit, and format tables in the rich textbox control.

- ✿ Once a table has been inserted into the control, users can modify the column and row properties, merge or split cells, and adjust various table properties like border, background color, dimensions, and alignment. The control also enables users to modify individual cell properties for complete control over their table design.

#### 5. **Undo / Redo**

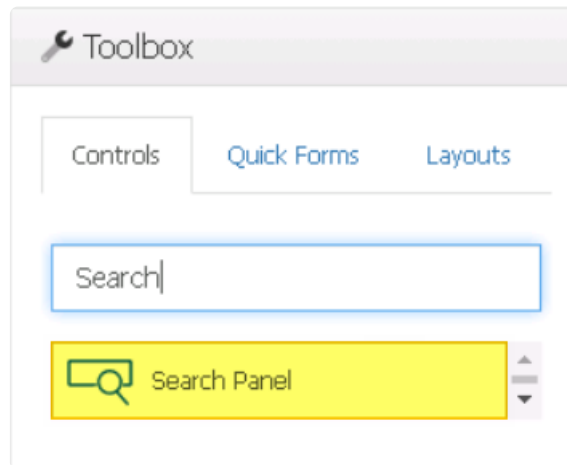
Users can undo or redo a previous action in the rich textbox control.

- ✿ Depending on the amount of text entered, a vertical scrollbar may appear on the side of the rich textbox to allow users to scroll through the contents. Additionally, users can resize this textbox by dragging at its bottom right corner.



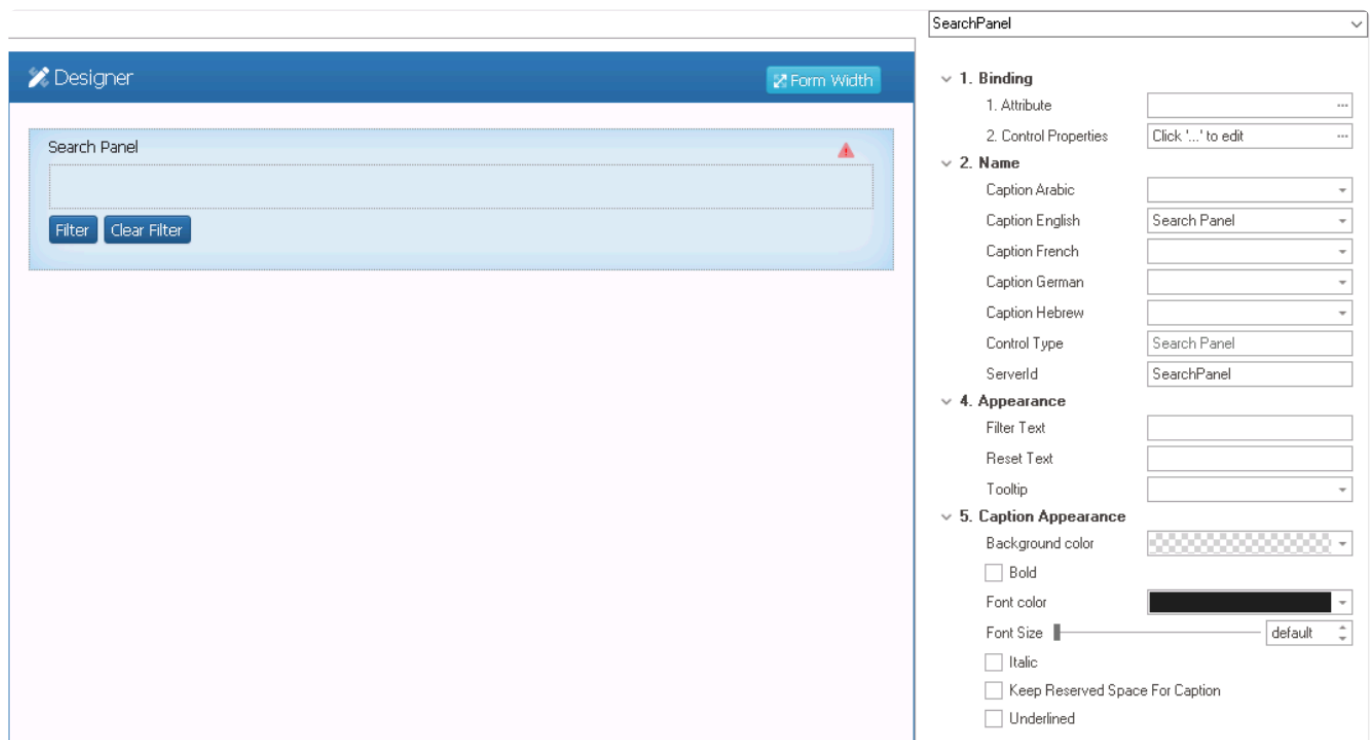
## 12.1.2.1.1.24. Search Panel

The search panel control enables users to quickly and easily filter through a list of items based on specific criteria, helping them find the information they need in an efficient manner.



! When creating a form to contain the search panel control, users can leave the **Entity** field empty. For more information on this matter, [click here](#).

When a search panel control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.

- For more information on the fields in this category, [click here](#).

! Users must link the search panel control to the same attribute as the corresponding grid control. For more information on this type of attribute, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Filter Text:** This field allows users to customize the text displayed inside the **Filter** button in the search panel.
    - Users can change the text to better reflect the purpose of the button or to display the text in a different language.
    - **Filter Text Preview: Web View**

#### Search Panel

|                                       |   |             |                                 |
|---------------------------------------|---|-------------|---------------------------------|
| Gap Analysis ID                       | Gap Analysis Name                           | Category    | <input type="checkbox"/> Active |
| <input type="text"/>                  | <input type="text"/>                        | Select... ▼ |                                 |
| <input type="button" value="Filter"/> | <input type="button" value="Clear Filter"/> |             |                                 |

✿ If this field is left blank, the default text 'Filter' will be displayed.

- **Reset Text:** This field allows users to customize the text displayed inside the **Clear Filter** button in the search panel.
  - Users can change the text to better reflect the purpose of the button or to display the text in a different language.
  - **Reset Text Preview: Web View**

#### Search Panel

|                                       |   |             |                                 |
|---------------------------------------|---|-------------|---------------------------------|
| Gap Analysis ID                       | Gap Analysis Name                           | Category    | <input type="checkbox"/> Active |
| <input type="text"/>                  | <input type="text"/>                        | Select... ▼ |                                 |
| <input type="button" value="Search"/> | <input type="button" value="Clear Filter"/> |             |                                 |

✿ If this field is left blank, the default text 'Clear Filter' will be displayed.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.

- **Search Panel Tooltip Preview: Web View**

Use the search panel to quickly find and filter data based on specific criteria.

Search Panel 

Gap Analysis ID

Gap Analysis Name

Category

☐ Active

Select... ▼

Filter

Clear Filter



A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

4. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.

- For more information on the fields in this category, [click here](#).

#### Search Panel Control Preview: Web View

##### Search Panel

Gap Analysis ID

Gap Analysis Name

Category

☐ Active

Select... ▼

Filter

Clear Filter

- To filter results according to the entered search criteria, click on the **Filter** button.
- To clear the filters and see the full data set again, click on the **Clear Filter** button.

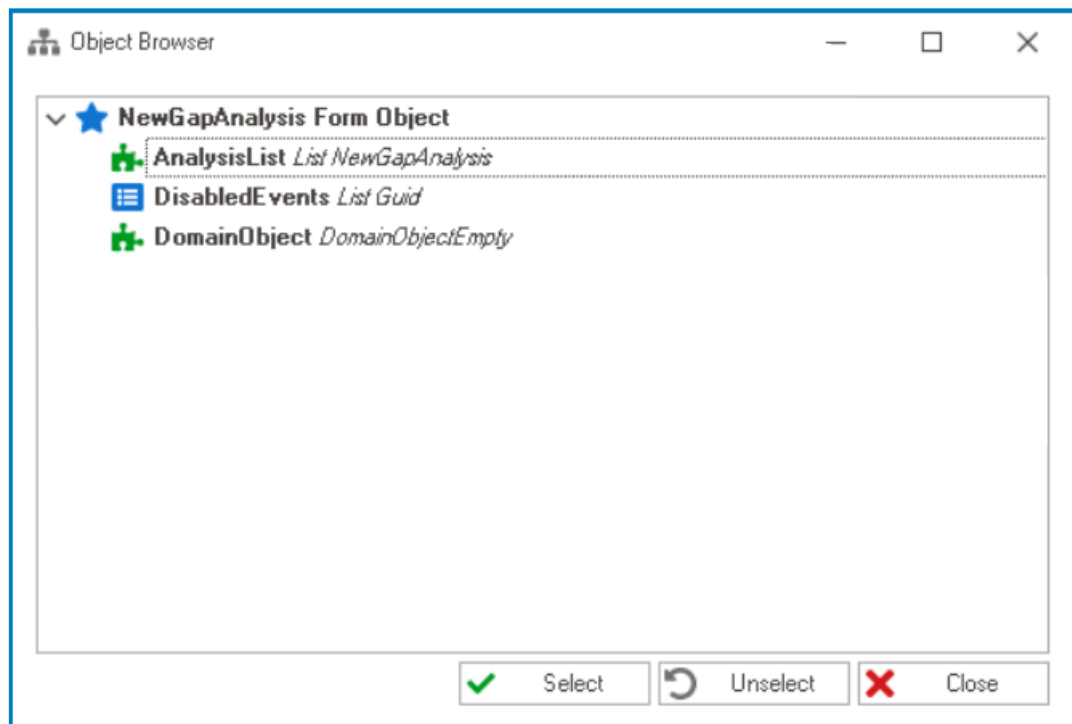


For detailed instructions on designing and building search panels, [click here](#).

## 12.1.2.1.1.24.1. Search Panel – Binding

In the **Binding** category, users can link a search panel to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the search panel control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated (form) entity's attributes—specifically, the attributes relevant to the search panel control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute



Users must link the search panel control to the same attribute as the corresponding grid control. For more information on this type of attribute, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- For more information on these properties and how to configure them, [click here](#) and navigate to the relevant section.

## 12.1.2.1.1.24.2. Search Panel – Build Components

---

Users who utilize search panels typically desire a simple form layout that focuses on essential components, such as a grid for presenting data from another form and a search panel for refining results.



When creating a form with such a minimalistic design, users can leave the **Entity** field empty. For more information on this matter, [click here](#).

To design a simple form with a search panel and grid, take the following steps:

### 1. Build Search Panel Components

To get started, if they have not already done so, users should add a search panel to the design environment. Once the search panel is added, users can proceed to build its components by dragging layouts and controls from the toolbox and dropping them inside the panel.



As long as the controls added to the search panel are linked to an attribute under the same entity associated with the grid, users can choose from a range of options, such as textboxes, dropdowns, date pickers, and checkboxes.

After adding the required controls to the search panel, users should assign a name to each control and link it to its corresponding attribute. If necessary, they should also configure the binding properties for each control to ensure that the data is displayed and updated correctly.

Finally, users can customize the search panel and its controls as needed.

The screenshot shows the 'Designer' application window. On the left, a 'Search Panel' is displayed with fields for 'Gap Analysis ID', 'Gap Analysis Name', 'Category' (a dropdown menu), and an 'Active' checkbox. Below these fields are 'Filter' and 'Clear Filter' buttons. On the right, a 'SearchPanel' properties panel is open, showing various settings:

- 2. Name:** Multiple caption fields for different languages (Arabic, English, French, German, Hebrew) and 'Control Type' and 'ServerId' set to 'Search Panel'.
- 4. Appearance:** Fields for 'Filter Text', 'Reset Text', and 'Tooltip'.
- 5. Caption Appearance:** Settings for 'Background color' (a checkerboard pattern), 'Font color' (black), 'Font Size' (a slider set to 'default'), and checkboxes for 'Is Bold' and 'Is Italic'.

## 2. Build Grid

Once the search panel is complete, users can move on to building the grid, which will display the results of the search. The grid will retrieve data from another form and update dynamically based on the search criteria entered in the search panel.



For detailed instructions on designing and building grids, please refer to the [corresponding section](#) in the manual.



For this type of form, users must link the grid to the same attribute as the one created for the search panel control (and vice versa).

## 3. Save Form & Generate Form Behavior

Once the search panel components and grid have been built and designed, users should save the form to preserve their work. In response to the following system prompt, users should select **Yes**.

The dialog box titled 'Generate Form Behavior' contains an information icon (i) and the text 'Generate a link in the main menu for this form?'. At the bottom, there are two buttons: 'Yes' and 'No'.



Confirming this prompt will allow for the automatic generation of the form behavior. For

more information on form behaviors, [click here](#).

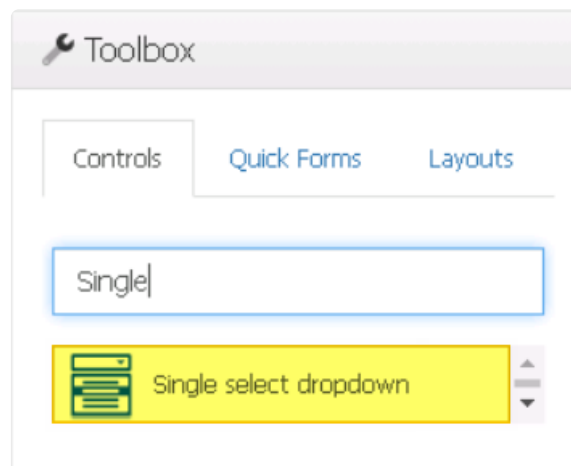


## 12.1.2.1.1.25. Single-Select Dropdown

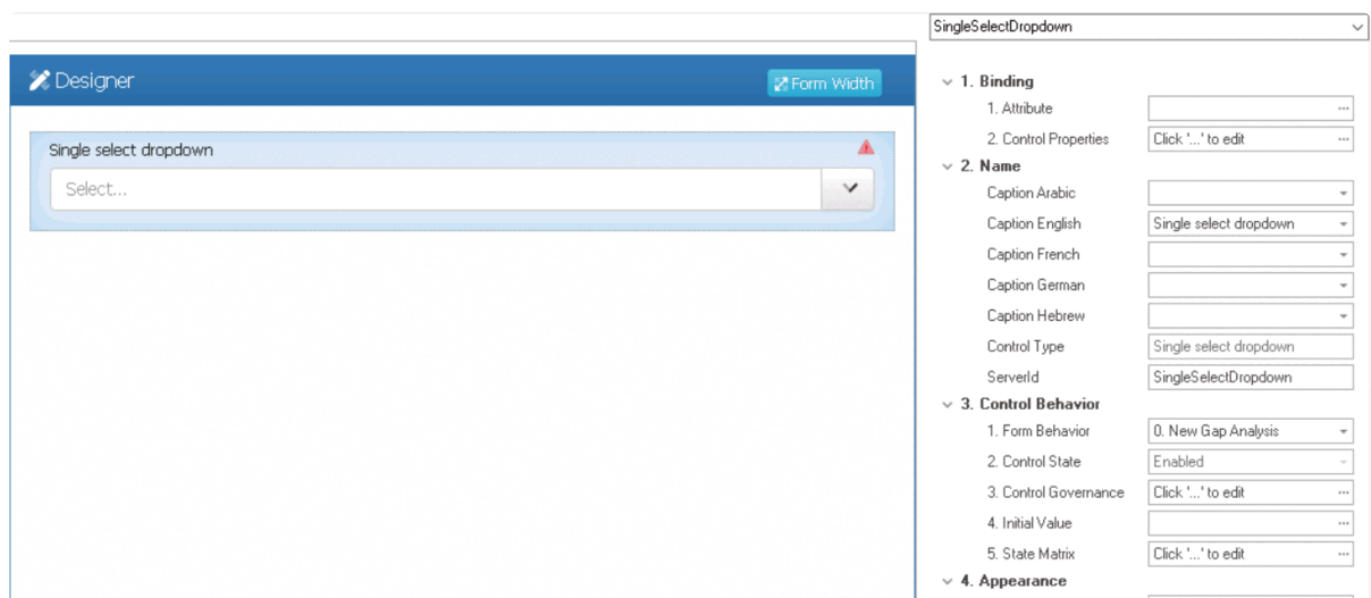
### Frequency



A single-select dropdown control is a user interface element that allows users to select one option from a list of predetermined choices.



When a single-select dropdown control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

- ✿ To display content in a single-select dropdown control, users can use string data types (such as **Char**, **NChar**, **NText**, **NVarChar**, or **Text**) for text content and numeric data types (such as **Decimal**, **Float**, **Int**, **Numeric**, or **Real**) for number content. If users want to display an existing list of items in the database, they can create a reference attribute that links to the list.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Control Governance**
    - In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
      - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
        - This window consists of two tabs:
          - [Selector](#)
          - [Validation](#)



Please note that users are limited to configuring only the required field validation properties for single-select dropdown controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).
4. **Appearance:** In this category, users can customize the visual appearance of the control.
    - For more information on these properties and how to configure them, [click here](#).
  5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
    - For more information on the fields in this category, [click here](#).

#### Single-Select Dropdown Control Preview: Web View

Frequency

Select...



For more information on how to use this field in the web application, [click here](#).

#### Quick Forms Application

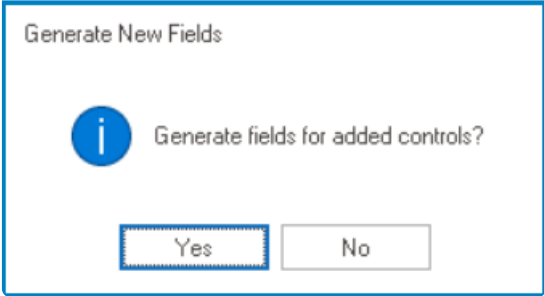
To apply the **Quick Forms** feature to single-select dropdown controls, take the following steps:

- Drag a single-select dropdown control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.


- To associate the control with an existing list in the database, specify the corresponding dropdown list in the **Server ID** field.
  - This can be done by entering a Server ID consisting of the name of the attribute followed by the name of the dropdown.
    - For example, "NameUser" would indicate that "Name" is the attribute and "User" is the reference to the list of users.

! If users prefer to create a custom list of static or dynamic items for the dropdown, they can skip this step and [configure the control](#) accordingly.

- Save the form and select **Yes** in response to the following prompt:



Generate New Fields

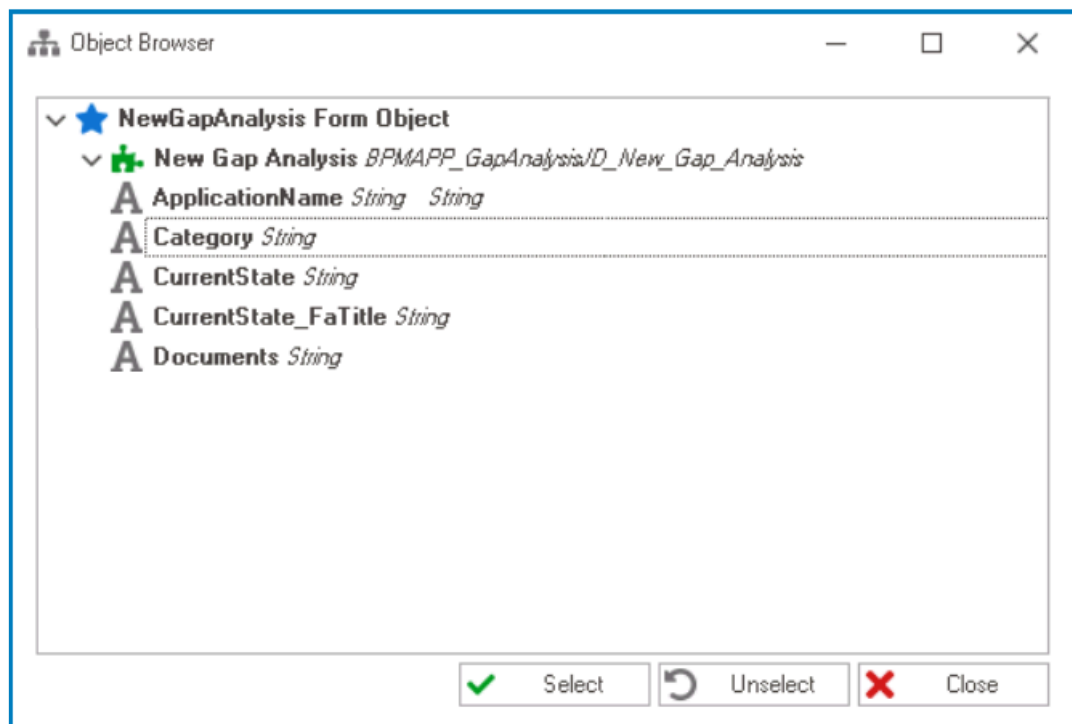
 Generate fields for added controls?

\* For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.25.1. Single-Select Dropdown – Binding

In the **Binding** category, users can link a single-select dropdown to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the single-select dropdown control.
  - To access the list of relevant attributes, click on the ellipses ( ... ) inside the field.



\* In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the single-select dropdown control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

\* To display content in a single-select dropdown control, users can use string data types

(such as **Char**, **NChar**, **NText**, **NVarChar**, or **Text**) for text content and numeric data types (such as **Decimal**, **Float**, **Int**, **Numeric**, or **Real**) for number content. If users want to display an existing list of items in the database, they can create a reference attribute that links to the list.

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. In the displayed window, navigate to the Single-Select tab.

**Control Properties**

Apply OK Cancel

**Form Control** SingleSelectDropdown  
**Entity and Attribute** DomainObject.Category  
**Object type** System.String

General Single select

[Empty] + ✎

☒ Load data asynchronously  
☒ Sort by Display member

Page size 1000  
 Load delay (ms) 100  
 Min characters for search 0  
 Search Type Con...

- Dropdown Data Source:** This field retrieves the list of dropdown data sources available in either the application or system. Depending on the data source chosen in this field, the available options for selection in the single-select dropdown control will vary.
  - To define a new dropdown data source, select the button.
    - It will display the [Query Editor](#) window, in which users can configure static or dynamic items.
  - To modify an existing dropdown data source, select the button.
  - To remove a dropdown data source, select the button.
    - This button will only be visible once users have added/selected a dropdown data source.

✿ If sourced from within the application, the dropdown data sources in this field will be retrieved from the [Dropdown Data Sources](#) section of the **Forms** module. If sourced from the system itself, they will be retrieved from the database directly.

- **Load Data Asynchronously:** This is a checkbox which, if selected, allows the dropdown data to be retrieved in the background while the user interacts with other parts of the application.
  - Selecting this checkbox will speed up the display of dropdown options, improving performance.
- **Sort By Display Member:** This is a checkbox which, if selected, allows users to sort the dropdown options by their display values, instead of their original order in the data source.
  - Users can specify the display member by editing the query in the [Query Editor](#) window
- **Page Size:** This field allows users to set the number of items that are displayed in the dropdown at one time.
  - By default, the size is set to 1000. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

✿ When the 'More Items' option is selected, the dropdown will update to display the number of items according to the number specified in this field, and will continue to update by that amount for subsequent selections until all items are displayed.

▪ **Tip: How to Change Page Size for All Dropdowns**

The information in this window is saved in the [MetaInfo](#) field of [BPMS\\_PhysicalFormBinding](#) as an XML document. Therefore, to change the page size for all dropdowns in the application, users can apply a query like this:

```
UPDATE [ray].[BPMS_PhysicalFormBinding]
SET MetaInfo = REPLACE(MetaInfo, '<MaxResults>20</MaxResults>', '<MaxResults>100</MaxResults>')
WHERE MetaInfo LIKE '%<MaxResults>20</MaxResults>%' AND
ApplicationName = 'yourAppName' AND MetaInfo LIKE
'%<AutoCompleteMeta%'
```

- **Load Delay (ms):** This field allows users to set the amount of time, in milliseconds, that the single-select dropdown control will wait before loading or displaying its options.
  - By default, this value is set to 100. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement

arrows.

- **Min Characters for Search:** This field allows users to specify the minimum number of characters a user must input into the search field before the search function is activated.
  - By default, the minimum value is set to 0. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.



By setting a minimum value in this field, users can not only help prevent unnecessary searches, but they can also ensure that a search query is specific enough to return relevant results.

- **Search Type:** This field displays the list of search types. It allows users to choose how the search function will match their search query with the available options.
  - Search Types: Contains, Starts With, or Ends With.
    - Contains: This option will return results that include the search query anywhere in the text.
    - Starts With: This option will return results that begin with the search query.
    - End With: This option will return results that end with the search query.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and exit the Control Properties window.
- **Cancel** This button allows users to cancel their changes and exit the Control Properties window.



In the General tab of this window, users can configure business rules for the single-select dropdown control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.1.25.2. Single-Select Dropdown – Control Governance

In the Selector tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of single-select dropdown controls.

1. **Browse Form Behavior:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'browse' functionality.
  - To define a new form behavior, select the button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing form behavior, select the button.
  - To remove a form behavior, select the button.
    - This button will only be visible once users have added/selected a form behavior.

✿ By selecting a form behavior in this field, users are essentially linking the control to a modal window. For more information on this feature, [click here](#).

2. **Apply:** This button allows users to save their changes.
3. **OK:** This button allows users to save their changes and close the window.
4. **Cancel:** This button allows users to cancel their changes and close the window.





In the Validation tab of this window, users can configure the required field validation properties for the single-select dropdown control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.1.25.3. Single-Select Dropdown – Web Display

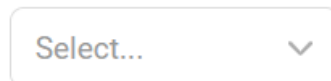
---

For those unfamiliar with using the single-select dropdown field in the **DBP** web application, the following instructions will guide you through the process:


### Single-Select Dropdown Field

A single-select dropdown field allows users to select only one value at a time from a predetermined list of options.

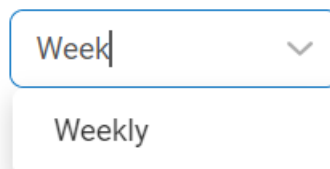
#### Frequency



There are two ways to select an item from the dropdown:

1. Click on the  icon. It will release the list of values. Users can then scroll through the options and make their selection.
2. Search for the value in the dropdown by inputting its name or keywords relating to it. Any items that match the search will be retrieved. Users can then click on the desired value to finalize their selection.

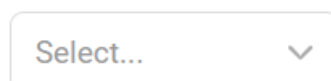
#### Frequency



### Single-Select Dropdown Field with Modal Window


If users have enabled the [browsing functionality](#) of the control, a modal window will be presented instead of the standard dropdown menu.

#### Folder



\* A modal window is a pop-up window that disables the main form, requiring users to interact with it before they can regain access. Users can interact with a modal typically by either completing the set task or exiting the window.

To use a single-select dropdown field linked to a modal window, follow these steps:

1. Click on the  icon. It will open a modal window containing the options available to choose from.
2. Navigate through the options and select the desired one by double-clicking on it.
  - Users will then be returned to the main form where they will find the field now populated with their selection.
3. Otherwise, to return to the main form without making a selection, click on the **Exit (X)** button.

In the example below, the modal window contains a tree view control to help users navigate through a hierarchical structure of options.

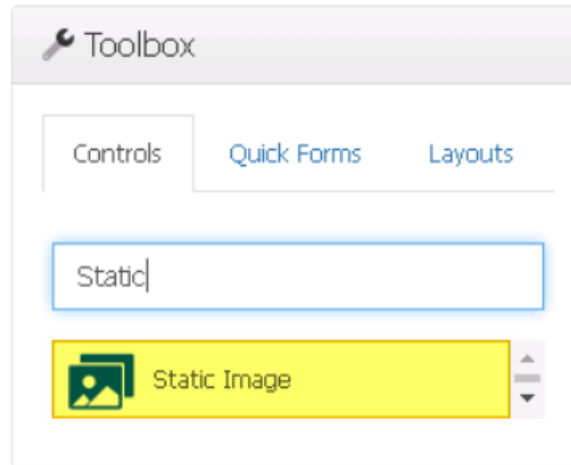
\* Users can determine the contents of the modal window by creating the form and linking the corresponding form behavior to the control.

### Folder

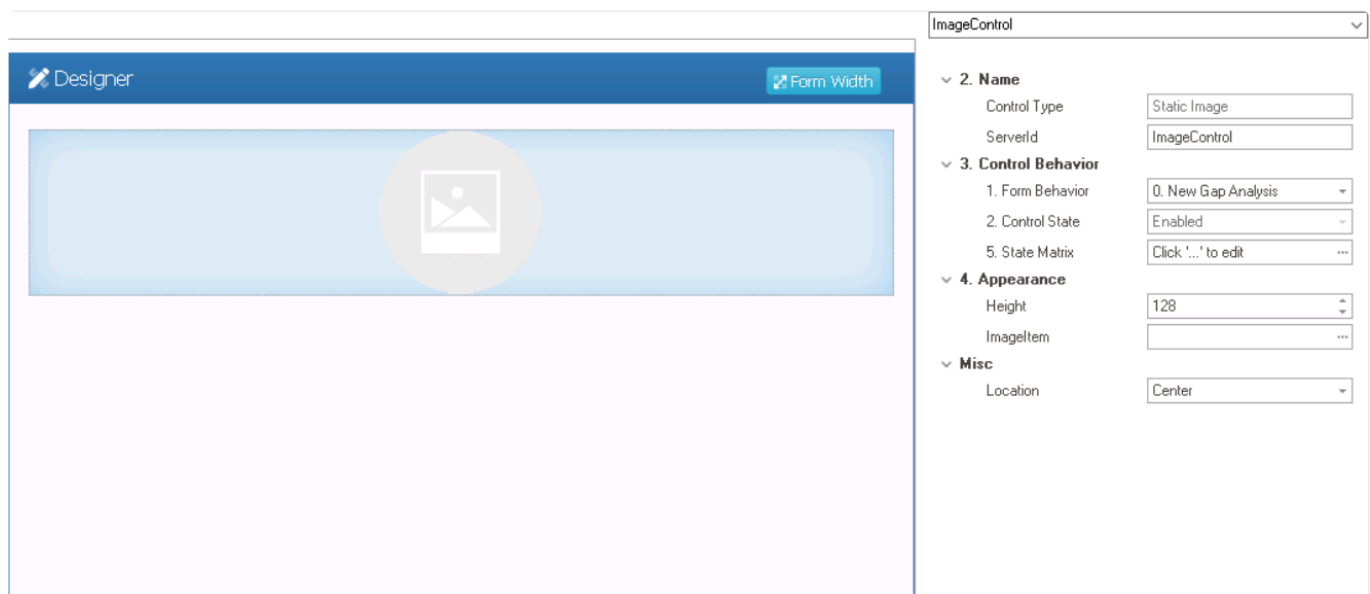
 

## 12.1.2.1.1.26. Static Image

The static image control enables users to display a fixed or pre-determined image within the form, providing a simple and efficient means of presenting visual content without the need for dynamic rendering or user-uploaded content.



When a static image control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

1. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
2. **Control Behavior:** In this category, users can configure the behavior of the control and define

how it interacts with the form or application.

- **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
  - Any state or validation properties set for the control will apply only to the selected form behavior.



Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (\*\*\*) inside the field.

| Form Behavior          | Control State |     |
|------------------------|---------------|-----|
| 1. Modify Gap Analysis | ✓ Enabled     | ... |
| 2. New Gap Analysis    | ✓ Enabled     | ... |
| 4. Review Gap Analysis | ✓ Enabled     | ... |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.



By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

3. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 128. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
  - **Image Item:** This field allows users to upload an image file to the system.
    - It will display the [Application Files](#) window, in which users can add and select the image file.
4. **Misc:** In this category, users can define additional properties for the control.
  - **Location:** This is a single-select dropdown field that allows users to choose the location of the control.
    - Location Options: Right, Center, or Left.
      - Right: This option places the control on the right-hand side of the screen.
      - Center: This option centers the control in the middle of the screen.
      - Left: This option places the control on the left-hand side of the screen.

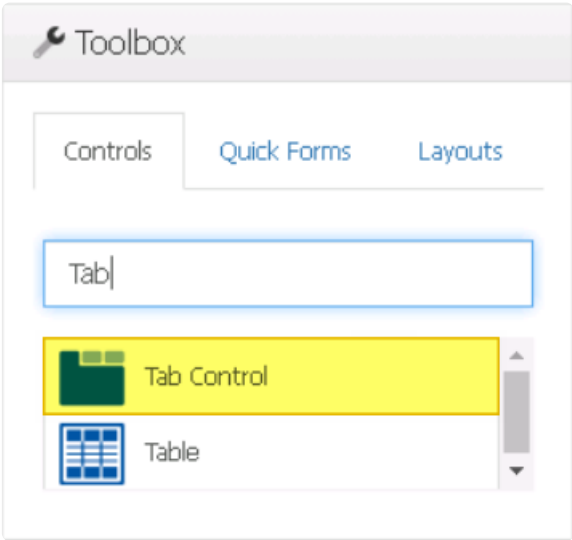


Users do not need to create or link an attribute for static image controls.

### Static Image Control Preview: Design View vs. Web View


# 12.1.2.1.1.27. Tab Control

The tab control is a user interface element that displays multiple tabs, each representing a different section of content or functionality. It allows users to easily switch between different views or modes within a form, providing a convenient way to organize and access information.



When an tab control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

1. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.

- Server ID refers to the name used to call the control in the code.
- 2. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state properties set for the control will apply only to the selected form behavior.



Only form behaviors linked to the associated form will be displayed in this dropdown.

- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.

| Form Behavior          | Control State |
|------------------------|---------------|
| 1. Modify Gap Analysis | ✓ Enabled     |
| 2. New Gap Analysis    | ✓ Enabled     |
| 4. Review Gap Analysis | ✓ Enabled     |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and selectable in the web application.
    2. Disabled: This option allows for the control to be visible but not selectable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
  - **Apply:** This button allows users to save their changes.
  - **OK:** This button allows users to save their changes and close the window.
  - **Cancel:** This button allows users to cancel their changes and close the window.



✿ By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.

- **Enable Lazy Loading:** This is a checkbox which, if selected, allows users to load tab content on-demand, rather than loading all the content upfront when the form is first loaded.

✿ If this checkbox is selected, the form will only load the content for the active tab and delay the loading of other tabs until they are selected by the user. This means that the initial loading time of the form is reduced, as the content for the inactive tabs is not loaded until needed. This can also mean, however, that there may be a slight delay when switching between tabs.

- **Tab Pages Behavior:** This field allows users to configure the behavior of individual tabs within the tab control.
  - To access the tab behavior properties, click on the ellipses (...) inside the field.

- **Name:** This field displays the name of the tab. It cannot be modified.
- **Caption:** This field displays the caption or title of the table. It cannot be modified.
- **Is Selected:** This checkbox determines whether the tab is pre-selected or not.
  - To pre-select a tab, users must check the **Is Selected** checkbox for that tab.

✿ When a user accesses a tab control, the tab that is pre-selected will be chosen by default. Only one tab can be pre-

selected at a time.

- **State:** This is a single-select dropdown field that displays the list of states. It allows users to modify the current state of the tab.
  - **States**
    1. Enabled: This option allows for the tab to be visible and selectable in the web application.
    2. Disabled: This option allows for the tab to be visible but not selectable in the web application.
    3. Invisible: This option allows for the tab to be hidden in the web application.

3. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Allow Looping:** This is a checkbox which, if selected, allows for a tab set to be looped so that when users reach the last tab in a set, the first tab is automatically next to be shown.
  - **Tab Looping Preview: Web View**

| Audit Details | Scope | Resources | Requirements | Agenda |
|---------------|-------|-----------|--------------|--------|
|---------------|-------|-----------|--------------|--------|

- **Fixed Tabs:** This is a checkbox which, if selected, allows for a tab set to be fixed so that all tabs are displayed on screen simultaneously. Each tab is placed at a fixed width, as determined by the total number of tabs in the set.
  - By default, if tabs are not fixed, they are set up as scrollable. A tab set is only scrollable, however, if it can no longer fit uniformly on screen.
- **Floating Tab Menu:** This is a checkbox which, if selected, allows for a tab set to remain in a fixed position as users scroll the page, “floating” above the content.
  - **Floating Tab Preview: Web View**

Home Page / Quality Management System / Audit / Requirements Management / Import Requirements from EPC

### Import Requirements from EPC

Import Control **Import Document** Import Process Import Capability Import Rule

In this tab, you can select which document(s) to import from EPC.

#### Folder

Process Docur ▾

Import All

Import Selected

Select All

Deselect All

#### EPC Documents

✎ Inline Edit

📄 Export to Excel

| Parent                      | Identifier | Name  | Selected For Import |
|-----------------------------|------------|---|---------------------|
|                             |            |   |                     |
| 13.4 Manage Change Reque... |            | 13.4.4 Assess and Evaluate the Change Request | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.5 Authorize the Change Request           | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.3 Review Request                         | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.1 Create and Record the Change Request   | ✓                   |
| 13.4 Manage Change Reque... |            | 13.4.2 Manage Emergency Change Requests       | ✓                   |



Rather than having borders on all sides, a floating tab set has only a bottom border.

- **Header at Top/Bottom:** This is a checkbox which, if selected, allows for the tab headers to be displayed at both the top and bottom of the form.
- **Tab Pages:** This field allows users to customize the appearance of individual tabs within the tab control.
  - To access these properties, click on the ellipses (...) inside the field.
    - It will display the [Tab Pages Editor](#) window.



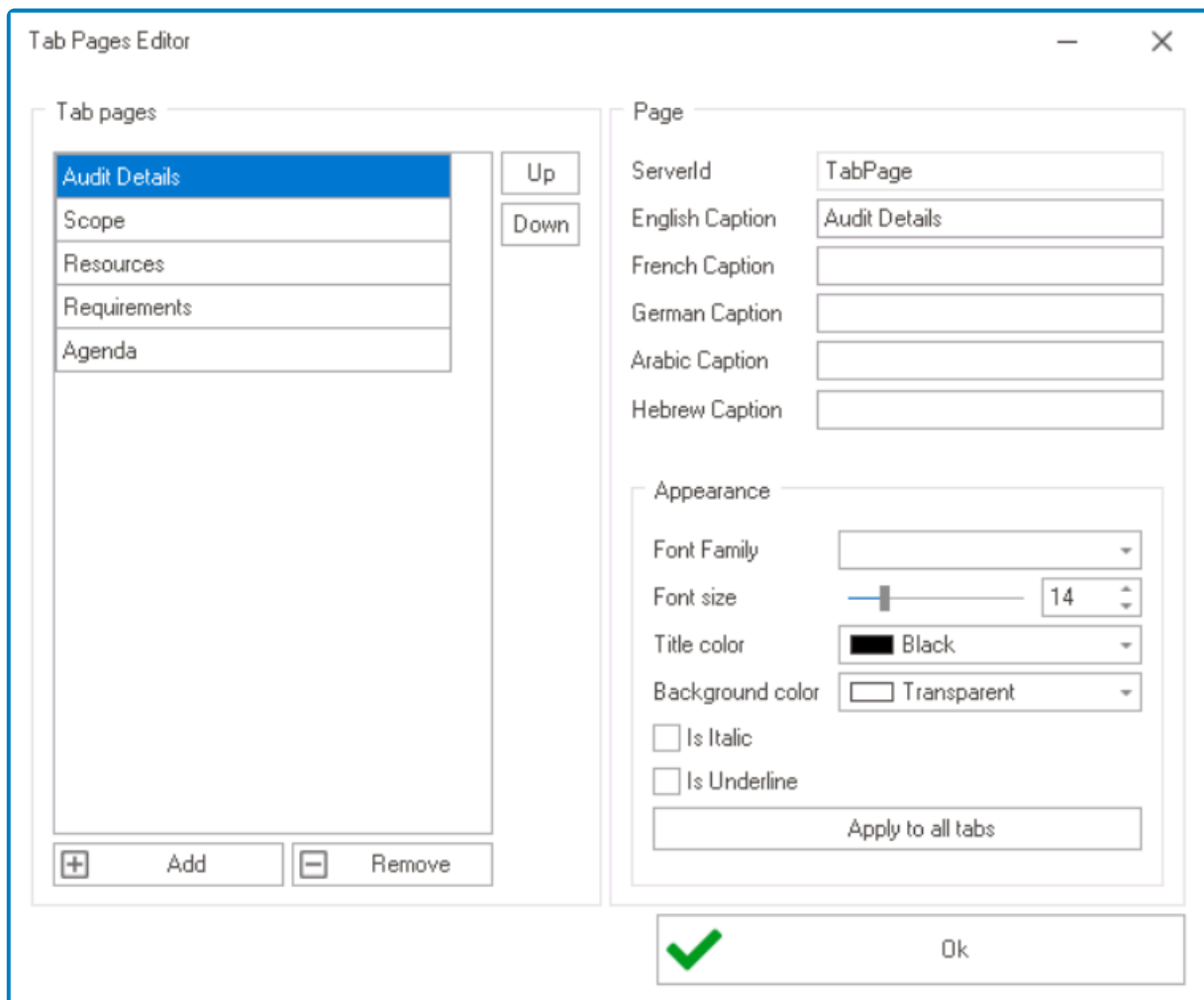
Once the tab control is added, users can proceed to build its components by dragging layouts and other controls from the toolbox and dropping them inside the tabs.

#### Tab Control Preview: Web View

|                      |       |           |              |        |
|----------------------|-------|-----------|--------------|--------|
| <b>Audit Details</b> | Scope | Resources | Requirements | Agenda |
|----------------------|-------|-----------|--------------|--------|

## 12.1.2.1.1.27.1. Tab Pages Editor

In the Tab Pages Editor window, users can customize the appearance of individual tabs within the tab control.



1. **Add:** This button allows users to add a new tab.
  - Once added, the new tab will be displayed in the **Tab Pages** box and its properties will be shown in the **Page** section.
2. **Remove:** This button allows users to remove a selected tab.
3. **Up / Down:** These buttons allow users to reorder the tabs in the list.
  - To move a tab up or down, select it from the list and click on the **Up** or **Down** button. This will move the selected tab one position up or down in the list, respectively.
4. **Server ID:** This field displays the Server ID for the tab. It cannot be modified.
  - Server ID refers to the name used to call the tab in the code.
5. **Caption:** This is a text field that allows users to input the displayed name or caption of the tab.
  - As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the tab, regardless of other language options that may be available.

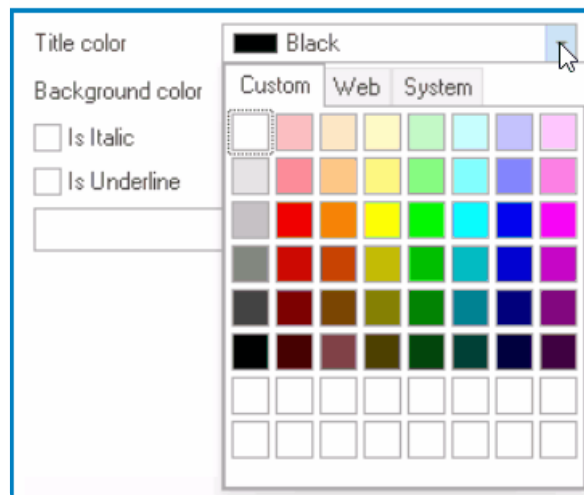


Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the

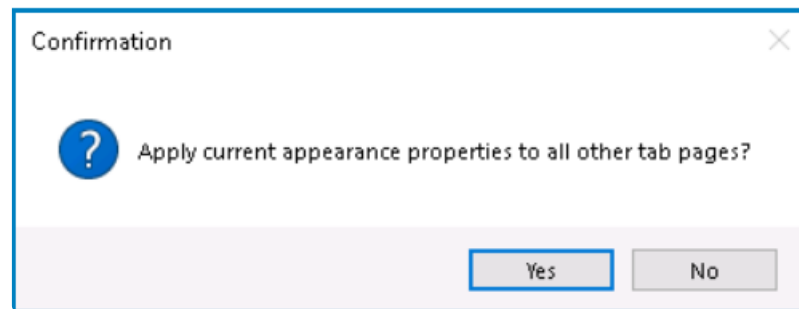
web server: English, French, German, Arabic, and Hebrew.

6. **Appearance:** In this section, users can customize the visual appearance of the tab.

- **Font Family:** This field allows users to choose the font family from a dropdown menu.
- **Font Size:** This field allows users to set the font size of the caption text.
  - By default, the font size is set to 14. To adjust this size, use either the slider or the numeric updown field.
    - In the numeric updown field, users can enter numbers either manually or by clicking the increment/decrement arrows.
- **Title Color:** This field allows users to choose the font color of the caption text from a dropdown menu.
  - It contains three tabs: Custom, Web, and System.



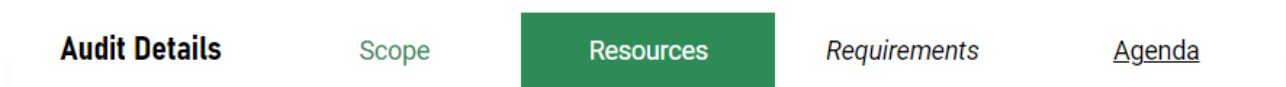
- In the Custom tab, users can choose a font color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
- In the Web tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
- In the System tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
- **Background:** This field allows users to choose the background color of the caption text from a dropdown menu.
  - It also has three tabs, similar to the **Title Color** field, where users can select a color from a color picker or a scrollable list of pre-defined options.
- **Is Italic:** This is a checkbox which, if selected, allows users to apply an italic or slanted effect to the caption text.
- **Is Underline:** This is a checkbox which, if selected, allows users to apply an underline effect to the caption text.
- **Apply to All Tabs:** This button allows users to apply the same caption appearance properties to all tabs within the tab control.
  - If this button is selected, users will be alerted of the action with the following confirmation message:



- **Yes or No:** To confirm the application, click on the **Yes** button. To cancel, click on the **No** button.

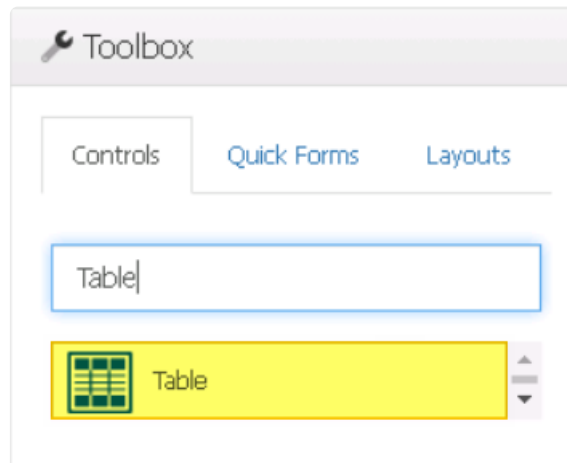
7. **OK:** This button allows users to save their changes and exit the Tab Pages Editor window.

### Customizing Tab Control: Web View



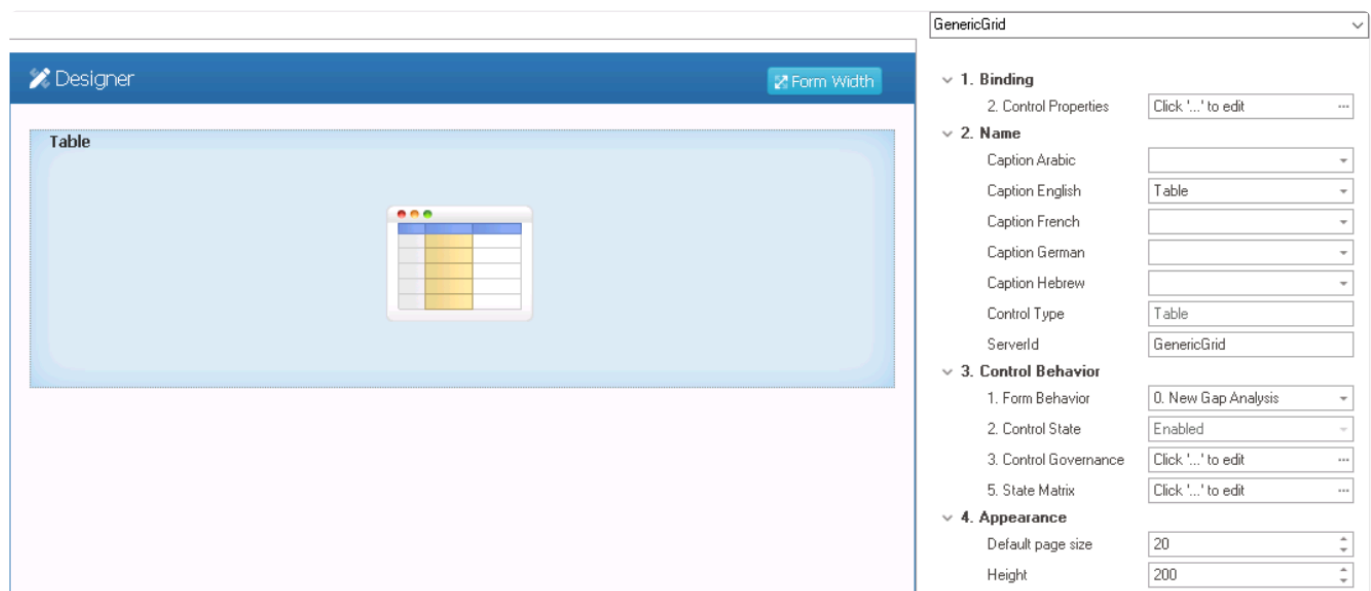
## 12.1.2.1.1.28. Table

Table controls are useful tools for displaying and manipulating large sets of data in a structured format. They enable users to sort, filter, add, edit, and interact with data in a tabular view, making it easier to visualize and analyze complex information.



✿ While grids are typically based on a single entity, tables are populated by data from various entities based on a specific query.

When a table control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can define the binding properties of the control.
  - **Control Properties:** This field allows users to configure the control properties.




- To access the control properties, click on the ellipses (...) inside the field.
  - It will display the [Control Properties](#) window.
- 2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
- 3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated to the control.
    - Any state properties set for the control will apply only to the selected form behavior.





Only form behaviors linked to the associated form will be displayed in this dropdown.

- **Control State:** This field displays the current state of the control.
  - **Control States**
    1. Enabled: This option allows for the control to be visible and editable in the web application.
    2. Disabled: This option allows for the control to be visible but not editable in the web application.
    3. Invisible: This option allows for the control to be hidden in the web application.
- **Control Governance:** This field allows users to configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field.



- **Enable View:** This is a checkbox which, if selected, allows users to enable view mode for the control.
  - If this checkbox is selected, it will enable the **View Form Behavior** field.
- **View Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
  - To define a new form behavior, select the  button.
    - It will display the [Form Behavior Properties](#) window.
  - To modify an existing form behavior, select the  button.
  - To remove a form behavior, select the  button.

 To set up the 'View Form', users only need to provide a name and a title.

- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.
- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (  ) inside the field.

| Form Behavior          | Control State |                          | View |
|------------------------|---------------|--------------------------|------|
| 1. Modify Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | ...  |
| 2. New Gap Analysis    | ✓ Enabled     | <input type="checkbox"/> | ...  |
| 3. Review Gap Analysis | ✓ Enabled     | <input type="checkbox"/> | ...  |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Control Governance:** This field allows users to configure the control governance properties.
  - If selected, it will display the Form / Behavior / Control window (described above).
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
- **Cancel:** This button allows users to cancel their changes and close the window.



By modifying the control state per form behavior, users can determine whether the control is made enabled, disabled, or invisible at different stages of the workflow.



In this window, users can also configure the governance properties for the control across multiple form behaviors. To gain an understanding of how these properties can be configured, [click here](#).

4. **Appearance:** In this category, users can customize the visual appearance of the control.

- **Default Page Size:** This field allows users to set the number of items that are displayed in the table at one time.
  - By default, the size is set to 20. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.



In the web application, a pagination bar appears at the bottom of a table when the number of items in the table exceeds the specified threshold set by the **Default Page Size** field. The pagination bar includes controls such as buttons for moving to the next or previous page, buttons for navigating to the first or last page, and a dropdown field for changing the default number of items displayed per page.

- **Height:** This is a numeric updown field that allows users to set the height of the control.
  - By default, the height is set to 400. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

### Table Control Preview: Web View

|   | Login                | Name                 | User Code            | Alternative Code     | Active                   | Email                  | Telephone            | Address              |
|---|----------------------|----------------------|----------------------|----------------------|--------------------------|------------------------|----------------------|----------------------|
|   | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="checkbox"/> | <input type="text"/>   | <input type="text"/> | <input type="text"/> |
| 1 | fred.bloggs          | Fred Bloggs          | FB                   |                      | true                     | fred.bloggs@interfa... | (800) 555-0100       | 123 Main Str...      |
| 2 | jan.janssen          | Jan Janssen          | JJ                   |                      | true                     | jan.janssen@interfa... | (800) 555-0101       | 456 Park Ave...      |
| 3 | jane.doe             | Jane Doe             | JD1                  |                      | true                     | jane.doe@interfa...    | (800) 555-0102       | 1800 Amphibi...      |
| 4 | joe.bloggs           | Joe Bloggs           | JB                   |                      | true                     | joe.bloggs@interfa...  | (800) 555-0103       | Avenida da Pa...     |
| 5 | john.doe             | John Doe             | JD2                  |                      | true                     | john.doe@interfa...    | (800) 555-0104       | 8 Rue du Nom...      |

⏮ ⏪

1

2

3

4

5

...

▶ ⏭

5 ▾




Export

1 - 5 of 11726 items ↺

## 12.1.2.1.1.28.1. Table – Control Properties

In the Table tab of the Control Properties window, users can build and customize the table.

\* In the General tab of this window, users can configure business rules for the single-select dropdown control. For more information on this matter, please [click here](#) and navigate to the relevant section.

1. **Caption:** This field displays the (English) caption of the table.
  - To modify the caption displayed in this field, refer to the [Name](#) category in the Control Properties panel.
2. **Connect to Data Source:** In this tab, users can link the appropriate data source to the table.
  - **Data Source:** This field retrieves the list of data sources (from both the [Dropdown Data Sources](#) section of the **Forms** module and the [KPIs](#) section of the **Entities** module). It allows users to select the data source for the table.
    - To define a new data source, select the  button.
      - It will display the [KPI Properties](#) window.
    - To modify an existing data source, select the  button.
    - To remove a data source, select the  button.
      - This button will only be visible once users have added/selected a data source.

\* The selected data source will provide the data for the table.

- **Parameters:** This is a table field that allows users to define parameters for the query.
    - Parameters are values that are used to filter or modify the data returned by the query.
      - Users can define the parameters in the table by specifying a parameter name and data type. Once the parameters are defined, users can provide values for the parameters by entering them in the [Value](#) column of the table.
    - When the query is executed, it uses the provided parameter values to filter or modify the data.
  - **Execute:** This button allows users to execute the query with the selected data source and parameters.
    - Once the query is executed, the resulting data will be displayed in the table.
3. **Color Alert:** In this tab, users can apply color alerts to the table, which can highlight specific rows or columns based on certain criteria.
    - **Add Alert:** This button allows users to add a new color alert to the table.
    - **Remove Alert:** This button allows users to remove a selected color alert.
    - **Alert Name:** This is a text field that allows users to input a name for the color alert.
    - **Filter:** This field allows users to define a criteria statement that evaluates to true or false for each row in the column.
      - The criteria can include typical operators and entity fields from the data source.

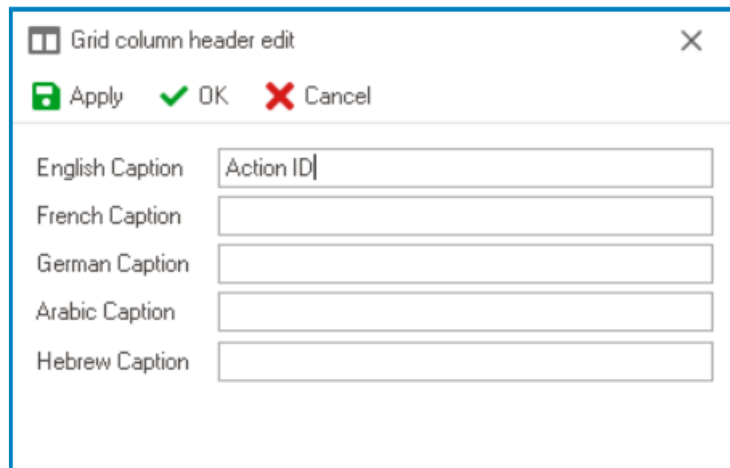


For example, the alert criteria  $[y2] > [y1]$  would highlight rows where the value in the 'y2' column is greater than the value in the 'y1' column.

- **Field:** This field retrieves the list of fields related to the data source, which users can use to specify the column that the color alert should be applied to.
  - **All:** This is a checkbox which, if selected, allows users to apply the color alert to all fields in the table.
    - If this checkbox is selected, the **Field** dropdown will be disabled.
  - **Color:** This field allows users to select a background color for the rows or columns that meet the alert criteria.
    - It contains two tabs: Custom and Web.
      - In the Custom tab, users can choose a background color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
      - In the Web tab, users can select a background color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
  - **Font Color:** This field allows users to select a font color for the rows or columns that meet the alert criteria.
    - It contains three tabs: Custom, Web, and System.
      - In the Custom tab, users can choose a font color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
      - In the Web tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
      - In the System tab, users can select a font color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
4. **Add Column:** This button allows users to add a new column to the table.
    - If selected, it will display the [Grid Column](#) window, in which users can define the properties of the column.
  5. **Remove Column:** This button allows users to remove a selected column in the table.
  6. **Generate Columns Automatically:** This is a checkbox which, if selected, allows users to automatically add all fields in the related query as columns in the table.
  7. **Sort Ascending:** This is a checkbox which, if selected, allows users to sort the table data in ascending order based on the selected column header.
  8. **Show Filter:** This is a checkbox which, if selected, allows users to display a filter row at the top of the table where they can filter the data based on specific criteria.
  9. **Show Aggregation Row:** This is a checkbox which, if selected, allows users to display aggregate values for selected columns in the table footer, such as sum or average.
  10. **Column Customization:** In this section, users can access and customize the columns populating the table.

| Position | Field           | Field Type       | Header           | Width | Align | Direction | Visible                             | Resizable                           | Sortable                            | Sort | Format  | Aggregate |
|----------|-----------------|------------------|------------------|-------|-------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|------|---------|-----------|
| 0        | ID              | System.String    | ID               | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |
| 0        | Description     | System.String    | Description      | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |
| 0        | Login           | System.String    | Login            | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |
| 0        | Name            | System.String an | Name             | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |
| 0        | UserCode        | System.String    | User Code        | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |
| 0        | AlternativeCode | System.String    | Alternative Code | 100   | left  | ltr       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 00   | Textbox | Empty     |

- **Position:** This field allows users to specify the order in which the column appears in the table.
- **Field:** This field displays the corresponding attribute for the column.
- **Field Type:** This field displays the data type of the attribute linked to the column.
- **Header:** This field allows users to specify the text that appears in the header of the column.
  - Users can input their preferred header directly into this field. Alternatively, they can click on the ellipses to display the Grid Column Header Edit window, in which they can input or modify the caption in all available languages.



The dialog box titled "Grid column header edit" contains a close button (X) in the top right corner. Below the title bar are three buttons: "Apply" (with a green icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon). The main area of the dialog has five text input fields, each preceded by a language label: "English Caption", "French Caption", "German Caption", "Arabic Caption", and "Hebrew Caption". The "English Caption" field contains the text "Action ID".

- As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the column, regardless of other language options that may be available.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Width:** This field allows users to set the width of the column in pixels.
- **Align:** This field allows users to specify the alignment of the text within the column. Users can choose their alignment preference from a dropdown menu.
  - **Alignment Options**
    1. Left: This option aligns the text to the left side of the column.
    2. Right: This option aligns the text to the right side of the column.
    3. Center: This option centers the text within the column.
    4. Unset: This option aligns the text based on the language or culture setting in the web application.

- **Direction:** This field allows users to specify the direction of the text within the column. Users can choose their direction preference from a dropdown menu.
  - **Direction Options**
    1. Left-to-Right: This option specifies that the text in the column should be read from left to right, in the standard direction for languages that use a Latin-based alphabet (such as English).
    2. Right-to-Left: This option specifies that the text in the column should be read from right to left, in the opposite direction to the standard direction for languages that use a Latin-based alphabet (such as Arabic or Hebrew).
    3. Unset: This option specifies that the text direction should be based on the language or culture setting in the web application.
- **Visible:** This is a checkbox which, if selected, allows for the column to be visible in the table.
- **Resizable:** This is a checkbox which, if selected, allows users to resize the width of the column in the table.
- **Sortable:** This is a checkbox which, if selected, allows users to sort the column by clicking on its header.
- **Sort:** This field allows users to set the initial sort order of the column. Users can choose the sort order from a dropdown menu.
  - By selecting a value from this dropdown, users are essentially assigning a priority level for the column.
- **Format:** This field allows users to specify the display format of the column.
  - To select or modify the display format, click on the ellipses (...) inside the field.
    - It will display the [Edit Type](#) window.
- **Aggregate:** This field allows users to specify how the data in the column should be aggregated. Users can choose their preferred aggregation option from a dropdown menu.
  - **Aggregation Options**
    1. Average: This option allows users to calculate the average of the values in the column.
    2. Empty: This option allows users to specify that the table should not perform any aggregation on the column.
    3. Sum: This option allows users to calculate the sum of the values in the column.

## 12.1.2.1.1.29. Textbox (Single-Line, Multi-Line & Password)

Textbox controls are interactive elements that allow users to input, modify, and display text in forms.

There are three main types of textbox controls in the program: single-line, multi-line, and password.

### 1. Single-Line Textbox

A single-line textbox control allows users to input short pieces of text, such as names or email addresses. In the web application, it appears as a small, rectangular box that can only display one line of text at a time.

**Audit Name**

### 2. Multi-Line Textbox

A multi-line textbox control allows users to input longer pieces of text, such as paragraphs or messages. In the web application, it appears as a larger rectangular box with a border that is designed to display multiple lines of text.

✿ Depending on the amount of text entered, a vertical scrollbar may appear on the side of the textbox to allow users to scroll through the contents.

### 3. Password Box

A password box control is a specialized type of single-line textbox that masks the text input by displaying dots instead of the actual characters. This is useful for entering passwords or other sensitive information that should not be visible to others.

**Password**

✿ Users can easily add any of these controls to the design environment by dragging and dropping them from the toolbox. Users also have the flexibility to switch between textbox



types at any point during the design process simply by selecting a different option from the corresponding dropdown in the textbox control properties.

When a textbox control is selected in the design environment, its associated properties are displayed in the Control Properties panel. **For more information on these properties, [click here](#).**

## 12.1.2.1.1.29.1. Textbox Control Properties

When a textbox control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

✱ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

✱ To create attributes for textbox controls, users need to choose data types such as **VarChar** (VarChar or VarChar\_Max) or **NVarChar** (NVarChar or NVarChar\_Max).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - For more information on these properties and how to configure them, [click here](#).

! Please note that when it comes to control governance properties, users are limited to configuring only the required field validation properties for textbox controls. It is not possible to set up required range validations.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Align:** This is a single-select dropdown field that allows users to choose the alignment of the text inside the control.
    - Alignment Options: Right, Center, Left, or Auto By Culture.
      - Auto By Culture: This option aligns the text based on the language or culture setting in the web application.
      - Right: This option aligns the text to the right side of the control.
      - Center: This option aligns the text to the center of the control.
      - Left: This option aligns the text to the left side of the control.
  - **Browser Autocomplete:** This is a checkbox which, if selected, allows users to enable the autocomplete feature for the control, based on data entered previously.
    - **Browser Autocomplete Preview: Web View**

✱ If the autocomplete feature is enabled, the browser will suggest

previously entered data to automatically fill in the input field when the user interacts with the textbox, such as by clicking on it or typing in it

- **Chars and Digits Only:** This is a checkbox which, if selected, allows users to restrict input in the textbox to only alphanumeric characters (letters and numbers).
  - Any special characters, such as punctuation marks or symbols, will not be allowed in the input field.
- **Latin Chars Only:** This is a checkbox which, if selected, allows users to restrict input in the textbox to only Latin alphabet characters.
  - Users will be allowed to input numbers and a limited set of special characters, such as punctuation marks and symbols for mathematical expressions. Any special characters outside this limited set, however, will not be allowed in the input field.
- **Mask:** This field allows users to specify a format for the input data in the textbox control. Users can either enter the format manually or select from a dropdown menu that includes two available formats:
  - aaa-aaa: This format is used when the input data needs to follow a specific pattern of six characters, with the first three characters being letters (a-z) and the second three characters also being letters (a-z), separated by a hyphen (-)
  - a9a/999: This format is used when the input data needs to follow a specific pattern of six characters, with the first and third characters being letters (a-z) and the second being a number (0-9), followed by a forward slash (/), and then three numbers (0-9).
  - **Textbox Mask Pattern Preview: Web View**

|   |   |   |
|---|---|---|
| Alphabetic Mask Pattern   | Alphanumeric Mask Pattern   | Custom Mask Pattern   |
| aaa-aaa <span style="float: right; font-size: 0.7em; cursor: pointer;">✕</span> | a9a/999 <span style="float: right; font-size: 0.7em; cursor: pointer;">✕</span> | aaa-99999-aaa <span style="float: right; font-size: 0.7em; cursor: pointer;">✕</span> |

- **Max Length:** This field allows users to specify the maximum number of characters that can be entered into the control.
  - By default, this field is left blank. To specify a maximum length of input, however, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
- **Placeholder:** This field allows users to input the placeholder text for the control.
  - Users can also directly add a placeholder text to the control simply by clicking on the inside of the control in the design environment.

Category

5. State Matrix

Click '...' to edit

4. Appearance

Placeholder Arabic

Placeholder English

Placeholder French

Placeholder German

Placeholder Hebrew

Tooltip

- As English is set as the default language in the system, any placeholder inputted in the English placeholder field will be used as the display placeholder for the control, regardless of other language options that may be available.
  - Similarly, any placeholder text entered in the design environment will automatically populate the English placeholder field.

\* Users can determine which languages are available for use in the Settings of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

- **Control Placeholder Text Preview: Web View**

**Email Address**

name@example.com

\* A placeholder text is the default value or text that is displayed inside the control until the user enters their own value.

- **Rows:** This field allows users to specify the number of visible rows for the multi-line textbox control, adjusting its height to display a certain amount of content at a time.
  - By default, the number of rows is set to 4. To adjust this value, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

! This field applies only to multi-line textboxes. For single-line textboxes and password boxes, the number of visible rows is fixed at one and cannot be modified.

- **Textbox Type:** This is a single-select dropdown field that allows users to choose the type of textbox.
  - Textbox Types: Single-Line, Multi-Line, or Password.

\* These textboxes also exist as separate controls that can be dragged from the toolbox and dropped onto the design environment.

- **Tooltip:** This field allows users to input a tooltip for the control.
  - In the web application, the tooltip is represented by the tooltip icon ( ? ) next to the control.
  - **Textbox Tooltip Preview: Web View**

Please enter your email address in the format name@example.com.

**Email Address** ?

\* A tooltip is a brief message or description that appears when the user hovers their cursor over the tooltip icon next to the control. This feature can help to provide additional context and guidance to the user as they interact with the form or application.

5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
- For more information on the fields in this category, [click here](#).

### Quick Forms Application

To apply the **Quick Forms** feature to textbox controls, take the following steps:

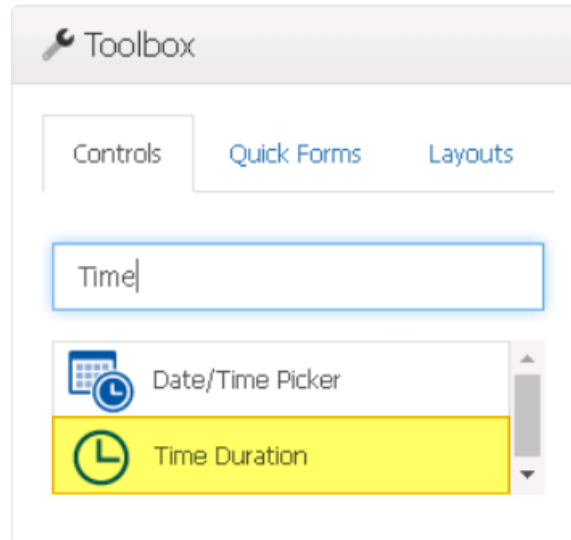
- Drag a textbox control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:



\* For more information on the **Quick Forms** feature, [click here](#).

## 12.1.2.1.1.30. Time Duration

The time picker control allows users to select or input a specific time.



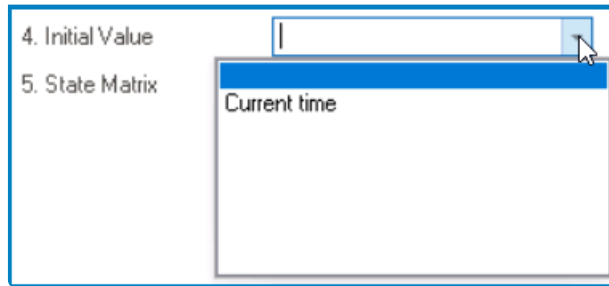
When a time duration control is selected in the design environment, its associated properties are displayed in the Control Properties panel.

✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - For more information on the fields in this category, [click here](#).

✿ To create attributes for time duration controls, users must use the **Time** data type.

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - For more information on these properties and how to configure them, [click here](#).
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **Time Duration Initial Value**  
When configuring the initial value for the time duration control, users can either set a specific time or select the Current Time option from the dropdown menu.



- For more information on these properties and how to configure them, [click here](#).

! Please note that when it comes to control governance properties, users are limited to configuring only the required field validation properties for time duration controls. It is not possible to set up required range validations.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - For more information on these properties and how to configure them, [click here](#).
5. **Caption Appearance:** In this category, users can customize the visual appearance of the control's caption.
  - For more information on the fields in this category, [click here](#).

#### Time Duration Control Preview: Web View

##### Start Time

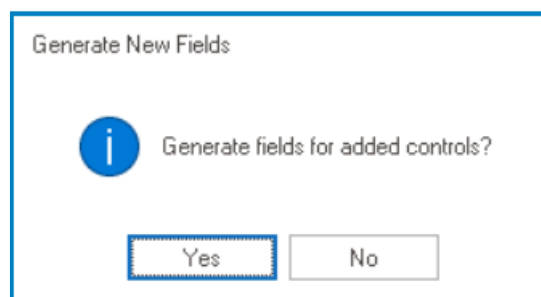


\* For more information on how to use this field in the web application, [click here](#).

#### Quick Forms Application

To apply the **Quick Forms** feature to time duration controls, take the following steps:

- Drag a time duration control from the toolbox and drop it onto the design environment.
- Provide a caption for the control.
- Save the form and select **Yes** in response to the following prompt:





For more information on the **Quick Forms** feature, [click here](#).



## 12.1.2.1.1.30.1. Time Duration Web Display

For those unfamiliar with using the time duration control in the **DBP** web application, the following instructions will guide you through the process:

### Start Time




A time duration control displays a list of pre-defined times for the user to select from. Users can enter a time into this field either manually or by selecting it from the list.

- **Instructions**

To display the list, click on the  icon.

### Start Time



00:00

00:30

01:00

01:30

02:00

02:30


03:00

03:30

04:00

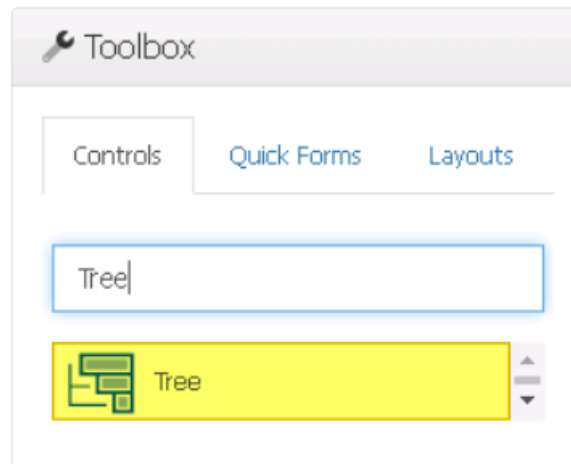
Once selected, the time will populate the field in the following manner:

### Start Time

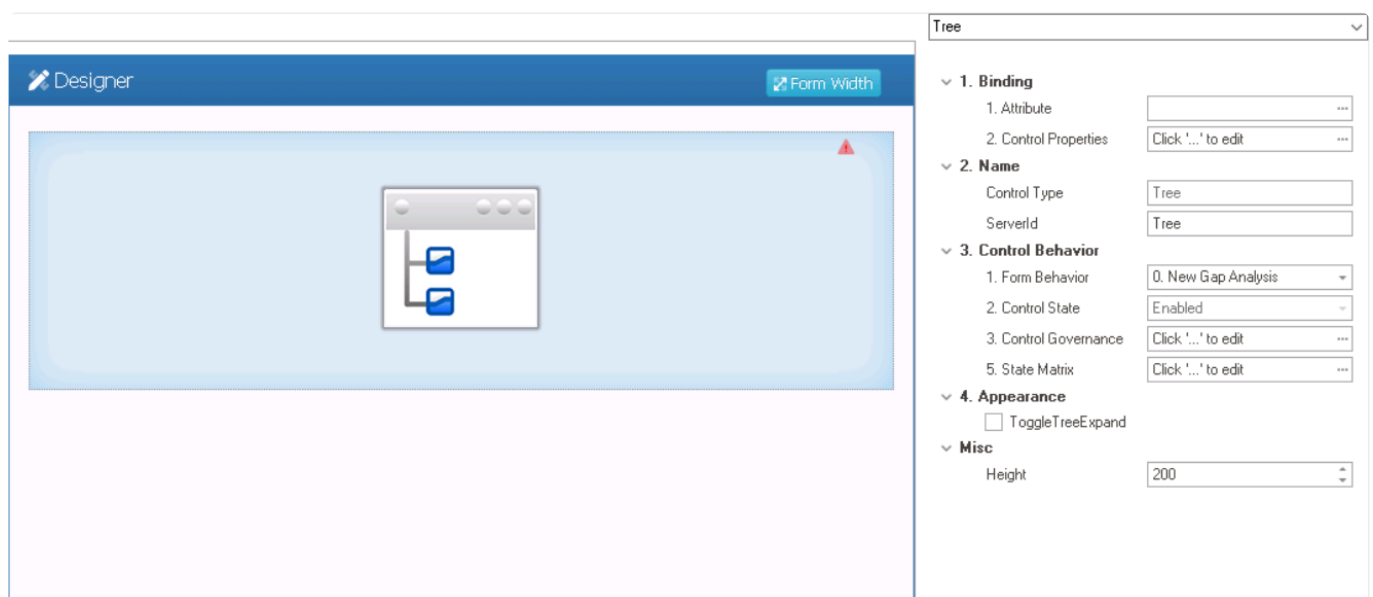


## 12.1.2.1.1.31. Tree

The tree control is a useful tool for displaying and navigating hierarchical data. With its intuitive structure and interactive features, this control simplifies the exploration, manipulation, and visualization of hierarchical information, making it an essential component for managing and presenting complex data relationships.



When a tree control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



✿ To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute and define its binding properties.
  - [For more information on the fields in this category, click here.](#)

 For more information on the required attributes for tree controls, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.

- **Control Governance**

- In the **Control Governance** field, users can configure various properties that govern the functionality of controls.
  - To access the control governance properties, click on the ellipses (...) inside the field. It will display the Form / Behavior / Control window.
    - This window consists of two tabs:
      - Tree Behavior
      - [Validation](#)



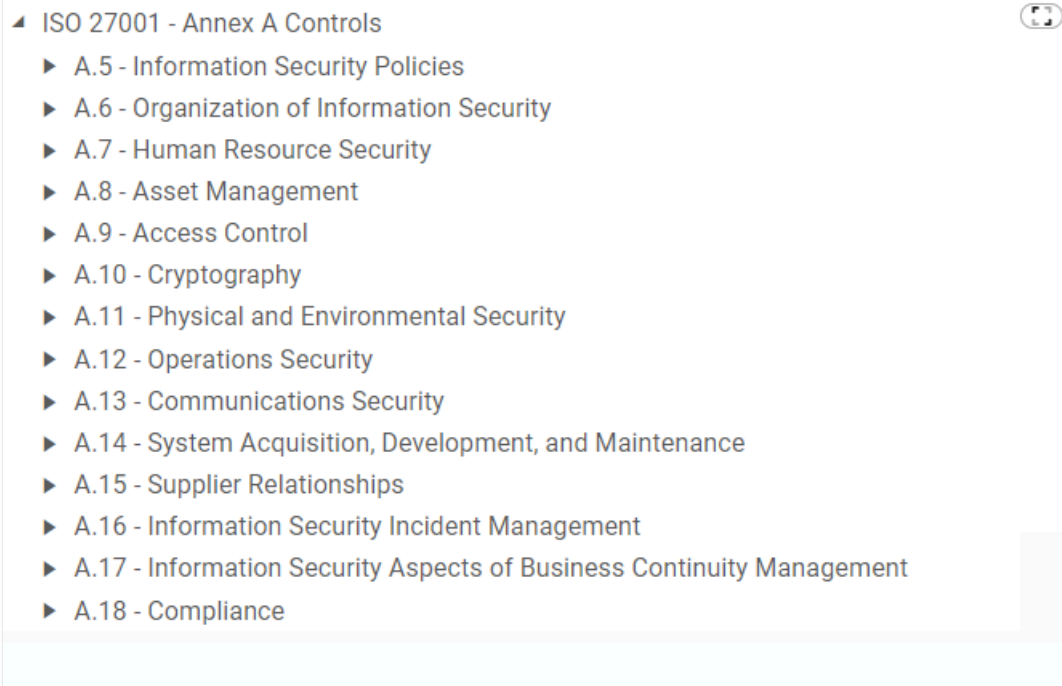
Please note that users are limited to configuring only the required field validation properties for tree controls and cannot set up required range validations.

- For more information on these properties and how to configure them, [click here](#).



Please note that the linked section contains information on setting an initial value, which is not applicable to tree controls and can be ignored.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Toggle Tree Expand:** This is a checkbox which, if selected, enables a toggle functionality that allows users to expand or collapse the tree nodes with a single click or interaction.
    - **Toggle Tree Expand Preview: Web View**
5. **Misc:** In this category, users can define additional properties for the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 200. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.

**Tree Control Preview: Web View**

# 12.1.2.1.1.31.1. Tree – Binding

In the **Binding** category, users can link a Gantt chart to its corresponding attribute and define its binding properties.

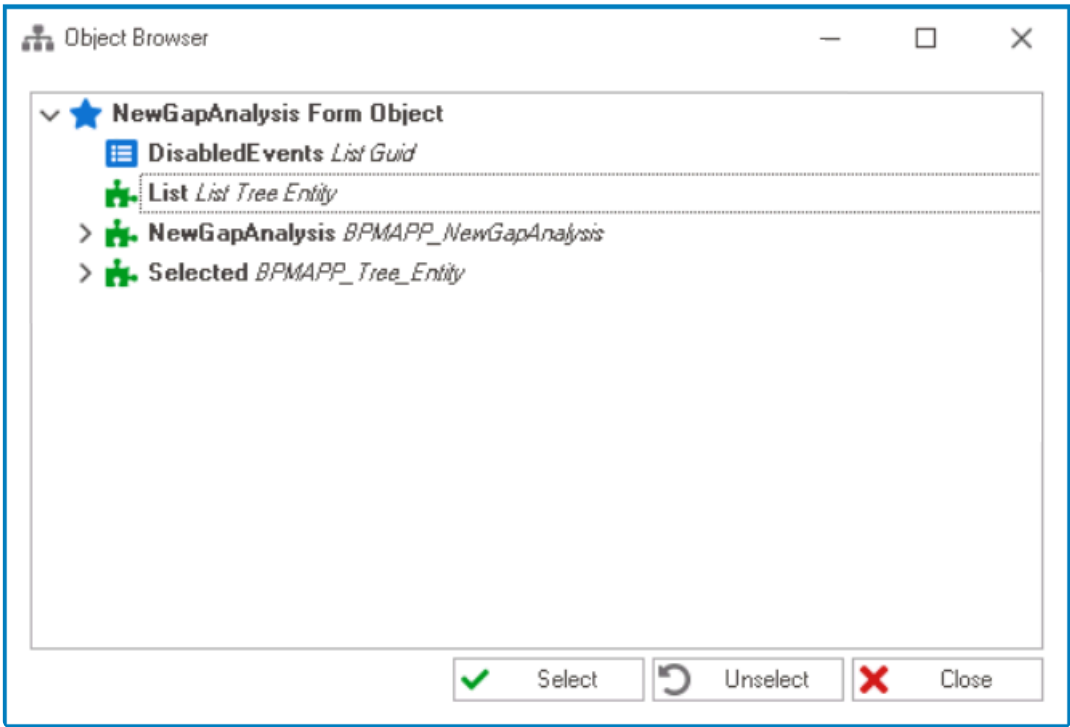
▼ 1. Binding

1. Attribute

2. Control Properties

Click '...' to edit

1. **Attribute:** This field allows users to select the attribute associated to the tree control.
- To access the list of relevant attributes, click on the ellipses ( ... ) inside the field.



✿

In the Object Browser window, users can access a list of the associated entity’s attributes—specifically, the attributes relevant to the tree control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

List

!

Users must bind the tree control to the first local attribute (**List**) created in the **Form Entity** tab. For more information on this matter, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- To access the control properties, click on the ellipses (...) inside the field. It will display the Control Properties window.
  - This window consists of three tabs:
    1. [General](#)
    2. [Tree](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Control Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

# 12.1.2.1.1.31.1.1. Tree – Key Components & Entity Conditions

This section of the manual outlines the key components and entity conditions that are integral to the functionality of the tree control.

## A. Key Components

There are several key components involved in the creation and utilization of the tree control.

These components include:

1. **Tree Entity**

This entity holds the data that is selected using the tree control. It must meet the specified requirements outlined in the **Entity Conditions** section.



This entity is linked to the tree form, acting as the data source from which information is extracted to populate the control.

2. **Tree Form**

This form serves as a platform for entering data into the tree entity. It allows users to input relevant information, such as item details and hierarchical relationships, which are stored in the entity for the tree control.

3. **Main Entity**

This entity is linked to the form where the tree control is being used or implemented. It represents the primary data object that the user is working with or collecting data for.



In the [provided instructions](#) for building these key components, this entity is created using the [Quick Forms](#) feature. Users, however, can choose to create the main entity manually.

4. **Main Form**

This form is where the tree control is implemented. It provides the overall interface and context for incorporating the tree functionality.

For the main form, users must create two local attributes in the [Form Entity](#) tab.

- The first local attribute is responsible for binding the tree control to the form. It acts as a container

or reference to a list of entities that meet the specified conditions outlined in the **Entity Conditions** section.

- The second local attribute is responsible for capturing and storing the currently selected item within the tree control. It acts as a reference to the selected item in the tree entity.



For more detailed instructions on how to create tree controls and their associated attributes, [click here](#).

---

## B. Entity Conditions

To qualify as a tree entity, four specific conditions must be met:

1. **Item Title**

The entity must have a data attribute that represents the title or label assigned to each item within the tree.

2. **Order**

The entity must have a data attribute that determines the positioning or sequence of items within the tree.

3. **Tooltip**

The entity must have a data attribute that allows additional contextual information to be associated with each item in the tree.

4. **Reference to Self (ParentID)**

The entity must have a reference attribute that points to itself. This self-reference enables connections within the tree entity, allowing items to reference and link to other items within the same tree.



This capability creates hierarchical relationships and enables navigation between related nodes.



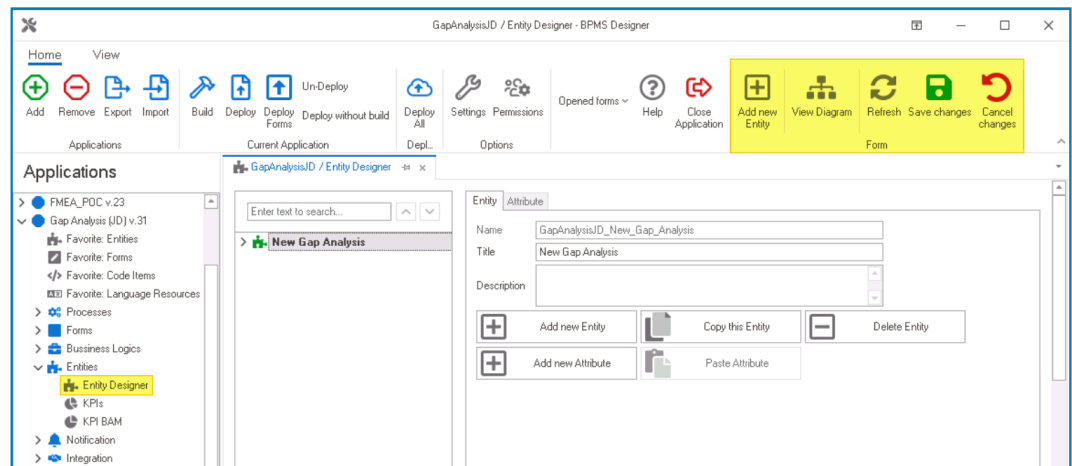
## 12.1.2.1.1.31.1.1.1. Tree – Attributes

To create tree controls and their associated attributes, follow these steps:

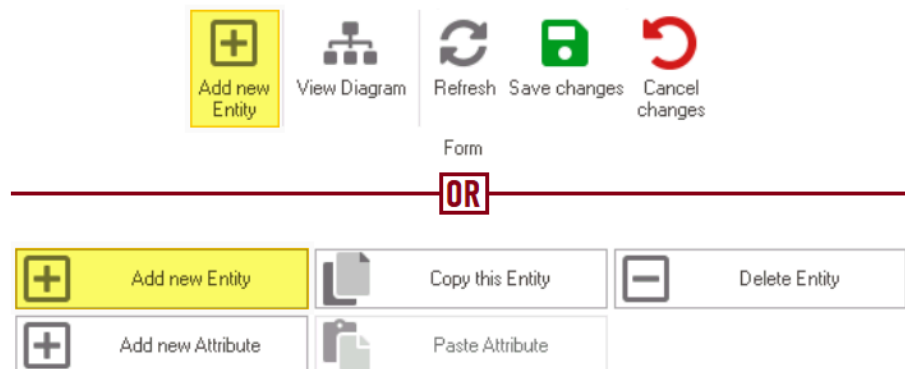
### 1. [Create Tree Entity](#)

#### • Access Entity Properties

- From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



- From either the Entity tab or the menu bar, select the **Add New Entity** option. It will display a set of fields in which you can define the properties of the new entity.



- In the fields of the Entity tab, users can define the properties of the newly created entity.

- In the **Name** field, provide a name for the entity.
- In the **Title** field, either specify a custom title for the entity or let the system automatically generate it based on the provided name.
- In the **Description** field, provide a description for the entity, if needed.

 For information on the required conditions for a tree entity, [click here](#).

## 2. [Create Attributes](#)

- **Access Attribute Properties**

- Once you have created the entity, select the **Add New Attribute** option from the Entity tab. It will direct you to the Attribute tab where you can define the properties of the new attribute.

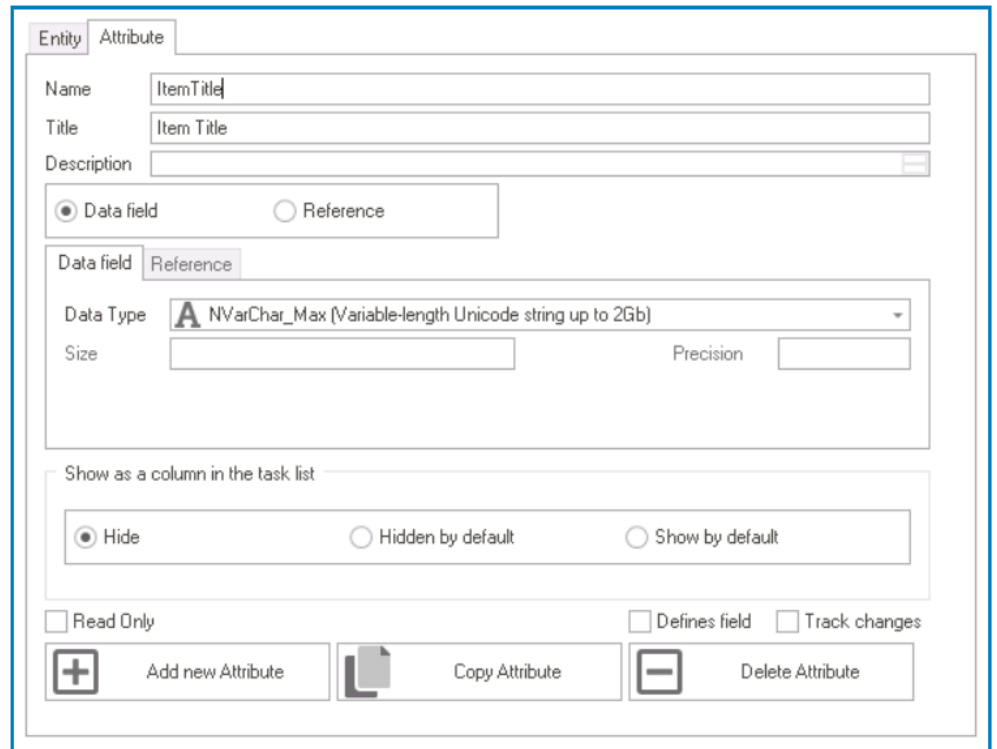
- In the fields of the Attribute tab, users can define the properties of the newly created attribute.

The screenshot shows the 'GapAnalysisJD / Entity Designer' application. On the left, a tree view shows 'New Gap Analysis' and 'Tree Entity' with 'NewAttribute' selected. The main panel is the 'Attribute' tab for 'NewAttribute'. It includes fields for Name, Title, and Description. The 'Data field' radio button is selected. The 'Data Type' dropdown is set to 'UniquelIdentifier (Globally unique identifier)'. The 'Show as a column in the task list' section has 'Show by default' selected. At the bottom, there are checkboxes for 'Read Only', 'Defines field', and 'Track changes', along with buttons for 'Add new Attribute', 'Copy Attribute', and 'Delete Attribute'.

✿ For tree controls, users must create three data attributes (item title, order, and tooltip) and one reference attribute (to itself—parent).

- **Create Data & Reference Attributes**

1. **Item Title:** This attribute is used to capture the title or label assigned to an item in the tree.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Date Field** option.
  - From the **Data Type** dropdown, select **NVarChar\_Max**.



2. **Order:** This attribute is used to capture the order or position of the item within the tree.


- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Data Field** option.
- From the **Data Type** dropdown, select an appropriate numeric data type, such as an integer or decimal, based on the ordering format needed.



For a sequential numbering system (e.g., 1, 2, 3), an integer data type would suffice. For a hierarchical numbering system (e.g., 1.1, 1.2, 1.3), a decimal data type can be used. Please note, however, that if a decimal data type is chosen, users will need to specify the size and precision.

3. **Tooltip:** This attribute is used to capture additional contextual information about the item in the tree.

- In the **Name** field, provide a name for the attribute.
- In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
- Choose the **Data Field** option.
- From the **Data Type** dropdown, select **NVarChar\_Max**.

4. **Parent:** This attribute is used to reference the tree entity.
- In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Choose the **Reference** option.
  - Click on the  button. It will display the Select Reference window. From the entity section, select **[AppName] – [TreeEntity]**. From the field section, select **ID**.

- Selecting the reference will redirect to the Attribute tab, where the entity section will display **[TreeEntity]** and the field section will display **ID**.

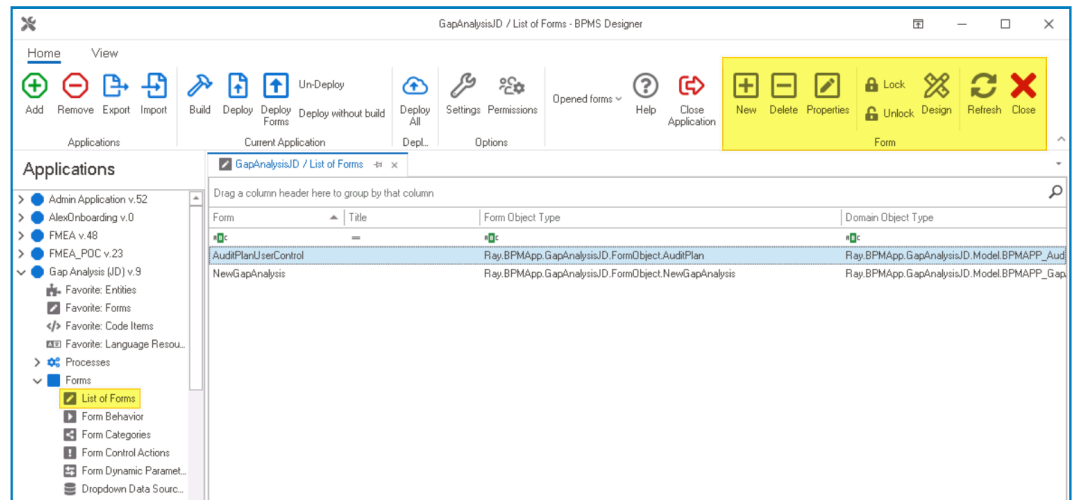
- Once users have created the necessary attributes for the tree control, the end-result should be the following list:

! Remember to select the **Save Changes** option from the menu bar to finalize the configuration.

### 3. [Create Tree Form](#)

- Access Form Properties**

- From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.

**Form Properties**

Apply OK Cancel

Name:

Form Entity:

Entity: [Please Select...] ▼

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- From the **Entity** dropdown, select the tree entity.
- Click **OK** and confirm both the prompt to start designing the form and the prompt to create a draft.



The purpose of this form is to allow data entry into the entity that holds data for the tree.

#### 4. Configure & Design Tree Form

- Once the tree form has been created and the Form Designer has been launched, users can proceed to finalize the form's configuration. There are two key steps to consider:
  - Change Parent Field from Grid to Dropdown**

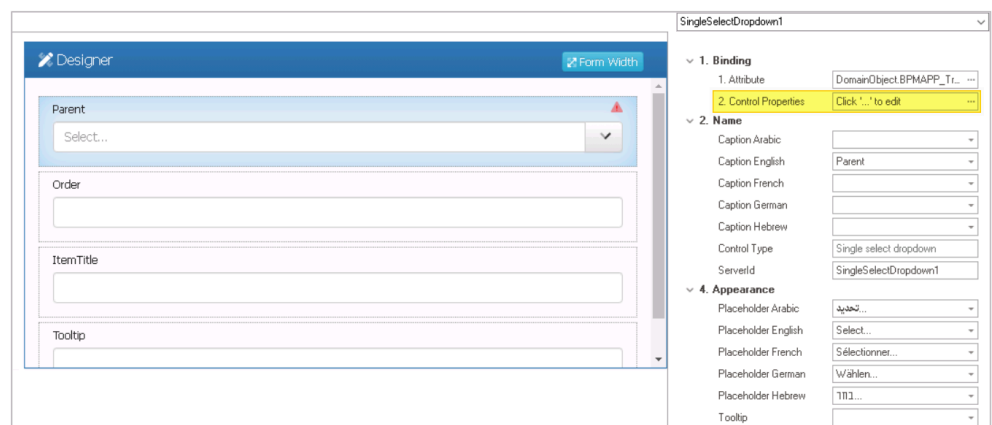
By default, foreign keys are linked to grid controls. To enable connections within the tree entity, however, users must replace the grid representation of the **Parent** field with a single-select dropdown. This can be achieved by removing the grid control, dragging and dropping a single-select dropdown control, and linking it to the **Parent** foreign key.

**!** Users must link the single-select dropdown control to the **ID** field of the **Parent** foreign key (ParentID).

- Configure Parent Dropdown Properties**

To ensure that the **Parent** dropdown accurately retrieves the required information, users must select and configure its data source. Follow these steps to accomplish this:

- Select the single-select dropdown control from the design environment. This will display its associated properties in the Control Properties panel.



- Click on the ellipses inside the **Control Properties** field, and then navigate to the Single-Select tab in the displayed window.



The screenshot shows the 'Control Properties' dialog box for a 'SingleSelectDropdown1' control. The 'Form Control' is 'SingleSelectDropdown1', the 'Entity and Attribute' is 'DomainObject.BPMAPP\_Tree\_EntityParent.Id', and the 'Object type' is 'System.Guid'. The 'General' tab is selected, and the 'Single select' sub-tab is active. The dropdown menu is currently empty. Checkboxes for 'Load data asynchronously' and 'Sort by Display member' are checked. The 'Page size' is set to 1000, 'Load delay (ms)' is 100, 'Min characters for search' is 0, and 'Search Type' is 'Con...'.

- Click on the button to access the [Query Editor](#), which allows users to define a new dropdown data source.

The screenshot shows the 'Query Editor' dialog box. The 'Static' checkbox is unchecked. The 'Title' field contains 'QS\_TreeSelectDataSource'. The 'Dynamic' tab is selected. The 'Connection' dropdown is set to 'BPMS'. Under 'DataBase Table', the 'Select Table' dropdown is set to 'BPMAPP\_Tree\_Entity', and the 'SQL Table' field contains 'SELECT \* FROM ray.BPMAPP\_Tree\_Entity'. The 'SQL' checkbox is checked, and the SQL query 'SELECT \* FROM ray.BPMAPP\_Tree\_Entity' is displayed in the text area. At the bottom, there are buttons for 'Execute', 'Dynamic', 'Edit Code', 'Designer', and 'Remove'. Below these, there is a section 'Indicate if your control is ComboBox' with 'Value Member' set to 'Id' and 'Display Member' set to 'ItemTitle'.

- In the **Title** field, provide a title for the dropdown data source.
- From the **Connection** dropdown, select **BPMS**.
- From the **Select Table** dropdown, select the **tree entity**.
  - Once the tree entity is selected, the **SQL Table** field will automatically populate with a sample SQL query that specifies

the desired table name and includes any necessary SQL syntax to retrieve the data for the dropdown.

- Copy the query from the **SQL Table** field and paste it into the command box.
  - Click on the **Execute** button to verify and execute the SQL query.
  - From the **Value Member** field, select **ID**.
  - From the **Display Member** field, select **Item Title**.
- Click on the **OK** button in both the Query Editor and Control Properties windows to save the changes made.
- After completing the form configuration, users can then proceed to design the form's layout.



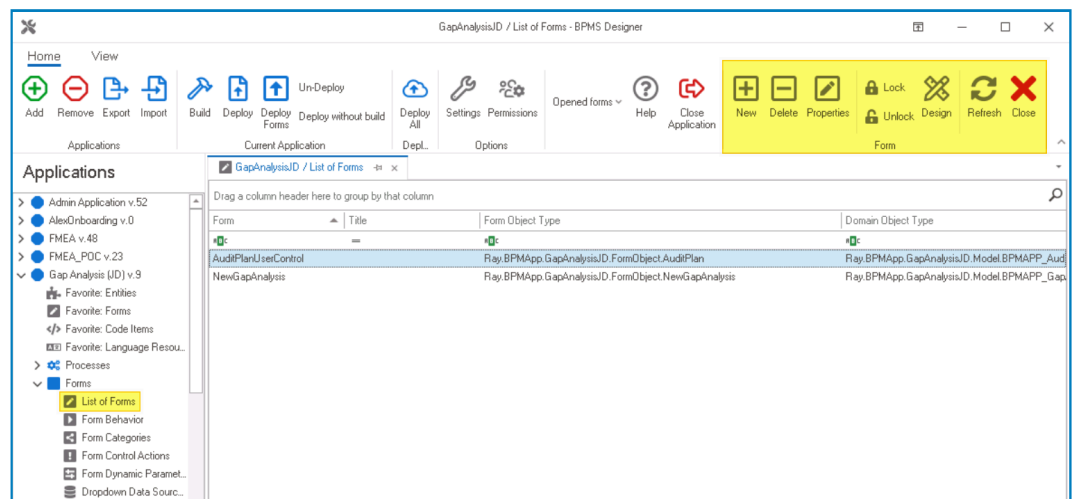
Given that this form will be used to populate the tree control, careful attention should be paid to its layout design to maximize usability.

- Click **Save Form** and confirm the prompt to generate a link in the main menu.

## 5. Create & Design Main Form

### • Access Form Properties

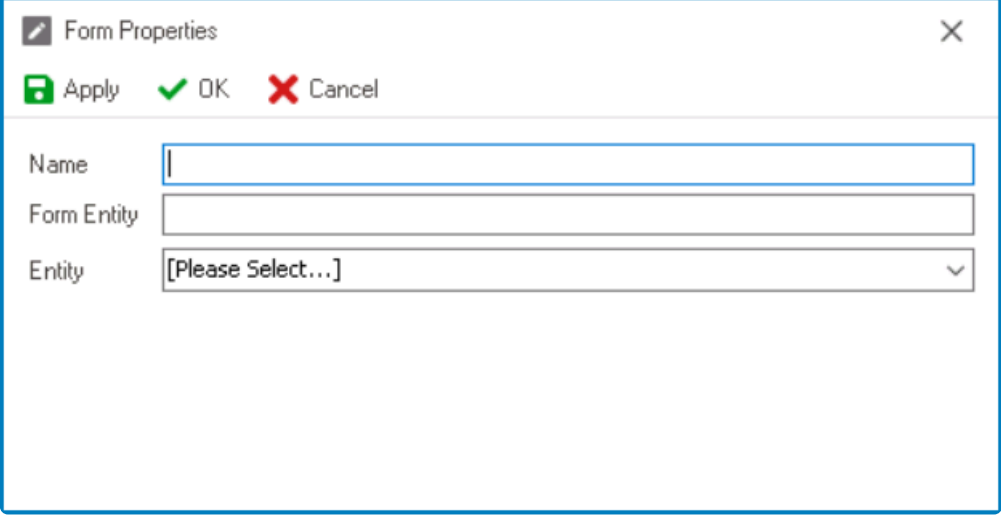
- From the Application Tree, navigate to **Forms > List of Forms** (double-click to open). It will display the List of Forms tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the Form Properties window.



- In the Form Properties window, users can create a new form for the application.



The **Form Properties** dialog box contains the following fields and controls:

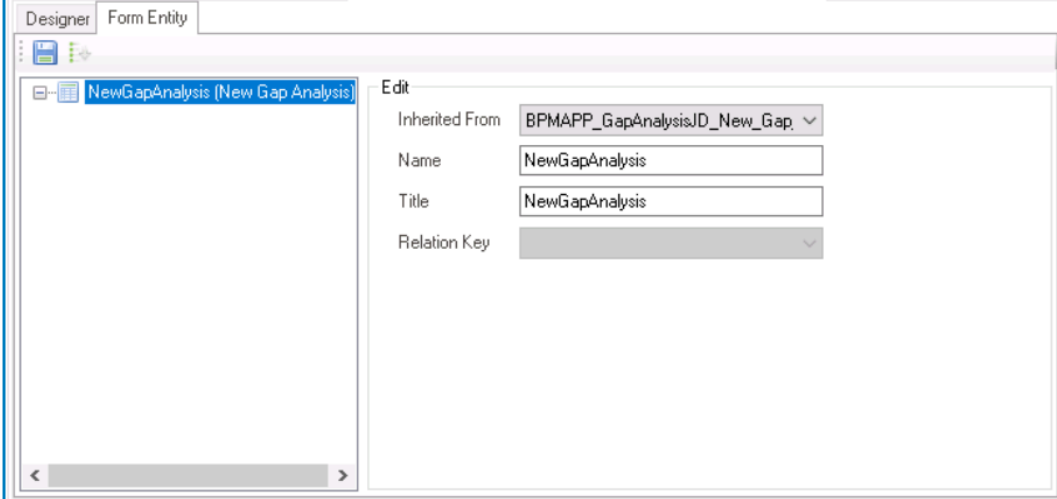
- Name:** A text input field for naming the form.
- Form Entity:** A text input field for specifying the form entity name.
- Entity:** A dropdown menu currently showing "[Please Select...]" with a downward arrow.
- Buttons:** "Apply" (with a floppy disk icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon).

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- Leave the **Entity** field empty.
- Click **OK** and confirm the prompt to start designing the form.

✿ The main form is where the tree control is used or implemented. After creating the main form and launching the [Form Designer](#), users should simply drag the tree control from the toolbox and drop it onto the design environment.

## 6. [Create Local Attributes](#)

- **Access Attribute Properties**
  - From the Form Designer, navigate to the Form Entity tab.



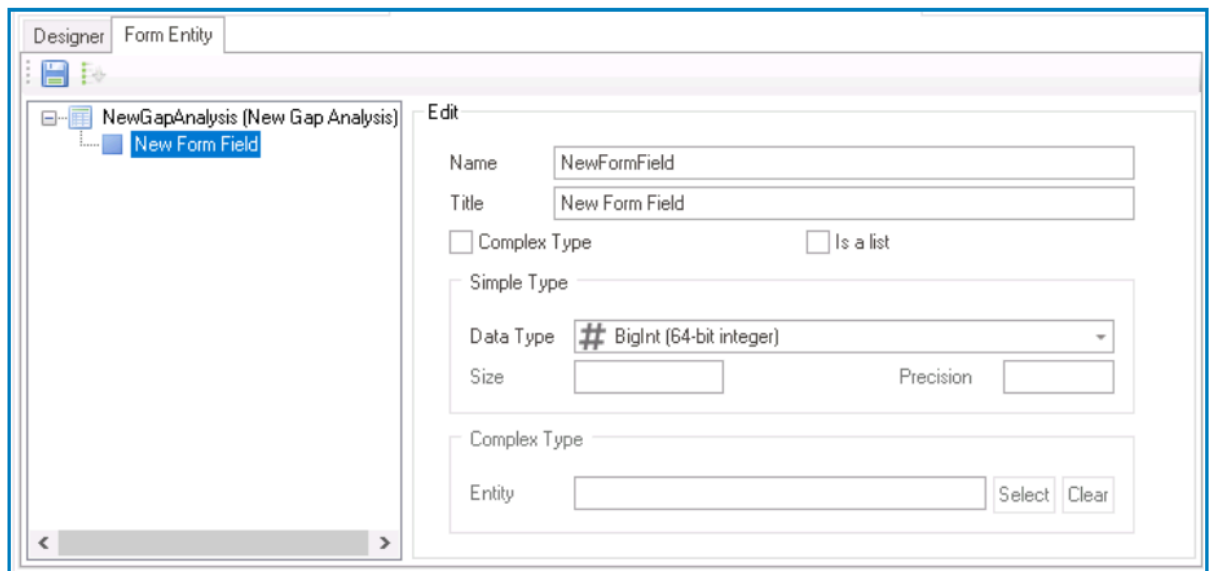
The **Form Designer** window shows the **Form Entity** tab. On the left, a tree control displays "NewGapAnalysis (New Gap Analysis)". On the right, the **Edit** panel contains the following properties:

- Inherited From:** A dropdown menu showing "BPMAPP\_GapAnalysisJD\_New\_Gap".
- Name:** A text field containing "NewGapAnalysis".
- Title:** A text field containing "NewGapAnalysis".
- Relation Key:** A dropdown menu.

- Right-click on the form entity and select **New Field** from the context menu. It will generate a new field below the form entity, which can then be customized and defined to suit the needs of the form.



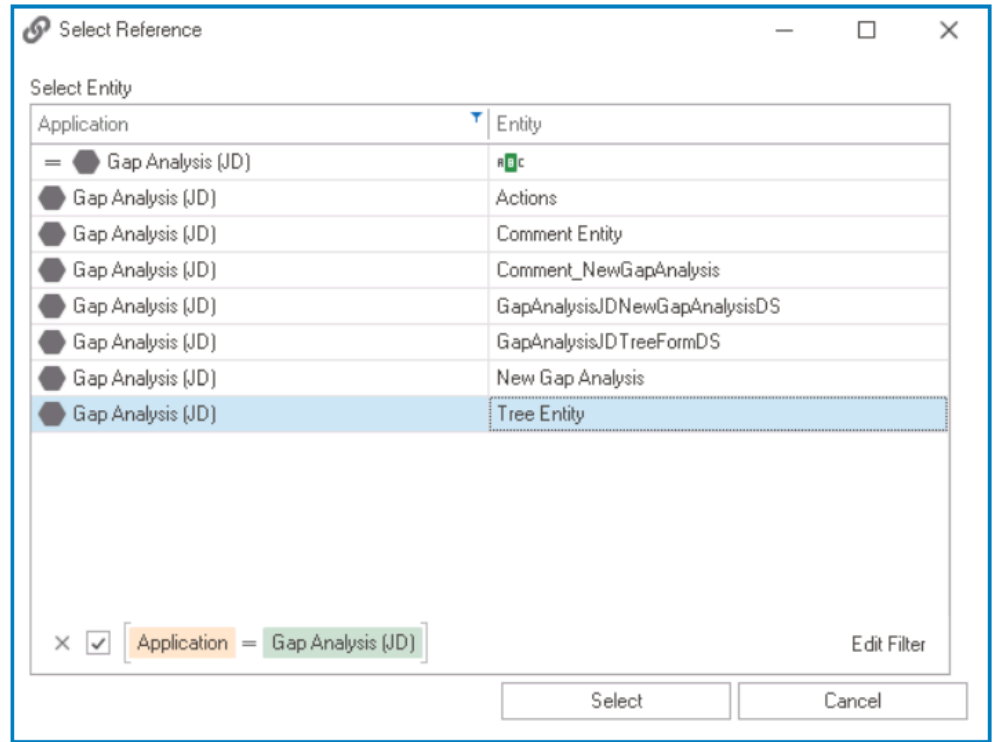
- In the Edit Properties pane, users can define the properties of the newly created attribute.



✿ For tree controls, users must create two local attributes (list and selected).

- **Create Local Attributes**

1. **List:** This attribute is used to bind the tree control to the form.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Select the **Complex Type** checkbox (because the control will reference data from another form).
  - Select the **Is a List** checkbox (because the control will hold multiple instances of the attribute's value).
  - In the **Complex Type** section, select the tree entity from which the attribute will reference data.



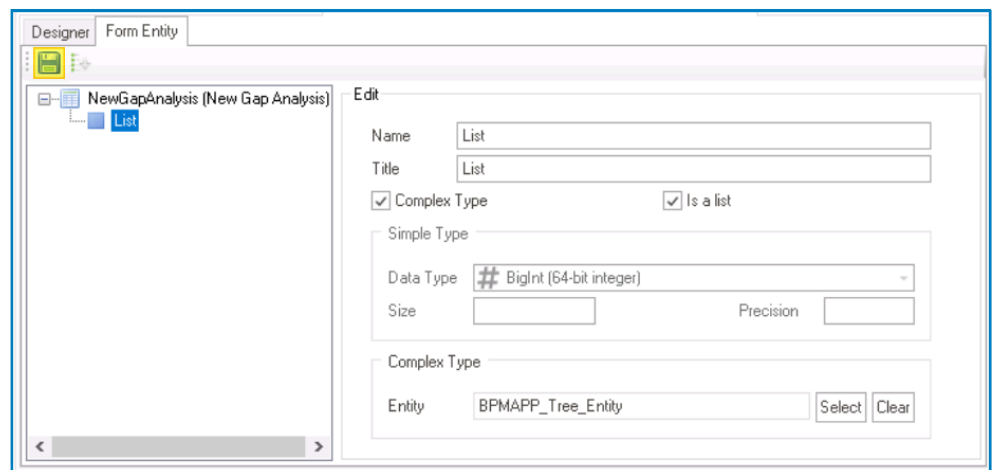
The 'Select Reference' dialog box is shown. It has a title bar with a lock icon and the text 'Select Reference'. Below the title bar is a 'Select Entity' section. This section contains a table with two columns: 'Application' and 'Entity'. The 'Application' column lists 'Gap Analysis (JD)' multiple times. The 'Entity' column lists various entities, with 'Tree Entity' highlighted in blue. At the bottom of the dialog, there is a filter bar showing 'Application = Gap Analysis (JD)' and an 'Edit Filter' button. There are also 'Select' and 'Cancel' buttons at the bottom right.

| Application         | Entity                        |
|---------------------|-------------------------------|
| = Gap Analysis (JD) | #c                            |
| Gap Analysis (JD)   | Actions                       |
| Gap Analysis (JD)   | Comment Entity                |
| Gap Analysis (JD)   | Comment_NewGapAnalysis        |
| Gap Analysis (JD)   | GapAnalysisJDNewGapAnalysisDS |
| Gap Analysis (JD)   | GapAnalysisJDTreeFormDS       |
| Gap Analysis (JD)   | New Gap Analysis              |
| Gap Analysis (JD)   | Tree Entity                   |

Filter: Application = Gap Analysis (JD) Edit Filter

Select Cancel

- Click on the  button to save your changes.



The 'Form Entity' window is shown. It has a title bar with 'Designer' and 'Form Entity'. On the left is a tree view showing 'NewGapAnalysis (New Gap Analysis)' and 'List'. On the right is an 'Edit' section. The 'Edit' section has fields for 'Name' (List) and 'Title' (List). There are checkboxes for 'Complex Type' (checked) and 'Is a list' (checked). Below these are fields for 'Data Type' (BigInt (64-bit integer)), 'Size', and 'Precision'. At the bottom, there is a 'Complex Type' section with an 'Entity' field (BPMAPP\_Tree\_Entity) and 'Select' and 'Clear' buttons.

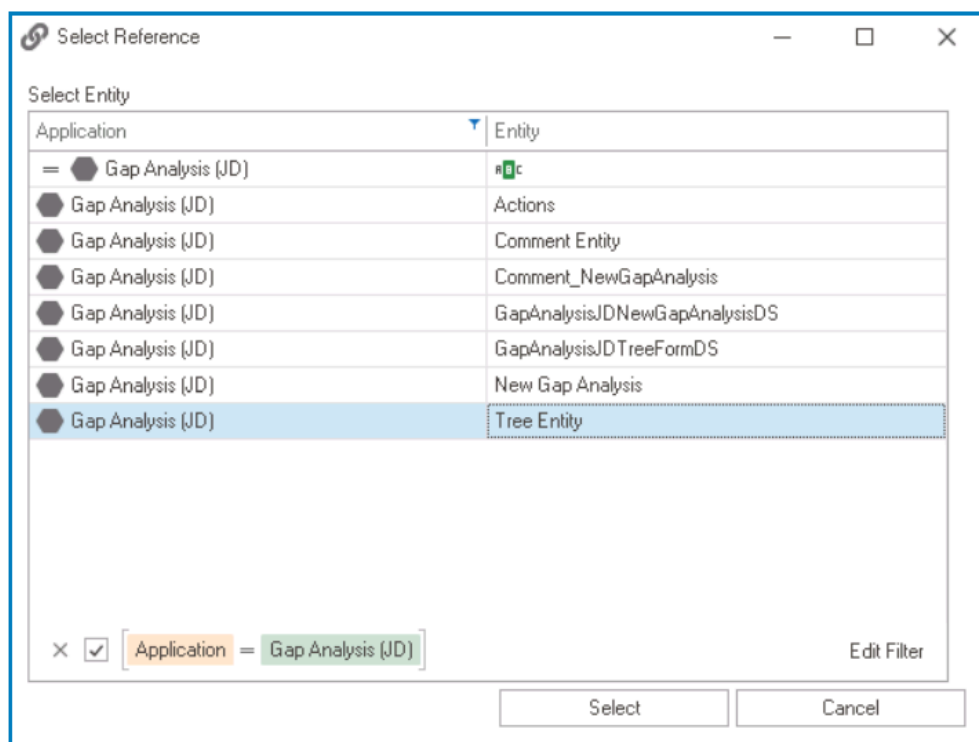
Name: List  
Title: List

☒ Complex Type ☒ Is a list

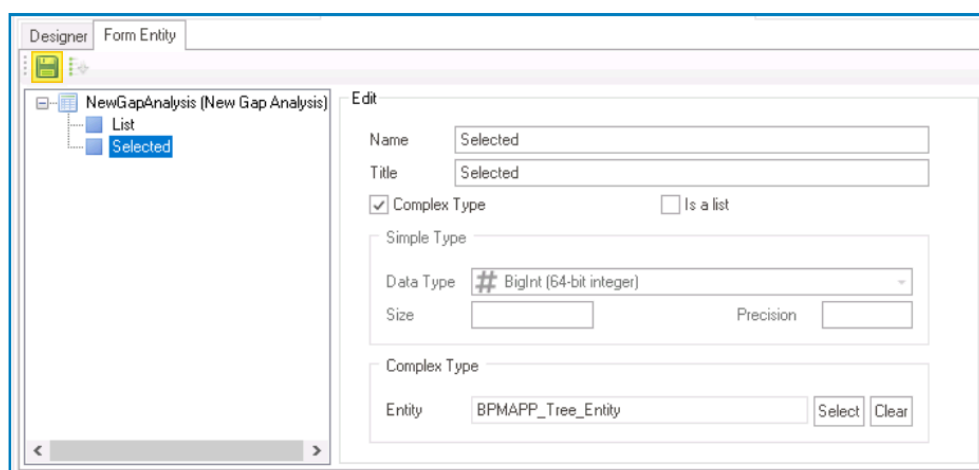
Simple Type  
Data Type: # BigInt (64-bit integer)  
Size: Precision:

Complex Type  
Entity: BPMAPP\_Tree\_Entity Select Clear

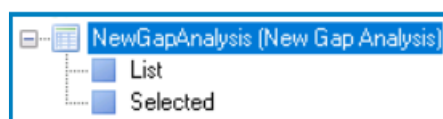
2. **Selected:** This attribute is used to capture and store the currently selected item within the tree.
  - In the **Name** field, provide a name for the attribute.
  - In the **Title** field, either specify a custom title for the attribute or let the system automatically generate it based on the provided name.
  - Select the **Complex Type** checkbox (because the control will reference data from another form).
  - In the **Complex Type** section, select the tree entity from which the attribute will reference data.



- Click on the  button to save your changes.



- Once users have created the necessary local attributes for the tree control, the end-result should be the following:



## 7. Configure Tree Control Binding Properties

- After creating the appropriate attribute, users can return to the Designer tab and define the binding properties for the tree control.




Once the control properties are configured, users must save the form. This will activate

the Quick Forms feature, prompting users to confirm the generation of both the main entity and the form behavior for the main form.

# 12.1.2.1.1.31.1.2. Control Properties – Tree Tab

In the Tree tab of the Control Properties window, users can build and customize the tree control.

 In the General tab of this window, users can configure business rules for the tree control. For more information on this matter, please [click here](#) and navigate to the relevant section.

Control Properties

Apply OK Cancel

Form ControlTree1

Entity and AttributeList

Object typeRay.BPMApp.GapAnalysisJD.Model.BPMAPP\_Tree\_Entity

GeneralTree

Root caption :Tree

Physical form :

\* nested Physical Form

☐ Base info

Foreign key :

Selected item field :

Hierarchy

Binding field :

Foreign fey :

Display field :

Tooltip

Weight

☐ Descending

1. **Root Caption:** This is a text field that allows users to specify the caption or label for the root node of the tree control. It represents the top-level or main node in the hierarchical structure.

- **Root Caption Preview: Web View**

Categories

2. **Physical Form:** This is a single-select dropdown field that retrieves the list of forms linked to the

© 2023 Interfacing Technologies Corporation

Page 754 of 1144



associated entity. It allows users to select the form from which data will be pulled to populate the tree control.

3. **Base Info:** This checkbox determines the structural organization of the tree associated with each node in a process.
  - If this checkbox is selected, it signifies the adoption of a **base information tree** structure, which allows for independent storage and management of data for each node.
    - The base information tree is created and maintained each time the main form is started or circulated within the process. The tree control operates in a manner where the modal form for each node is directly stored or sent, without relying on the saving of the main form.



While hierarchical relationships can still exist within the base information tree, the storage and management of nodes are not directly dependent on parent-child relationships.

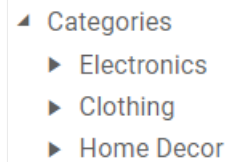
- If the checkbox is left unselected, the tree structure follows a **parent-child relationship** model.
  - In this structure, a tree contains nodes that are stored and managed based on their relationship as parent and child. Each time the parent form (parent node) begins or is circulated in the process, the child form (child node) is stored and managed accordingly.



This relationship indicates that the child form is dependent on the parent form in terms of storage and submission.

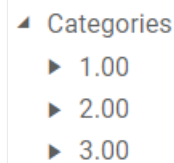
4. **Foreign Key:** This is a single-select dropdown field that retrieves the list of reference attributes available in the associated entity. It allows users to select the attribute that can serve as a foreign key, establishing the relationship between the tree control and the associated entity.
5. **Selected Item Field:** This is a single-select dropdown field that retrieves the list of relevant local attributes associated with the main form. It allows users to select the attribute that will capture and store the currently selected item within the tree control.
6. **Hierarchy:** In this section, users can configure the hierarchy settings for the tree control, specifying the attributes that define the hierarchical structure of the data.
  - **Binding Field:** This is a single-select dropdown field that retrieves the list of relevant local attributes associated with the main form. It allows users to select the attribute that will bind the tree control to the form, serving as a reference to the list of entities and enabling the display of hierarchical data within the tree.
  - **Foreign Key:** This is a single-select dropdown field that retrieves the list of relevant reference attributes available in the associated entity. It allows users to select the attribute that can serve as a foreign key, establishing relationships between nodes in the tree control.
  - **Display Field:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute whose value will be displayed as text on individual nodes in the tree control.
    - **Display Field Preview: Web View**
      - For example, if the selected attribute is **Item Title**, the control will update

the display of each node to showcase its corresponding title.



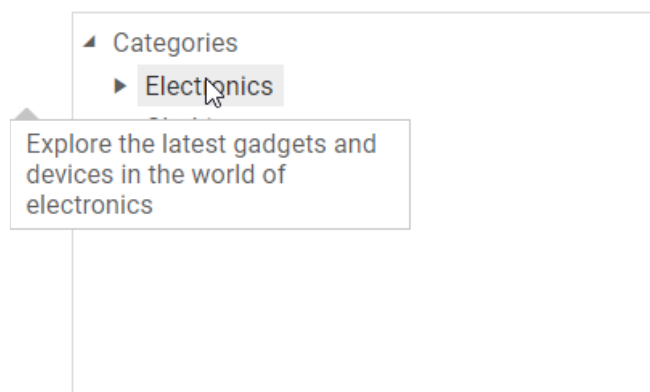
```
graph TD; Categories[Categories] --> Electronics[Electronics]; Categories --> Clothing[Clothing]; Categories --> HomeDecor[Home Decor];
```

- Similarly, if the selected attribute is **Order**, the control will update the display of each node to showcase its corresponding order value.



```
graph TD; Categories[Categories] --> 1.00[1.00]; Categories --> 2.00[2.00]; Categories --> 3.00[3.00];
```

- **Tooltip:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute that will provide additional contextual information when users hover over or interact with specific nodes in the tree control.
  - **Tooltip Preview: Web View**



- **Weight:** This is a single-select dropdown field that retrieves the list of attributes available in the associated entity. It allows users to select the attribute representing the weight or priority of the nodes in the tree control.
  - This attribute is used for sorting or organizing nodes based on their importance or logical order within the tree hierarchy.
- **Descending:** This checkbox determines the sorting order of nodes within the tree control.

- If this checkbox is selected, the nodes will be sorted in descending order based on the selected weight.
- If this checkbox is left unselected, the nodes will be sorted in ascending order based on the selected weight.

### Example: Nested Attributes Selection

The screenshot shows the 'Control Properties' dialog box for a 'Tree' control. The 'General' tab is selected. The 'Form Control' is 'Tree1', 'Entity and Attribute' is 'List', and 'Object type' is 'Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_Tree\_Entity'. The 'Physical form' is set to 'TreeForm' with a note '\* nested Physical Form'. The 'Base info' checkbox is checked. The 'Hierarchy' section contains several fields: 'Foreign key' is 'BPMAPP\_Tree\_EntityParent', 'Selected item field' is 'Selected', 'Binding field' is 'BPMAPP\_Tree\_EntitiesParent', 'Foreign fey' is 'BPMAPP\_Tree\_EntityParent', 'Display field' is 'ItemTitle', 'Tooltip' is 'Tooltip', and 'Weight' is 'Order'. The 'Descending' checkbox is unchecked.

Control Properties

Apply OK Cancel

Form Control Tree1

Entity and Attribute List

Object type Ray.BPMApp.GapAnalysisJD.Model.BPMAPP\_Tree\_Entity

General Tree

Root caption : Categories

Physical form : TreeForm

\* nested Physical Form

☒ Base info

Foreign key : BPMAPP\_Tree\_EntityParent

Selected item field : Selected

Hierarchy

Binding field : BPMAPP\_Tree\_EntitiesParent

Foreign fey : BPMAPP\_Tree\_EntityParent

Display field : ItemTitle

Tooltip Tooltip

Weight Order ☐ Descending

# 12.1.2.1.31.2. Tree – Control Governance

In the Tree Behavior tab of the Form / Behavior / Control window, users can configure properties that govern the functionality of tree controls.

Form [AuditPlanList] / Behaviour [Audit Plan List] / Control [Tree1]

Apply

OK

Cancel

Control Name **Tree1**

Tree Behaviour

Validation

☐ Enable Edit

Edit Form

[Empty]

☐ Enable New

New Form

[Empty]

☐ Enable View

View Form

[Empty]

☐ Enable Delete

Delete Form

[Empty]

☐ Inline New

Import Form

[Empty]

☐ Inline Edit

Inline Edit Form

[Empty]

☒ Enable import from Excel

Import Form

[Empty]

☒ Enable export from Excel

☐ Enable Select (only in selector forms)

Please note that the **Inline New** and **Inline Edit** fields in this window are disabled for the tree control.

1. **Enable Edit:** This is a checkbox which, if selected, allows users to enable the **Edit** function for the tree.
- If this checkbox is selected, it will enable the **Edit Form** field.

**Edit Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'edit' functionality.

To define a new form behavior, select the button.

It will display the [Form Behavior Properties](#) window.




To modify an existing form behavior, select the button.

To remove a form behavior, select the button.

! To set up the 'Edit Form', users only need to provide a name and a title.

\* With **Edit** enabled, users can edit items or nodes in the tree using a modal window linked to the tree.




2. **Enable New:** This is a checkbox which, if selected, allows users to enable the **Add** function for the tree.

- If this checkbox is selected, it will enable the **New Form** field.
  - **New Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'new' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'New Form', users only need to provide a name, a title, and select the **Starter** checkbox.

\* With **Add** enabled, users can add items or nodes to the tree using a modal window linked to the tree.




3. **Enable View:** This is a checkbox which, if selected, allows users to enable the **Show** function for the tree.

- If this checkbox is selected, it will enable the **View Form** field.
  - **View Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'view' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'View Form', users only need to provide a name and a title.




\* With **View** enabled, users can view items or nodes in the tree using a modal window linked to the tree.

4. **Enable Delete:** This is a checkbox which, if selected, allows users to enable nodes to be deleted in the tree.

- If this checkbox is selected, it will enable the **Delete Form** field.
  - **Delete Form:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'delete' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! Unless users need to add custom functionality when deleting, it is not necessary for them to set up the 'Delete Form'. If users do need to add custom functionality when deleting, however, they will need to create the 'Delete Form' and then create a [post-form action](#) and attach it to the selected form behavior. It's important to note that the pre-form actions of the delete form behavior will not be called.

5. **Enable Import from Excel:** This is a checkbox which, if selected, allows users to enable both the **Excel Template** and **Import from Excel** functions for the grid.

- If this checkbox is selected, it will enable the **Import Form** field.
  - **Import Form:** This is a single-select dropdown field that retrieves the list of starter form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior linked to the control's 'inline add' functionality.
    - To define a new form behavior, select the  button.
      - It will display the [Form Behavior Properties](#) window.
    - To modify an existing form behavior, select the  button.
    - To remove a form behavior, select the  button.

! To set up the 'Import Form', users only need to provide a name, a title, and select the **Starter** checkbox.

\* When this option is enabled, users can download an Excel template that contains the appropriate column headers for the control, and then populate the template with their data. Users can then upload the completed template into the control to import the data.

6. **Enable Export from Excel:** This is a checkbox which, if selected, allows users to enable the **Export to Excel** function for the grid.

\* When this option is enabled, users can export the data from the control into an Excel file.

7. **Enable Select:** This is a checkbox which, if selected, allows users to enable the selection of tree data from another form using a dropdown control.

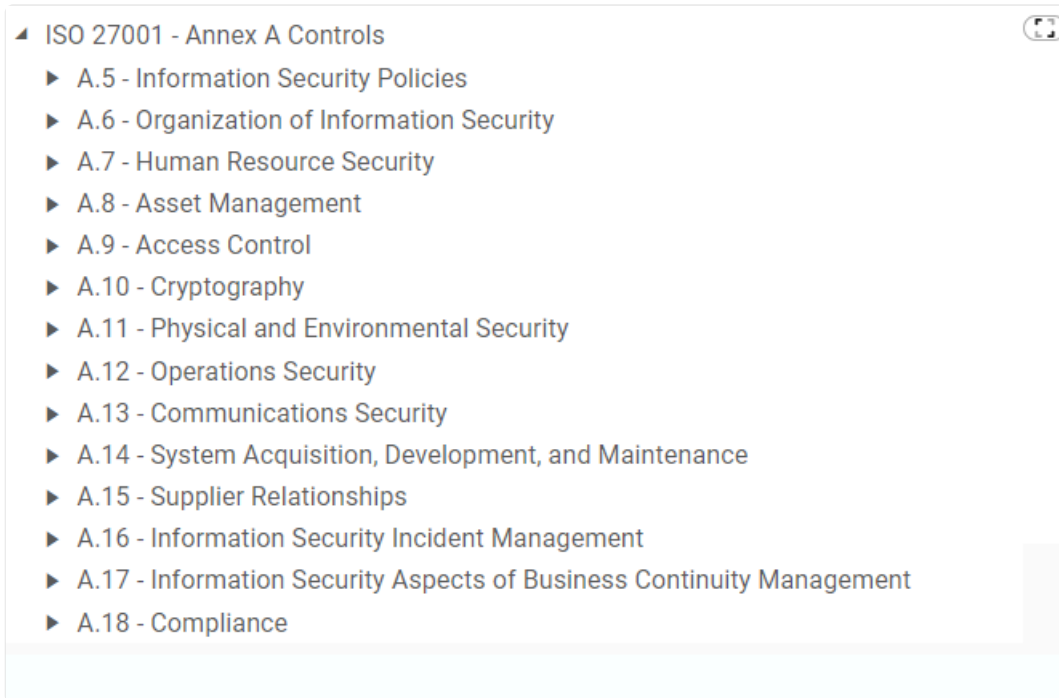
✿ When this option is enabled, users can interact with a tree displayed in a separate form and select items by double-clicking on them. The selected value will then be returned and populated in the associated dropdown control.

8. **Apply:** This button allows users to save their changes.
9. **OK:** This button allows users to save their changes and close the window.
10. **Cancel:** This button allows users to cancel their changes and close the window.

✿ In the Validation tab of this window, users can configure the required field validation properties for the tree control. For more information on this matter, please [click here](#) and navigate to the relevant section.

## 12.1.2.1.1.31.3. Tree – Web Display


For those unfamiliar with using the tree control in the **DBP** web application, the following instructions will guide you through the process:



### 1. Expanding and Collapsing Individual Nodes

- To expand a node and reveal its child nodes, click on the arrow icon located to the left of the node. To collapse a node and hide its child nodes, click on the arrow icon again.

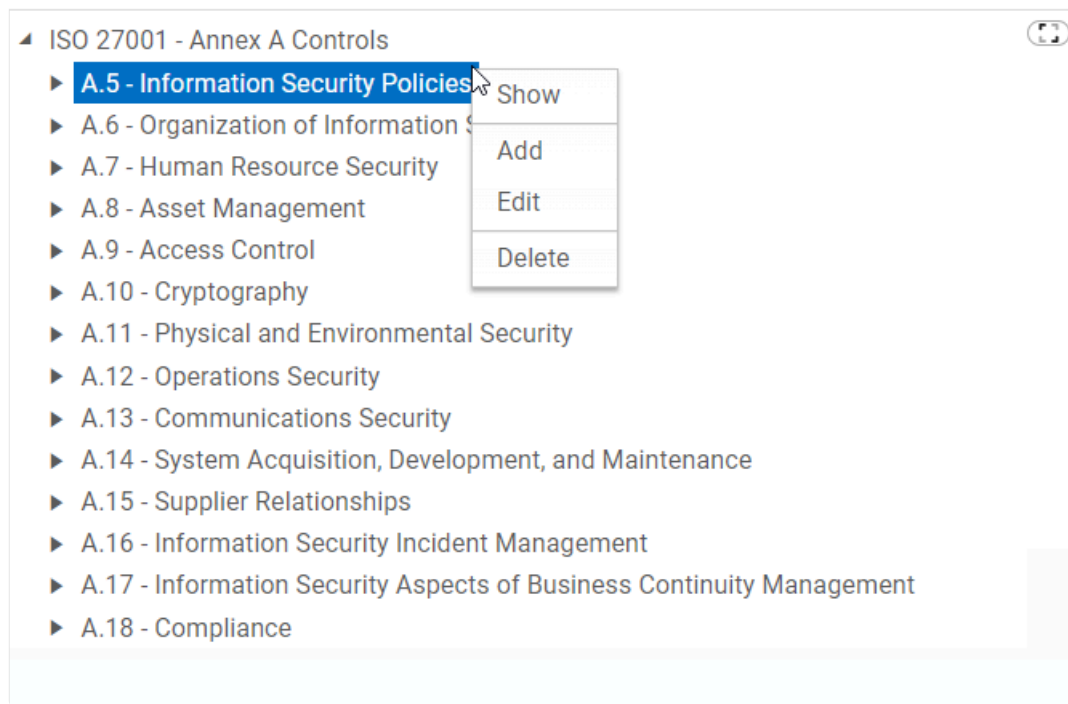
### 2. Expand/Collapse All Nodes

- To expand or collapse all nodes within the tree, click on the  button located to the right of the control.

### 3. Access Tree Context Menu

- To access the context menu, right-click on a specific node within the tree control.





- **Show:** This option allows users to view the information associated with the selected node.
  - If selected, it will open a pop-up window with the 'View Form'.
  - In viewing mode, users typically cannot make any modifications.

! Users can configure the 'View Form' in the Control Governance tab. The 'View Form' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the tree attributes.

- **Add:** This option allows users to add a new node as a child of the selected node.
  - If selected, it will open a pop-up window with the 'New Form'.
    - In this form, users can provide the required information for the node.
  - At the bottom of the form, there is the following button:

Save and Return

- **Save and Return:** This button allows users to save the node and return to the main form.
- Once they have saved their work and returned to the main form, users will find the new node they have created now added to the tree.

! Users can configure the 'New Form' in the Control Governance tab. The 'New Form' is a form behavior that determines the appearance and functionality of the modal form in the web

application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the tree attributes.

- **Edit:** This option allows users to edit the information of the selected node.
  - If selected, it will open a pop-up window with the 'Edit Form'.
    - This form is typically a duplicate of the 'New Form', with the exception being that the fields are all populated with previously inputted information.
  - After saving their changes and returning to the form, users will find the node in the tree updated with the latest modifications.



Users can configure the 'Edit Form' in the [Control Governance](#) tab. The 'Edit Form' is a form behavior that determines the appearance and functionality of the modal form in the web application. Please note, however, that the actual contents and design of the form are pulled from the custom form that contains the tree attributes.

- **Delete:** This option allows users to remove the selected node from the tree control, along with its associated child nodes.
  - If this button is selected, users will be alerted of the action with the following confirmation message:

**Confirm**

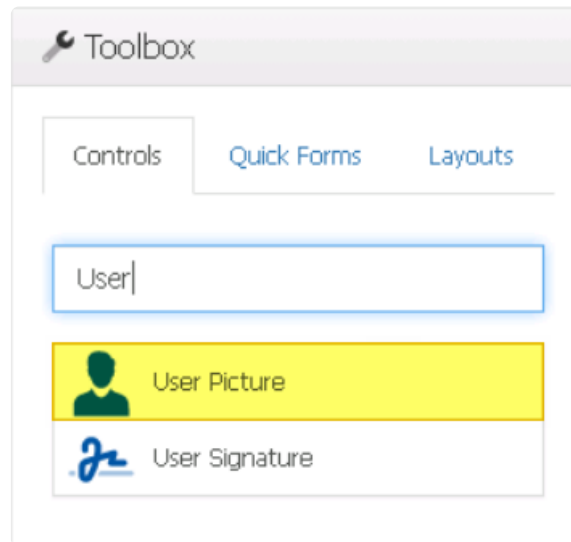
Are you sure you want to delete?

Submit Cancel

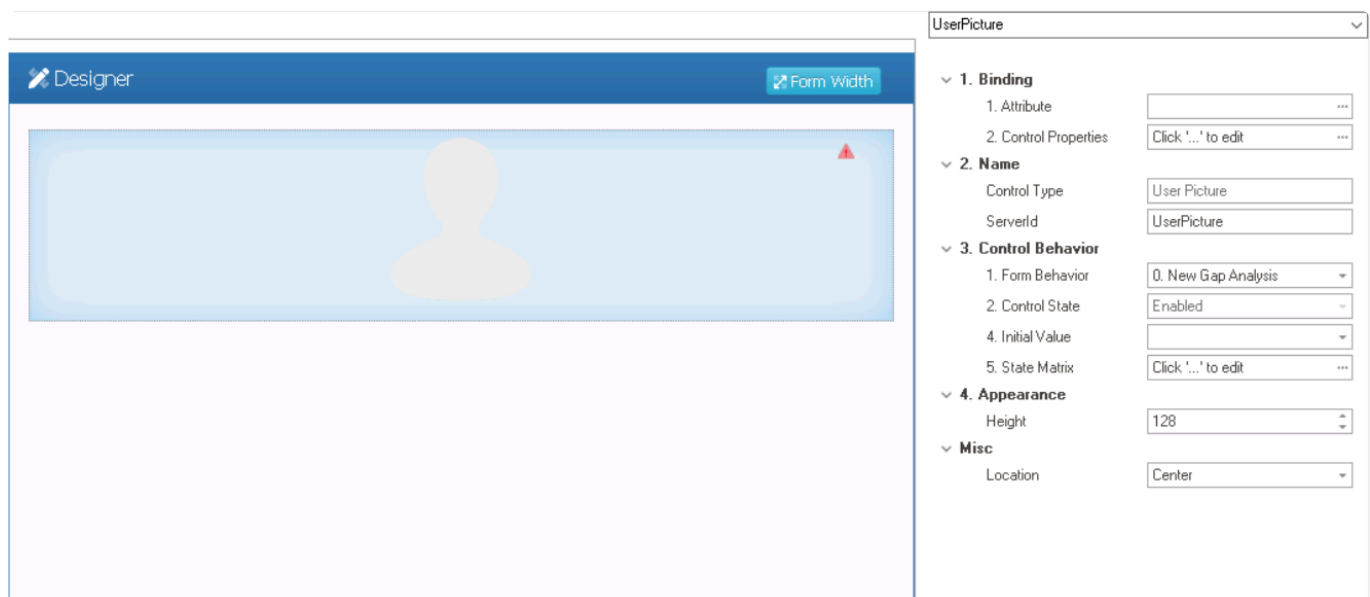
- Click on either the **Submit** button to confirm the deletion or the **Cancel** button to cancel it.

## 12.1.2.1.1.32. User Picture

With the user picture control, user pictures can be dynamically displayed on forms.




When a user picture control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



 To access the **Control Behavior** category, users must first create the form behavior.

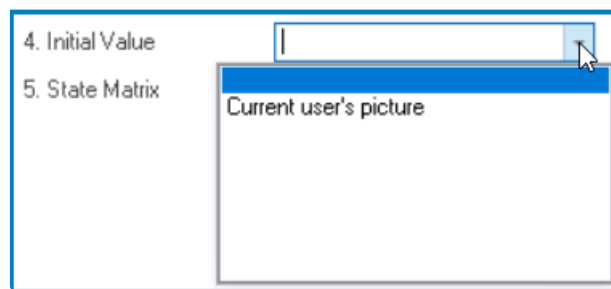
1. **Binding:** In this category, users can link a control to its corresponding attribute.
  - For more information on the fields in this category, [click here](#).

 To enable user picture controls to retrieve and display user pictures in the system, a reference attribute must be created linking the control to all user accounts in the

database. For more information on creating this type of attribute, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **User Picture Initial Value**

An option for the user picture control is to set the initial value to the current user's picture. This means that when the form is opened, the control will automatically display the picture for the user associated with the current session.



- For more information on these properties and how to configure them, [click here](#).

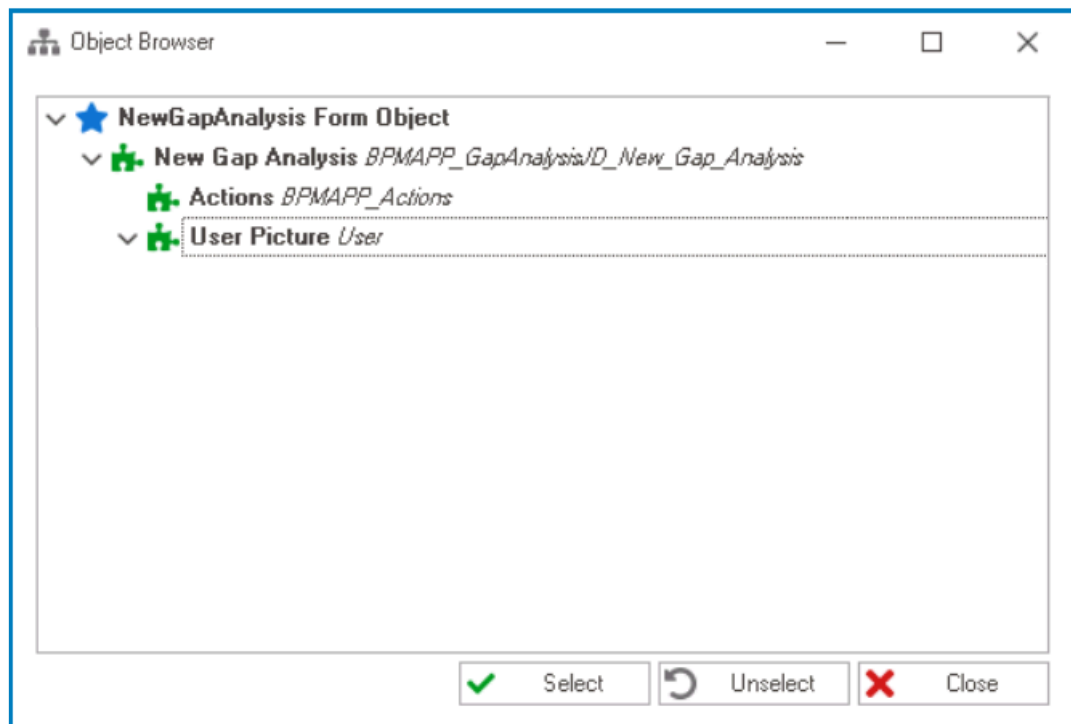
! Please note that the linked section contains information on control governance properties that are not applicable to user picture controls and can be disregarded.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 128. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Misc:** In this category, users can define additional properties for the control.
  - **Location:** This is a single-select dropdown field that allows users to choose the location of the control.
    - Location Options: Right, Center, or Left.
      - Right: This option places the control on the right-hand side of the screen.
      - Center: This option centers the control in the middle of the screen.
      - Left: This option places the control on the left-hand side of the screen.

## 12.1.2.1.1.32.1. User Picture – Binding

In the **Binding** category, users can link a user picture control to its corresponding attribute and define its binding properties.

1. **Attribute:** This field allows users to select the attribute associated to the user picture control.
  - To access the list of relevant attributes, click on the ellipses ( **...** ) inside the field.



\* In the Object Browser window, users can access a list of the associated (form) entity's attributes—specifically, the attributes relevant to the user picture control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

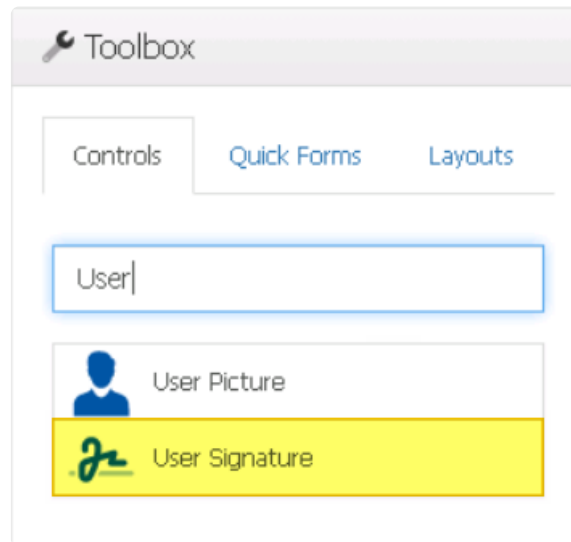
! The user picture control must be linked to the corresponding user reference attribute. For more information on this type of attribute, [click here](#).

2. **Control Properties:** This field allows users to configure the control properties.

- For more information on these properties and how to configure them, [click here](#) and navigate to the relevant section.

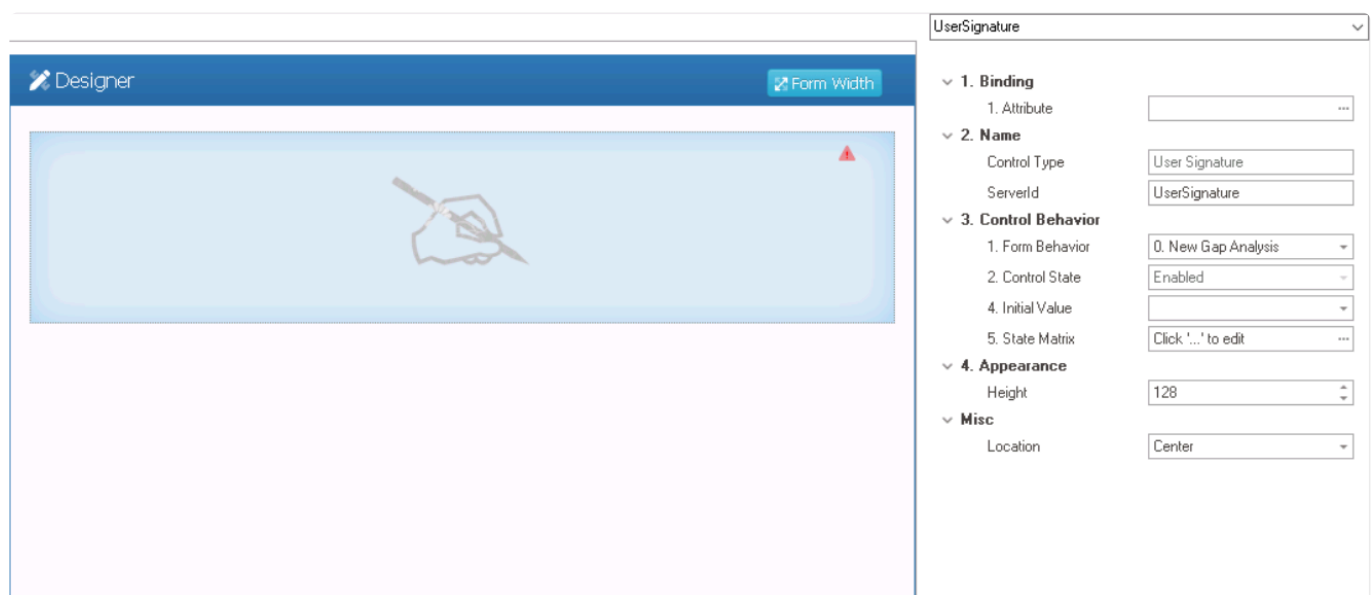
## 12.1.2.1.1.33. User Signature

With the user signature control, user signatures can be dynamically displayed on forms.



\* User signatures are set up in the [Users](#) tab of the **Organization Management (Admin)** application.

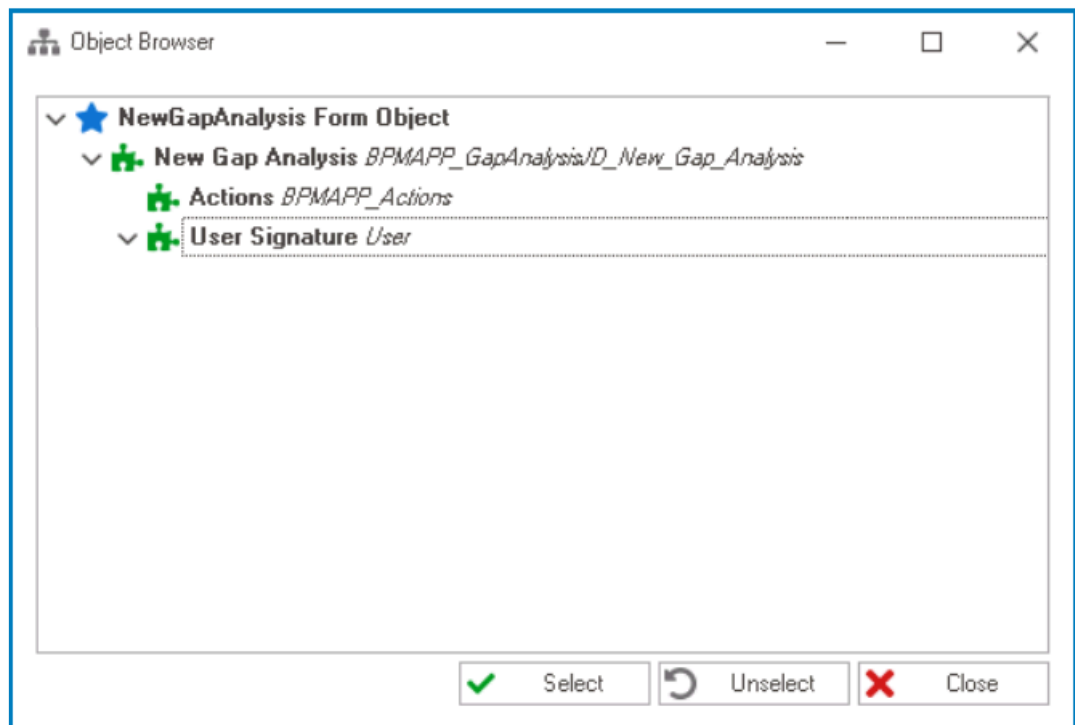
When a user signature control is selected in the design environment, its associated properties are displayed in the Control Properties panel.



\* To access the **Control Behavior** category, users must first create the form behavior.

1. **Binding:** In this category, users can link a control to its corresponding attribute.
  - **Attribute:** This field allows users to select the attribute associated to the control.

- To access the list of relevant attributes, click on the ellipses (...) inside the field.



✿ In the Object Browser window, users can access a list of the associated entity's attributes—specifically, the attributes relevant to the control. To select an attribute, either click on the **Select** button or double-click on the entry from the list.

- Once an attribute has been selected, it will populate the **Attribute** field.

1. Attribute

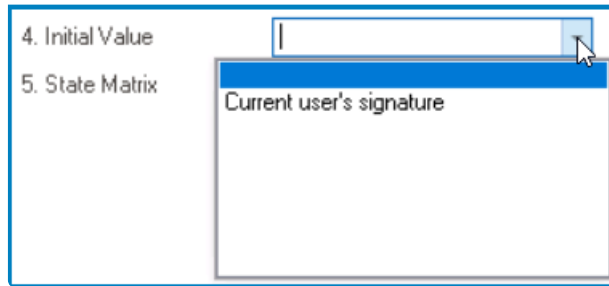
DomainObject.UserUserSig... \*\*\*

! To enable user signature controls to retrieve and display user signatures in the system, a reference attribute must be created linking the control to all user accounts in the database. For more information on creating this type of attribute, [click here](#).

2. **Name:** In this category, users can access basic properties required for the control's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the control in the code.
3. **Control Behavior:** In this category, users can configure the behavior of the control and define how it interacts with the form or application.
  - **User Signature Initial Value**

An option for the user signature control is to set the initial value to the current user's signature. This means that when the form is opened, the control will automatically display the signature for the user associated with the current session.





- For more information on these properties and how to configure them, [click here](#).

! Please note that the linked section contains information on control governance properties that are not applicable to user signature controls and can be disregarded.

4. **Appearance:** In this category, users can customize the visual appearance of the control.
  - **Height:** This is a numeric updown field that allows users to set the height of the control.
    - By default, the height is set to 128. To adjust this size, users can enter numbers into the field either manually or by clicking the increment/decrement arrows.
5. **Misc:** In this category, users can define additional properties for the control.
  - **Location:** This is a single-select dropdown field that allows users to choose the location of the control.
    - Location Options: Right, Center, or Left.
      - Right: This option places the control on the right-hand side of the screen.
      - Center: This option centers the control in the middle of the screen.
      - Left: This option places the control on the left-hand side of the screen.

#### User Signature Control Preview: Web View

- If the initial value is set to the current user's signature, the user signature control will function in the following manner:
- If the user signature is instead set to be dependent on another control, such as a dropdown, its function will vary based on the user's selection.

## 12.1.2.1.2. Quick Forms

---

In the Quick Forms tab of the toolbox, users are provided access to a set of controls for which the **Quick Forms** feature can be applied.



Each control has its own set of properties. To view the properties for a specific control, please refer to the corresponding section listed below.

1. [Checkbox](#)
2. [Date Time Picker](#)
3. [File Attachment](#)
4. [Grid](#)
5. [Multi-Select Dropdown](#)
6. [Multi-Line Text Box](#)
7. [Numeric Text Box](#)
8. [Password Box](#)
9. [Radio Button List](#)
10. [Rich Text Box](#)
11. [Single-Select Dropdown](#)
12. [Text Box](#)
13. [Time Duration](#)



To learn more about the **Quick Forms** feature and how it can be generally applied, [click here](#).

## 12.1.2.1.2.1. Quick Forms Feature

**Quick Forms** is a powerful feature that streamlines the process of creating and designing forms by automating the generation of entities, attributes, and form behaviors.

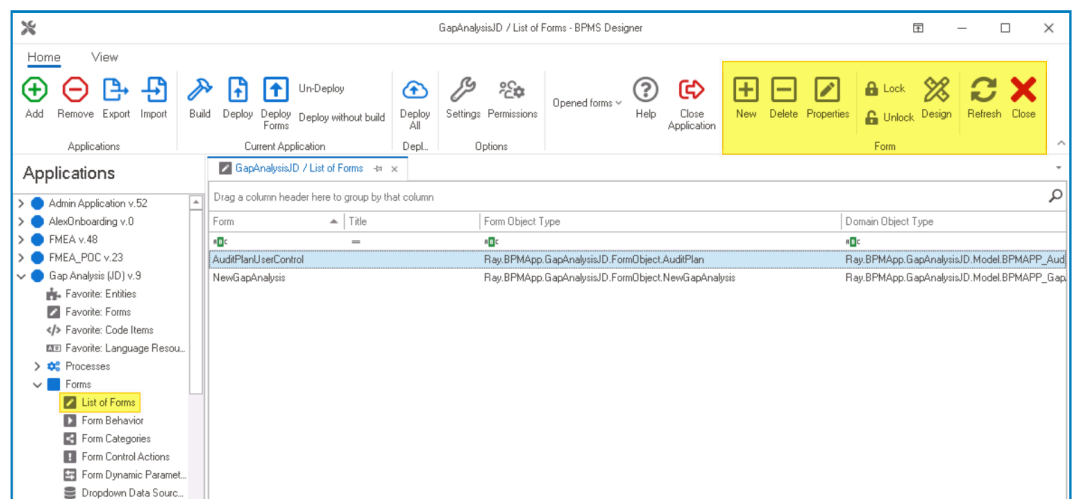
✿ In the [Quick Forms](#) tab of the toolbox, users are provided access to the controls that can be used in conjunction with the **Quick Forms** feature to streamline the form development process.

Typically, to build the application, users must take the following steps:

1. [Create Entity](#)
2. [Add Attributes](#)
3. [Create Form](#)
4. [Design Form](#)
5. [Create Form Behavior](#)

With **Quick Forms**, however, building an application is far less complex and time-consuming. Instead of the five-step process required by the traditional approach, **Quick Forms** simplifies the process into just two easy steps:

1. [Create Form](#)
  - **Access Form Properties**
    - From the [Application Tree](#), navigate to **Forms > List of Forms** (double-click to open). It will display the [List of Forms](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** option. It will display the [Form Properties](#) window.



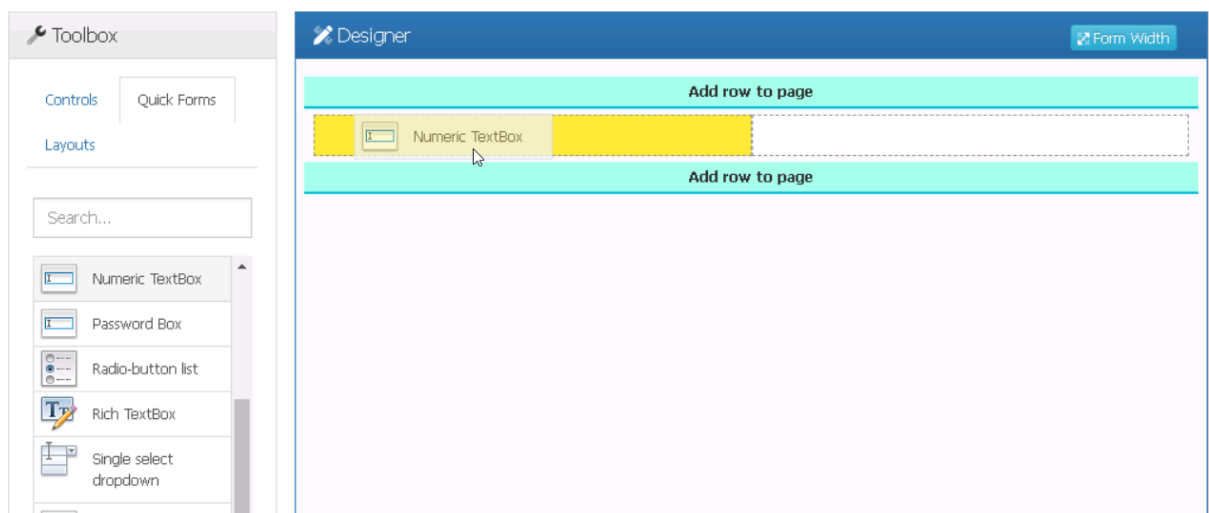
- In the Form Properties window, users can create a new form for the application.

- In the **Name** field, provide a name for the form.
- In the **Form Entity** field, either specify a custom name for the form entity or let the system automatically generate it based on the provided form name.
- Leave the **Entity** field empty.
- Click **OK** and confirm the prompt to start designing the form.

✿ Once the form is saved in the Form Designer, the entity will automatically be generated with the same name as the form.

## 2. Design Form

- In the Form Designer, drag the required controls from the toolbox and drop them onto the design environment.



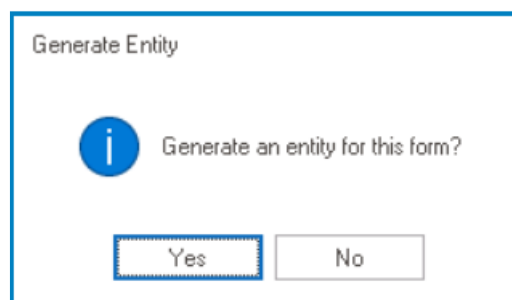
- Provide a caption for each control. These captions will be used to create the names for the corresponding attributes.

! Certain controls, like dropdowns, require extra clarification to determine what information they display. To learn how to configure these controls in accordance with the **Quick Forms** feature, please refer to the [list of controls](#) provided for guidance.

- **Example: Dropdown Control Configuration**

To enable a dropdown to display the relevant information, users must specify the corresponding dropdown list in the Server ID. This can be done by entering a Server ID consisting of the name of the attribute followed by the name of the dropdown. For example, "NameUser" would indicate that "Name" is the attribute and "User" is the reference to the list of users.

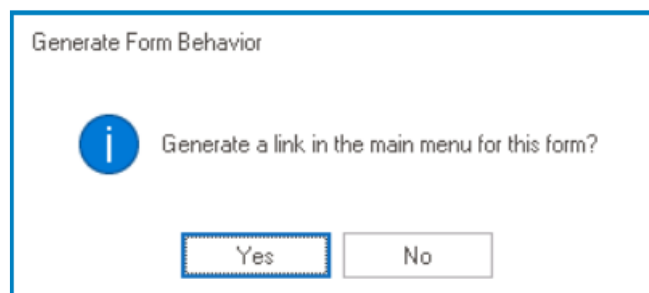
- Once the controls have been added and labeled, and the user saves the form, the system will present two inquiries.
  1. **Generate an entity for this form?**



- To generate the entity and its attributes automatically, users must select **Yes** in response to this prompt.

\* To generate entities, the system uses the name of the associated form in adherence with the [Application Name][Form Name] format.


2. **Generate a link in the main menu for this form?**



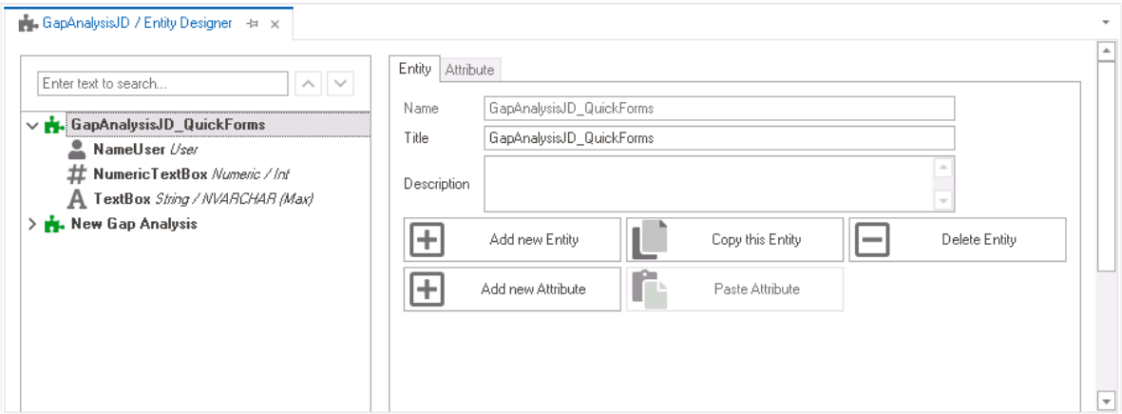
- To generate the form behavior automatically, users must select **Yes** in response

to this prompt.

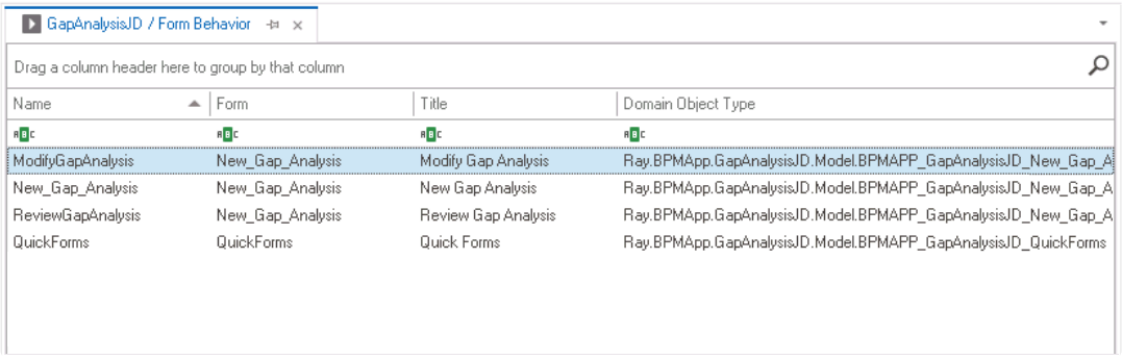
- By default, the form behavior will then be set up as a starter, and the user will be designated as an admin.

 To generate form behaviors, the system uses the name of the associated form (without any additional modifiers).

- Once the user has confirmed their selections for both prompts, the form will close.
  - To access the generated entity and attributes, navigate to the [Entity Designer](#) section of the **Entities** module.



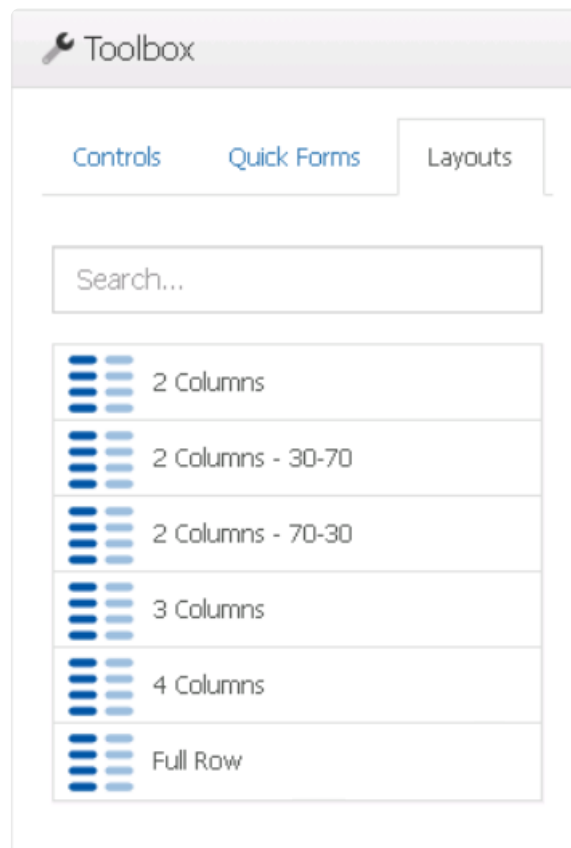
- To access the generated form behavior, navigate to the [Form Behavior](#) section of the **Forms** module.



| Drag a column header here to group by that column |                  |                     |  |
|---|------------------|---------------------|--|
| Name  | Form             | Title               | Domain Object Type   |
| ModifyGapAnalysis                                 | New_Gap_Analysis | Modify Gap Analysis | Ray.BPMApp.GapAnalysisJD.Model.BPMAPP_GapAnalysisJD_New_Gap_A  |
| New_Gap_Analysis                                  | New_Gap_Analysis | New Gap Analysis    | Ray.BPMApp.GapAnalysisJD.Model.BPMAPP_GapAnalysisJD_New_Gap_A  |
| ReviewGapAnalysis                                 | New_Gap_Analysis | Review Gap Analysis | Ray.BPMApp.GapAnalysisJD.Model.BPMAPP_GapAnalysisJD_New_Gap_A  |
| QuickForms  | QuickForms       | Quick Forms         | Ray.BPMApp.GapAnalysisJD.Model.BPMAPP_GapAnalysisJD_QuickForms |

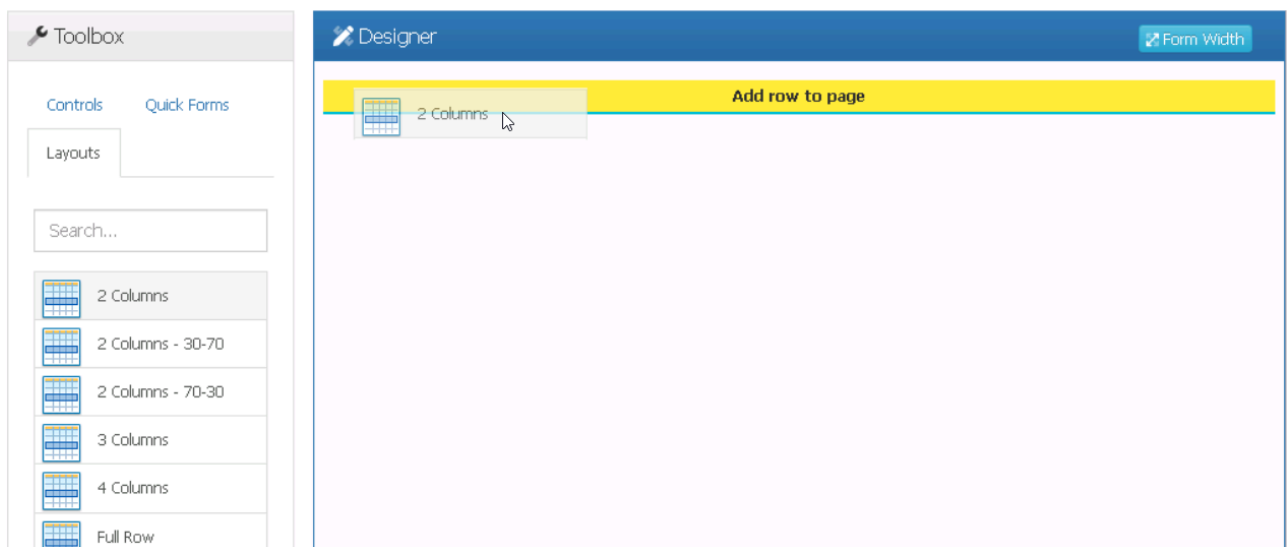
## 12.1.2.1.3. Layouts

In the Layouts tab of the toolbox, users are provided access to a variety of layouts that can be used for designing a form.



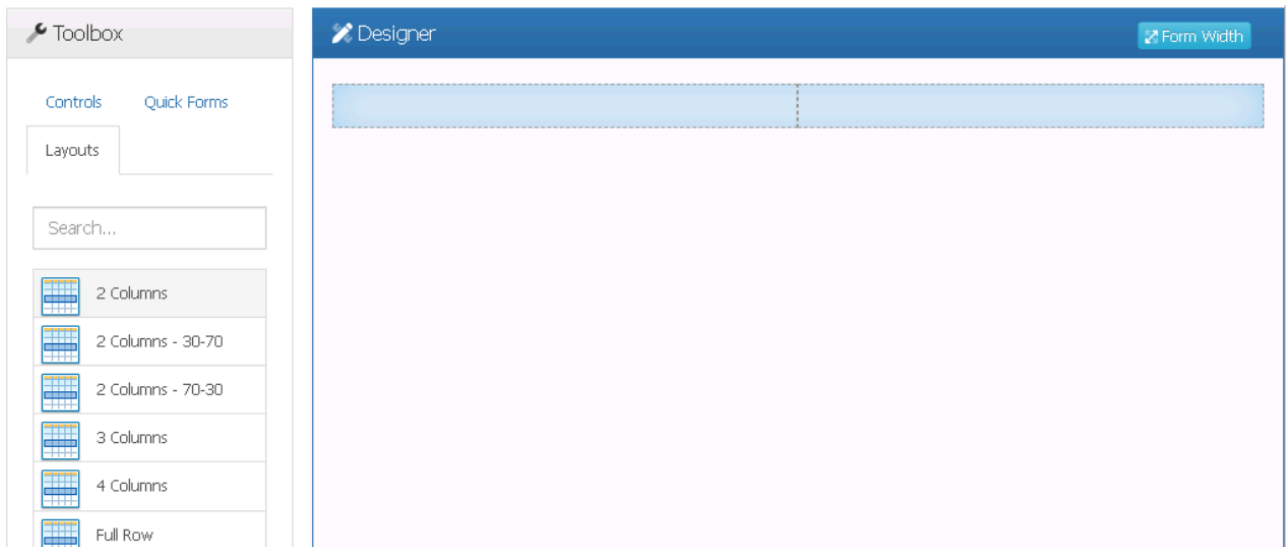
To add layouts to the design environment, take the following steps:

1. From the toolbox, select a layout style and drag it towards a viable drop location in the design environment.



- ✿ As users drag an object towards the design environment, the viable drop locations will be indicated by the **Add Row to Page** fields.

2. Drop the layout style onto the viable area.



- ✿ Make sure the **Add Row to Page** field is highlighted in yellow before dropping the object. Otherwise, the placement operation will not succeed.

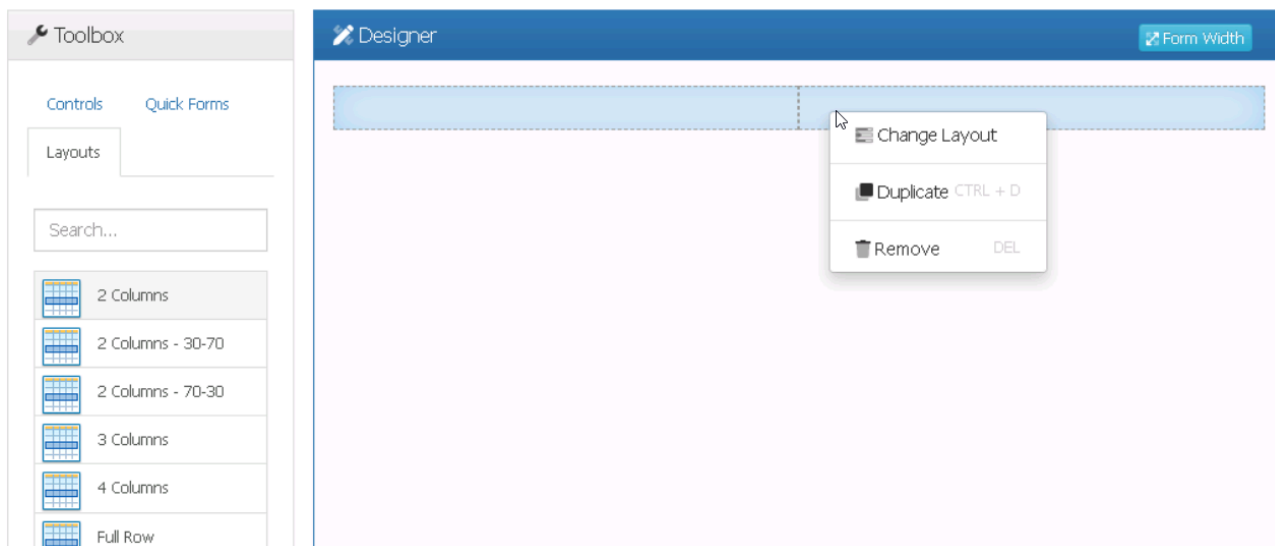
3. In this manner, users can design the layout of the form, adding as many layout styles as needed.

- ✿ When adding an object to an existing column in the design environment, the viable drop locations will be highlighted in yellow. When adding an object in between existing layouts, the viable drop locations will be indicated by the **Insert Here** fields.

Once a layout has been added to the design environment, users can perform the following actions:

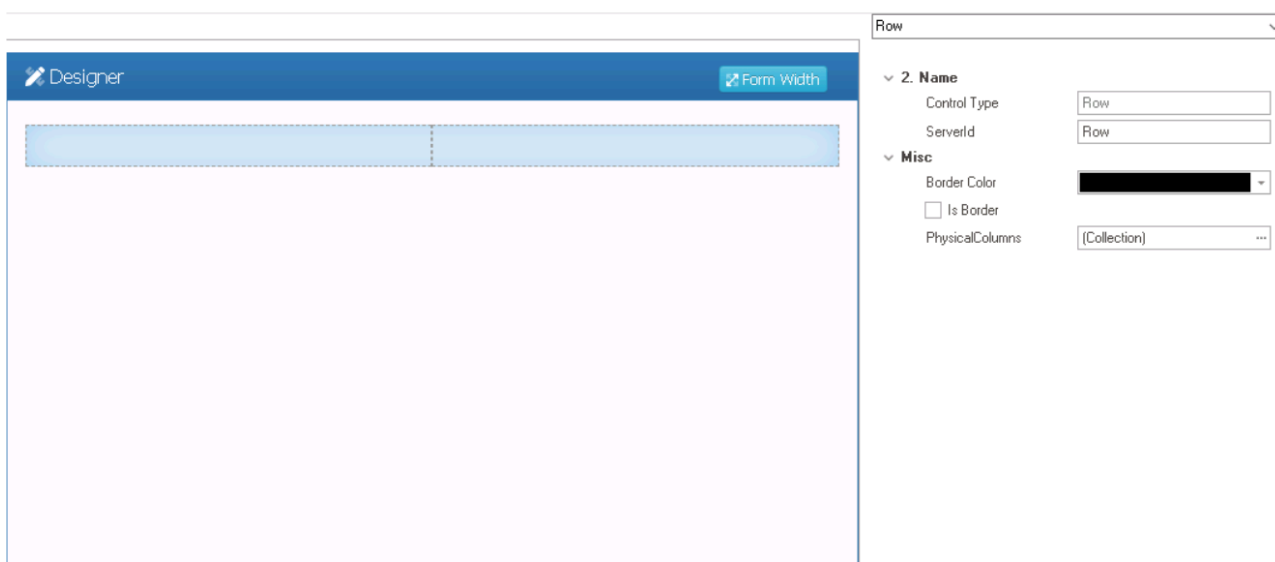
1. **Change, Duplicate, or Remove Layout**





- For more information on the options in this context menu, [click here](#).

## 2. Configure Properties



- For more information on how to configure the layout properties, [click here](#).

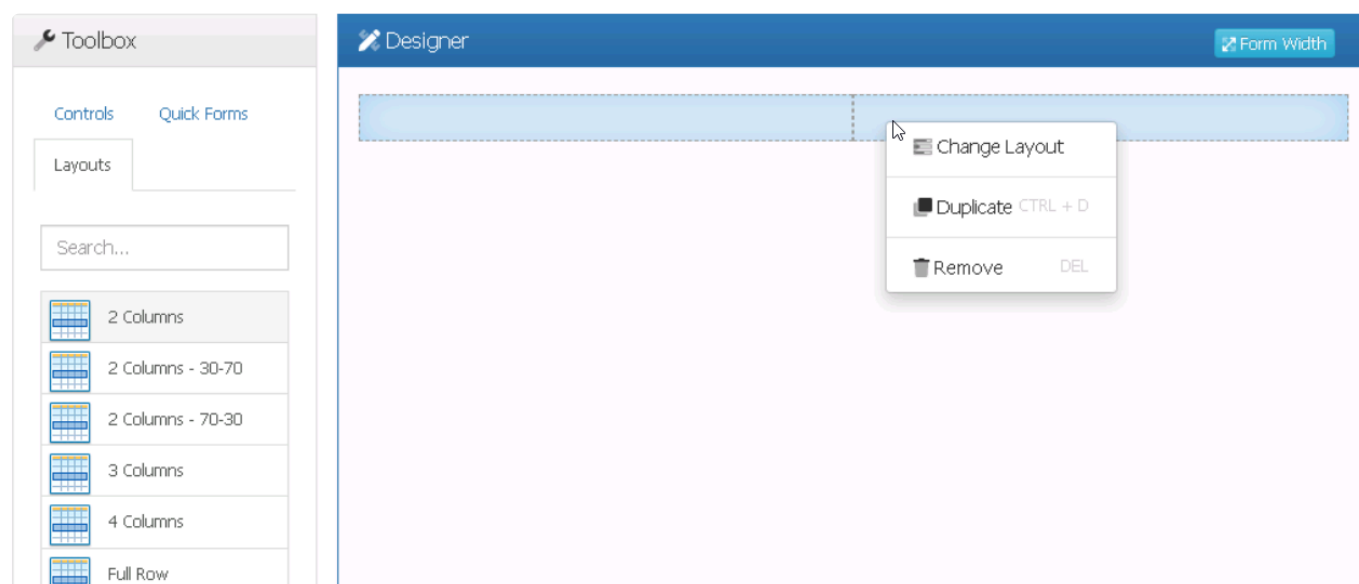


It is important to note that these layout styles are used solely for organizing and structuring the layout of controls within the form. Put simply, while the layout selected will impact how controls are categorized in the web application, the columns themselves will not be visible to the end-user.

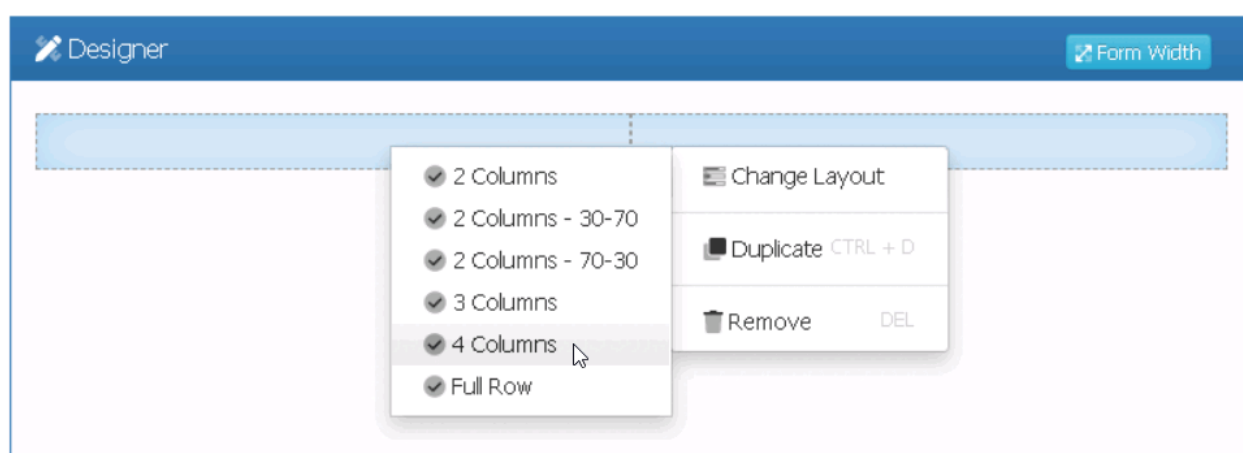
## Layout Preview: Design View vs. Web View

## 12.1.2.1.3.1. Layout Context Menu

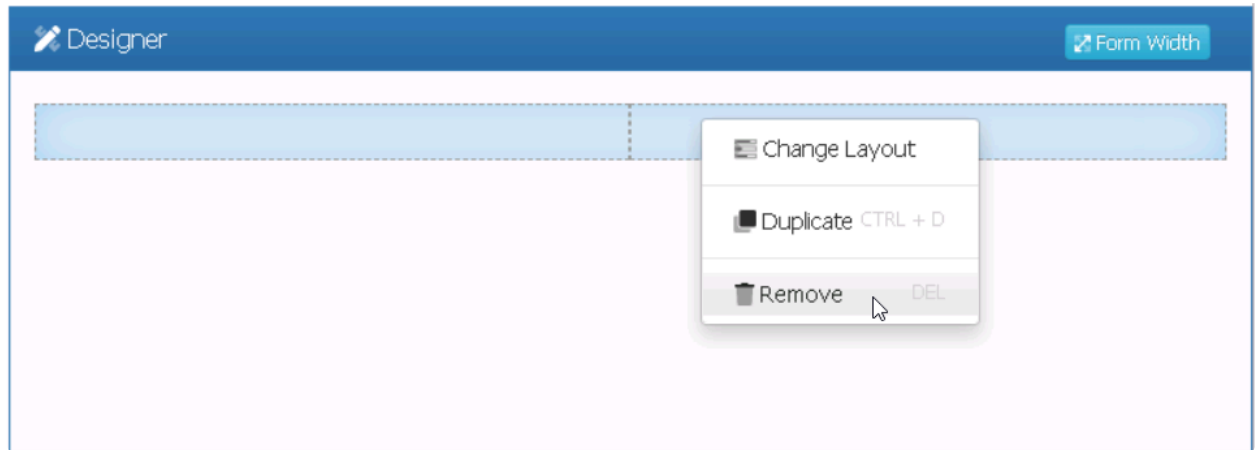
To change, duplicate, or remove a layout, right-click on the layout and select the corresponding option from the context menu.



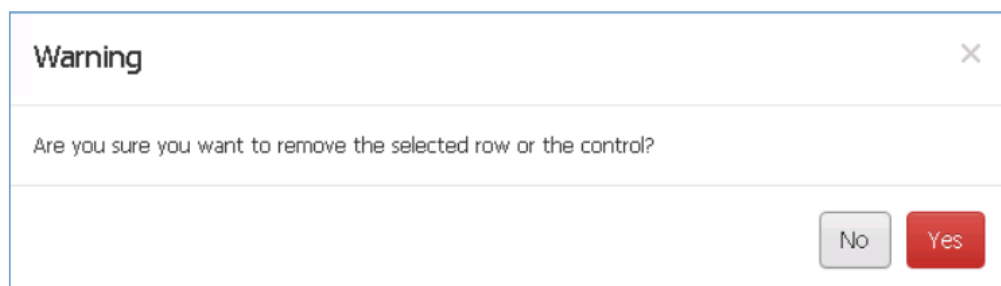
1. **Change Layout:** This option allows users to change the selected layout, replacing it with another layout from the toolbox.



2. **Duplicate Layout:** This option allows users to duplicate the selected layout, creating a copy below it.
3. **Remove Layout:** This option allows users to remove the selected layout.



- If this option is selected, users will be alerted of the action with the following confirmation message:



- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.

## 12.1.2.1.3.2. Layout Properties

When a layout is selected in the design environment, its associated properties are displayed in the Control Properties panel.

\* To access the **Control Behavior** category, users must first create the form behavior. Otherwise, only the **Name** and **Misc** categories will be shown.

1. **Name:** In this category, users can access basic properties required for the layout's identification and proper referencing in the code.
  - **Control Type:** This field displays the control type. It cannot be modified.
    - The control type for all layout styles is set to 'row'.
  - **Server ID:** This is a text field that allows users to input the Server ID for the control.
    - Server ID refers to the name used to call the layout in the code.

\* By default, the Server ID is set to the control type. Users, however, can modify this identifier as needed to ensure that the control is properly referenced and interacts correctly with other components in the code.

2. **Control Behavior:** In this category, users can define the behavior of the layout.
  - **Form Behavior:** This is a single-select dropdown field that retrieves the list of form behaviors (from the [Form Behavior](#) section of the **Forms** module). It allows users to select the form behavior associated with the layout.
    - Only form behaviors linked to the associated form will be displayed in this dropdown.
  - **Control State:** This field displays the current state of the layout.
    - **Control States**
      1. **Enabled:** This option allows for the content of the layout to be visible and editable in the web application.
      2. **Disabled:** This option allows for the content of the layout to be visible but

not editable in the web application.

3. **Invisible:** This option allows for the content of the layout to be hidden in the web application.

- **State Matrix:** This field allows users to configure the control state properties.
  - To access the control state properties, click on the ellipses (...) inside the field.

| Form Behavior          | Control State |     |
|------------------------|---------------|-----|
| 1. Modify Gap Analysis | ✓ Enabled     | ... |
| 2. New Gap Analysis    | ✓ Enabled     | ... |
| 4. Review Gap Analysis | ✓ Enabled     | ... |

- **Form Behavior:** This field displays the form behavior.
  - All form behaviors linked to associated form will be listed in this window.
- **Control State:** This is a single-select dropdown field that displays the list of control states. It allows users to modify the current control state of the form behavior.
- **Apply:** This button allows users to save their changes.
- **OK:** This button allows users to save their changes and close the window.
  - If a control state has been modified, a prompt will appear asking users whether they want to apply the same state to the child controls inside the layout.

Set control state

Do you want to set states for child controls?

Yes No

- If **Yes** is selected, the chosen control state will be applied to all the controls inside the layout.
  - This means that the behavior of all the child controls will be updated to match the control state of the parent layout.
- If **No** is selected, the chosen control state will not be applied to the controls inside the layout.
  - This means that the child controls will retain their previous

behavior, regardless of any changes made to the parent layout's control state.

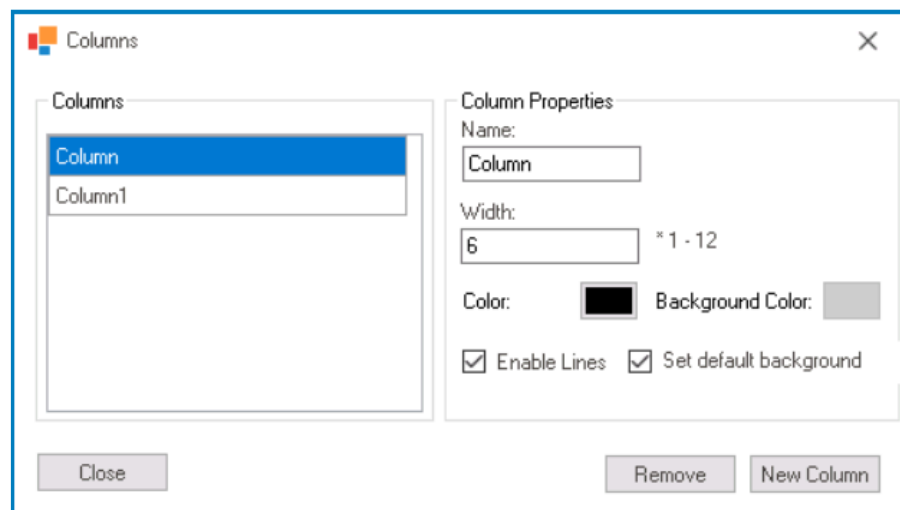
- **Cancel:** This button allows users to cancel their changes and close the window.



By modifying the control state per form behavior, users can determine which sections of the form are made visible and/or editable at different stages of the workflow.

3. **Misc:** In this category, users can define additional properties for the layout.

- **Border Color:** This is a single-select dropdown field that allows users to choose a border color for the layout.
  - It contains two tabs: Custom and Web.
    - In the Custom tab, users can choose a border color by selecting a color from a color picker. The color picker offers a selection of theme colors, standard colors, and the ability to create custom colors.
    - In the Web tab, users can select a border color from a scrollable list. The list displays a range of specific colors, each with its own name, allowing users to easily choose from pre-defined options.
    - After choosing a border color, the layout will update to reflect the new color scheme.
- **Is Border:** This is a checkbox which, if selected, allows users to display a border around the layout.
  - In order to apply a border color to the layout, users must first select the **Is Border** checkbox.
- **Physical Columns:** This field allows users to configure the column properties.
  - To access the column properties, click on the ellipses (...) inside the field.



- **Name:** This field displays the name of the selected column.
  - Users can modify this field as needed.
- **Width:** This field displays the width of the selected column.
  - Users can modify this field as needed. They must, however, ensure that the value entered falls within the range of 1-12.
- **Color:** This field allows users to set the border color of the selected column.
  - If selected, it will display the Color window, in which users can access a color

picker.

- This color picker offers a selection of basic colors and the ability to create custom colors.
- **Background Color:** This field allows users to set the background color of the selected column.
  - If selected, it will display the Color window, in which users can access a color picker.
    - This color picker offers a selection of basic colors and the ability to create custom colors.
  - In order to apply a background color to the column, users must first select a border color.
- **Enable Lines:** This is a checkbox which, if selected, allows users to enable lines or borders around the columns.
  - In order to apply a border color to the column, users must first select the **Enable Lines** checkbox.
- **Set Default Background:** This is a checkbox which, if selected, allows users to reset the column background color to its default state.
  - By default, the background color for columns is transparent.
- **New Column:** This button allows users to add a new column.
  - Before adding a new column, users must adjust the width of existing columns to ensure the total width of all columns does not exceed 12.
- **Remove:** This button allows users to remove a selected column.
- **Close:** This button allows users to close the Columns window.



The total width of all columns in the list cannot be greater than 12.

### Layout Border Color Preview: Web View

#### Quick Form

| Analysis ID | Analysis Name |
|-------------|---------------|
|             |               |

### Column Border and Background Color Preview: Web View

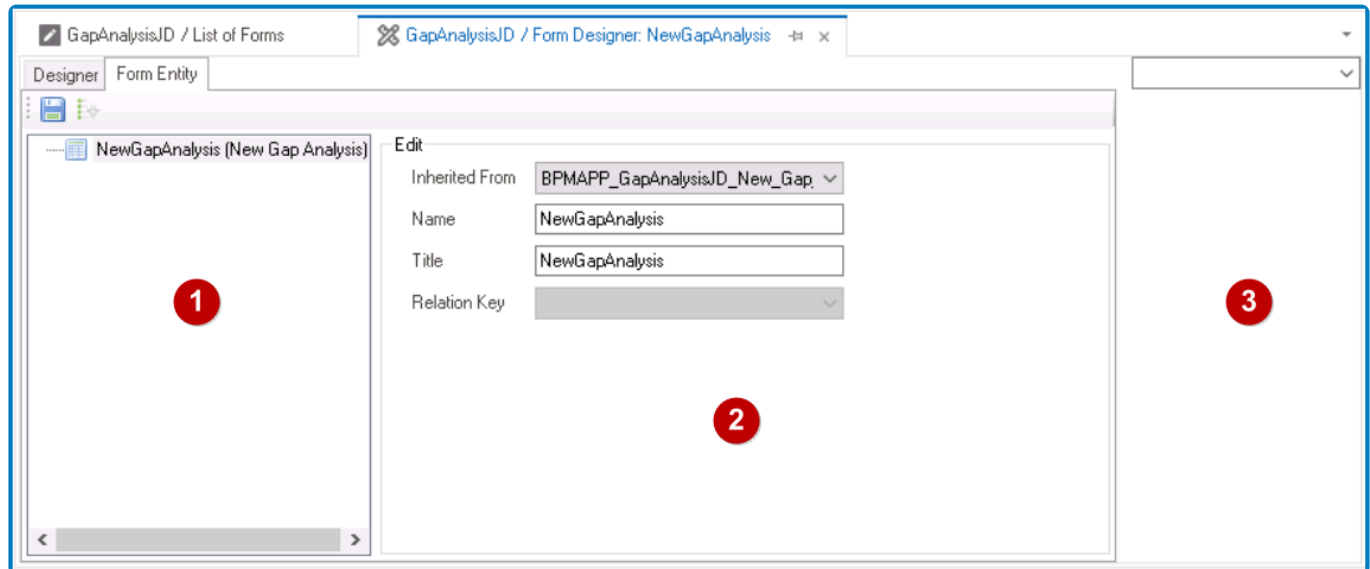
#### Quick Form

| Analysis ID | Analysis Name |
|-------------|---------------|
|             |               |

## 12.1.2.2. Form Entity Tab

In the Form Entity tab, users can modify the form properties and create local attributes for the form.

- ✿ Local attributes are specific to the form itself. They are used to store data that is relevant to the form, rather than to the entire application.

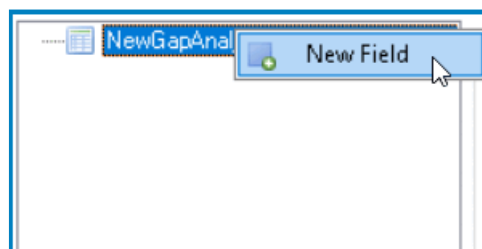


There are three primary components to the Form Entity tab:

### 1. Attribute Tree

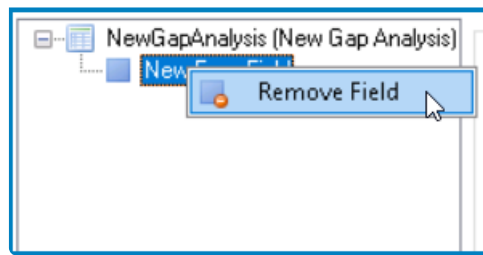
In the Attribute Tree, users can manage the attributes defined for the form entity, including creating and deleting form-specific local attributes.

- To create a local attribute for the form, right-click on the form entity and select **New Field** from the context menu. It will generate a new field below the form entity, which can then be customized and defined to suit the needs of the form.

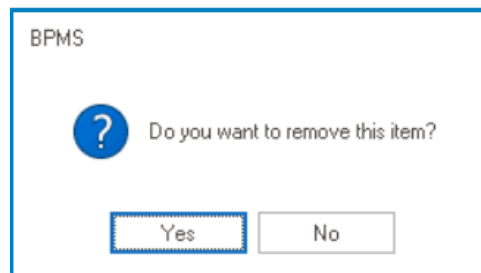


- To remove a local attribute, right-click on the attribute and select **Remove Field** from the context menu.





- If this option is selected, users will be alerted of the action with the following confirmation message:

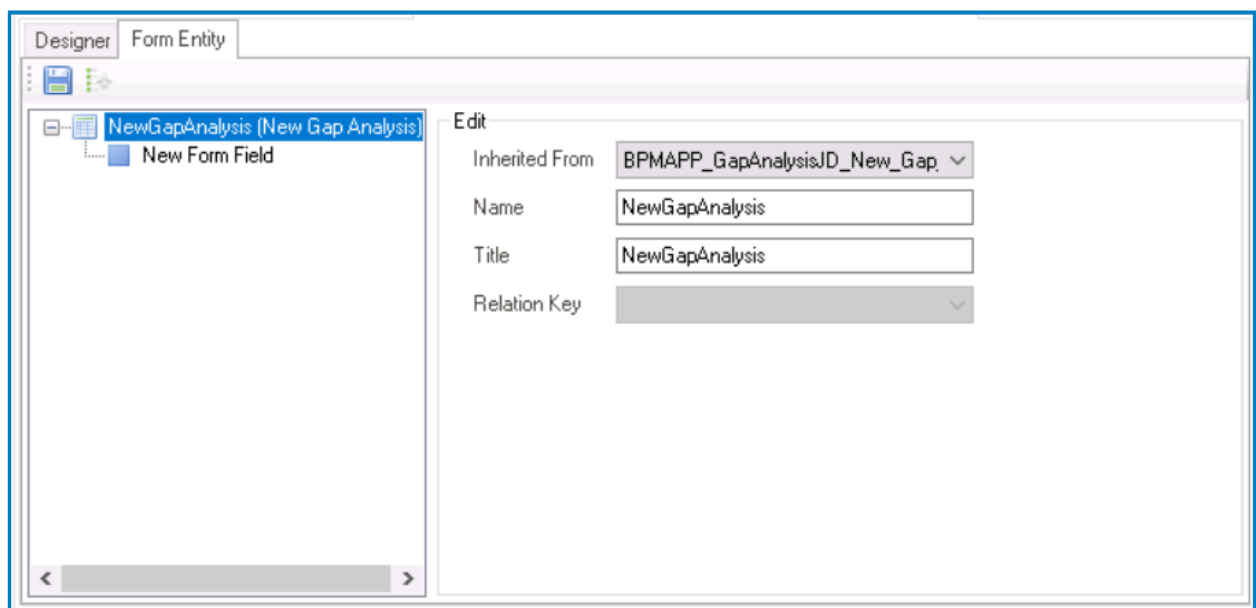


- **Yes or No:** To confirm the removal, click on the **Yes** button. To cancel, click on the **No** button.

## 2. Edit Properties

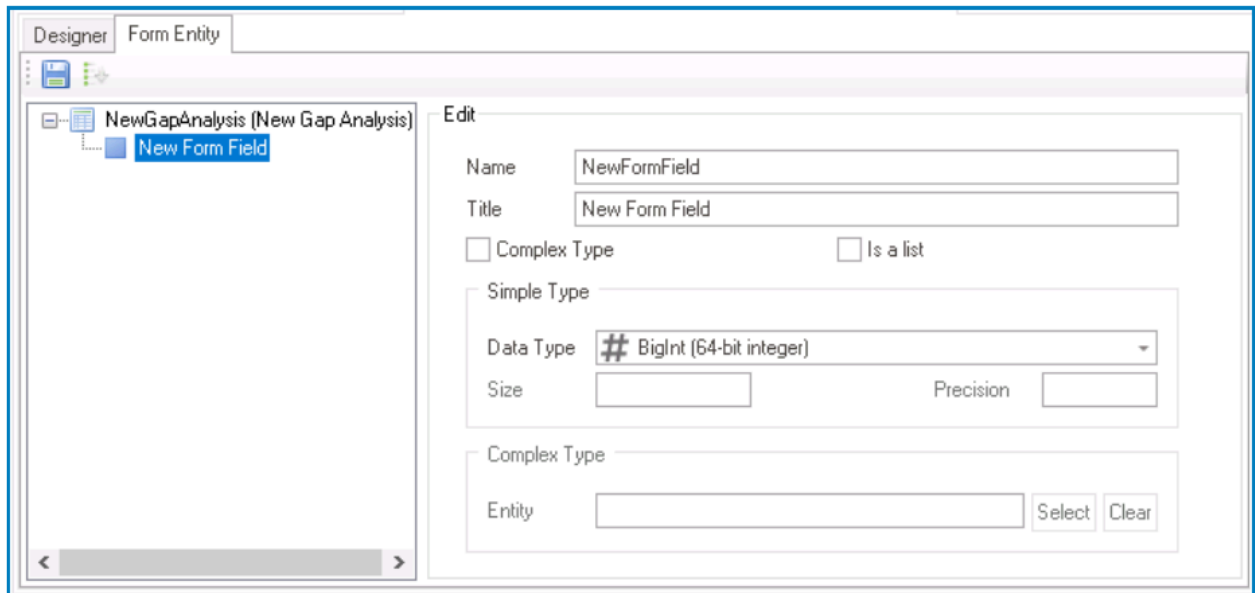
In the Edit Properties pane, users can modify the properties of the currently selected attribute or form entity.

- To modify the properties of the form entity, select it from the Attribute Tree. This will display the associated properties in the Edit Properties pane.



- **Inherited From:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select or modify the entity associated with the form.
- **Name:** This field displays the name of the form entity.
  - Users can modify this field as needed.

- **Title:** This field displays the title of the form entity.
    - Users can modify this field as needed.
  - **Relation Key:** This field displays the relation key for the form.
    - Users cannot modify this field.
- To define or modify the properties of an attribute, select it from the Attribute Tree. This will display the associated properties in the Edit Properties pane,



- **Name:** This is a text field that allows users to input the name of the attribute.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A form must have a unique name. It cannot have the same name as another form in the application.
- **Title:** This is a text field that allows users to input the title of the attribute.
  - The title inputted in this field will be used to identify the attribute in the Attribute Tree.
  - Unlike with names, there are no rules determining the validity of titles.
- **Complex Type:** This is a checkbox which, if selected, allows the control to reference data from another form or entity.
  - This field is used when the control is not directly tied to the form.
- **Is a List:** This is a checkbox which, if selected, allows the control to hold multiple instances of the attribute's value, effectively creating a list of data.
  - This field is used when the control is expected to display and/or manage a collection of data.
- **Simple Type:** In this section, users can define the properties of an attribute that stores a single, simple value within the form.
  - **Data Type:** This is a single-select dropdown field that displays the list of data types. It allows users to select the data type for the attribute.
    - By choosing the appropriate data type, users can ensure that the attribute's value is properly formatted and validated within the form.
  - **Size:** This is a numeric field that allows users to input the size of the data type associated with the attribute.

- Depending on the data type selected, this field may be enabled or disabled. For data types that require a specific size (such as Binary, Char, and Decimal), users must input a size within the given range of values.

Simple Type

Data Type **T** Binary (Fixed-length binary data)

Size  (1-8000) Precision

- **Precision:** This is a numeric field that allows users to input the precision of the decimal places for decimal or numeric data types associated with the attribute.
  - This field will only be enabled if Decimal or Numeric (Decimal) are selected as data types.

Simple Type

Data Type **#** Decimal (Decimal)

Size  (1-8000) Precision

- **Complex Type:** In this section, users can select the form or entity from which the attribute will reference data.
  - **Select:** This button allows users to select the referenced form or entity.
    - It will display the Select Reference window.

Select Reference

Select Entity

| Application       | Entity                          |
|-------------------|---------------------------------|
| =                 | mc                              |
| Gap Analysis (JD) | Actions                         |
| Gap Analysis (JD) | Audit Plan                      |
| System            | Departments                     |
| System            | Firms                           |
| Gap Analysis (JD) | FormObject_Audit Plan           |
| Gap Analysis (JD) | GapAnalysisJD_QuickForm         |
| Gap Analysis (JD) | GapAnalysisJDNew_Gap_AnalysisDS |
| Gap Analysis (JD) | New Gap Analysis                |
| System            | Individuals                     |
| System            | Offices                         |
| System            | Payment                         |
| System            | Principal Groups                |
| System            | Principals                      |
| Gap Analysis (JD) | QuickForm                       |
| System            | Roles                           |
| System            | Users                           |

Select Cancel



- **Select or Cancel:** To confirm the selection, click on the Select button. To cancel the selection, click on the Cancel button.
- Once selected, the form or entity will populate the **Entity** field.

Complex Type

Entity

- **Clear:** This button allows users to clear the selected form or entity from the **Entity** field.

✿ A simple type attribute allows the control to manage and display a single value that is contained within the form itself. A complex type attribute allows the control to retrieve and display data from an external source.

-  : This button allows users to save their changes.
-  : This button allows users to automatically sort all attributes listed in the tree in ascending order.

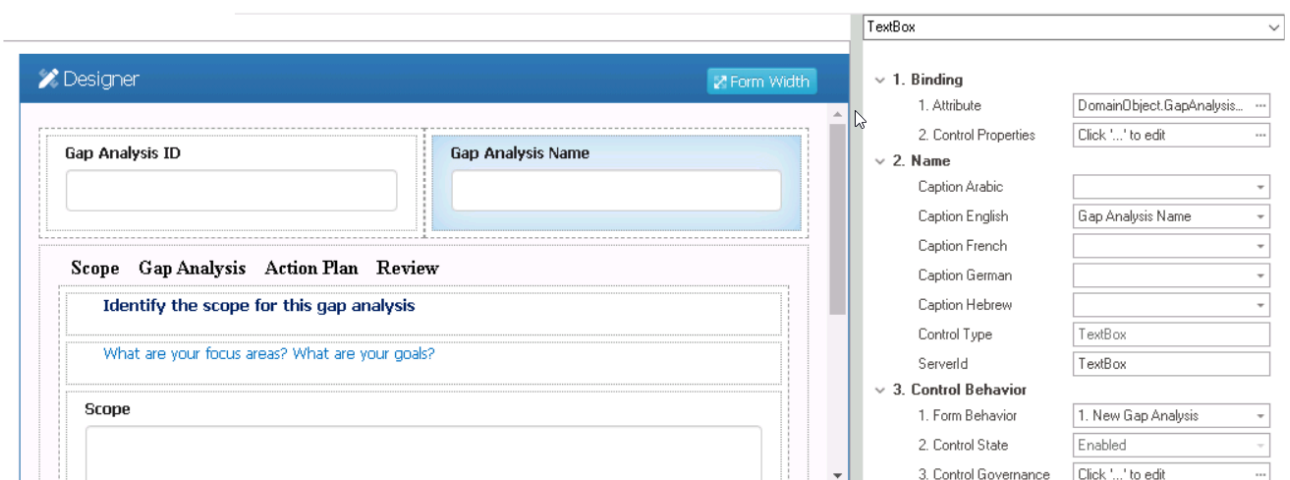
### 3. Control Properties

In the Control Properties panel, users can view and modify the properties of a selected layout or control. To select a layout or control, either click on it from the design environment or choose it from the dropdown menu in the panel.

✿ Depending on the item selected, the properties displayed in this panel may differ.

To resize this panel, take the following steps:

- Hover the cursor over the edge of the panel's left-side border (as shown below).



- Click and drag the cursor until you get the desired size.

Designer

Form Width

Gap Analysis ID

Gap Analysis Name

TextBox

Scope

Gap Analysis

Action Plan

Review

Identify the scope for this gap analysis

What are your focus areas? What are your goals?

Scope

TextBox

1. Binding

1. Attribute

DomainObject.GapAnalysis...

2. Control Properties

Click '...' to edit

2. Name

Caption Arabic

Caption English

Gap Analysis Name

Caption French

Caption German

Caption Hebrew

Control Type

TextBox

ServerId

TextBox

3. Control Behavior

1. Form Behavior

0. Modify Gap Analysis

2. Control State

Enabled

3. Control Governance

Click '...' to edit

## 12.1.2.3. Tips and Tricks for Customizing Templates

To customize templates with advanced features, the following tips and tricks may be useful.

### 1. Conditional Blocks

Users can utilize conditional blocks to hide a section in the Word or PDF document.

To add a conditional block, use the following format:

```
<conditional @attributeName>
block content
</conditional>
```



Use `@attributeName` to access the entity (domain object) and `@@attributeName` to access the form entity (form object).

To avoid extra line breaks, it's best to add the conditional tags directly next to your content.

For example:

```
<conditional @attributeName>Content Line1
Content Line2
Content Line3</conditional>
```

### Conditional Blocks: Examples

### 2. Watermarks

To add watermarks, insert `<conditionalWatermark @attributeName/>` anywhere in the document.



Please note that if more than one watermark is added, only the last one will be processed. All other tags, however, will be removed from the resulting Word or PDF document.




As with other conditional tags, use `@attributeName` to access the entity (domain object) and `@@attributeName` to access the form entity (form object).

### 3. Column Sorting in Grids

To sort a grid according to an attribute (or column), add either of the following symbols in the first cell of the first row (header row most of the time):

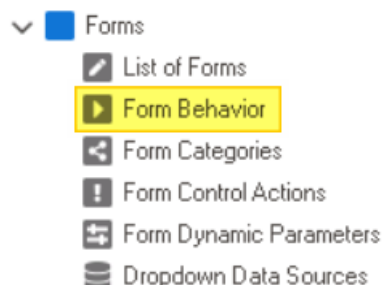
- ^attributeName for ascending
- ~attributeName for descending

 Currently, we only support sorting by one attribute.

 Please note that when generating a template in the [Form Designer](#), this sorting tag will be automatically added to the grid according to the grid control configuration.

## 12.2. Form Behavior

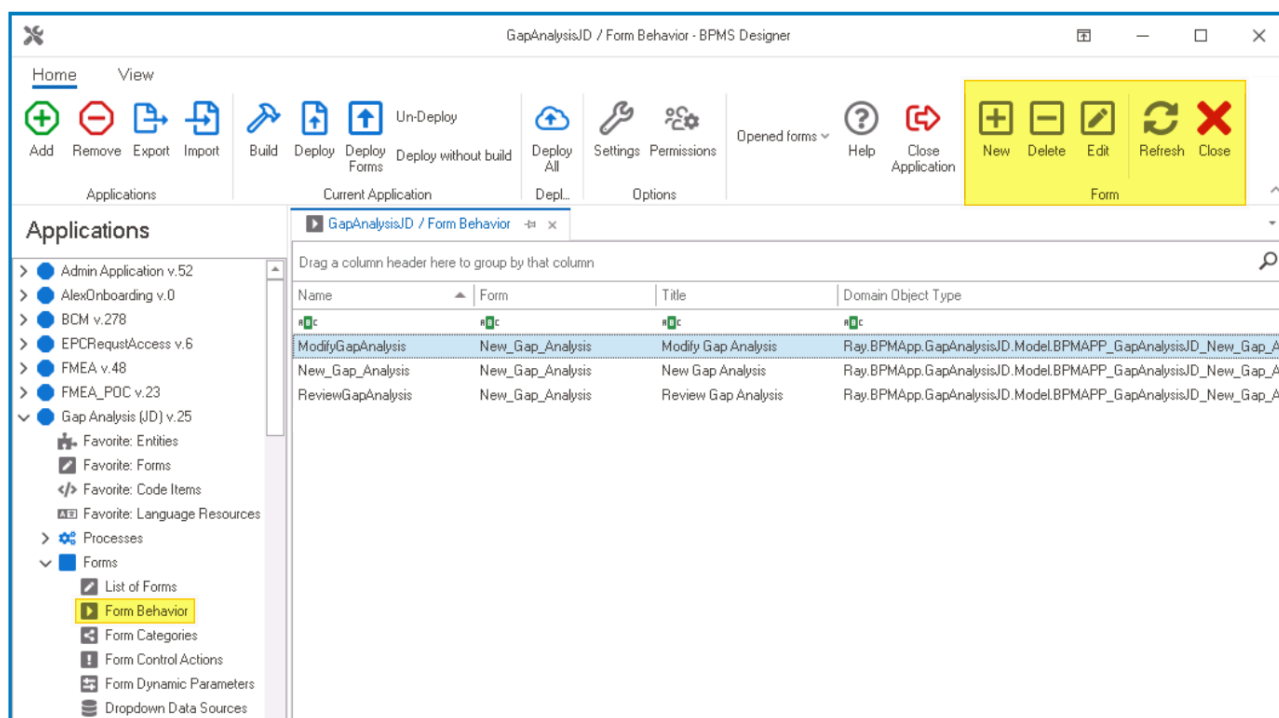
In the **Form Behavior** section of this module, users can create custom behaviors for their application's forms. By defining these behaviors, users can control how forms are displayed in the web application, how data is collected, and how responses are handled.



\* Form behaviors are rules that determine how forms behave and operate in the web application. By customizing form behaviors, users can create forms that are optimized to meet their specific requirements.

To manage forms in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > Form Behavior** (double-click to open). It will display the [Form Behavior](#) tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined form behaviors for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

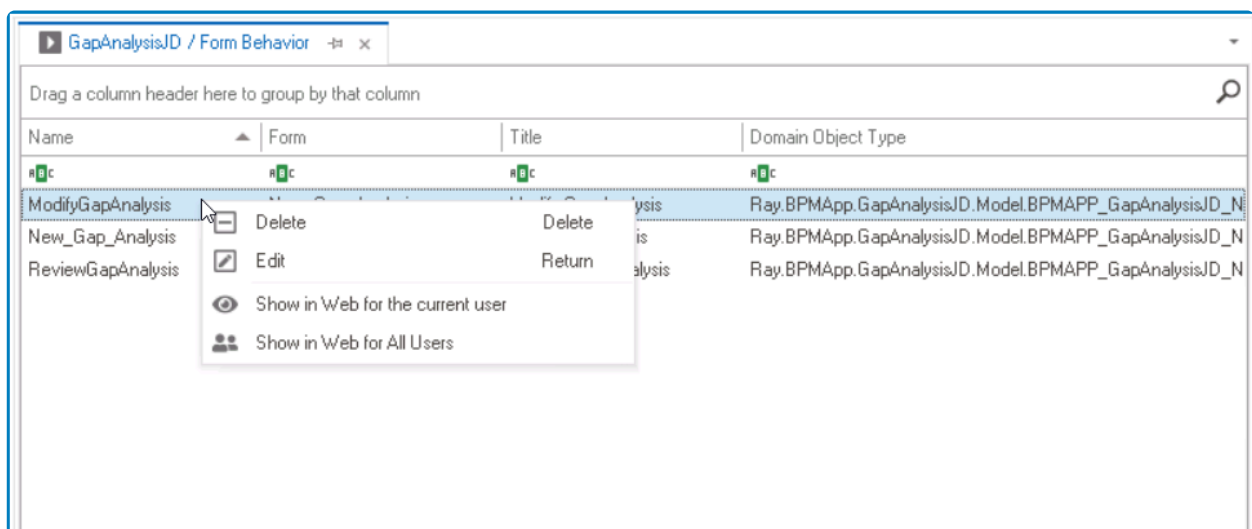


2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new form behavior.
  - If selected, it will display the [Form Behavior Properties](#) window.
- **Delete:** This button allows users to delete an existing form behavior.
  - To use this function, first select a form behavior from the list.
- **Edit :** This button allows users to edit the properties of an existing form behavior.
  - If selected, it will display the [Form Behavior Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a form behavior from the list.
- **Refresh :** This button allows users to refresh the [Form Behavior](#) tab.
- **Close:** This button allows users to close the [Form Behavior](#) tab.

**Note:** To access the form behavior context menu, right-click on a form behavior in the list. It will display the following options:



- **Delete:** This option allows users to delete the selected form behavior.
- **Edit :** This option allows users to edit the properties of the selected form behavior.
- **Show in Web for the Current User:** This option allows the form to be accessible in the web application to the user currently logged into the Designer.
- **Show in Web for All Users:** This option allows the form to be accessible in the web application to all users.

## 12.2.1. Form Behavior Properties

---

In the Form Behavior Properties window, users can create a new form behavior for the application.

There are three tabs to consider in this window:

1. [General](#)
2. [Advanced](#)
3. [Title](#)



To save changes per tab, click on the **Apply** button. To save changes and exit the Form Behavior Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 12.2.1.1. General Tab

In the General tab, users can define the basic properties of the form.

1. **Name:** This is a text field that allows users to input the name of the form behavior. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A form behavior must have a unique name. It cannot have the same name as another form behavior linked to the entity.
2. **Starter:** This is a checkbox which, if selected, allows users to mark a form as a “starter form.”
  - If this checkbox is selected, it will enable the following fields: **Show in Form Menu**, **View Order**, **Enable Tracking**, and **Show Tracking Code**.

✿ A starter form is a form that is used to initiate a new process or workflow. By marking a form as a starter, users can ensure that it is easily accessible in the web application, whereas non-starter forms can only be accessed as stages within an existing process or workflow.

3. **Title:** This is a text field that allows users to input the title of the form behavior.
- The title inputted in this field will be the displayed title of the form in the **DBP** web application.



Users can override this displayed title in the [Title](#) tab of the window.

- Unlike with names, there are no rules determining the validity of titles.
4. **Form:** This is a single-select dropdown field that retrieves the list of forms (from the [List of Forms](#) section of the **Forms** module). It allows users to select the form to which they want to apply the form behavior.
5. **Save Button at the Bottom:** This is a checkbox which, if selected, allows for a **Save** button to be displayed at the bottom of the form.

Submit form

Save Form



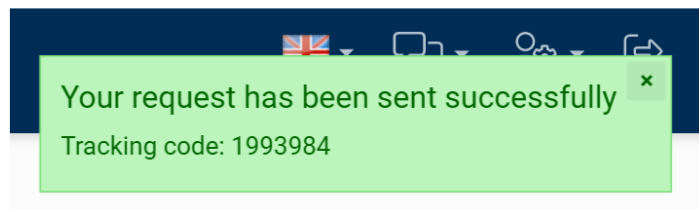
For this feature to work, users must also enable a draft for the form. To enable a draft, select the **Draft Enabled** checkbox and create a corresponding task type.


6. **Show in Form Menu:** This is a checkbox which, if selected, allows for the form to be displayed in the form menu of the web application.
- If this checkbox is selected, it will enable all fields inside the **Receivers** section of the window.
7. **View Order:** This is a numeric field that allows users to determine the order in which the form appears in the form menu or its assigned category.

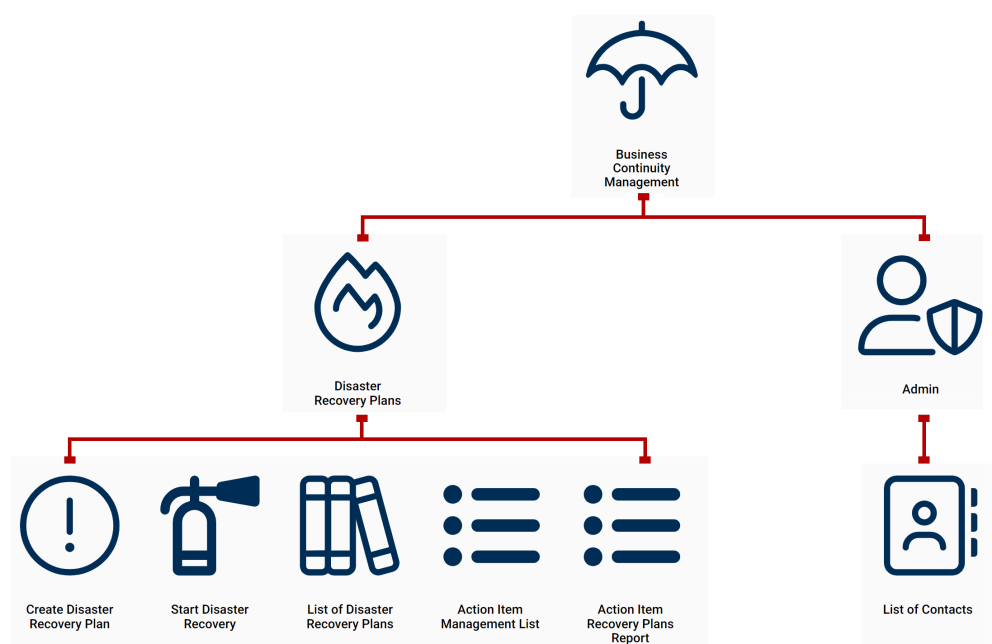
8. **Enable Tracking:** This is a checkbox which, if selected, allows users to enable tracking of form submissions.
- In the **DBP** web application, users can track form submissions using the [Tracking](#) form.

\* To be able to utilize this feature, users must also select the **Show Tracking Code** checkbox.

9. **Show Tracking Code:** This is a checkbox which, if selected, allows the unique tracking code to be displayed on the form after submission, enabling users to track the form's progress and completion.
- The form's tracking code is displayed in the top right corner of the screen.



10. **Receivers:** In this section, users can specify who can access the form and where the form will be stored in the web application.
- Category:** This field allows users to assign a category to the form, which determines the folder where the form will be stored in the web application. This helps to keep forms organized and easily accessible.
    - To select a form category, click on the  button. It will display the Task Admin Form Category window.
      - For more information on the contents of this window, [click here](#).
    - Using categories, users can group multiple forms that are related to the same subject, making it easier to organize and locate them in the web application.
      - Form Categories Example: Web View**




*This menu map illustrates the organization of forms in the Business Continuity Management category, which is a top-level category in the web*


*application. Under this category, there are two subcategories: Disaster Recovery Plans and Admin. The Disaster Recovery Plans subcategory contains five forms, while Admin has one form.*


- **Receivers:** This field allows users to define the receiver(s) authorized to access the form. These receivers are individual users, roles, or groups that are explicitly identified by the user.




The screenshot shows a form titled 'Receiver(s)'. Inside, there is a 'Category' dropdown menu currently set to 'Forms'. To the right of the dropdown is a small button with two dots. Below these is a large, empty rectangular box intended for a list of receivers. At the bottom right of the box are two buttons: a plus sign (+) and a minus sign (-).

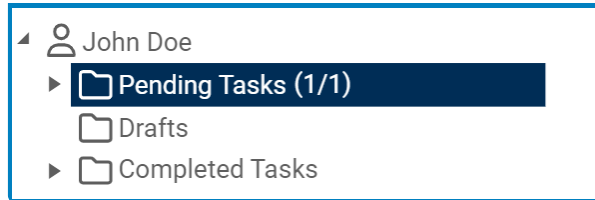
-  : This button allows users to add a new receiver.
  - It will display the Receivers window, from which users can select the specific receiver.
    - **For more details on the Receivers window, see [previous instance](#).**
  - Once a receiver has been added, it will populate the **Receivers** field.

This screenshot shows the same 'Receiver(s)' form as before, but now the list box contains one entry, 'John Doe', which is highlighted with a blue background. The '+' and '-' buttons remain at the bottom right.


-  : This button allows users to remove a selected receiver.

 Users can also determine the accessibility of the form in the web application using the [form behavior context menu](#).

11. **Draft Enabled:** This is a checkbox which, if selected, allows users to save the form as a draft.
  - To set up the draft, users must create the corresponding task type.
    - To create the task type, select the  button.
      - It will display the [Task Type Properties](#) window.
    - To modify the task type, select the  button.
    - To remove the task type, select the  button.
  - After saving the form in the web application, users can access the draft in the **Drafts** folder of the **Task Navigation Pane**.



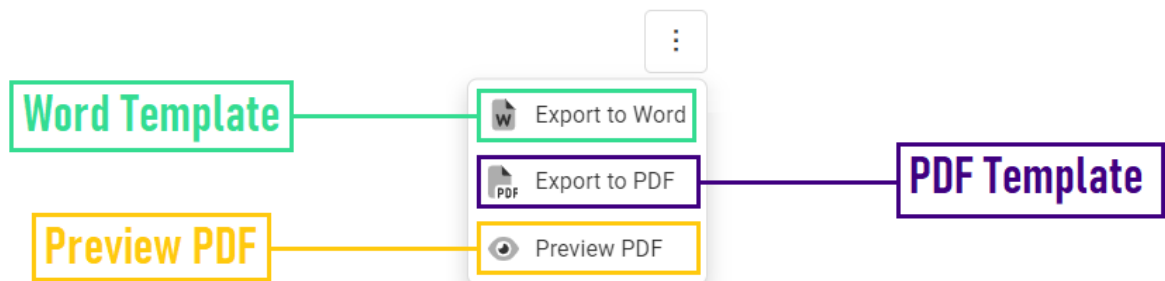
12. **Word Template:** This field allows users to add a template to the form.

- To upload a template file, select the  button.
  - It will display the [Word Template Editor](#) window, in which users can upload template files in different languages.

**!** Please note that while the downloaded template file can be either a Word or PDF document, the uploaded template file must be in Word format.

13. **File Type:** This is a single-select dropdown field that displays the list of file types. It allows users to select the desired file type(s) for the downloadable template.

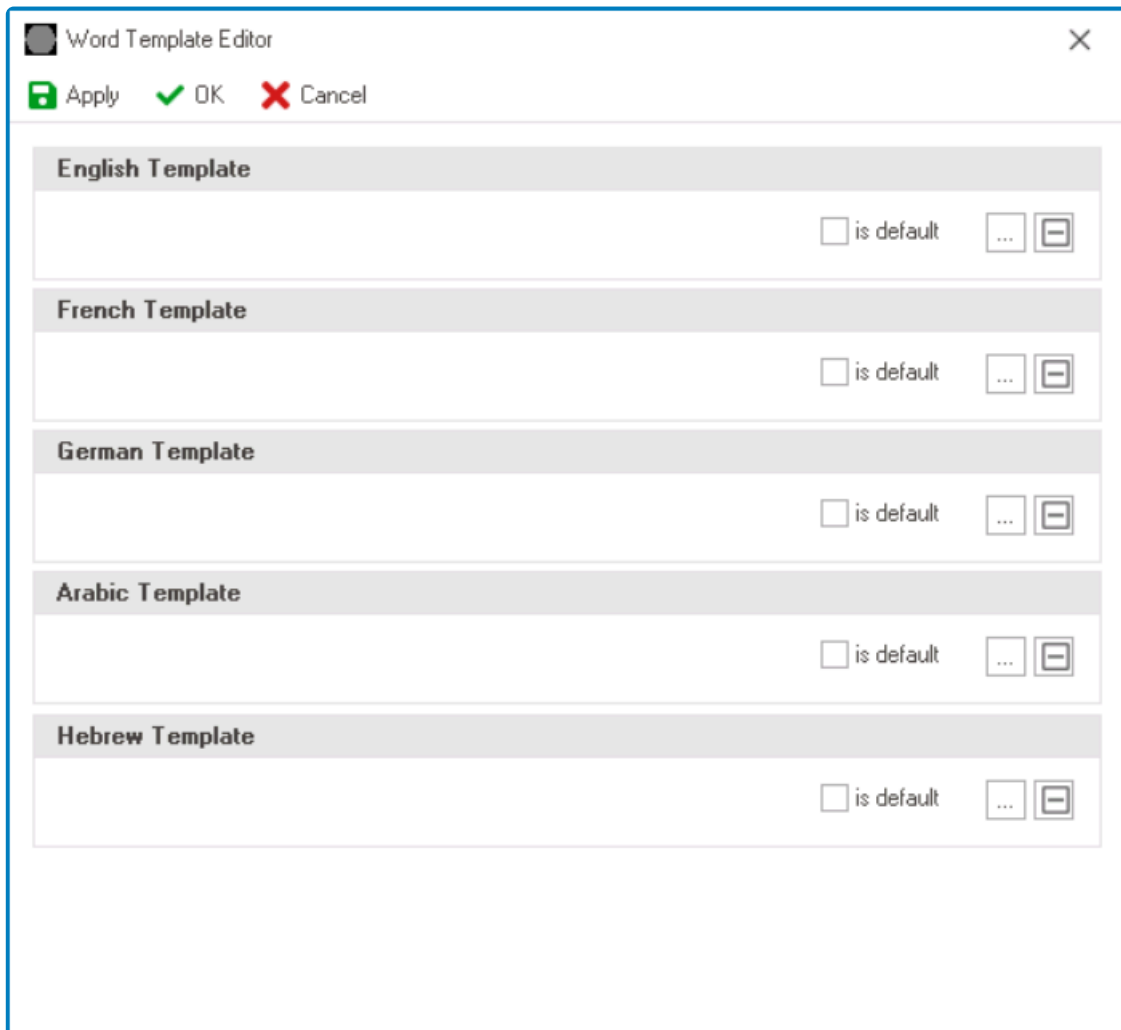
- File Types: Word, PDF, or All Output Types.
  - Word: This option allows users to download the template in Microsoft Word format.
  - PDF: This option allows users to download the template in Portable Document Format (PDF).
  - All Output Types: This option allows users to download the template in all available file formats, including Word and PDF.
- When enabled, templates can be accessed using the kebab (or three dots) menu located in the upper rightmost corner of the form.





- As outlined in the above image, using the available options in the menu, users can download a Word template, download a PDF template, or preview a PDF template.

## 12.2.1.1.1. Word Template Editor

In the Word Template Editor window, users can upload template files in different languages, enabling them to create and customize templates for use in a multilingual environment



1. **Template:** For each language, users can upload a separate template file.
  - **Is Default:** This is a checkbox which, if selected, allows users to set a specific language template file as the default.
  -  : This button allows users to upload a template file for the language.
    - It will display the [Application Files](#) window, in which users can add and select the template file.
  -  : This button allows users to remove a template file.
2. **Apply:** This button allows users to save their changes.
3. **OK:** This button allows users to save their changes and close the window.
4. **Cancel:** This button allows users to cancel their changes and close the window.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.



## 12.2.1.2. Advanced Tab

In the Advanced tab, users can define the basic properties of the form.

**Form Behaviour Properties**

Apply OK Cancel

General Advanced Title

1 Form Object GapAnalysisJDNewGapAnalysisDS (GapAnalysisJDNewGapAnalysisDS)

2 Domain Object NewGapAnalysis (NewGapAnalysis)

3 Pre-Form Action [Empty]

4 Post-Form Action [Empty]

5 ☐ Show possible event (used by base info form)

6 Buttons

- ☒ Needs Save and Close button (only for modal forms)
- ☒ Needs Save and New button (only for modal forms)
- ☒ Enable navigation buttons ( only for modal forms )
- ☒ Needs Submit Button
- ☐ Close modal form on apply

1. **Form Object:** This field displays the form object or form entity associated with the selected form.
2. **Domain Object:** This field displays the domain object or entity associated with the selected form.

✿ Users can modify the form selection in the [General](#) tab of the window.

3. **Pre-Form Action:** This is a single-select dropdown field that retrieves the list of pre-form actions associated with the form object (from the [Pre-Form Actions](#) section of **Actions**). It allows users to link a pre-form action to the form.

! If the **Starter** checkbox is selected for the form behavior, only pre-form (starter) actions will be displayed in this dropdown.

4. **Post-Form Action:** This is a single-select dropdown field that retrieves the list of post-form actions associated with the form object (from the [Post-Form Actions](#) section of **Actions**). It allows users to link a post-form action to the form.

! If the **Starter** checkbox is selected for the form behavior, only post-form (starter) actions will be displayed in this dropdown.

5. **Show Possible Event:** This is a checkbox which, if selected, allows
6. **Buttons:** In this section, users can determine which buttons are displayed on the form.
  - **Needs Save and Close Button:** This is a checkbox which, if selected, allows for the **Save and Return** button to be displayed on the modal form.
    - When this button is selected, the form data will be saved and the modal form will be closed, returning users to the main form.

A blue rectangular button with rounded corners and a white border, containing the text "Save and Return" in white.


- **Needs Save and New Button:** This is a checkbox which, if selected, allows for the **Save and New** button to be displayed on the modal form.
  - When this button is selected, the form data will be saved and a new modal form will be opened for data entry.

An orange rectangular button with rounded corners and a white border, containing the text "Save and New" in white.

- **Enable Navigation Buttons:** This is a checkbox which, if selected, allows users to display navigation buttons on the modal form.
  - These buttons allow users to navigate between different records or data entries in the form.



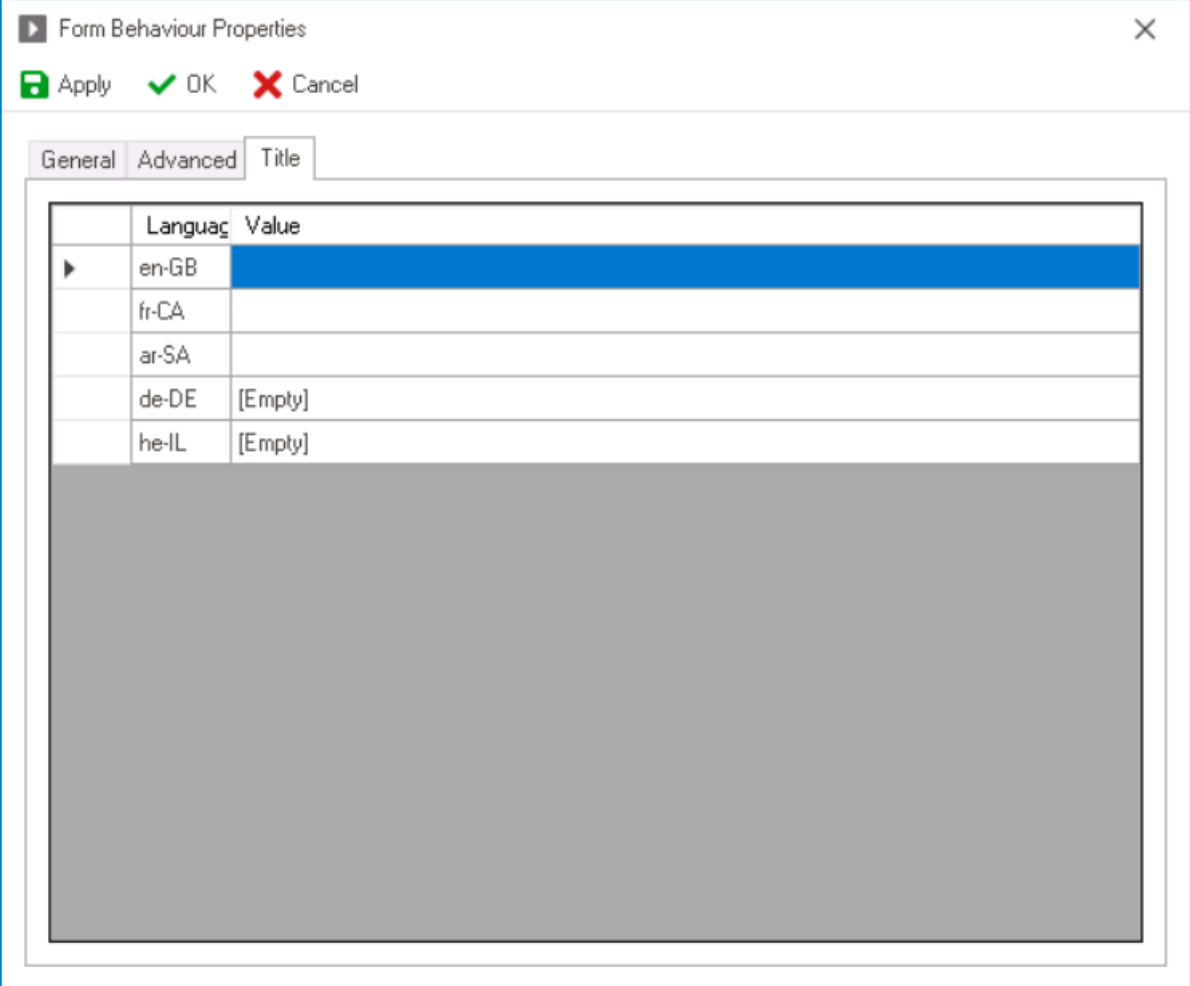
- **Needs Submit Button:** This is a checkbox which, if selected, allows for the **Submit** button to be displayed on the form.
  - When this button is selected, the form data will be submitted for processing.

A blue rectangular button with rounded corners and a white border, containing the text "Submit form" in white.

- **Close Modal Form on Apply:** This is a checkbox which, if selected, allows for the modal form to be closed upon clicking the **Apply** button.

## 12.2.1.3. Title Tab

In the Title tab, users can set the title for the form in different languages. As English is set as the default language in the system, any title inputted in the English language field will be used as the display title for the form, regardless of other language options that may be available.









| Language | Value   |
|----------|---------|
| en-GB    |         |
| fr-CA    |         |
| ar-SA    |         |
| de-DE    | [Empty] |
| he-IL    | [Empty] |

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

A form's title is a descriptive name which helps to identify and differentiate it from other forms in the **DBP** web application. It appears at the top of the form and is used as a reference for users when working with the form.

## Sample Form

|   |   |   |   |
|---|---|---|---|
| Analysis ID   | Analysis Name   | Analysis Date   | Analysis Type   |
| <input type="text"/>  | <input type="text"/>  | <input type="text" value="2024-01-01"/>  | <div>Select...</div> |
| Data Collection Method  | Data Analysis Method  | Analysis Report Format  | Category  |
| <div>Select...</div> | <div>Select...</div> | <div>Select...</div>                     | <div>Select...</div> |

### Description

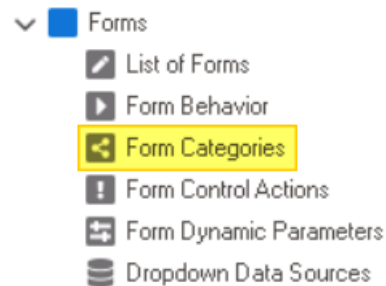
### Analysis Description

[illegible]

[Submit form](#)

## 12.3. Form Categories

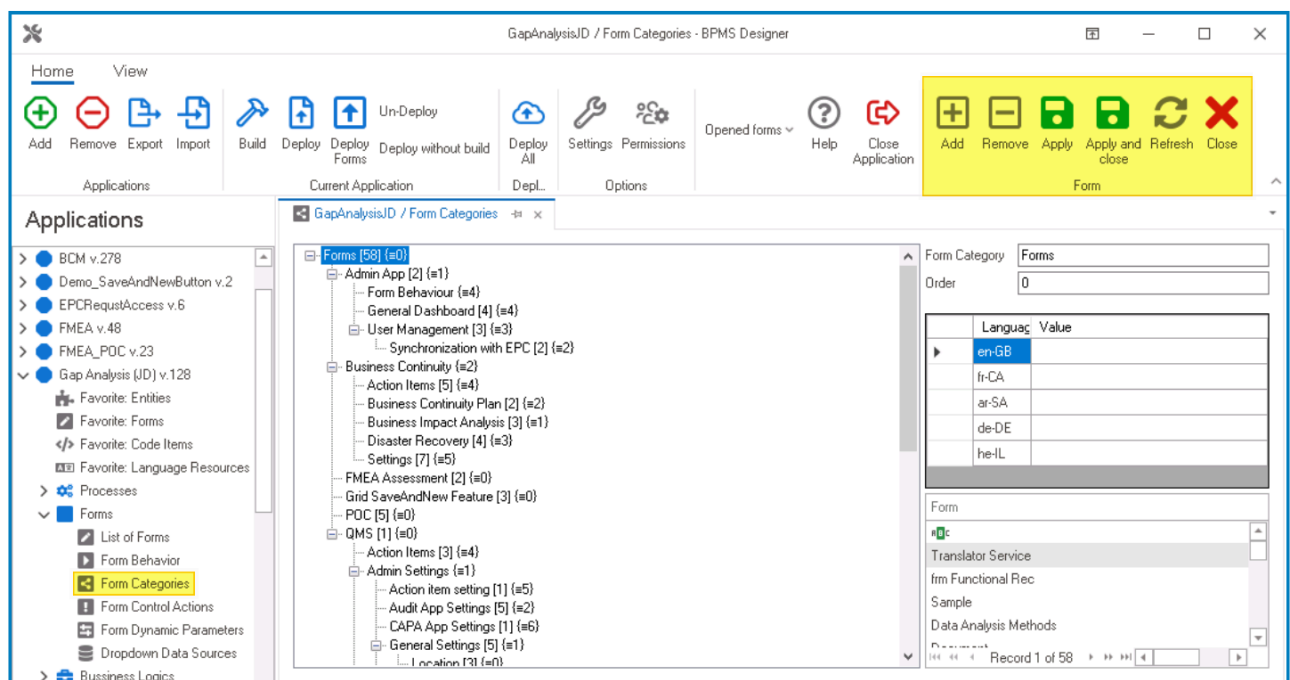
In the **Form Categories** section of this module, users can create a hierarchical structure of categories to organize and store their forms in the **DBP** web application.



- ✿ With the ability to create multiple layers of categories, users can group related forms together in a hierarchical structure that reflects their workflow or organizational needs. Each form category represents a new folder where the forms can be stored in the web application, and there is no limit to the number of levels that can be used.

To manage form categories in the program, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > Form Categories** (double-click to open). It will display the **Form Categories** tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined form categories in the program. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **Add:** This button allows users to add a new form category.
  - If selected, it will generate a new category at the bottom of the list, which can then be customized and defined according to the user's preferences.
    - For more information on defining the form category properties, [click here](#).
- **Remove:** This button allows users to remove an existing form category.
  - To use this function, first select a form category from the list.
- **Apply:** This button allows users to save their changes.
- **Apply and Close:** This button allows users to close the Form Categories tab.
- **Refresh:** This button allows users to refresh the Form Categories tab.
- **Close:** This button allows users to close the Form Categories tab.



Users can also create form categories directly within the properties of the corresponding form behavior. For more information on how to create form categories in this way, [click here](#).

## 12.3.1. Form Category Properties

In the Form Category Properties pane, users can define or modify the properties of the selected form category.

1. **Form Category:** This is a text field that allows users to input a title for the form category.
2. **Order:** This is a numeric field that allows users to determine the order in which the form category appears in the web application.
  - By setting a value in this field, users can arrange the form categories and subcategories in the desired order, based on their workflow or organizational needs.
    - **Form Category Order Example: Design View**

✿ All listed categories adhere to the Form Category [No. of Forms] {≡ Order} format. For example, **Action Items [5] {≡ 4}**.

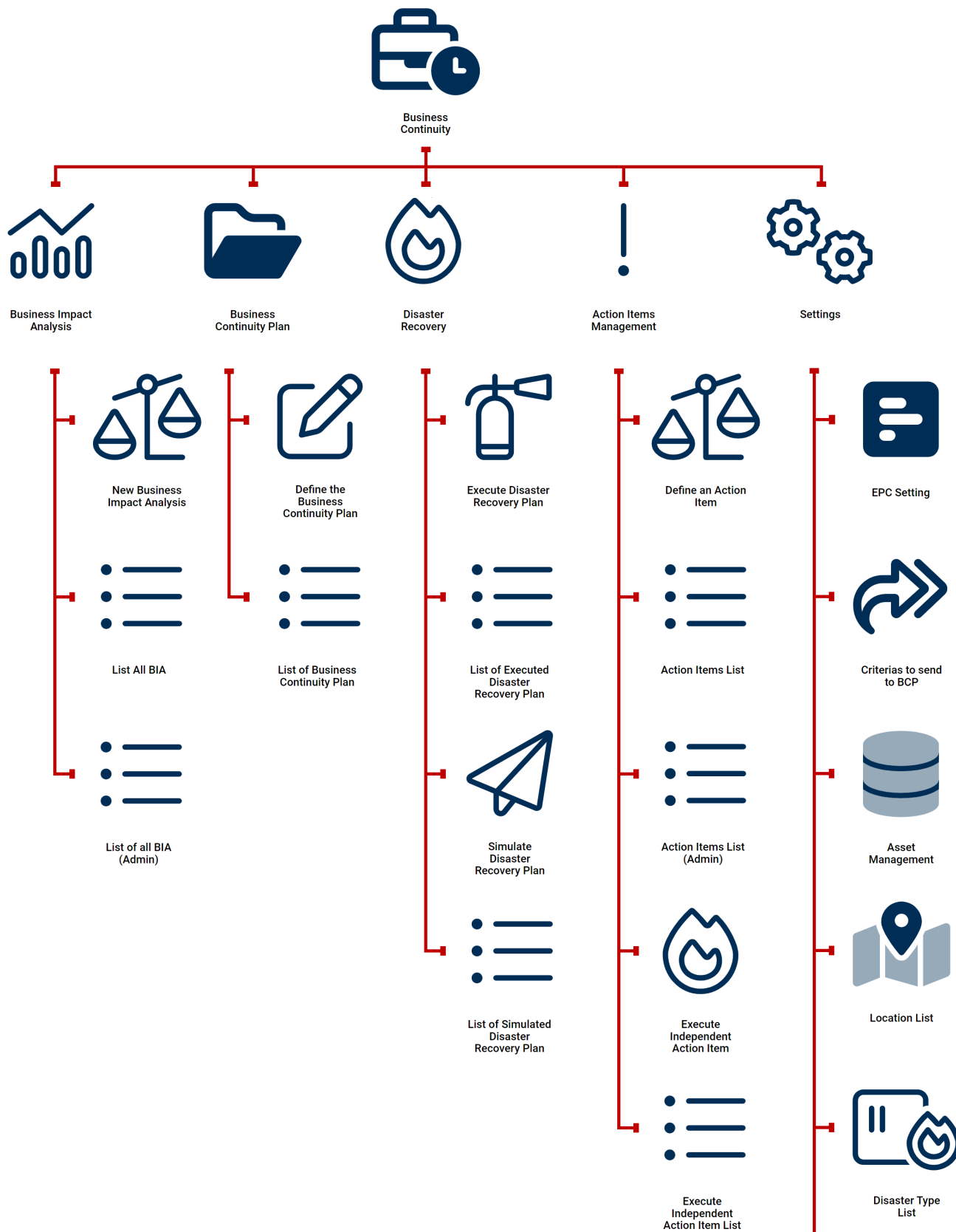
3. **Translation:** This is a table field that allows users to set the title for the form category in different languages.
  - As English is set as the default language in the system, any title inputted in the English language field will be used as the display title for the category, regardless of other language options that may be available.

✿ Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

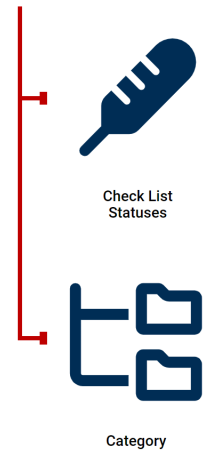
4. **Form:** This table displays all forms assigned to the selected category.

- Users can assign categories to their forms in the properties of the corresponding form behavior.
  - For more information on how to assign categories to forms, [click here](#).

### Form Categories Hierarchy: Web View







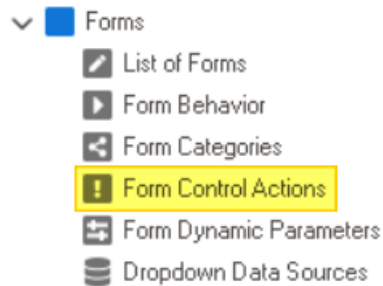
*This menu map illustrates the organization of forms in the Business Continuity category, which is a top-level category in the web application. Under this category, there are five subcategories: Business Impact Analysis, Business Continuity Plan, Disaster Recovery, Action Items Management, and Settings. The number of forms within each subcategory is as follows: Business Impact Analysis (3), Business Continuity Plan (2), Disaster Recovery (4), Action Items Management (5), and Settings (7). This hierarchical structure helps users to easily locate and manage their forms based on their specific needs.*



To save your changes, click on the **Apply** button in the menu bar. To save changes and exit the [Form Categories](#) tab, click on the **Apply and Close** button.

## 12.4. Form Control Actions

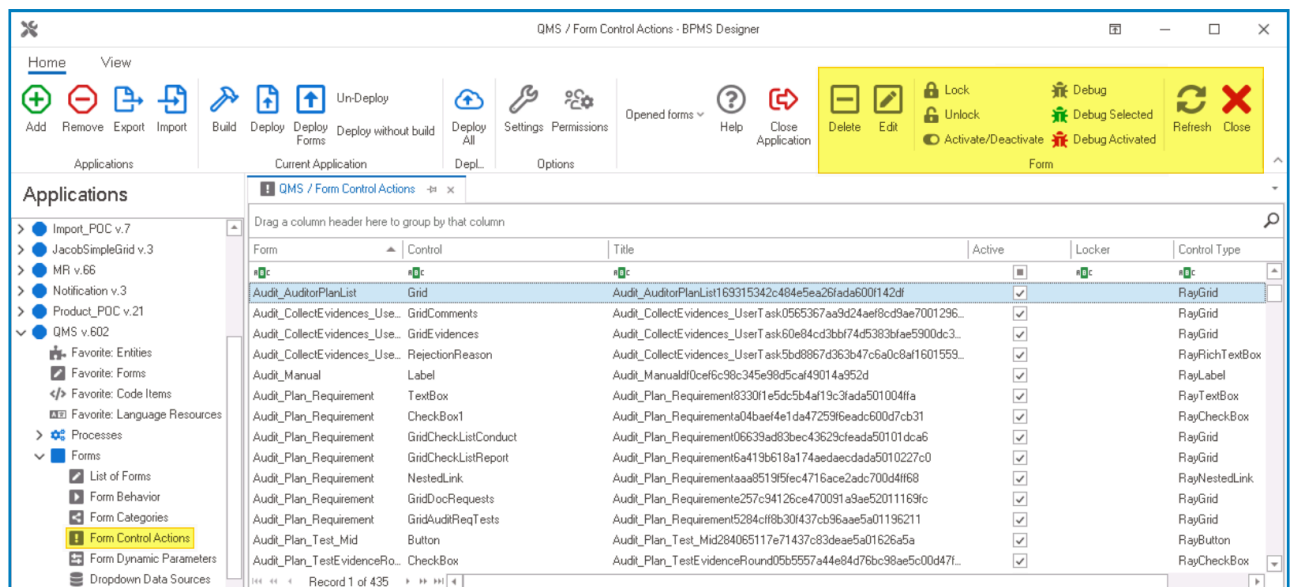
In the **Form Control Actions** section of this module, users can access and manage a comprehensive list of the actions that have been created by adding code or rules to controls or forms. This provides a streamlined and efficient way to monitor and control the behavior of the form or control.



✿ For more information on how to add rules, [click here](#). For more information on how to add code to a specific control, [click here](#).

To manage form control actions in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > Form Control Actions** (double-click to open). It will display the **Form Control Actions** tab and expand the menu bar to show additional options.



✿ In this tab, users can access a list of all defined form control actions in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



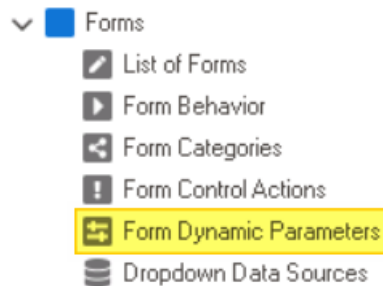
- **Delete:** This button allows users to delete an action.
  - To use this function, first select an action from the list.
- **Edit:** This button allows users to edit the code associated with the action.
  - If selected, it will launch the [Code Designer](#), in which users can edit and customize the code specifying the behavior of the control or form.

✿ Users can write code to dynamically show or hide form controls based on user input or selection. For example, a radio button control in a form can be configured to show or hide a multi-line textbox depending on the user's response to a question.

- **Lock:** This button allows users to lock the action, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the action.
  - If selected, access restriction on the action will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the action.
- **Debug All:** This button allows users to debug all actions in the list.
- **Debug Selected:** This button allows users to debug only the selected action in the list.
- **Debug Activated:** This button allows users to debug only the actions that are tagged as active.
- **Refresh :** This button allows users to refresh the [Form Control Actions](#) tab.
- **Close:** This button allows users to close the [Form Control Actions](#) tab.

## 12.5. Form Dynamic Parameters

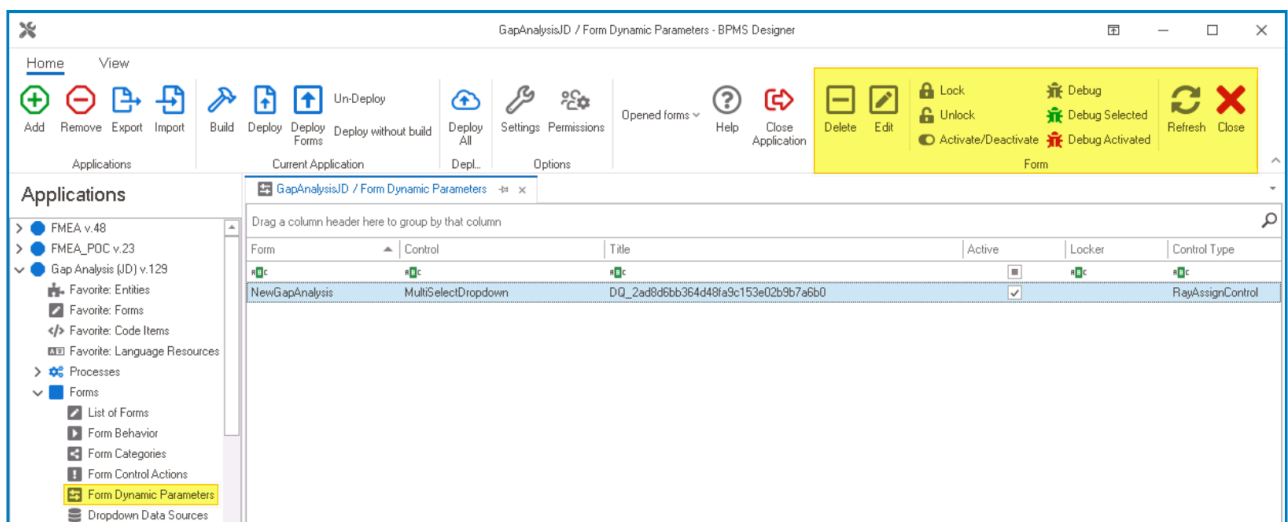
In the **Form Dynamic Parameters** section of this module, users can access and manage a list of the dynamic parameters used to modify the behavior of forms within the application.



\* Form dynamic parameters refer to values that are generated or changed dynamically in a form based on user input or other factors. These parameters can include controls such as dropdowns, checkboxes, radio buttons, and text fields, as well as hidden fields that contain data that is not visible to the user.

To manage form dynamic parameters in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > Form Dynamic Parameters** (double-click to open). It will display the **Form Dynamic Parameters** tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined form dynamic parameters in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



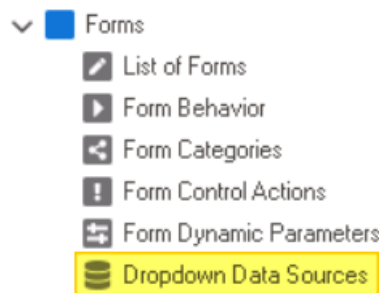
- **Delete:** This button allows users to delete a dynamic parameter.
  - To use this function, first select a dynamic parameter from the list.
- **Edit:** This button allows users to edit the code associated with the dynamic parameter.
  - If selected, it will launch the [Code Designer](#), in which users can edit and customize the code specifying the behavior of the control or form.

✿ Users can write code to dynamically show or hide form controls based on user input or selection. For example, a dynamic parameter can be used to display different options in a dropdown list based on the user's previous selections.

- **Lock:** This button allows users to lock the dynamic parameter, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the dynamic parameter.
  - If selected, access restriction on the dynamic parameter will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the dynamic parameter.
- **Debug All:** This button allows users to debug all dynamic parameters in the list.
- **Debug Selected:** This button allows users to debug only the selected dynamic parameter in the list.
- **Debug Activated:** This button allows users to debug only the dynamic parameters that are tagged as active.
- **Refresh :** This button allows users to refresh the [Form Dynamic Parameters](#) tab.
- **Close:** This button allows users to close the [Form Dynamic Parameters](#) tab.

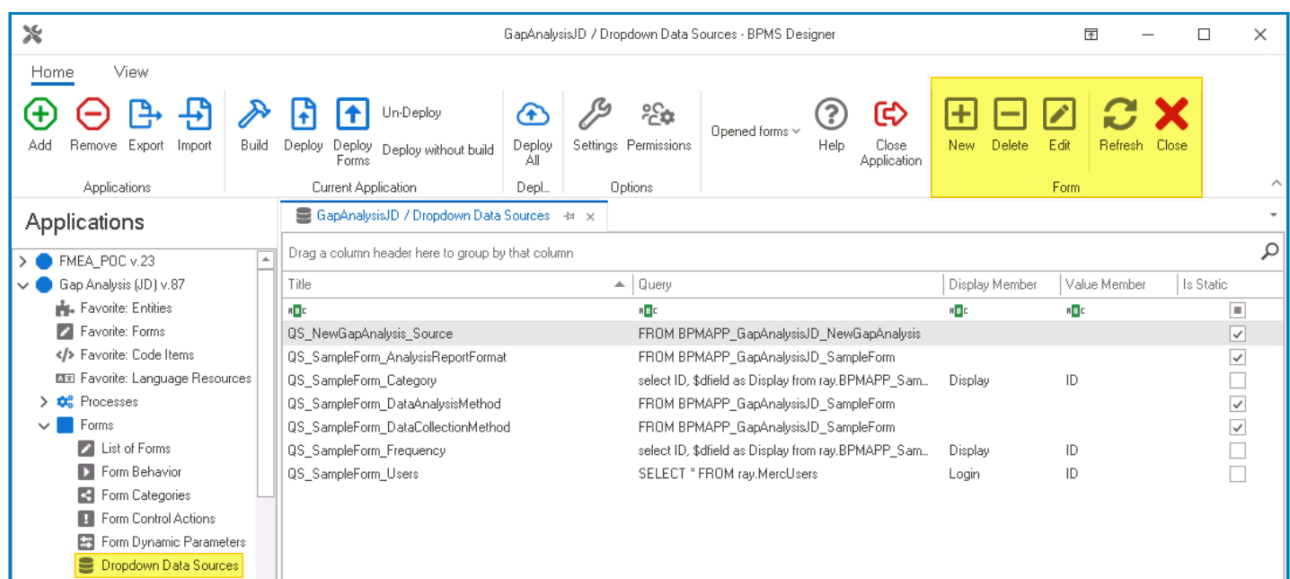
## 12.6. Dropdown Data Sources


In the **Dropdown Data Source** section of this module, users can create dropdown data sources for the application.



To manage dropdown data sources in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Forms > Dropdown Data Sources** (double-click to open). It will display the **Dropdown Data Sources** tab and expand the menu bar to show additional options.



 In this tab, users can access a list of all dropdown data sources defined for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new dropdown data source.

- If selected, it will display the [Query Editor](#) window.
- **Delete:** This button allows users to delete an existing dropdown data source.
  - To use this function, first select a dropdown data source from the list.
- **Edit :** This button allows users to edit an existing dropdown data source.
  - To use this function, first select a dropdown data source from the list.
- **Refresh :** This button allows users to refresh the [Dropdown Data Sources](#) tab.
- **Close:** This button allows users to close the [Dropdown Data Sources](#) tab.



Users can also create a dropdown data source directly within the properties of the corresponding dropdown. For more information on how to create dropdown data sources in this way, [click here](#).



When defining dropdown data sources, be sure to use meaningful names that clearly identify their purpose and contents.

## 12.6.1. Query Editor

---

In the Query Editor window, users can create a new dropdown data source for the application.

There are two tabs to consider in this window:

1. [Dynamic](#)
2. [Static](#)



To enable the Static tab (and simultaneously disable the Dynamic tab), select the **Static** checkbox at the top of the window.



To save changes per tab, click on the **Apply** button. To save changes and exit the Query Editor window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.



## 12.6.1.1. Dynamic Tab

In the Dynamic tab, users can create a query that pulls data from a database or other external source to populate the dropdown options.

- ✿ This type of dropdown is useful for displaying information that may change frequently or depend on other fields in the form. When the query is executed, the dropdown will display the most up-to-date information available from the external source.

The screenshot shows the 'Query Editor' dialog box with the 'Dynamic' tab selected. The 'Static' checkbox is unchecked. The 'Title' field is empty. The 'Connection' dropdown is set to 'BPMS'. Under the 'DataBase Table' section, 'Select Table' and 'SQL Table' are empty, and the 'SQL' checkbox is checked. Below this is a large text area for the SQL query. At the bottom of the 'DataBase Table' section are buttons for 'Execute', 'Dynamic', 'Edit Code', 'Designer', and 'Remove'. Below the 'DataBase Table' section is a section titled 'Indicate if your control is ComboBox' with 'Value Member' and 'Display Member' dropdowns.

1. **Static:** This is a checkbox which, if selected, will activate the [Static](#) tab and redirect users to it, while disabling the Dynamic tab.
2. **Title:** This is a text field that allows users to input the title of the dropdown data source.
  - A dropdown data source must have a unique name. It cannot have the same name as another dropdown data source in the application.
3. **Connection:** This field allows users to choose the data source connection that will be used to retrieve data for the dropdown options.
  - By selecting a connection from this dropdown, users can establish a link between the query and the external data source.

! Users must select the **BPMS** option as the data source connection.

4. **Select Table:** This field retrieves the list of tables (or entities) available in either the application or system. It allows users to choose the table that will be used to retrieve data for the dropdown options.

\* If created in the application, the tables displayed in this field will be retrieved from the **Entity Designer** section of the **Entities** module. If the tables already exist in the system, they will be retrieved from the database directly.

5. **SQL Table:** Once a table has been selected, this field will automatically be populated with a sample SQL query that specifies the table name and any necessary SQL syntax to retrieve the data for the dropdown options.
- To use this sample, copy and paste it into the command box.

The screenshot shows a 'DataBase Table' dialog box. It has a 'Select Table' dropdown menu with 'MercUsers' selected. Below it, the 'SQL Table' field contains the query 'SELECT \* FROM ray.MercUsers'. A checkbox labeled 'SQL' is checked. At the bottom, there are buttons for 'Execute', 'Dynamic', 'Edit Code', 'Designer', and 'Remove'.

6. **SQL:** This is a checkbox which, if selected, allows users to write their own SQL query in the command box.
- Execute:** This button allows users to verify and execute the SQL query.
    - If selected, it will display the [Query Executor](#) window.
  - Edit Code:** This button allows users to launch the [Code Designer](#), in which users can write code specifying custom functions or methods to manipulate the data retrieved by the SQL query.
    - For more details on how to customize SQL queries using the [Code Designer](#), [click here](#).
  - Designer:** This button allows users to launch the [Rule Designer](#), in which they can define rules that specify the conditions and actions for the SQL query.
    - For more details on how to customize SQL queries using the [Rule Designer](#), [click here](#).
  - Remove:** This button allows users to remove the code.

\* To view a list of helpful sample queries, [click here](#).

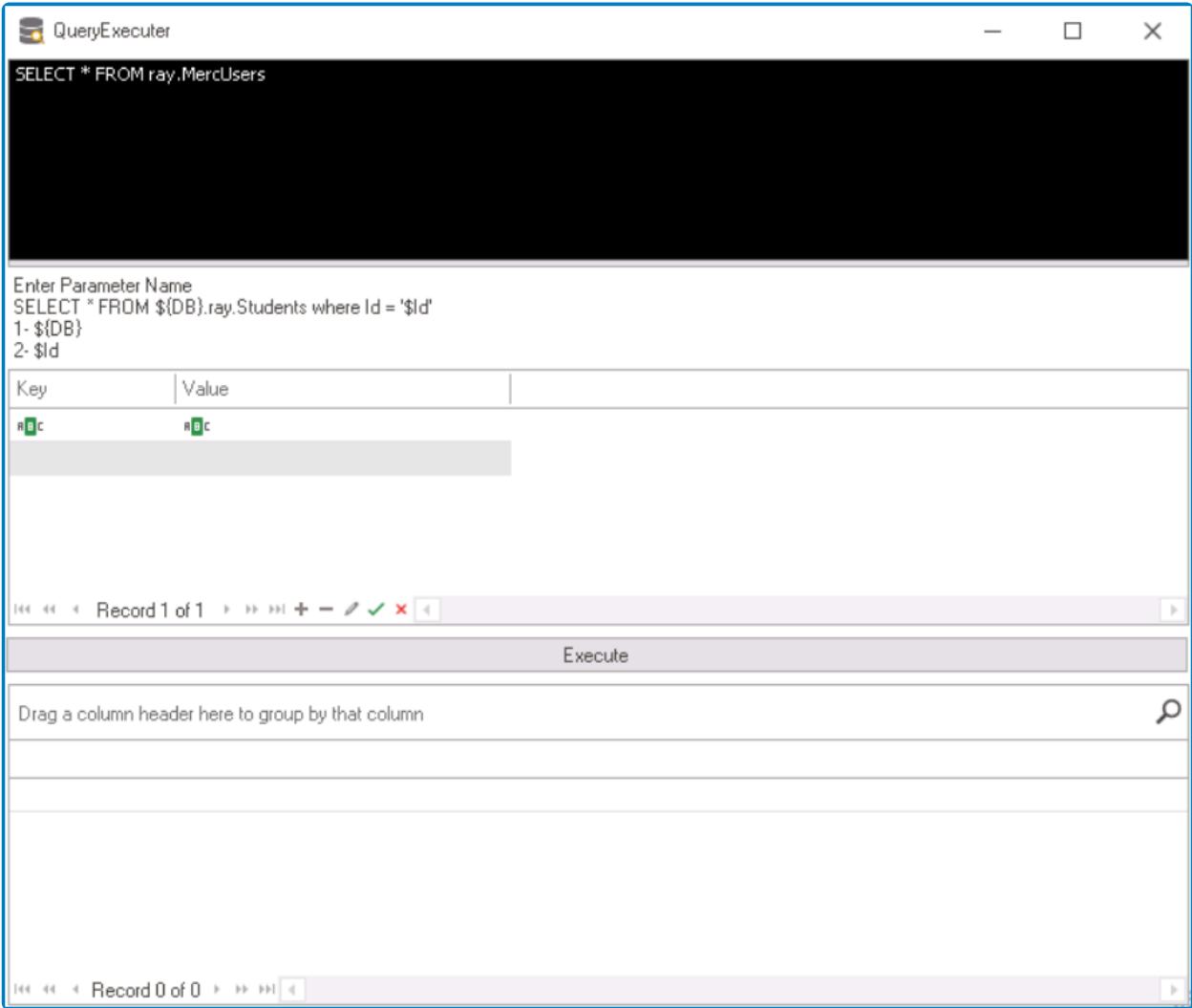
7. **Value Member:** This field retrieves the list of value members relevant to the selected table. It allows users to choose the value member for the dropdown control.
  - The value member is the actual data that is stored and manipulated in the system.
8. **Display Member:** This field retrieves the list of display members relevant to the selected table. It allows users to choose the display member for the dropdown control.
  - The display member is the visible representation of the value member that is shown to the user in the user interface.




If you are creating a combo box control, please select values for the **Value Member** and **Display Member** fields to define the data associated with each item in the dropdown list and how it is presented to the user.

# 12.6.1.1.1. Query Executor


In the Query Executor window, users can verify and execute the SQL query.



 For guidance on how to manipulate the data presented in this window, [click here](#).

In the table below the query field, there are two columns: Key and Value.

- If your query contains parameter placeholders, enter the parameter names in the Key column and the corresponding values for the parameters in the Value column.
- Click on the **Execute** button.
- The program will execute the modified query with the parameter values you entered and display the results in the table below.

 If your query does not contain any parameter placeholders, you do not need to enter any values for the Key and Value parameters in the table. You can simply click on the **Execute** button to run the query and display the results in the table below.

## 12.6.1.1.2. Sample Queries

This section provides a collection of sample queries that can be used to retrieve useful information from the database. These queries are designed to help users better understand how to query the database and retrieve the information they need. Please refer to the following list of queries for more information.

### 1. Retrieve Users Belonging to a Specific Group

To retrieve a list of users that belong to a specific group, apply either of the following queries:

- **SQL**

```
Select top 100 percent u.Id, u.Title
from ray.MercPrincipals u
join ray.MercPrincipalGroupMembers gp on gp.PrincipalID=u.ID
join ray.MercPrincipals g on g.ID=gp.GroupID
where u.MemberType=2 and u.Active=1 and g.Title='All users'
order by u.Title
```

- **HQL**

```
select u FROM User u JOIN u.PrincipalGroups g where g.Title='Auditors'
```

### 2. Retrieve All Active Tasks For All Applications

To retrieve a list of all active tasks for all applications, apply the following query:

```
select i.Id, ray.DecodeML(i.Subject, 'en') InstanceName, CreationDate, p.Title
AssignedTo, t.ApplicationName, ray.DecodeML(t.Subject, 'en') TaskTypeName, p
v.FaTitle ProcessName
from ray.BPMS_TaskInstance i
left join ray.MercPrincipals p on i.PrincipalId=p.ID
left join ray.BPMS_TaskType t on i.TaskTypeId=t.Id
left join ray.BPMS_Process pv on t.ProcessId=pv.Id
```

### 3. Retrieve All Closed Tasks For All Applications

To retrieve a list of all closed tasks for all applications, apply the following query:

```
select i.Id, ray.DecodeML(i.Subject, 'en') InstanceName, CreationDate, i.DoneDate,
i.Duration, i.Status, p.Title AssignedTo, u.Title CompletedBy, t.Applicat
```

```
ionName, ray.DecodeML(t.Subject, 'en') TaskTypeName  
from ray.BPMS_TaskInstanceCompleted i  
left join ray.MercPrincipals p on i.PrincipalId=p.ID  
left join ray.MercPrincipals u on i.UserId=u.ID  
left join ray.BPMS_TaskType t on i.TaskTypeId=t.Id
```

## 12.6.1.2. Static Tab

In the Static tab, users can manually enter the list of options that will be available in the dropdown.

- ✿ This type of dropdown is useful for displaying a fixed set of options that are unlikely to change frequently or depend on other fields in the form. The options are entered directly by the user and will not change unless the user manually updates them.

The screenshot shows the 'Query Editor' window with the 'Static' tab selected. At the top, there are buttons for 'Apply', 'OK', and 'Cancel'. Below these, the 'Static' checkbox is checked. A 'Title' text field is present. The main area is divided into two sections: 'Items' on the left and 'Item' on the right. The 'Items' section contains a large empty list box with 'Up' and 'Down' buttons to its right, and 'Add' and 'Remove' buttons at the bottom. The 'Item' section contains a table with columns for 'Value' and various language captions: English, French, German, Arabic, and Hebrew. Each caption has an associated text input field.

1. **Static:** This is a checkbox which, if selected, will activate the Static tab and redirect users to it, while disabling the [Dynamic](#) tab.
2. **Title:** This is a text field that allows users to input the title of the dropdown data source.
  - A dropdown data source must have a unique name. It cannot have the same name as another dropdown data source in the application.
3. **Add:** This button allows users to add a new dropdown item.
  - Once added, the new item will be displayed in the **Items** box and its properties will be shown in the **Item** section.

The screenshot displays two panels in the DBP Designer. The 'Items' panel on the left features a list box with 'Item 0' selected, and buttons for 'Up', 'Down', 'Add', and 'Remove'. The 'Item' panel on the right contains a table of fields for the selected item:

| Item            |              |
|-----------------|--------------|
| Value           | Item0        |
| English Caption | Item 0       |
| French Caption  | Élément 0    |
| German Caption  | Gegenstand 0 |
| Arabic Caption  | العنصر 0     |
| Hebrew Caption  | 0 יט 0       |

- **Value:** This is a text field that allows users to input the data value for the dropdown item.
  - This value corresponds to a specific item in the dropdown list. When users select the item, the corresponding value is sent to the server, allowing it to be processed in the backend.



It's important to make sure that the value entered matches the item selected and is in the correct format for the data type you are working with (e.g., "true" or "false" for boolean values, or "0" or "1" for integer values). It is recommended to keep the value simple, as it will be used to write business logic if necessary.

- **Caption:** This is a text field that allows users to input the displayed name or caption of the dropdown item.
  - As English is set as the default language in the system, any caption inputted in the English caption field will be used as the display caption for the item, regardless of other language options that may be available.



Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.



Please note that for the translation to be effective on the corresponding control, the form must be saved and closed beforehand.



4. **Remove:** This button allows users to remove a selected dropdown item.
5. **Up / Down:** These buttons allow users to reorder the items in the list.
  - To move an item up or down, select it from the list and click on the **Up** or **Down** button. This will move the selected item one position up or down in the list, respectively.

# 13. Business Logics

---

In the **Business Logics** module, users are provided access to the following elements:



1. [Rules](#)
2. [Actions](#)
3. [Guards](#)
4. [Code Messages](#)

# 13.1. Rules

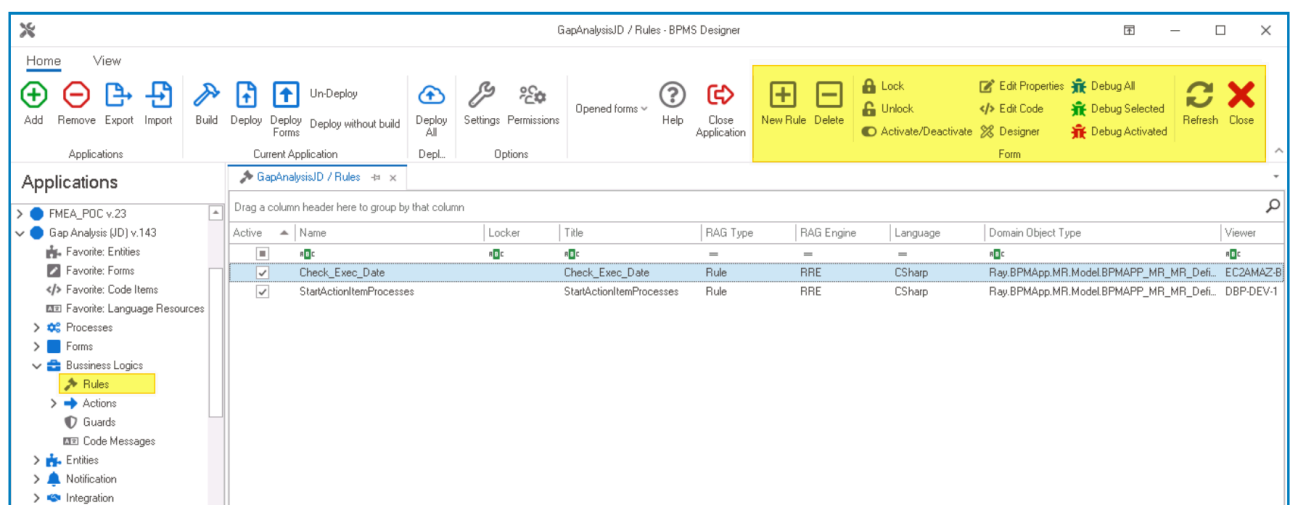
In the **Rules** section of this module, users can create and customize rules for the application.



\* A rule refers to a set of instructions that take specific inputs, apply a defined process, and ultimately yield a desired output. These rules can be defined and executed in real-time as part of a process or subprocess.

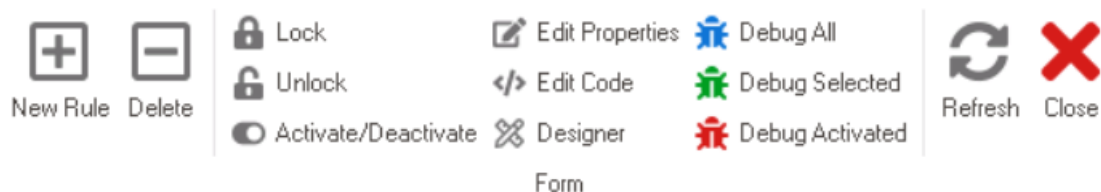
To manage rules in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Business Logics > Rules** (double-click to open). It will display the **Rules** tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined rules in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new rule.

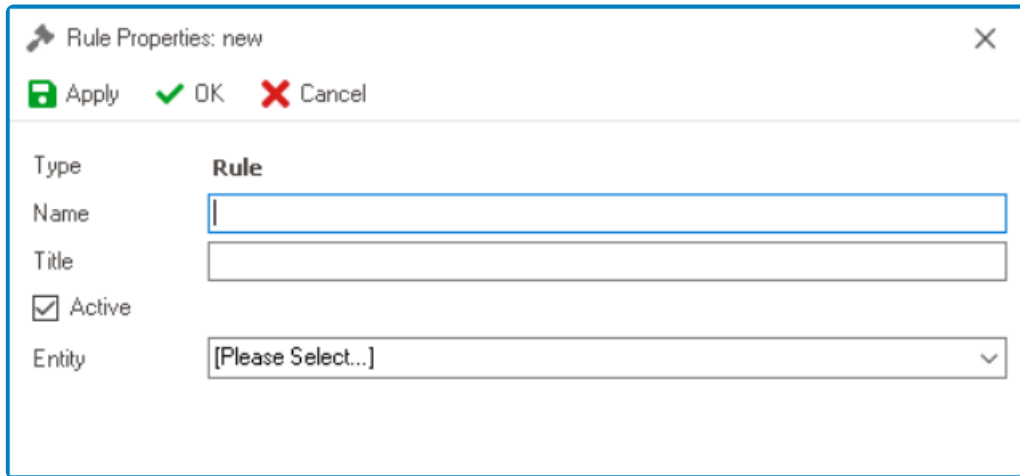
- If selected, it will display the [Rule Properties](#) window.
- **Delete:** This button allows users to delete a rule.
  - To use this function, first select a rule from the list.
- **Lock:** This button allows users to lock the rule, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the rule.
  - If selected, access restriction on the rule will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the rule.
- **Edit Properties:** This button allows users to edit the properties of an existing rule.
  - If selected, it will display the [Rule Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a rule from the list.
- **Edit Code:** This button allows users to launch the [Code Designer](#), in which users can write or edit code specifying the behavior of the rule.
  - [For more details on how to define rules using the Code Designer, click here.](#)
- **Designer:** This button allows users to launch the [Rule Designer](#), in which they can define rules that specify the conditions and actions for the rule.
  - [For more details on how to define rules using the Rule Designer, click here.](#)
- **Debug All:** This button allows users to debug all rules in the list.
- **Debug Selected:** This button allows users to debug only the selected rule in the list.
- **Debug Activated:** This button allows users to debug only the rules that are tagged as active.
- **Refresh :** This button allows users to refresh the [Rules](#) tab.
- **Close:** This button allows users to close the [Rules](#) tab.



By utilizing rules, businesses can automate decision-making processes, enforce regulations, and ensure consistency and accuracy in their operations.

## 13.1.1. Rule Properties

In the [Rule Properties](#) window, users can create a new rule for the application.

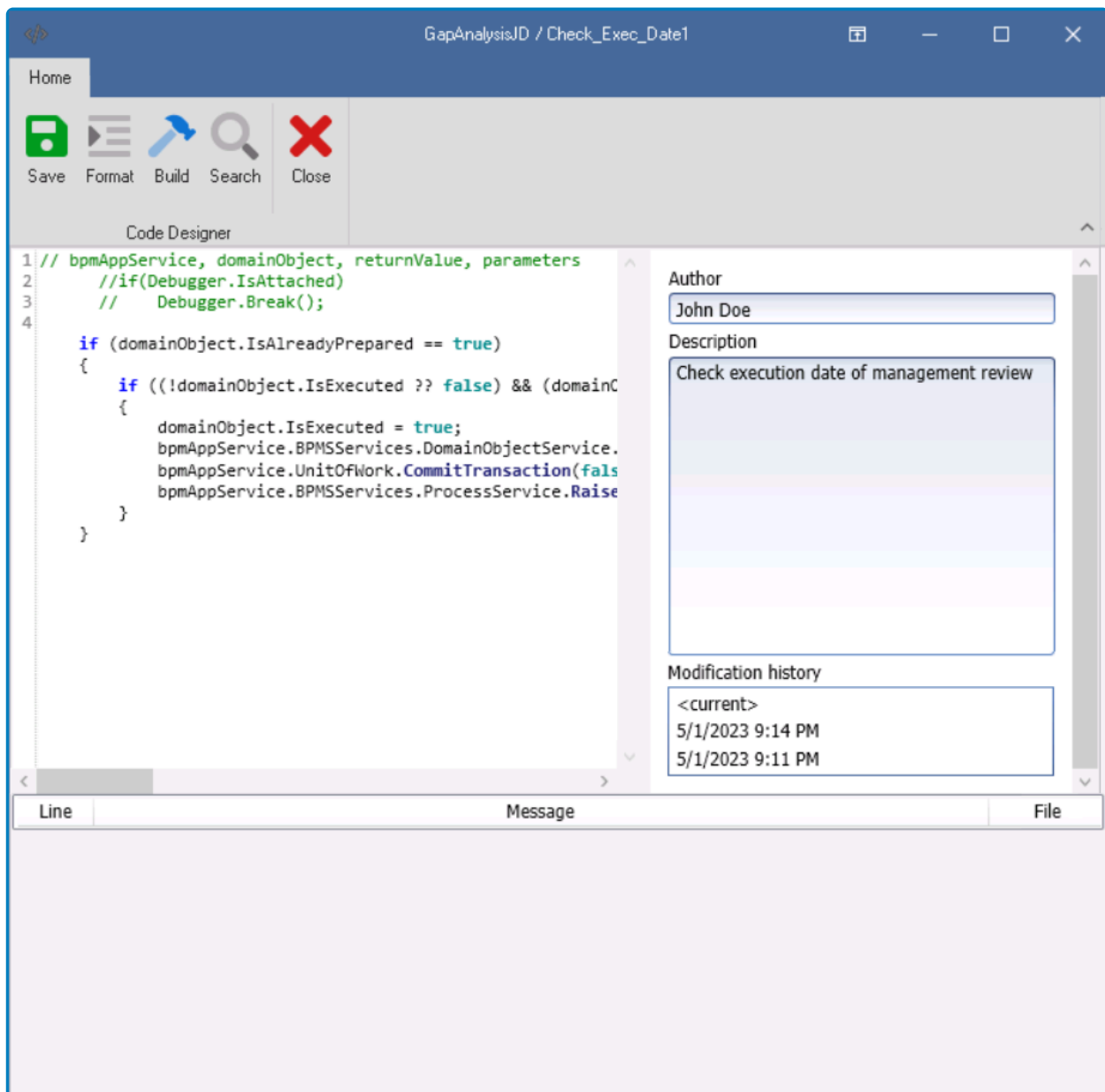


1. **Name:** This is a text field that allows users to input the name of the rule. It is a **mandatory** field.
  - There are the following guidelines determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A rule must have a unique name. It cannot have the same name as another rule in the program.
2. **Title:** This is a text field that allows users to input the title of the rule.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the rule out as active.
4. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the rule.

\* To save changes, click on the **Apply** button. To save changes and exit the [Rule Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 13.1.2. Code Designer

In the Code Designer, users can write and build code specifying the behavior of a given rule, action, or guard. This code is scripted using the C# programming language.



1. **Save:** This button allows users to save any changes they have made to their code.
  - If this button is selected, a validity check will automatically be performed and the results will be displayed in the **Message** field.
2. **Format:** This button allows users to automatically structure and format their code according to certain predefined standards or formatting rules.
  - By clicking on this button, users can quickly apply formatting changes to their code without having to manually adjust indentation, spacing, or other formatting elements.
3. **Build:** This button allows users to compile and build their code.
  - During this process, the code will be checked for errors, and any issues or errors encountered will be reported in the **Message** field. If the validity check is successful and there are no errors, the **Message** field will display a message indicating that the build was successful.

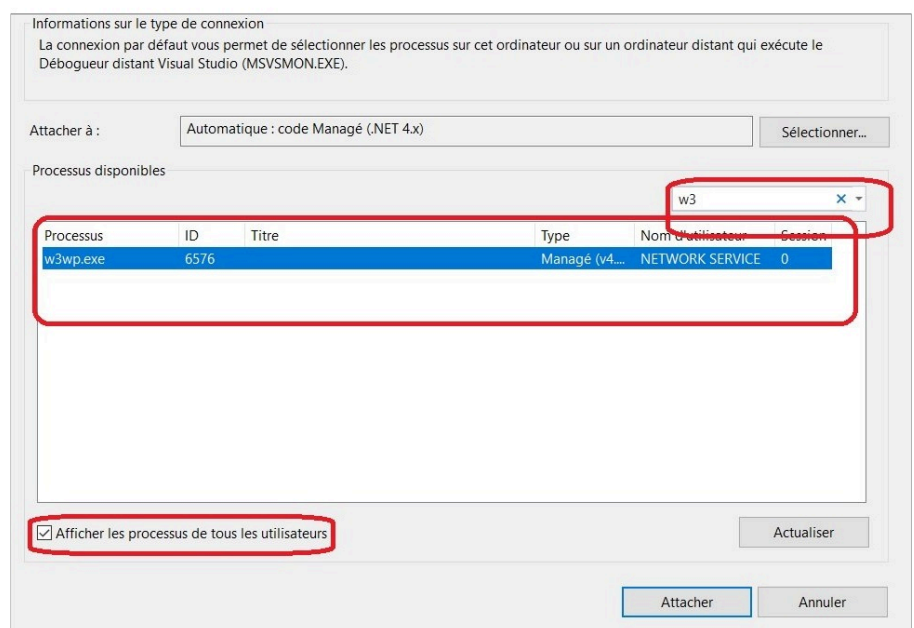
4. **Search:** This button allows users to search for and find specific keywords, phrases, or pieces of code within their code file. Users can also replace text with a new string or code snippet by entering the search term and the new string or code snippet in the appropriate fields.
- If this button is selected, it will display the [Find](#) window.

\* The search results will be updated to reflect the changes, and users can choose to replace individual instances or all instances of the search term.

5. **Close:** This button allows users to close the [Code Designer](#).
6. **Code Environment:** This is the area where users can write and edit their code.
- In the code environment, there are comments written in the code that suggest which objects or variables can be used to accomplish certain tasks. These comments are intended to help programmers understand how the code works and to make modifications or additions to the code more easily.
    - **Debugging Code with Debugger.Break()**
      - “if (Debugger.IsAttached) — Debugger.Break()” is a line of code that is currently commented out in the code environment. When uncommented, it causes the code to break at that point when the code is run in Visual Studio’s debugger.

\* This can be useful for troubleshooting and identifying errors in the code.

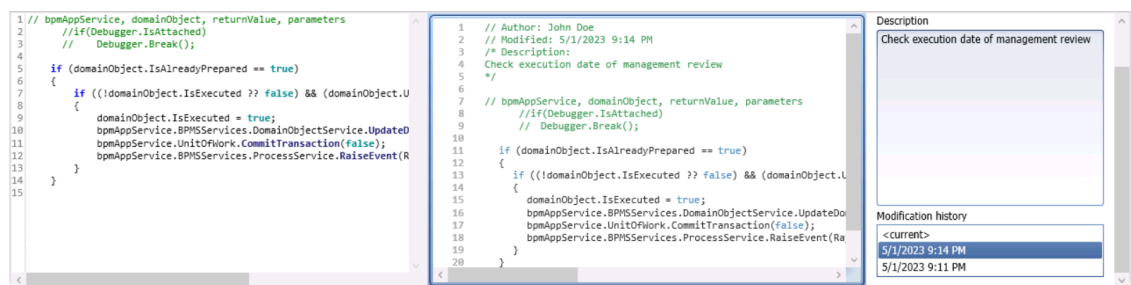
- In order for the code to break at that point when using the Visual Studio debugger, the programmer needs to have Visual Studio open and running, and they need to have attached the debugger to the process that is running the code.



- ✿ Depending on the type of business logic being written (such as a rule, action, or guard), users may have access to different APIs and features to incorporate into their code.

7. **Code Metadata:** In this panel, users can view and manage the metadata for their code, including authorship, descriptions, and modification history.

- **Author:** This field allows users to input their name, which will be displayed as the author of the code modification.
  - Once saved, a user's name remains displayed in the panel until a new name is provided.
- **Description:** This field allows users to input a brief description of the task or code they are working on.
  - Once saved, a user's description remains displayed in the panel until a new description is provided.
- **Modification History:** This field displays a log of all changes made to the code, including the date and time of each modification.
  - When a specific entry is selected in this field, a small panel appears highlighting the section of the code that was modified, along with the name of the author and their corresponding description, date, and time.



8. **Message:** This is an area where messages and errors are displayed after a user performs an action, such as saving or building their code. This field is useful for debugging and troubleshooting code.

- When the user selects the **Save** or **Build** button, the program automatically performs a validity check on the code.
  - If there are errors in the code, they will be displayed in the **Message** field.
  - If the code is valid, a message indicating success will be displayed.

- ✿ For sample scripts and comprehensive demonstrations highlighting the various use cases and applications of BPMS services, [click here](#).



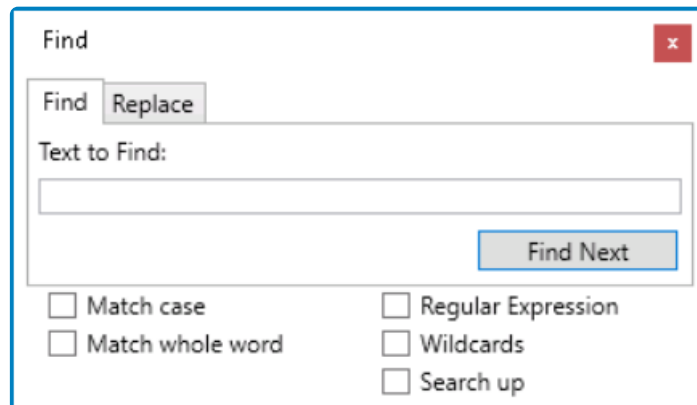
## 13.1.2.1. Find

In the Find window, users can find and replace text within their code file.

There are two tabs to consider in this window: Find and Replace.

### Find Tab

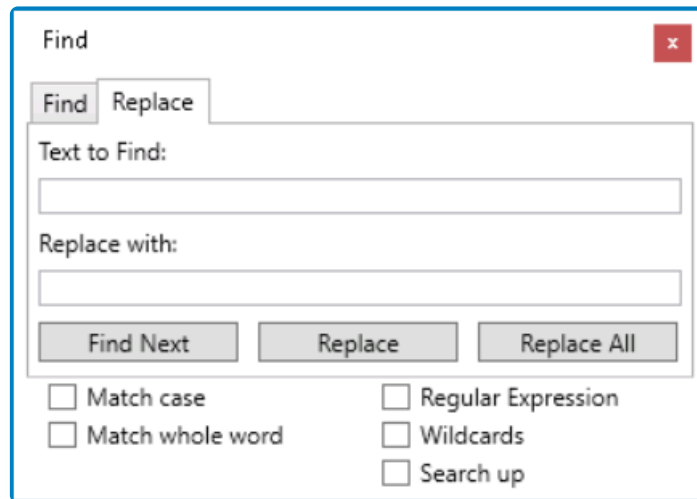
In this tab, users can search for and find specific keywords, phrases, or pieces of code within their code file.



1. **Text to Find:** This is a text field that allows users to enter the search term or keyword that they want to find in their code.
2. **Find Next:** This button allows users to find the next occurrence of the search term in the code.
3. **Match Case:** This is a checkbox which, if selected, allows users to perform a case-sensitive search for the search term.
4. **Match Whole Word:** This is a checkbox which, if selected, allows users to search for the search term as a whole word rather than as part of a larger word.
5. **Regular Expression:** This is a checkbox which, if selected, allows users to use regular expressions in their search term.
6. **Wildcards:** This is a checkbox which, if selected, allows users to use wildcards in their search term.
7. **Search Up:** This is a checkbox which, if selected, allows users to search for the search term in the opposite direction from the current cursor position.

### Replace Tab

In this tab, users can replace specific keywords, phrases, or pieces of code with a new string or code snippet.



1. **Text to Find:** This is a text field that allows users to enter the search term or keyword that they want to find in their code.
2. **Replace With:** This is a text field that allows users to enter the text or code that they want to replace the search term with.
3. **Find Next:** This button allows users to find the next occurrence of the search term in the code.
4. **Replace:** This button allows users to replace the current occurrence of the search term with the text or code entered in the **Replace With** field.
5. **Replace All:** This button allows users to replace all occurrences of the search term with the text or code entered in the **Replace With** field.
6. **Match Case:** This is a checkbox which, if selected, allows users to perform a case-sensitive search for the search term.
7. **Match Whole Word:** This is a checkbox which, if selected, allows users to search for the search term as a whole word rather than as part of a larger word.
8. **Regular Expression:** This is a checkbox which, if selected, allows users to use regular expressions in their search term.
9. **Wildcards:** This is a checkbox which, if selected, allows users to use wildcards in their search term.
10. **Search Up:** This is a checkbox which, if selected, allows users to search for the search term in the opposite direction from the current cursor position.

## 13.1.2.2. Scripts

---

Within this section, you will find a collection of sample scripts designed to enhance your understanding and utilization of BPMS services.

1. [Sample – Dynamically Add Control Validation](#)
2. [Sample – Copy Files from One File Attachment Control to Another](#)
3. [Sample – Store & Retrieve Custom Server Files](#)
4. [Sample – Add Template to Email](#)
5. [Sample – Send Email By Code](#)
6. [Sample – Extract Activities ID from BPMSPProcessInfoString](#)
7. [Sample – Terminate Active Tasks for Specific DomainObject](#)
8. [BPMS Services – Use Cases and Applications](#)

## 13.1.2.2.1. Sample – Dynamically Add Control Validation

---

This sample script demonstrates how to dynamically add and remove a required control (or required field) validator.

### Applicability

This functionality is only available within the control script as it requires access to the `formControl` object.

### Code Sample

```
// Set validator  
formControl.ControlName.SetRequiredFieldValidate("Enter value for Control Name", "group1"); // second parameter is irrelevant for now
```

```
// Clear validator  
formControl.ControlName.ClearRequiredFieldValidate();
```

## 13.1.2.2.2. Sample – Copy Files from One File Attachment Control to Another

---

This sample script demonstrates how to copy files from one file attachment control to another.

### Code Sample

```
var files = bpmAppService.BPMSServices.DmsService.GetFileFromControl(domainObject.Documents);
    foreach(var file in files)
    {
        if(domainObject.BPMAPP_Audit_Mid_PlanExecute_RequirementRequirementAudit.Documents == null)
        {
            domainObject.BPMAPP_Audit_Mid_PlanExecute_RequirementRequirementAudit.Documents = Guid.NewGuid().ToString();
            bpmAppService.BPMSServices.DomainObjectService.SaveObject(domainObject.BPMAPP_Audit_Mid_PlanExecute_RequirementRequirementAudit);
        }
        bpmAppService.BPMSServices.DmsService.AddAttachFile(bpmAppService.BPMSServices.DmsService.GetFileContent(domainObject.Documents, file.Id), file.Name, domainObject.BPMAPP_Audit_Mid_PlanExecute_RequirementRequirementAudit.Documents);
    }
}
```

## 13.1.2.2.3. Sample – Store & Retrieve Custom Server Files

This sample script demonstrates the **Custom Server Files** feature, which allows users to store and retrieve custom files, as well as reference them conveniently using a URL.

✿ By default, custom files are stored in the local folder on the web server, (i.e., c:\inetpub\wwwroot\dbp\TempFiles\xxx).

### Configuration

The storage type and precise location are configured in the web.config file. Some settings are borrowed from the [file attachment storage configuration](#), including AWS configuration.

```
<!--*Possible values: DB, LocalFolder, RemoteFolder, AWS*-->
<add key="faStorageFileFolder" value="FileAttachment" />
<add key="faStorageRemoteFolder" value="\\remoteserver\shared\FileAttachments" />
<add key="faAWSBucket" value="itc-dev-bpms-build-repo" />
<add key="faAWSBucketFolder" value="FileAttachment" />
<add key="faAWSRegion" value="USEast2" />
<add key="faAWSAccessKeyId" value="BlaBlaBla" />
<add key="faAWSSecretAccessKey" value="BlaBlaBlaBla" />
<!--* * * Custom Server Files Storage * * *-->
<add key="faCFStorageType" value="AWS" />
<add key="faCFStorageFileFolder" value="CustomFiles" />
<add key="faCFStorageRemoteFolder" value="\\remoteserver\shared\CustomFiles" />
<add key="faCFAWSBucketFolder" value="CustomFiles" />
```

## 13.1.2.2.4. Sample – Add Template to Email

This sample script demonstrates how to attach a template file to an email.

### Applicability

This functionality is only available within the control script as it requires access to the formControl object.

### Prerequisites

To utilize this feature, users must first upload the template file in the Designer. The file is attached to a specified behavior, allowing users to have a different template per form behavior for the same form.

✿ For more information on how to set up templates per form behavior, [click here](#).

### Code Sample

```
// Prepare attachment
//
byte[] content = bpmAppService.BPMSServices.DmsService.GetAppFileContent("ES
R Template - 2021-04-21.docx", formObject); // the name of the file uploaded
SortedList _attachList = new SortedList();
_attachList.Add(domainObject.BPMAPP_Hatch_PMR_ProjectProjectRef.SAPNo + ".doc
x", content); // this is the name of file in email
//
// Send the email
//
var subject = "The subject";
var body = "The body";
var email = "recipient@site.com";
bpmAppService.BPMSServices.NotificationService.Send(subject, body, Notificatio
nAgent.Email, true, email, _attachList);
```

## 13.1.2.2.5. Sample – Send Email By Code

---

This sample script demonstrates how to send an email.

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

Passing the email address of the starter user:

```
// Set subject, body & recipient
//
var subject = "The subject of the email";
var body = "The body of the email";
var isAsync = true; // Set to true to send the email asynchronously
var recipient = bpmAppService.BPMSServices.MercuryService.StarterUser.Email;
//
// Send the email
//
bpmAppService.BPMSServices.NotificationService.Send(subject, body, NotificationAgent.Email, isAsync, recipient);
```



## 13.1.2.2.6. Sample – Extract Activities ID from BPMSProcessInfoString

This sample script demonstrates how to extract the **BPMS\_Activities ID** from the **BPMSProcessInfoString** field using an SQL query. This can be useful when you need to retrieve the current state of an object within a process (domainObject.CurrentState).

### Applicability

This method is applicable when you need to retrieve an object by its current state.

! If you already know the ID of the object, however, it is recommended to use the **DomainObjectService.GetDomainObject** method instead.

### Code Sample

The **BPMSProcessInfoString** field contains XML content, making it challenging to extract the Activities ID. You can achieve this by using the following query:

To achieve this, use the following query:

```
SELECT          CAST(B.[BPMSProcessInfoString] AS XML).value('/BPMInfos[1]/ProcessInfosList[1]/ProcessInfos[1]/Activities[1]/ActivityInfo[1]/Id[1]', 'NVARCHAR(36)') AS BPMS_ActivitiesId
FROM            ray.BPMAPP_BCM_BCP B
```

To retrieve the **CurrentState** and **CurrentState\_FaTitle**, you can pass the result to the following query:

```
SELECT          Name AS CurrentState, FaTitle AS CurrentState_FaTitle
FROM            ray.BPMS_Activities
WHERE           Id = @BPMS_ActivitiesId
```

## 13.1.2.2.7. Sample – Terminate Active Tasks for Specific DomainObject

---

This sample script demonstrates how to terminate a pending or active task for a specified domain object.

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

```
// Get active tasks from database
//
var id = domainObject.Id.ToString();
var query = String.Format("SELECT * FROM [ray].[BPMS_TaskInstance] WHERE [Stat
us] = 'Active' AND ObjectId = '{0}'", id);
var tasks = bpmAppService.BPMSServices.SqlService.ExecuteSqlQueryFromDB<Ray.BP
MS.Model.TaskInstance>(query);
//
// Terminate all active tasks
//
if (tasks != null && tasks.Count > 0)
{
    foreach (var task in tasks)
    {
        var taskObject = bpmAppService.BPMSServices.DomainObjectService.Ge
tDomainObject<Ray.BPMS.Model.TaskInstance>(task.Id);
        var user = bpmAppService.BPMSServices.MercuryService.GetUser(taskO
bject.StarterUserId);
        bpmAppService.BPMSServices.TaskService.SendToTerminated(domainObje
ct, user);
    }
}
```

## 13.1.2.2.8. BPMS Services – Use Cases and Applications

**BPMS Services** encompass a wide range of functionalities and capabilities that are essential for the smooth operation of the platform. Each service serves a specific purpose and provides valuable features to enhance the overall user experience and enable efficient workflow management.

Some of the key services include:

### 1. ActivityService

| Service Method                    | Service Method Signature   | Purpose | Documentation |
|-----------------------------------|--|---------|---------------|
| <b>GetCurrentStateByProcessId</b> | string <b>GetCurrentStateByProcessId</b> (string currentState, Guid processId) | —       | —             |
| <b>GetCurrentState</b>            | string <b>GetCurrentState</b> (string currentState, Guid processVersionId)     | —       | —             |

### 2. CalendarService

Connection to Outlook and local timesheet

| Service Method                      | Service Method Signature  | Purpose                      | Documentation |
|-------------------------------------|---|------------------------------|---------------|
| <b>CreateEvent</b>                  | ServiceResult <b>CreateEvent</b> (string subject, string body, DateTime date, TimeSpan start, TimeSpan end, List<Guid> users = null, string room = null, bool allowOverlap = false)                   | Create a meeting/event       | —             |
| <b>CreateDateRangeFullDayEvents</b> | ServiceResult <b>CreateDateRangeFullDayEvents</b> (string subject, string body, DateTime startDate, DateTime endDateExcluded, List<Guid> users = null, string room = null, bool allowOverlap = false) | —                            | —             |
| <b>CheckAvailability</b>            | List<EventView> <b>CheckAvailability</b> (DateTime date, TimeSpan start, TimeSpan end, List<Guid> users = null, string room = null)   | Check if users are available | —             |
| <b>CheckAvailability</b>            | List<EventView> <b>CheckAvailability</b> (DateTime startDate, DateTime endDate, List<Guid> users = null, string room = null)  | Check if users are available | —             |

|                                 |  |                             |   |
|---------------------------------|--|-----------------------------|---|
| <b>GetSuggestedMeetingTimes</b> | List<EventView><br><b>GetSuggestedMeetingTimes</b><br>(DateTime date, int duration, List<Guid> users = null, string room = null) | Get suggested meeting times | — |
| <b>DeleteEvents</b>             | ServiceResult <b>DeleteEvents</b> (string meetingIds)  | Delete meeting              | — |

### 3. DataSourceService

public DataTable ExecuteQuery(BaseTemplate connection, string query)

public string GenerateSelectQuery(DataTable dataTable, List<KeyValuePair<string, string>> schemadeclaration)

### 4. DelegationService

### 5. DmsService

Access to file storages, file attachment controls

| Service Method                   | Service Method Signature  | Purpose                               | Documentation                 |
|----------------------------------|---|---------------------------------------|-------------------------------|
| <b>GetFileFromControl</b>        | List<DmsFile> <b>GetFileFromControl</b><br>(string groupId, bool isWithContent = true)                  | Get file from file attachment control | <a href="#">Documentation</a> |
| <b>GetFileContent</b>            | byte[] <b>GetFileContent</b> (string groupId, Guid fileId)  | —                                     | —                             |
| <b>GetFileStream</b>             | Stream <b>GetFileStream</b> (string groupId, Guid fileId)   | —                                     | —                             |
| <b>SaveAttachFile</b>            | Guid <b>SaveAttachFile</b> (byte[] buffer, string name)   | —                                     | —                             |
| <b>SaveAttachFile</b>            | void <b>SaveAttachFile</b> (byte[] buffer, string name, Guid fileId)                                    | —                                     | —                             |
| <b>AddAttachFile</b>             | Guid <b>AddAttachFile</b> (byte[] buffer, string name, string groupId)                                  | —                                     | —                             |
| <b>SaveLocalFileAsAttachment</b> | Guid <b>SaveLocalFileAsAttachment</b><br>(string filePath)  | —                                     | —                             |
| <b>GetWordFormat</b>             | string <b>GetWordFormat</b> (string fileName, string appName, bool isPdf = false, bool isPrint = false) | —                                     | —                             |
| <b>GetAppFileContent</b>         | byte[] <b>GetAppFileContent</b> (string fileName, IFormObject formObject)                               | —                                     | —                             |

|                                |  |   |   |
|--------------------------------|--|---|---|
| <b>CustomFilesSaveFile</b>     | void <b>CustomFilesSaveFile</b> (string folder, string filename, byte[] filedata)            | — | — |
| <b>CustomFilesSaveFile</b>     | void <b>CustomFilesSaveFile</b> (string filename, byte[] filedata, params string[] paths)    | — | — |
| <b>CustomFilesSaveFile</b>     | void <b>CustomFilesSaveFile</b> (string folder, string filename, Stream inputStream)         | — | — |
| <b>CustomFilesSaveFile</b>     | void <b>CustomFilesSaveFile</b> (string filename, Stream inputStream, params string[] paths) | — | — |
| <b>CustomFilesDeleteFolder</b> | void <b>CustomFilesDeleteFolder</b> (string folder)  | — | — |
| <b>ParseWord</b>               | List<string Header, string Section> <b>ParseWord</b> (DmsFile file)                          | — | — |
| <b>ZipFolder</b>               | void <b>ZipFolder</b> (string folderPath, string zipFullFileName)                            | — | — |
| <b>UnzipFile</b>               | void <b>UnzipFile</b> (string zipFilePath, string unzipFolderPath)                           | — | — |
| <b>GetFlatFileList</b>         | List<FileInfo> <b>GetFlatFileList</b> (string path)  | — | — |

## 6. DomainObjectService

Manipulating with entity objects (Domain Objects – DO)

| Service Method            | Service Method Signature  | Purpose                                       | Documentation                 |
|---------------------------|---|---|-------------------------------|
| <b>GetDomainObject</b>    | T <b>GetDomainObject</b> <T>(Guid objectId)                                 | Get domain object by type and ID              | —                             |
| <b>CreateDomainObject</b> | T <b>CreateDomainObject</b> <T>()   | Create a new domain object                    | <a href="#">Documentation</a> |
| <b>CreateDomainObject</b> | DomainObjectBase <b>CreateDomainObject</b> (Type domainObjectType)          | Create a new domain object                    | —                             |
| <b>CreateDomainObject</b> | DomainObjectBase <b>CreateDomainObject</b> (Type domainObjectType, Guid id) | Create a new domain object with predefined ID | —                             |
| <b>DeleteDomainObject</b> | void <b>DeleteDomainObject</b> (DomainObjectBase domainObject)              | Delete domain object                          | —                             |

|                                |  |  |   |
|--------------------------------|--|--|---|
| <b>SaveDomainObject</b>        | void <b>SaveDomainObject</b><br>(DomainObjectBase<br>domainObject)                             | —  | — |
| <b>SaveDomainObject</b>        | void <b>SaveDomainObject</b><br>(DomainObjectBase<br>domainObject, Guid Id)                    | —  | — |
| <b>UpdateDomainObject</b>      | void <b>UpdateDomainObject</b><br>(DomainObjectBase<br>domainObject)                           | —  | — |
| <b>GetAll</b>                  | IList<T> <b>GetAll</b> <T>()   | Get all domain objects<br>of a specific type                           | — |
| <b>GetFirst</b>                | T <b>GetFirst</b> <T>()  | Get the first domain<br>object of a specific type                      | — |
| <b>GetEvents</b>               | PossibleEvent <b>GetEvents</b><br>(DomainObjectBase<br>domainObject)                           | Get list of events<br>available at the current<br>state of the process | — |
| <b>GetDomainObjectState</b>    | string <b>GetDomainObjectState</b><br><T>(Guid objectId)                                       | Get the current object<br>state  | — |
| <b>GetDomainObjectTracking</b> | string <b>GetDomainObjectTracking</b><br><T>(Guid objectId)                                    | Get object process<br>tracking number                                  | — |
| <b>ReadFirstSheetOfExcel</b>   | DataTable<br><b>ReadFirstSheetOfExcel</b> (byte[]<br>data, int indexIncrement = 0)             | —  | — |
| <b>ReadOfExcel</b>             | DataTable <b>ReadOfExcel</b> (byte[]<br>data, string sheetName, int<br>indexIncrement = 0)     | —  | — |
| <b>ReadOfExcel</b>             | DataTable <b>ReadOfExcel</b> (string<br>fileName, string sheetName, int<br>indexIncrement = 0) | —  | — |

## 7. EPCService

### Connection to EPC

| Service Method        | Service Method Signature   | Purpose                      | Documentation                 |
|-----------------------|--|------------------------------|-------------------------------|
| <b>GetAccessToken</b> | string <b>GetAccessToken</b> (string baseUrl, string<br>userName, string password) <sup>obsolete</sup>               | Get EPC<br>access token      | —                             |
| <b>Login</b>          | bool <b>Login</b> (string baseUrl, string userName, string<br>password, out List<string> errors) <sup>new 14.1</sup> | Log in to EPC<br>using local | <a href="#">Documentation</a> |

|                      |  |   |                               |
|----------------------|--|---|-------------------------------|
|                      |  | settings                                  |                               |
| <b>Login</b>         | bool <b>Login</b> (out List<string> errors) <sup>new 14.1</sup>  | Log in to EPC using global settings       | <a href="#">Documentation</a> |
| <b>EPCGlobalURL</b>  | string <b>EPCGlobalURL</b> {get;} <sup>new 14.1</sup>  | Get <b>Base URL</b> from global settings  | —                             |
| <b>EPCGlobalUser</b> | string <b>EPCGlobalUser</b> {get;} <sup>new 14.1</sup>   | Get <b>Username</b> from global settings  | —                             |
| <b>GetContent</b>    | string <b>GetContent</b> (string baseUrl, string envId, string accessToken)  | Get raw content from EPC call             | —                             |
| <b>GetObjects</b>    | List<EPCObject> *GetObjects*(string api, Dictionary<string, object> parameters = null, string language = "en") <sup>new 14.1</sup> | Get a list of objects (environments etc.) | <a href="#">Documentation</a> |
| <b>GetUsers</b>      | List<EPCUser> <b>GetUsers</b> (string language = "en") <sup>new 14.1</sup>   | Get a list of users                       | <a href="#">Documentation</a> |
| <b>ActionCreate</b>  | bool <b>ActionCreate</b> (EPCAction action, Guid envId) <sup>new 14.1</sup>  | Create an Action item                     | —                             |
| <b>ActionDelete</b>  | bool <b>ActionDelete</b> (string actionId) <sup>new 14.1</sup>   | Delete an Action item                     | —                             |

## 8. **FolderService**

## 9. **FormService**

Works with forms, form behaviors, form controls

| Service Method              | Service Method Signature   | Purpose   | Documentation |
|-----------------------------|--|---|---------------|
| <b>GetFormPrincipals</b>    | List<FormPrincipal><br><b>GetFormPrincipals</b> (Guid formBehaviourId)           | Get a list of the principals who have access to the form behavior | —             |
| <b>AddPrincipalsToForm</b>  | void <b>AddPrincipalsToForm</b> (Guid formBehaviourId, List<string> principals)  | Make form behavior accessible for given principals                | —             |
| <b>RemoveFormPrincipals</b> | void <b>RemoveFormPrincipals</b> (Guid formBehaviourId, List<string> principals) | Remove principals from form behavior assignees                    | —             |

|                                |   |  |                               |
|--------------------------------|---|--|-------------------------------|
| <b>RemoveAllFormPrincipals</b> | void <b>RemoveAllFormPrincipals</b> (Guid formBehaviourId)  | Remove all principals from form behavior assignees           | —                             |
| <b>SaveFormProperties</b>      | void <b>SaveFormProperties</b> (Guid formBehaviourId, string title, Guid? category, int? order, List<Principal> principals) | Change form behavior properties                              | —                             |
| <b>GetTemplates</b>            | List<(string language, Guid id, string filename, byte[] content)><br><b>GetTemplates</b> (Guid formBehaviourId)             | Get a list of export templates attached to the form behavior | —                             |
| <b>SetTemplate</b>             | void <b>SetTemplate</b> (Guid formBehaviourId, string language, string filename, byte[] content)                            | Add an export template to the form behavior                  | —                             |
| <b>SetFilter</b>               | void <b>SetFilter</b> (string sourceControlId, List<ClauseItem> filters) <sup>new 14.0</sup>                                | Set a filter for a grid on a form like search panel does     | <a href="#">Documentation</a> |
| <b>SetFilter</b>               | void <b>SetFilter</b> (IControl control, List<ClauseItem> filters) <sup>new 14.0</sup>                                      | Set a filter for a grid on a form like search panel does     | <a href="#">Documentation</a> |
| <b>ClearFilter</b>             | void <b>ClearFilter</b> (string sourceControlId) <sup>new 14.0</sup>  | Clear a grid filter  | <a href="#">Documentation</a> |
| <b>ClearFilter</b>             | void <b>ClearFilter</b> (IControl control) <sup>new 14.0</sup>  | Clear a grid filter  | <a href="#">Documentation</a> |

10. **HQLService**11. **LanguageService**12. **LogService**13. **MercuryService**14. **MessageService**15. **MiscService**16. **NotificationService**17. **ProcessService**[Service Method – RaiseEvent](#)



18. **PushNotificationService**

19. **RuleService**

20. **SQLService**

[Service Method – ExecuteSQLQuery](#)

21. **SystemService**

22. **TaskService**

23. **TimeEventService**

24. **TimesheetService**

25. **TrackingService**

26. **TranslationService**

27. **WebConnectorService**

## 13.1.2.2.8.1. Method – GetFileFromControl (DmsService)

---

This sample script demonstrates how to retrieve files from the file attachment control. The input parameter is the value from the attribute attached to the file attachment control. When “isWithContent” is set to false, it returns only information about the files without loading the file content. This approach avoids impacting the web server’s memory and CPU usage when dealing with large data.

### Syntax

```
List<DmsFile> GetFileFromControl(string groupId, bool isWithContent = true)
```

```
public class DmsFile
{
    public Guid Id { get; set; }
    public string Name { get; set; }
    public string FileContent { get; set; }
    public string FileType { get; set; }
    public Guid FolderId { get; set; }
    public bool IsLockedForCurrentUser { get; set; }
    public string Creator { get; set; }
    public string FilePath { get; set; }
    public int Size { get; set; }
    public short VersionNo { get; set; }
    public FileChangeMode ChangeMode { get; set; }
    public DateTime CreateDate { get; set; }
    public string Description { get; set; }
    public bool IsPublish { get; set; }
}
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Samples

```
var file = bpmAppService.BPMSServices.DmsService.GetFileFromControl(domainObject.ExcelFile)[0];
var table = bpmAppService.BPMSServices.DomainObjectService.ReadOfExcel(Convert.FromBase64String(file.FileContent), "Sheet1");
```

```
var res = new List<string>();
```

```
if (!string.IsNullOrEmpty(domainObject.VendorManualMultiple))
{
    var files = bpmAppService.BPMSServices.DmsService.GetFileFromControl(domainObject.VendorManualMultiple, false);
    foreach (var f in files)
        res.Add("'" + f.Name + "'");
}
var pars = string.Join(" ", res);
parameters.Add("q", pars);
```

```
if (domainObject.IsVendorManualMultiple.HasValue && domainObject.IsVendorManualMultiple.Value)
{
    var basePath = System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPath("~/TempFiles"), "CustomFiles", domainObject.Id.ToString());

    if (System.IO.Directory.Exists(basePath))
        System.IO.Directory.Delete(basePath, true);
    System.IO.Directory.CreateDirectory(basePath);
    if (!string.IsNullOrEmpty(domainObject.VendorManualMultiple))
    {
        var files = bpmAppService.BPMSServices.DmsService.GetFileFromControl(domainObject.VendorManualMultiple);
        foreach (var f in files)
        {
            var fname = System.IO.Path.Combine(basePath, f.Name);

            if (System.IO.File.Exists(fname))
                System.IO.File.Delete(fname);
            System.IO.File.WriteAllBytes(fname, System.Convert.FromBase64String(f.FileContent));
        }
    }
}
```

## 13.1.2.2.8.2. Method – CreateDomainObject (DomainObjectService)

---

This sample script demonstrates how to create a new domain object.

### Syntax

```
bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject<T>()
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

```
var a = bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject<Ray.BPMApp.ACAN.Model.BPMAPP_ACAN_ProductionPermitAuthorization>();  
a.UserAuthorizer = bpmAppService.BPMSServices.MercuryService.CurrentUser;  
a.Date = DateTime.Now;  
a.IsDeleted = false;  
bpmAppService.BPMSServices.DomainObjectService.SaveDomainObject(a);  
bpmAppService.UnitOfWork.CommitTransaction();
```



If you use the **CreateDomainObject** method with **CommitTransaction** in pre-form actions, use **CommitTransaction(false)** to keep the NHibernate session open.

## 13.1.2.2.8.3. Method – Login (EPCTestService)

This sample script demonstrates how to log in to EPC using local or global settings. It returns a boolean value indicating whether the login was successful or not. An additional method returns a list of errors if some settings are not valid.

The **Login()** method shares the same EPC session for all calls from all users on the server.

### Syntax

```
// Login to EPC using local settings
bool Login(string baseUrl, string userName, string password, out List<string> errors)
// Login to EPC using global settings
bool Login(out List<string> errors)
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Samples

```
var sb = new StringBuilder();
sb.AppendLine("Connect to EPC using local settings");
var settings = bpmAppService.BPMSServices.DomainObjectService.GetFirst<Ray.BPMApp.Playground.Model.BPMAPP_Playground_EPCSettings>();
if (settings == null)
    sb.AppendLine("EPC settings do not exist");
sb.AppendLine(string.Format("EPC URL: 0 as 1", settings.URL, settings.UserName));
var errors = new List<string>();
if (bpmAppService.BPMSServices.EPCService.Login(settings.URL, settings.UserName, settings.Password, out errors))
    sb.AppendLine("Successfully!");
else
{
    sb.AppendLine("Login failed!");
    foreach (var e in errors)
        sb.AppendLine(e);
}
formObject.Log = sb.ToString();
```

```
var sb = new System.Text.StringBuilder();
sb.AppendLine("Connect to EPC using global settings");
sb.AppendLine(string.Format("EPC URL: 0 as 1", bpmAppService.BPMSServices.EPCS
```

```
ervice.EPCGlobalURL, bpmAppService.BPMSServices.EPCService.EPCGlobalUser));  
var errors = new List<Istring>();  
if (bpmAppService.BPMSServices.EPCService.Login(out errors))  
    sb.AppendLine("Successfully");  
else  
{  
    sb.AppendLine("Login failed");  
    foreach (var e in errors)  
        sb.AppendLine(e);  
}  
formObject.Log = sb.ToString();
```

## 13.1.2.2.8.4. Method – GetObjects (EPService)

---

This sample script demonstrates how to retrieve a list of all objects (environments, etc.) from EPC.

- **api** – relative API (without server name)
- **parameters** – additional call parameters
- **language** – call language

### Syntax

```
List<IEPCObject> GetObjects(string api, Dictionary<Istring, object> parameters = null, string language = "en")
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

```
var sb = new StringBuilder();
var errors = new List<Istring>();
if (bpmAppService.BPMSServices.EPCService.Login(out errors))
{
    var objects = bpmAppService.BPMSServices.EPCService.GetObjects("api/v1/admin/environments");
    if (objects != null && objects.Any())
        foreach (var e in objects)
            sb.AppendLine(string.Format("Environment id 0 - 1", e.nodeId, e.name));
}
else
    foreach (var e in errors)
        sb.AppendLine(e);
formObject.Log = sb.ToString();
```

## 13.1.2.2.8.5. Method – GetUsers (EPCService)

---

This sample script demonstrates how to retrieve a list of users from EPC.

- **language** – call language

### Syntax

```
List<IEPCUser> GetUsers(string language = "en")
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

```
var sb = new StringBuilder();
var errors = new List<string>();
if (bpmAppService.BPMSServices.EPCService.Login(out errors))
{
    var users = bpmAppService.BPMSServices.EPCService.GetUsers();
    if (users != null)
        foreach (var u in users)
            sb.AppendLine(string.Format("user id 0 - 1", u.princip
alId, u.name));
}
else
    foreach (var e in errors)
        sb.AppendLine(e);
formObject.Log = sb.ToString();
```



## 13.1.2.2.8.6. Method – SetFilter / ClearFilter (FormService)

---

This sample script demonstrates how to set or clear a filter for a grid on a form, similar to how a search panel operates.

- **sourceControlId** – grid control ServerId
- **control** – control from formControls
- **filters** – list of filters (ClauseItem {string Property; ClauseEnum Clause; object Value})

### Syntax

```
void SetFilter(string sourceControlId, List<ClauseItem> filters)
void SetFilter(IControl control, List<ClauseItem> filters)
void ClearFilter(string sourceControlId)
void ClearFilter(IControl control)
```

### Applicability

This functionality is only available within the control script as it requires access to the formControl object.

### Code Samples

```
// set filter
var filters = new List<ClauseItem>();
filters.Add(new ClauseItem("Quantity", ClauseEnum.Equals, 5));
bpmAppService.BPMSServices.FormService.SetFilter(formControl.Lines, filters);
```

```
// clear filter
bpmAppService.BPMSServices.FormService.ClearFilter(formControl.Lines);
```

## 13.1.2.2.8.7. Method – RaiseEvent (ProcessService)

---

This sample script demonstrates how to raise an event in system state.

### Syntax

```
bpmAppService.BPMSServices.ProcessService.RaiseEvent (BaseEvent baseEvent, DomainObjectBase domainObject)
bpmAppService.BPMSServices.ProcessService.RaiseEvent (BaseEvent baseEvent, DomainObjectBase domainObject, User user)
```

### Applicability

This functionality is only available within the system state rule.

### Code Sample

```
bpmAppService.BPMSServices.ProcessService.RaiseEvent (Ray.BPMApp.ACAN.Events.BPMAPP_ACAN_ProductionPermit.ListenEvents.AEAFFromMOCReview, domainObject, domainObject.UserReviewer);
```

### Automatic Router:

```
if (condition)
    bpmAppService.BPMSServices.ProcessService.RaiseEvent (Ray.BPMApp.ACAN.Events.BPMAPP_ACAN_ProductionPermit.ListenEvents.AEAToMOCReview, domainObject);

else
    bpmAppService.BPMSServices.ProcessService.RaiseEvent (Ray.BPMApp.ACAN.Events.BPMAPP_ACAN_ProductionPermit.ListenEvents.Submit2, domainObject);
```

## 13.1.2.2.8.8. Method – ExecuteSQLQuery (SQLService)

---

This sample script demonstrates how to execute an SQL query and get a result.

### Syntax

```
IList ExecuteSqlQuery(string query)
IList ExecuteSqlQuery(string query, Dictionary Parameters)
IList ExecuteSqlQueryFromDB(string query)
IList ExecuteSqlQueryFromDB(string query, Dictionary Parameters)
```

### Applicability

This functionality is available anywhere users can write C# code.

### Code Sample

```
var cnt = (int)bpmAppService.BPMSServices.SqlService.ExecuteSqlQuery("select c
ount(*) from ray.BPMAPP_ACAN_Document where groupId = '" + domainObject.GroupI
d.ToString() + "'") [0];
```

```
var q = string.Format("Select Document from ray.BPMAPP_ACAN_DocumentOwners whe
re Owner ='0'", formObject.Owner);
var filteredIDs = bpmAppService.BPMSServices.SqlService.ExecuteSqlQuery(q);
includedId.AddRange(filteredIDs.Cast());
```

```
var fleetList = bpmAppService.BPMSServices.SqlService.ExecuteSqlQueryFromDB(st
ring.Format("select * from ray.BPMAPP_ACAN_ExternalUserPermissionFleet where
[UserPermission] = '0'", profile.Id));
```

```
var seq = bpmAppService.BPMSServices.SqlService.ExecuteSqlQuery("select isnul
l(max(DrawingSeqMajor),29999) + 1 seqno from ray.BPMAPP_ACAN_Document") [0];
formObject.NextMajor = (int)seq;
```

## 13.1.3. Rule Designer

In the Rule Designer, users can define rules that specify the conditions and actions for a given rule, action, or guard. The Rule Designer essentially allows users to define a script without writing code.

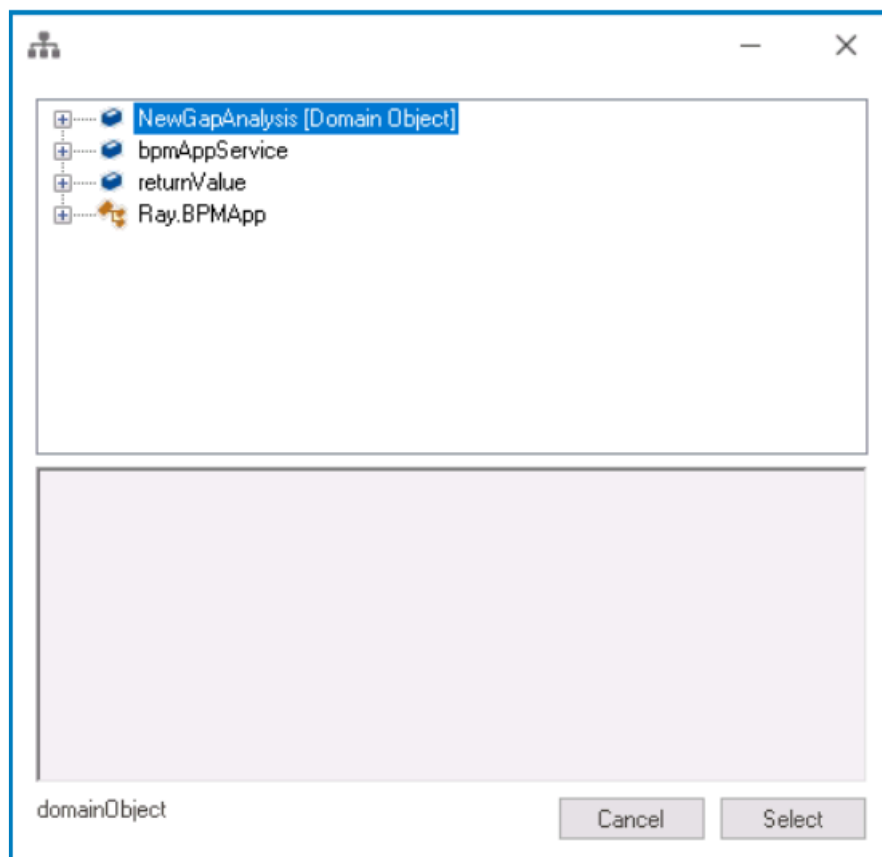
The screenshot shows the 'Rule Designer' interface for a rule named 'Check Execution Date'. The interface includes a toolbar with a checkmark, a green square, and a red X. The main area is divided into three sections: 'Conditions', 'Actions', and 'Errors/Output'. The 'Conditions' section has an 'Active' checkbox and a rule editor with dropdowns and a trash icon. The 'Actions' section has 'Call Method' and 'Set Property' buttons, and an editor showing 'returnValue' assigned to a variable. The 'Errors/Output' section has two tabs, 'Errors' and 'Output', and a large empty text area.

1. **Conditions:** In this section, users can specify the conditions for the rule, action, or guard.
  - **Active:** This is a checkbox which, if selected, allows users to enable the application of conditions for the rule, action, or guard.
    - If this checkbox is selected, it will enable the following fields:

☒ Active

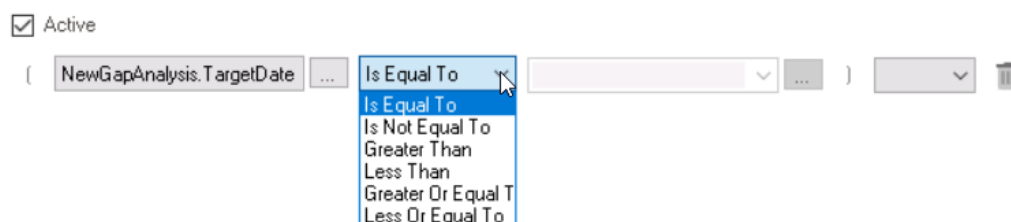
( [ ] ... [ ] ... [ ] ... ) [ ] [ ]

- To add a new value, click on the [ ] button. It will display the following window, in which users can select from a list of available values to use in the condition.



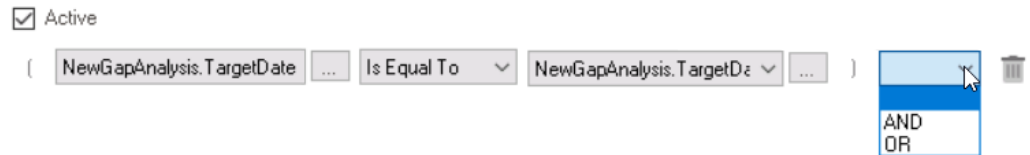
\* These values include attributes of the associated entity or services related to it, depending on the context of the rule, action, or guard. To select a value, either click on the **Select** button or double-click on the entry from the list.

- Between two values in a condition, users must select a comparison operator (such as “Is Equal To”, “Is Not Equal To”, “Contains”, or other available operators) to specify the relationship between the values.



\* Depending on the first value selected, the list of comparison operators displayed in the dropdown may vary.

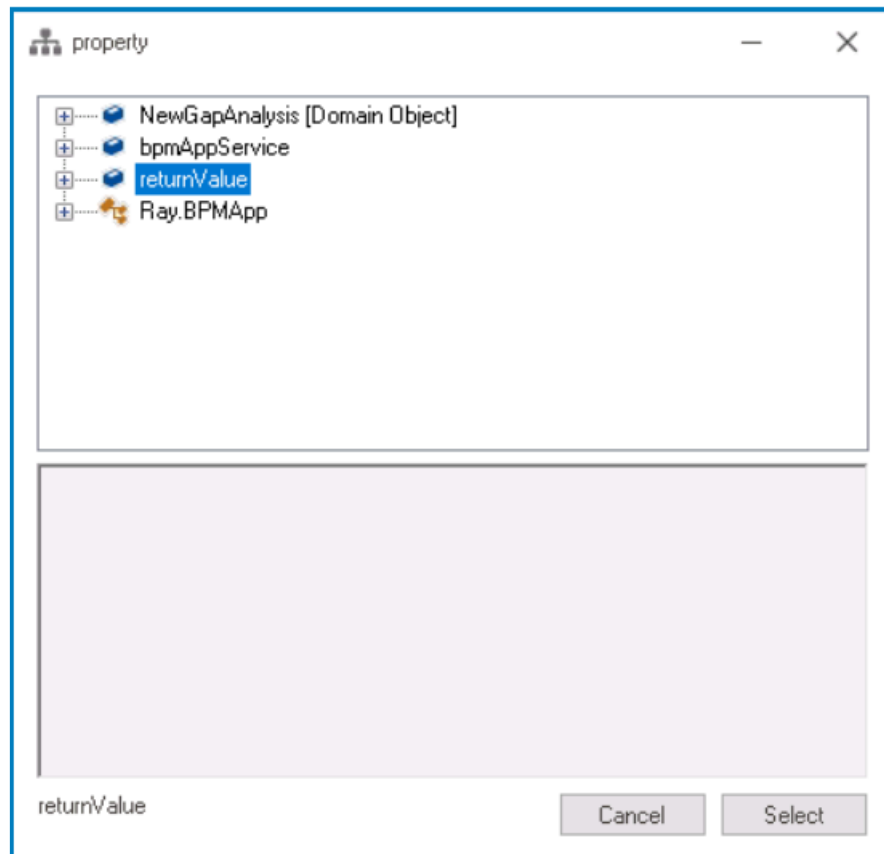
- Once a condition is defined, users can add another condition and specify the relationship between them using logical operators such as “AND” or “OR”.



- To delete a condition, click on the button.

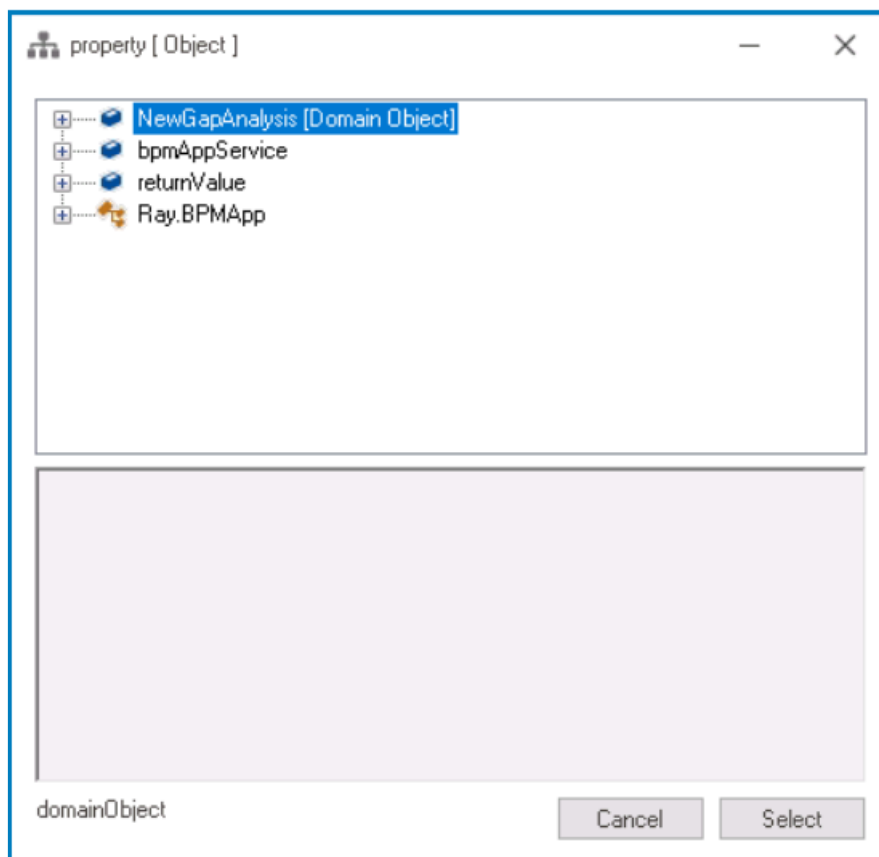
2. **Actions:** In this section, users can specify the actions for the rule, action, or guard.

- To add a new property, click on the button. It will display the following window, in which users can select from a list of available values to use in the action.



✿ To select a property, either click on the **Select** button or double-click on the entry from the list.

- For each property value, users must select a corresponding property object. To select a property object, click on the second button. It will display the following window, in which users can select from a list of available values to use in the action.



✿ To select a property object, either click on the **Select** button or double-click on the entry from the list.

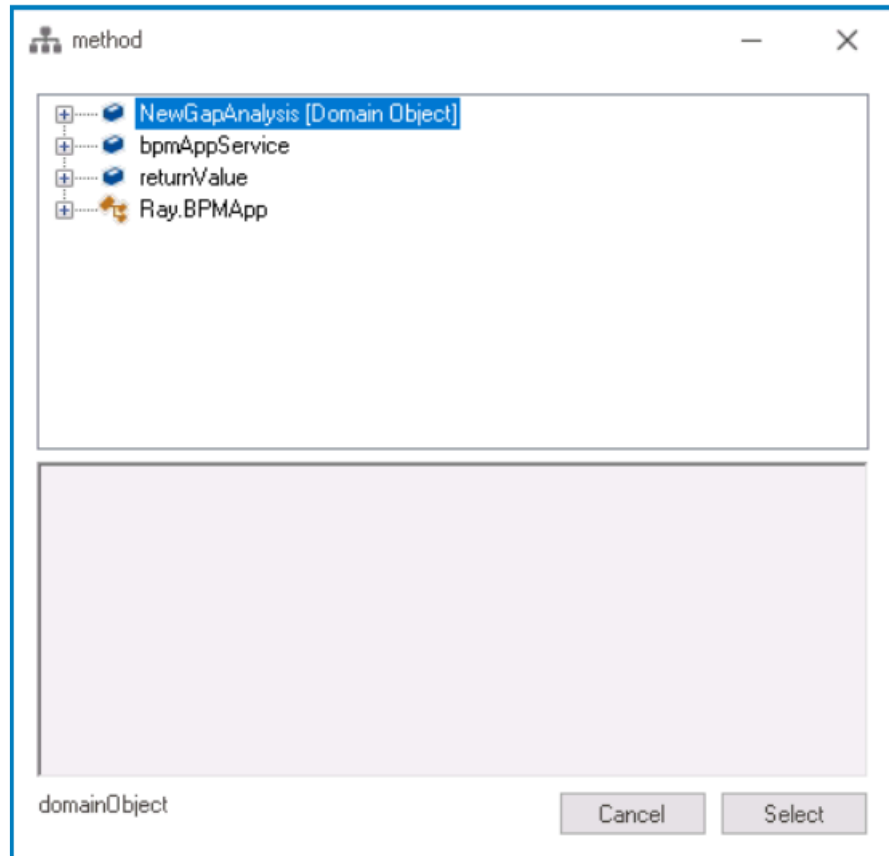
- To set a new action, click on the **Set Property** button.
  - It will display a new set of fields for specifying a property value and its corresponding object.
- To execute a method on an action, click on the **Call Method** button.
  - It will display the following set of fields:

Call Method    Set Property




returnValue ▼ ... =    ▼ ...

▼ ...

- To select a method, click on the ... button. It will display the following window, in which users can select from a list of available methods.



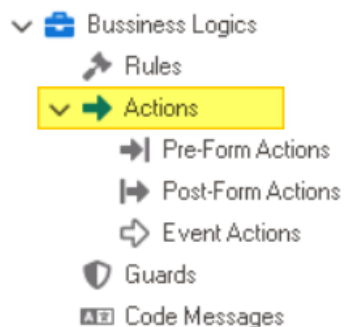
To select a value, either click on the **Select** button or double-click on the entry from the list.

3. **Errors:** When users validate their design, any errors or issues will be displayed in this tab.
  - To check for errors in their design, users must first select the **Validate** button.
4. **Output:** When users validate their design and it is successful, the output of the design will be displayed in this tab.
  - To review the output of their design, users must first select the **Validate** button.
5. **Validate:** This button allows users to validate their design.
  - Icon: 
  - If selected, the system will automatically perform a validity check on the design, and any errors or issues are displayed in the **Errors** tab. If the design is valid, the **Output** tab will display the set conditions and actions.
6. **Save:** This button allows users to save their design.
  - Icon: 
7. **Close:** This button allows users to close the Rule Designer.
  - Icon: 



## 13.2. Actions

In the **Actions** section of this module, users can create and customize actions for the application.



\* An action refers to a specific task that should be performed either before the form is loaded or submitted, after the form is submitted, or during the form submission process.

Users can create three types of actions in the program:

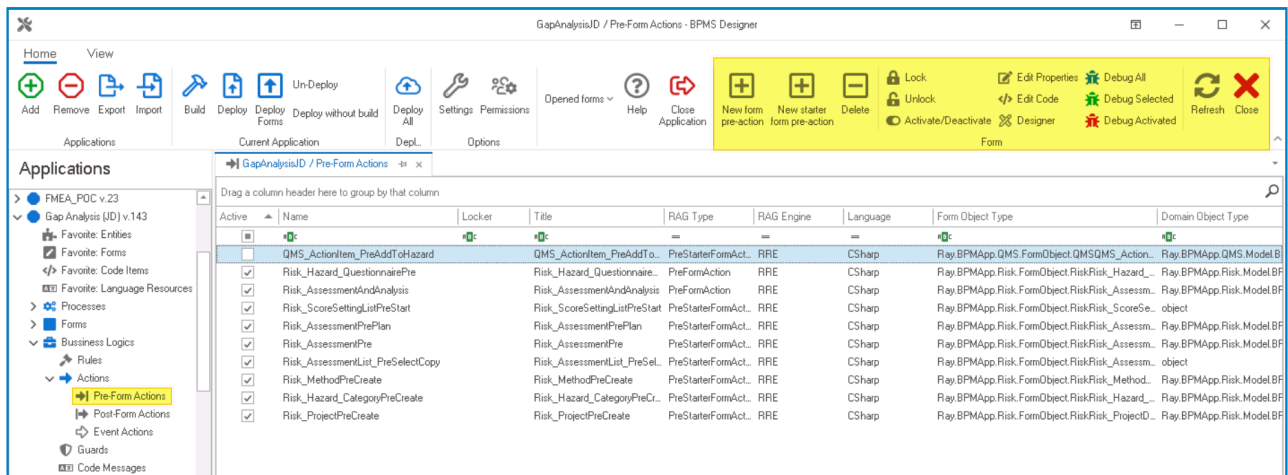
1. [Pre-Form Actions](#)
2. [Post-Form Actions](#)
3. [Event Actions](#)

## 13.2.1. Pre-Form Actions

A pre-form action is an action that takes place before the loading or submission of a form.

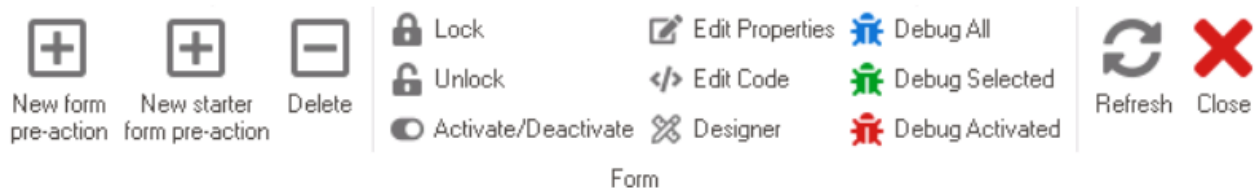
To manage pre-form actions in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Business Logics > Actions > Pre-Form Actions** (double-click to open). It will display the Pre-Form Actions tab and expand the menu bar to show additional options.



In this tab, users can access a list of all defined pre-form actions in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New Form Pre-Action:** This button allows users to define a new pre-form action.
  - If selected, it will display the [Pre-Form Action Properties](#) window.
- **New Starter Form Pre-Action:** This button allows users to define a new pre-form (starter) action.
  - If selected, it will display the [Pre-Form \(Starter\) Action Properties](#) window. This window is largely a duplicate of the [Pre-Form Action Properties](#) window, with the exception being that it allows users to create actions specifically aimed at starter forms.

A starter form is a form that is used to initiate a new process or workflow.

- **Delete:** This button allows users to delete a pre-form action.
  - To use this function, first select a pre-form action from the list.
- **Lock:** This button allows users to lock the pre-form action, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the pre-form action.
  - If selected, access restriction on the pre-form action will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the pre-form action.
- **Edit Properties:** This button allows users to edit the properties of an existing pre-form action.
  - If selected, it will display the Pre-Form Action Properties window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a pre-form action from the list.
- **Edit Code:** This button allows users to launch the Code Designer, in which users can write or edit code specifying the behavior of the pre-form action.
  - For more details on how to define pre-form actions using the Code Designer, [click here](#).



Among other uses, pre-form actions can be utilized to set up conditional events in a process. For information on how to set up conditional events using code, [click here](#).

- **Designer:** This button allows users to launch the Rule Designer, in which they can define rules that specify the conditions and actions for the pre-form action.
  - For more details on how to define pre-form actions using the Rule Designer, [click here](#).
- **Debug All:** This button allows users to debug all pre-form actions in the list.
- **Debug Selected:** This button allows users to debug only the selected pre-form action in the list.
- **Debug Activated:** This button allows users to debug only the pre-form actions that are tagged as active.
- **Refresh :** This button allows users to refresh the Pre-Form Actions tab.
- **Close:** This button allows users to close the Pre-Form Actions tab.



To execute a pre-form action in the web application, users must link it to the corresponding form behavior. For more information on this matter, [click here](#).

### Pre-Form Action Example

- For example, suppose we have an attribute or field named 'Date', and we want that field to be automatically filled with the current date. To achieve this, we can define an action like the one below:
  - `domainObject.Date = bpmAppService.HelperServices.DateTimeService.CurrentDate;`

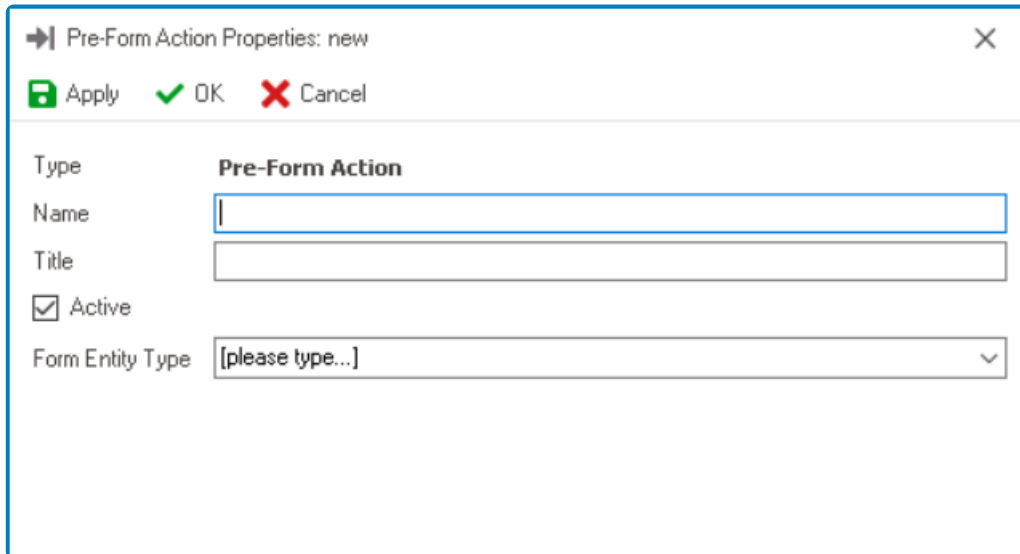


This action sets the value of the 'Date' field to the current date, and it will be

executed automatically before the form is loaded.

## 13.2.1.1. Pre-Form Action Properties

In the [Pre-Form Action Properties](#) window, users can create a new pre-form action for the application.



1. **Name:** This is a text field that allows users to input the name of the pre-form action. It is a **mandatory** field.
  - There are the following guidelines determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A pre-form action must have a unique name. It cannot have the same name as another pre-form action in the program.
2. **Title:** This is a text field that allows users to input the title of the pre-form action.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the pre-form action out as active.
4. **Form Entity Type:** This is a single-select dropdown field that retrieves the list of form entities (from the [List of Forms](#) section of the **Forms** module). It allows users to select the form entity associated with the pre-form action.

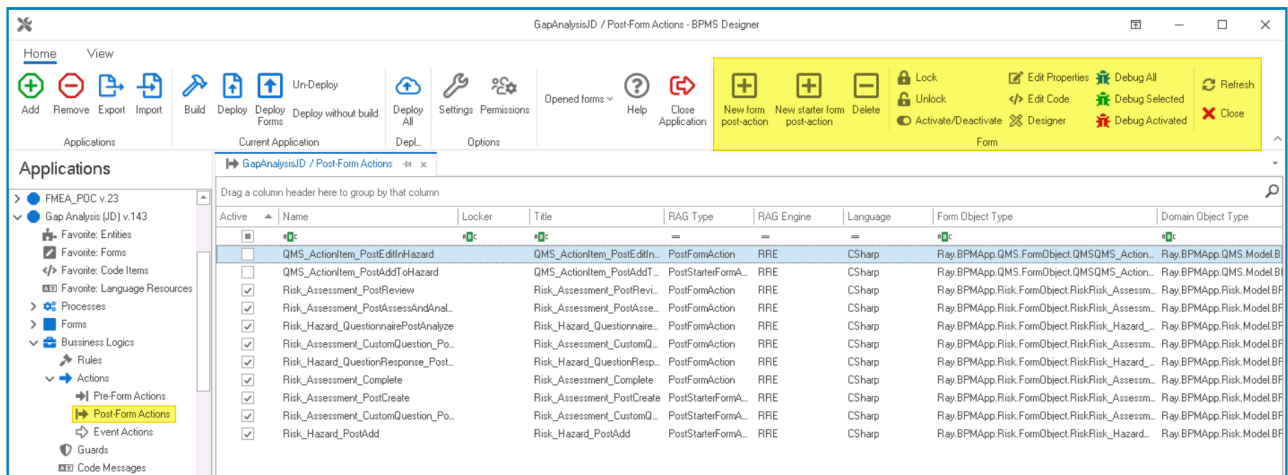
\* To save changes, click on the **Apply** button. To save changes and exit the [Pre-Form Action Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 13.2.2. Post-Form Actions

A post-form action is an action that takes place after the submission of a form.

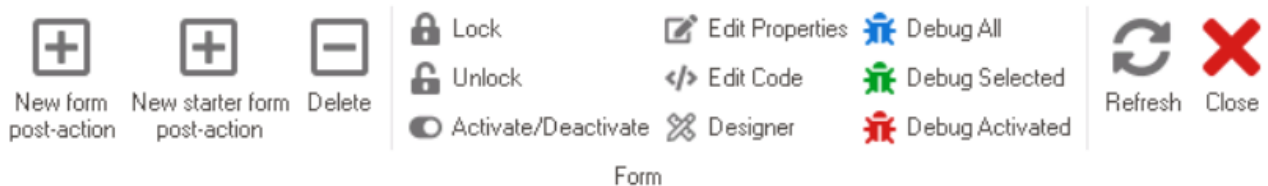
To manage post-form actions in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Business Logics > Actions > Post-Form Actions** (double-click to open). It will display the Post-Form Actions tab and expand the menu bar to show additional options.



In this tab, users can access a list of all defined post-form actions in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New Form Post-Action:** This button allows users to define a new post-form action.
  - If selected, it will display the [Post-Form Action Properties](#) window.
- **New Starter Form Post-Action:** This button allows users to define a new post-form (starter) action.
  - If selected, it will display the [Post-Form \(Starter\) Action Properties](#) window. This window is largely a duplicate of the [Post-Form Action Properties](#) window, with the exception being that it allows users to create actions specifically aimed at starter forms.

A starter form is a form that is used to initiate a new process or workflow.

- **Delete:** This button allows users to delete a post-form action.
  - To use this function, first select a post-form action from the list.
- **Lock:** This button allows users to lock the post-form action, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the post-form action.
  - If selected, access restriction on the post-form action will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the post-form action.
- **Edit Properties:** This button allows users to edit the properties of an existing post-form action.
  - If selected, it will display the Post-Form Action Properties window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a post-form action from the list.
- **Edit Code:** This button allows users to launch the Code Designer, in which users can write or edit code specifying the behavior of the post-form action.
  - [For more details on how to define post-form actions using the Code Designer, click here.](#)
- **Designer:** This button allows users to launch the Rule Designer, in which they can define rules that specify the conditions and actions for the post-form action.
  - [For more details on how to define post-form actions using the Rule Designer, click here.](#)
- **Debug All:** This button allows users to debug all post-form actions in the list.
- **Debug Selected:** This button allows users to debug only the selected post-form action in the list.
- **Debug Activated:** This button allows users to debug only the post-form actions that are tagged as active.
- **Refresh :** This button allows users to refresh the Post-Form Actions tab.
- **Close:** This button allows users to close the Post-Form Actions tab.



To execute a post-form action in the web application, users must link it to the corresponding form behavior. [For more information on this matter, click here.](#)

### Post-Form Action Example

- For example, suppose we have a mandatory field called 'Name of Employee', and we want an error message to be displayed whenever the user tries to submit the form without filling the field. To achieve this, we can define an action like the one below:
  - If (domainObject.NameOfEmployee = null)
 

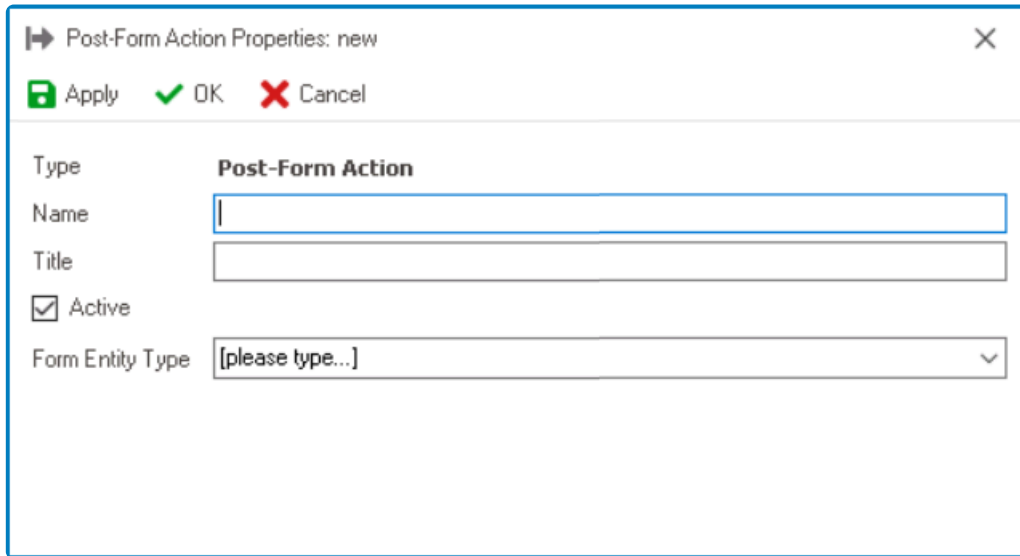
```
{ actionResult.Result = Rayvarz.BPMS.RuleModel.Result.Error; actionResult.Message = "Please select the Name of Employee";
```



This action sets the appropriate message for the 'Name of Employee' field, and it will be executed automatically when the user attempts to submit the form.

## 13.2.2.1. Post-Form Action Properties

In the [Post-Form Action Properties](#) window, users can create a new post-form action for the application.



The screenshot shows a dialog box titled "Post-Form Action Properties: new". At the top, there are three buttons: "Apply" (with a green square icon), "OK" (with a green checkmark icon), and "Cancel" (with a red X icon). Below the buttons, the dialog contains the following fields:

- Type:** A label followed by the text "Post-Form Action".
- Name:** A text input field with a blue border.
- Title:** A text input field.
- Active:** A checkbox that is currently checked.
- Form Entity Type:** A dropdown menu showing "[please type...]" with a downward arrow.

1. **Name:** This is a text field that allows users to input the name of the post-form action. It is a **mandatory** field.
  - There are the following guidelines determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A post-form action must have a unique name. It cannot have the same name as another post-form action in the program.
2. **Title:** This is a text field that allows users to input the title of the post-form action.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the post-form action out as active.
4. **Form Entity Type:** This is a single-select dropdown field that retrieves the list of form entities (from the [List of Forms](#) section of the **Forms** module). It allows users to select the form entity associated with the post-form action.

\* To save changes, click on the **Apply** button. To save changes and exit the [Pre-Form Action Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.



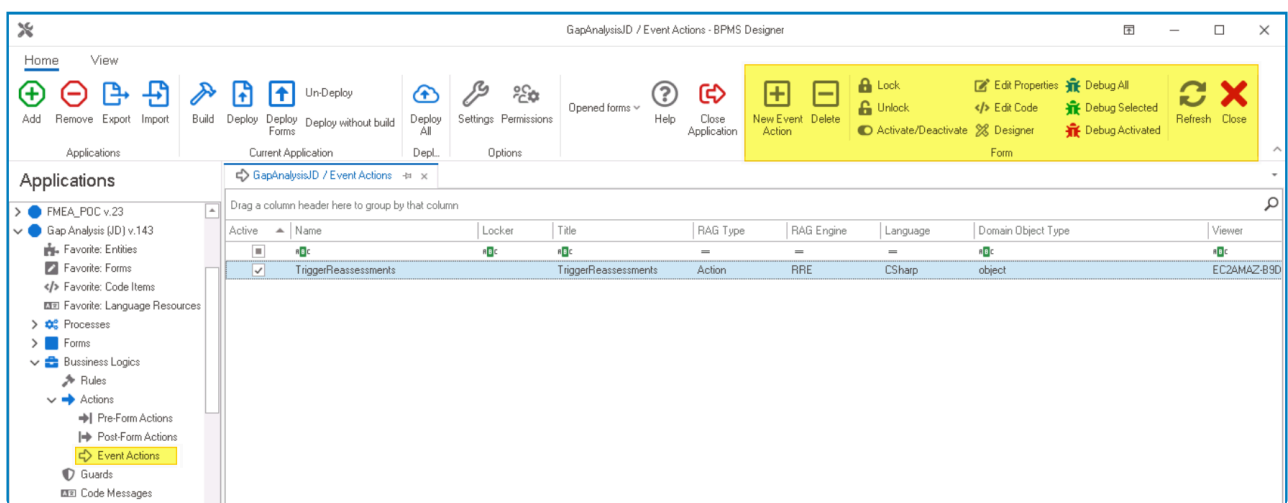
## 13.2.3. Event Actions

An event action is an action that takes place simultaneously with the event.

- \* An event is essentially a type of action or trigger that occurs during the execution of a process. Although it can affect the process flow and be affected by it, an event remains independent of the process. For more information on how events function in a process, [click here](#).

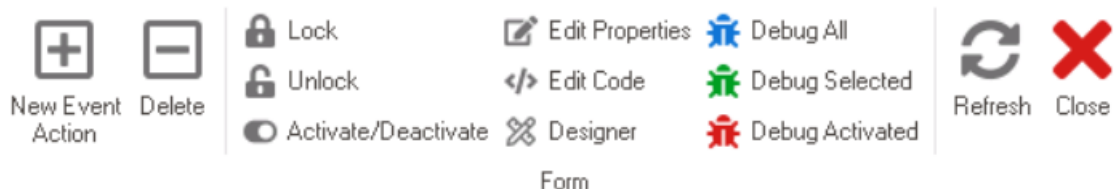
To manage event actions in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Business Logics > Actions > Event Actions** (double-click to open). It will display the Event Actions tab and expand the menu bar to show additional options.



- \* In this tab, users can access a list of all defined event actions in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New Event Action:** This button allows users to define a new event action.
  - If selected, it will display the [Event Action Properties](#) window.
- **Delete:** This button allows users to delete an event action.
  - To use this function, first select an event action from the list.
- **Lock:** This button allows users to lock the event action, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the event action.
  - If selected, access restriction on the event action will be lifted.

- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the event action.
- **Edit Properties:** This button allows users to edit the properties of an existing event action.
  - If selected, it will display the Event Action Properties window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select an event action from the list.
- **Edit Code:** This button allows users to launch the Code Designer, in which users can write or edit code specifying the behavior of the event action.
  - [For more details on how to define event actions using the Code Designer, click here.](#)
- **Designer:** This button allows users to launch the Rule Designer, in which they can define rules that specify the conditions and actions for the event action.
  - [For more details on how to define event actions using the Rule Designer, click here.](#)
- **Debug All:** This button allows users to debug all event actions in the list.
- **Debug Selected:** This button allows users to debug only the selected event action in the list.
- **Debug Activated:** This button allows users to debug only the event actions that are tagged as active.
- **Refresh :** This button allows users to refresh the Event Actions tab.
- **Close:** This button allows users to close the Event Actions tab.



Users can also create and link an event action directly within the properties of the corresponding event. [For more information on this matter, click here.](#)

## 13.2.3.1. Event Action Properties

In the [Event Action Properties](#) window, users can create a new event action for the application.

Action Properties: new

Apply OK Cancel

Type **Action**

Name

Title

☒ Active

Entity

Scope

Drag a column header here to group by that column

| Title                | Name              | Access Name                   |
|----------------------|-------------------|-------------------------------|
| Approve              | Approve           | Ray.BPMApp.GapAnalysisJD.S... |
| Submit               | Submit            | Ray.BPMApp.GapAnalysisJD.S... |
| Send for Review      | SendforReview     | Ray.BPMApp.GapAnalysisJD.S... |
| Send Back for Review | SendBackforReview | Ray.BPMApp.GapAnalysisJD.S... |
| Reject               | Reject            | Ray.BPMApp.GapAnalysisJD.S... |
| Submit               | Submit            | Ray.BPMApp.GapAnalysisJD.S... |

Record 1 of 16

Event name

Process


Process Version

State Name

1. **Name:** This is a text field that allows users to input the name of the event action. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An event action must have a unique name. It cannot have the same name as another event action in the program.
2. **Title:** This is a text field that allows users to input the title of the event action.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the event action out as active.
4. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the event action.

5. **Scope:** This table displays all defined events in the application (as retrieved from the [Events](#) section of the **Processes** module). It allows users to assign a scope to the event action.
- To select a scope from the table, double-click on it.
    - When a scope is selected from the table, the associated access name will automatically populate the **Event Name** field.
  - Assigning a scope to an event action is optional. If a scope is assigned, the action will execute automatically when the corresponding event occurs. If not, the action must be executed manually.

✿ If a scope is assigned, the event action can be linked to a scheduled event. For more information on scheduled events, [click here](#).

6. **Event Name:** When a scope is selected from the table, the associated access name will automatically populate this field.
- To remove the event name, click on the  button.
7. **Process:** This is a single-select dropdown field that retrieves the list of processes (from the [Process Design](#) section of the **Processes** module). It allows users to select the process during which the event action will be executed.

✿ If a process is not selected from this dropdown, the event action will be executed in all processes associated with the chosen entity.

8. **Process Version:** This is a single-select dropdown field that displays the list of versions associated with the selected process. It allows users to select the version of the process during which the event action will be executed.
- Depending on the process selected, the process versions displayed in this dropdown may vary.
    - If -1 is selected as a process version, the event action will be executed in all versions of the selected process.

✿ If a process version is selected from this dropdown, the event action will be triggered only for that particular version of the process.

9. **State Name:** This is a single-select dropdown field that displays the list of states in the selected process. It allows users to select the state during which the event action will be executed.
- Depending on the process and process version selected, the states displayed in this dropdown may vary.

✿ If a state is not selected from this dropdown, the event action will be assigned to all states associated with the chosen process.

### Event Action Execution:

| Event | Process          | Version          | State          | Outcome   |
|-------|------------------|------------------|----------------|---|
| Event | Any              | Any (-1)         | Any            | The event action will be triggered for all processes and states (associated with the chosen entity)     |
| Event | Specific Process | Any (-1)         | Any            | The event action will be triggered for only one process, but in all versions and states                 |
| Event | Specific Process | Specific Version | Any            | The event action will be triggered for only one version of a particular process, but in all states      |
| Event | Specific Process | Any              | Specific State | The event action will be triggered for only one state of a particular process                           |
| Event | Specific Process | Specific Version | Specific State | The event action will be triggered for a specific state in a particular version of a particular process |



To save changes, click on the **Apply** button. To save changes and exit the [Event Action Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 13.3. Guards

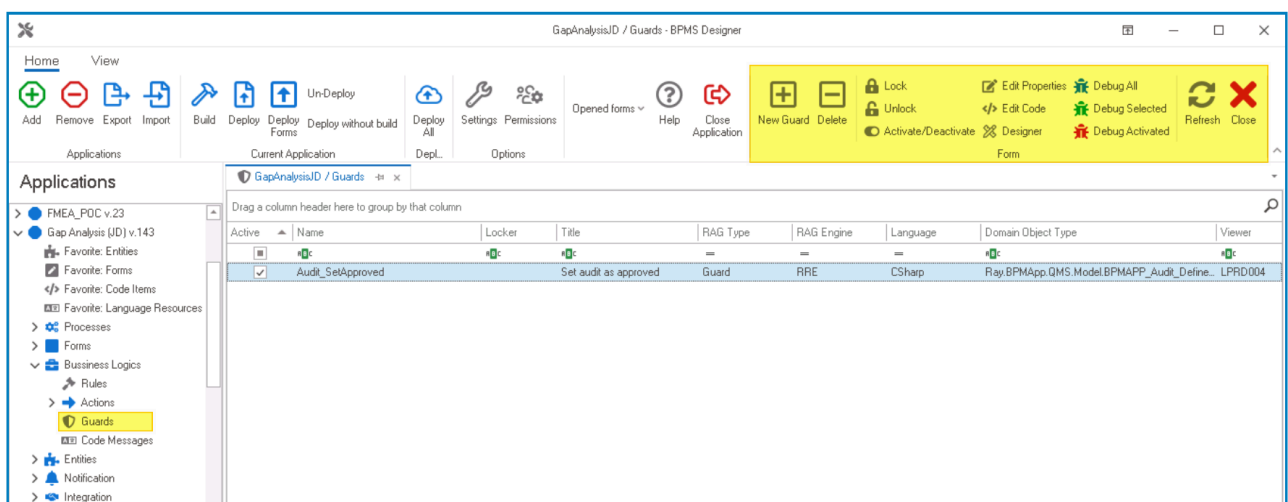
In the **Guards** section of this module, users can create and customize guards for the application.



- Similar to an [event action](#), a guard is an action that takes place simultaneously with the event. A guard, however, can also prevent the occurrence or execution of an event.

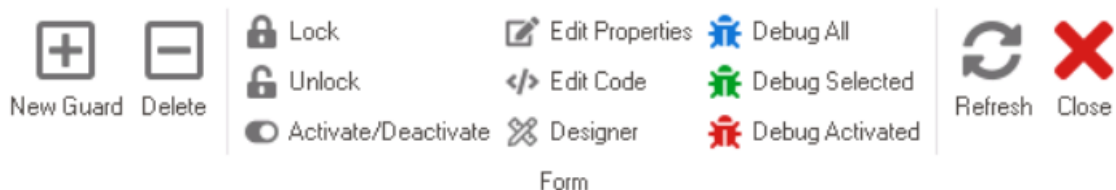
To manage guards in the application, take the following steps:

- From the [Application Tree](#), navigate to **Business Logics > Guards** (double-click to open). It will display the **Guards** tab and expand the menu bar to show additional options.



- In this tab, users can access a list of all defined guards in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



- New Event Action:** This button allows users to define a new guard.
  - If selected, it will display the [Guard Properties](#) window.

- **Delete:** This button allows users to delete a guard
  - To use this function, first select a guard from the list.
- **Lock:** This button allows users to lock the guard, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the guard.
  - If selected, access restriction on the guard will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the guard.
- **Edit Properties:** This button allows users to edit the properties of an existing guard.
  - If selected, it will display the Guard Properties window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select an guard from the list.
- **Edit Code:** This button allows users to launch the Code Designer, in which users can write or edit code specifying the behavior of the guard.
  - For more details on how to define guards using the Code Designer, [click here](#).
- **Designer:** This button allows users to launch the Rule Designer, in which they can define rules that specify the conditions and actions for the guard.
  - For more details on how to define guards using the Rule Designer, [click here](#).
- **Debug All:** This button allows users to debug all guards in the list.
- **Debug Selected:** This button allows users to debug only the selected guard in the list.
- **Debug Activated:** This button allows users to debug only the guards that are tagged as active.
- **Refresh :** This button allows users to refresh the Guards tab.
- **Close:** This button allows users to close the Guards tab.



Users can also create and link a guard directly within the properties of the corresponding event. For more information on this matter, [click here](#).

## 13.3.1. Guard Properties

In the Guard Properties window, users can create a new guard for the application.

Guard Properties: new

Apply OK Cancel

Type: **Guard**

Name:

Title:

☒ Active

Entity: [Please Select...]

Scope

Drag a column header here to group by that column

| Title                | Name              | Access Name                 |
|----------------------|-------------------|-----------------------------|
| Approve              | Approve           | Ray.BPMApp.GapAnalysisJD.Si |
| Submit               | Submit            | Ray.BPMApp.GapAnalysisJD.Si |
| Send for Review      | SendforReview     | Ray.BPMApp.GapAnalysisJD.Si |
| Send Back for Review | SendBackforReview | Ray.BPMApp.GapAnalysisJD.Si |
| Reject               | Reject            | Ray.BPMApp.GapAnalysisJD.Si |
| Submit               | Submit            | Ray.BPMApp.GapAnalysisJD.Si |

Record 1 of 16

Event name:

Process: [Empty]

Process Version:

State Name:

1. **Name:** This is a text field that allows users to input the name of the guard. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A guard must have a unique name. It cannot have the same name as another guard in the program.
2. **Title:** This is a text field that allows users to input the title of the guard.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the guard out as active.
4. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the guard.
5. **Scope:** This table displays all defined events in the application (as retrieved from the [Events](#)



section of the **Processes** module). It allows users to assign a scope to the guard.

- To select a scope from the table, double-click on it.
  - When a scope is selected from the table, the associated access name will automatically populate the **Event Name** field.
- Assigning a scope to a guard is optional. If a scope is assigned, the guard will execute automatically when the corresponding event occurs. If not, the guard must be executed manually.

✿ If a scope is assigned, the guard can be linked to a scheduled event. For more information on scheduled events, [click here](#).

6. **Event Name:** When a scope is selected from the table, the associated access name will automatically populate this field.

- To remove the event name, click on the  button.

7. **Process:** This is a single-select dropdown field that retrieves the list of processes (from the [Process Design](#) section of the **Processes** module). It allows users to select the process during which the guard will be executed.

✿ If a process is not selected from this dropdown, the guard will be executed in all processes associated with the chosen entity.

8. **Process Version:** This is a single-select dropdown field that displays the list of versions associated with the selected process. It allows users to select the version of the process during which the guard will be executed.

- Depending on the process selected, the process versions displayed in this dropdown may vary.
  - If -1 is selected as a process version, the guard will be executed in all versions of the selected process.

✿ If a process version is selected from this dropdown, the guard will be triggered only for that particular version of the process.

9. **State Name:** This is a single-select dropdown field that displays the list of states in the selected process. It allows users to select the state during which the guard will be executed.

- Depending on the process and process version selected, the states displayed in this dropdown may vary.

✿ If a state is not selected from this dropdown, the guard will be assigned to all states associated with the chosen process.

## Guard Execution:

| Guard | Process          | Version          | State          | Outcome  |
|-------|------------------|------------------|----------------|--|
| Guard | Any              | Any (-1)         | Any            | The guard will be triggered for all processes and states (associated with the chosen entity)     |
| Guard | Specific Process | Any (-1)         | Any            | The guard will be triggered for only one process, but in all versions and states                 |
| Guard | Specific Process | Specific Version | Any            | The guard will be triggered for only one version of a particular process, but in all states      |
| Guard | Specific Process | Any              | Specific State | The guard will be triggered for only one state of a particular process                           |
| Guard | Specific Process | Specific Version | Specific State | The guard will be triggered for a specific state in a particular version of a particular process |



To save changes, click on the **Apply** button. To save changes and exit the Guard Properties window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 13.4. Code Messages

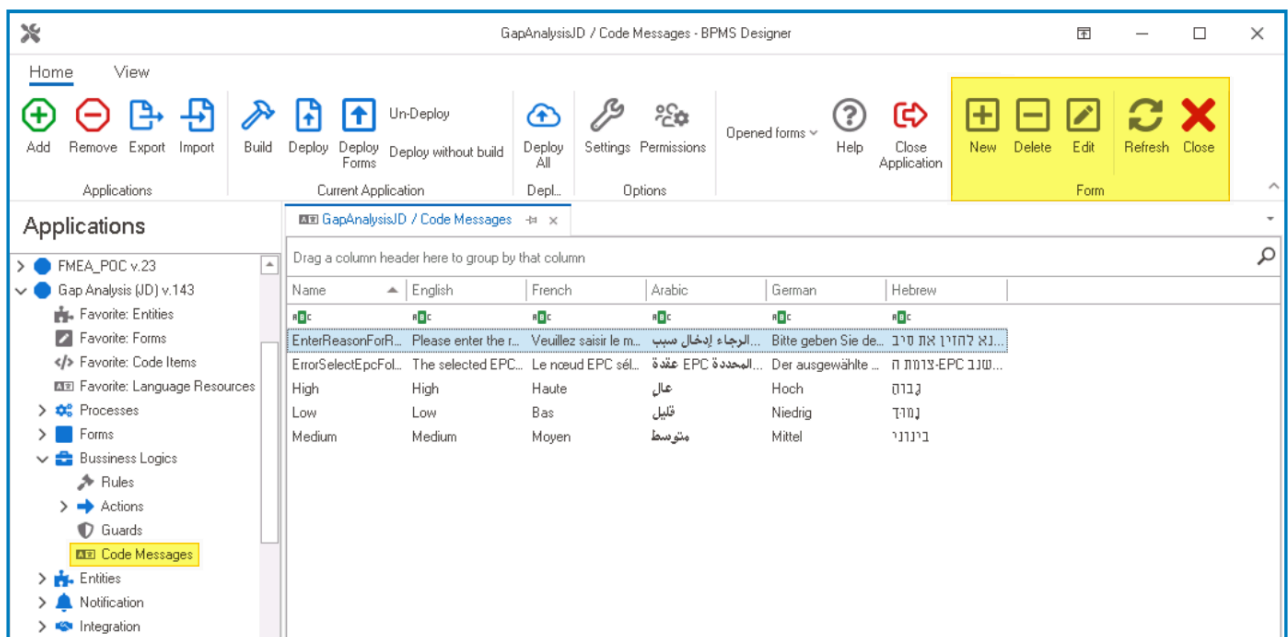
In the **Code Messages** section of this module, users can create code messages for the application.



- \* Code messages are messages or alerts that provide information about the status or behavior of the web application. They can be used to provide feedback to the user or to signal an error or problem that needs attention. Code messages can take various forms, such as error messages, warning messages, or informational messages.

To manage code messages in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Business Logics > Code Messages** (double-click to open). It will display the Code Messages tab and expand the menu bar to show additional options.



- \* In this tab, users can access a list of all defined code messages in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

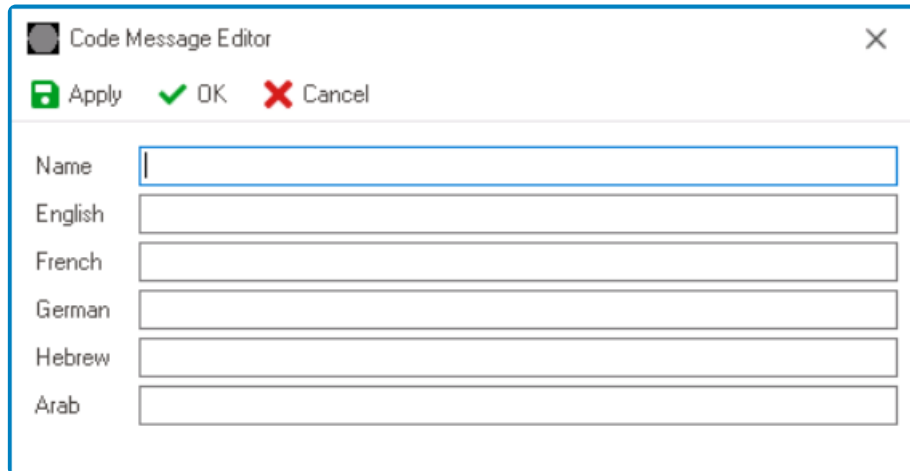
2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new code message.
  - If selected, it will display the Code Message Editor window.
- **Delete:** This button allows users to delete an existing code message.
  - To use this function, first select a code message from the list.
- **Edit :** This button allows users to edit the properties of an existing code message.
  - If selected, it will display the Code Message Editor window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a code message from the list.
- **Refresh :** This button allows users to refresh the Code Messages tab.
- **Close:** This button allows users to close the Code Messages tab.

## 13.4.1. Code Message Editor

In the Code Message Editor window, users can create a new code message for the application.



1. **Name:** This is a text field that allows users to input the name of the code message. It is a mandatory field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A code message must have a unique name. It cannot have the same name as another code message in the application.
2. **Message:** This is a text field that allows users to input the code message.
  - Users can input the message in different languages.

✿ Users can determine which languages are available for use in the Settings of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Hebrew, and Arabic, and Hebrew.

3. **Apply:** This button allows users to save their changes.
4. **OK:** This button allows users to save their changes and close the window.
5. **Cancel:** This button allows users to cancel their changes and close the window.

## 14. Entities

---

The **Entities** module is an essential tool for managing processes, forms, and data storage in the application. It generates the necessary data structures, including tables and relationships, within the database to store all process and form data. Each entity created through this module results in a new table being added to the database.

In the **Entities** module, users are provided access to the following elements:



1. [Entity Designer](#)
2. [KPIs](#)
3. [KPI BAM](#)

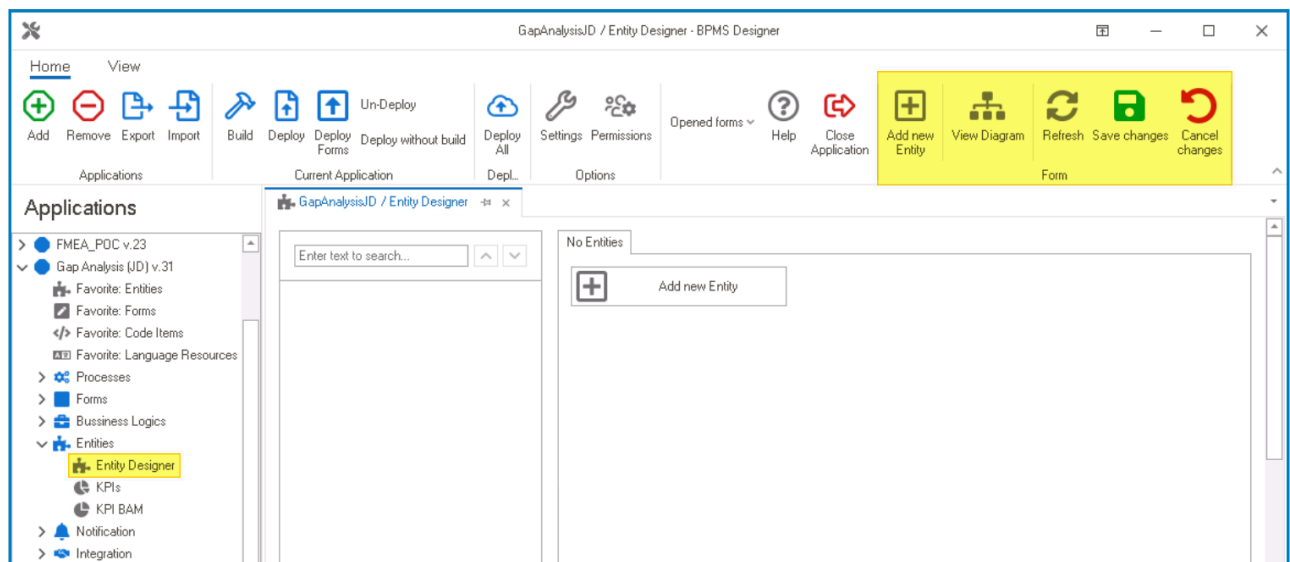
# 14.1. Entity Designer

The Entity Designer is a tool which enables users to create and customize entities and attributes within the application. With this tool, users can define the data structures that will be used to store process and form data.

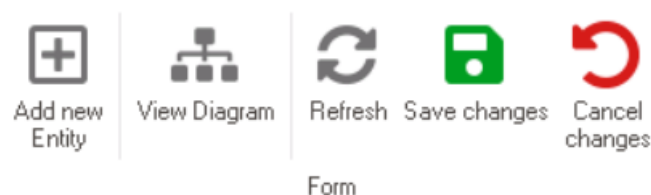


To access the Entity Designer, take the following steps:

1. From the [Application Tree](#), navigate to **Entities > Entity Designer** (double-click to open). It will display the Entity Designer tab and expand the menu bar to show additional options.



2. From the menu bar, select the desired option.



- **Add New Entity:** This button allows users to add a new entity.
  - If selected, the No Entities subtab will be replaced with the Entity and Attribute subtabs, which allow users to define the new entity and its attributes.

1. [Entity Tab](#)
2. [Attribute Tab](#)

✿ Users can also add a new entity by selecting the **Add New Entity**

button in the No Entities subtab.

- **View Diagram:** This button allows users to view the Entity-Relationship (ER) diagram.
  - For more information on this diagram, [click here](#).
- **Refresh:** This button allows users to refresh the Entity Designer tab.
- **Save Changes:** This button allows users to save their changes.
- **Cancel Changes:** This button allows users to cancel their changes.



## 14.1.1. Add New Entity

In the Entity subtab, users can define the properties of the newly created entity as well as manage the structure of the Entity Tree.

This subtab is divided into three key components:

### 1. Entity Tree

In this section of the tab, users can view a hierarchical representation of the entities and their associated attributes in the application.

The screenshot shows the Entity subtab interface. On the left is the Entity Tree, which is currently empty except for a search box at the top labeled "Enter text to search...". A red circle with the number "1" is placed over the tree area. On the right is the Entity details panel, which has two tabs: "Entity" (selected) and "Attribute". The "Entity" tab contains fields for Name (filled with "GapAnalysisJD\_NewEntity"), Title, and Description. Below these fields are four buttons: "Add new Entity", "Copy this Entity", "Delete Entity", and "Add new Attribute".

- Users can expand and collapse the tree nodes to view or hide the associated attributes of each entity.
- Users can also search the list of entities and their attributes by using the search box.

This screenshot shows the Entity subtab interface with search results. The search box at the top of the Entity Tree is filled with "New Gap Analysis". The tree is expanded, showing a list of entities. The entity "New Gap Analysis" is highlighted in yellow. Below it, several attributes are listed, including "AnalysisScope", "Category", "DataAnalysisMethods", "GapAnalysisID", "GapAnalysisName", "Source", and "TargetDate". The Entity details panel on the right shows the "Entity" tab with fields for Name (filled with "NewGapAnalysis"), Title (filled with "New Gap Analysis"), and Description. The same four buttons are present at the bottom.

- All the titles containing the search value will be highlighted in yellow. Users can then click on the up or down arrow buttons to navigate to the next or previously matched title,

respectively.

## 2. Entity Properties

In this section of the tab, users can define the properties of the entity.

- **Name:** This is a text field that allows users to input the name of the entity. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters.
    - An entity must have a unique name. It cannot have the same name as another entity in the program.
- **Title:** This is a text field that allows users to input the title of the entity. It is a **mandatory** field.
  - The title inputted in this field will be used as the display title for the entity.
  - Unlike with names, there are no rules determining the validity of titles.
- **Description:** This is a multi-line text field that allows users to input a description for the entity.

! To finalize the configuration of the entity, select the **Save Changes** option from the menu bar. To cancel, select the **Cancel Changes** option.

## 3. Entity Menu

In this section of the tab, users can manage the entity and its attributes.

- **Add New Entity:** This button allows users to add a new entity.
- **Copy This Entity:** This button allows users to copy the selected entity.
  - If selected, a new entity with all the associated attributes of the original will be automatically created.
    - Users can then modify the properties of the duplicate entity as needed.
    - By default, however, the name of the duplicate entity will follow this format: OriginalName\_1.

**!** Please ensure that the name of the duplicate entity is unique and does not match any existing entity name.

- **Delete Entity:** This button allows users to delete the selected entity.
  - If selected, users will be alerted of the action with a confirmation message.
    - To confirm the deletion, click on the **Yes** button. To cancel, click on the **No** button.
- **Add New Attribute:** This button allows users to add a new attribute under the selected entity.
  - If selected, it will redirect users to the [Attribute](#) tab, wherein they can define the properties of the newly created attribute.
- **Paste Attribute:** This button allows users to paste the selected attribute.
  - To enable this button, users must first select an attribute from the Entity Tree and click on the **Copy Attribute** button (located in the [Attribute](#) tab).

## 14.1.1.1. Add New Attribute

In the Attribute subtab, users can define the properties of the newly created attribute as well as manage the structure of the Entity Tree.

✿ An attribute is a characteristic or property of an entity that describes its features or qualities. For example, “color” is an attribute of “hair”, indicating its hue or shade. To facilitate data storage and management, attributes are stored as table fields in the database. Each attribute can be linked to a corresponding form control.


This subtab is divided into two key components:

### 1. Attribute Properties


In this section of the tab, users can define the properties of the attribute.

- **Name:** This is a text field that allows users to input the name of the attribute. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters.
    - An attribute must have a unique name. It cannot have the same name as another attribute within the entity.
- **Title:** This is a text field that allows users to input the title of the attribute. It is a **mandatory** field.
  - The title inputted in this field will be used as the display title for the attribute.
  - Unlike with names, there are no rules determining the validity of titles.
- **Description:** This is a multi-line text field that allows users to input a description for the attribute.

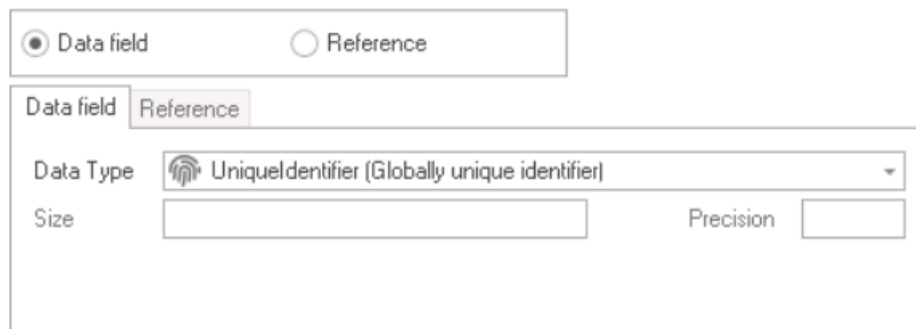
- **Attribute Type:** This is a radio button that allows users to select only one option from a set of mutually exclusive options. It is a **mandatory** field. Users can choose to create either a data attribute or a reference attribute.
  - If the **Data Field** option is selected, it will enable the Data Field tab.

 Data attributes are used to store simple, self-contained values that are unique to the entity.


- If the **Reference** option is selected, it will enable the Reference tab.

 Reference attributes are used to create more complex relationships between entities. By creating a reference attribute linking to another entity, users can establish a relationship between different entities.



- **Data Field:** In this tab, users can define the properties of the data attribute.



- **Data Type:** This field allows users to choose the data type of the attribute from a dropdown menu. The selected data type will determine the SQL field type of the attribute in the database.
  - Attributes are linked to form controls according to their data type.
    - For more information on form controls and their compatible data types, [click here](#).
- **Size:** This field allows users to set the size of the attribute.
  - Depending on the attribute selected, this field may be enabled or disabled.
- **Precision:** This field allows users to set the precision of the attribute.
  - This field is only enabled for the Decimal and Numeric data type options.

 Precision refers to the number of digits after the decimal point.

- **Reference:** In this tab, users can define the properties of the reference attribute.

- **Entity:** This field automatically displays the referenced entity.
  - For an entity to appear in this field, users must first make their selection.
-  : This button allows users to remove the selected entity and field.
-  : This button allows users to select the entity and field.
  - If clicked, it will display the Select Reference window, where users can choose an application or system entity and select a specific field or attribute within it.

| Application       | Entity           |
|-------------------|------------------|
| =                 | g                |
| System            | Departments      |
| System            | Firms            |
| System            | Individuals      |
| Gap Analysis (JD) | New Gap Analysis |
| System            | Offices          |
| System            | Payment          |
| System            | Principal Groups |
| System            | Principals       |
| System            | Roles            |
| System            | Users            |

| Select field                    |
|---------------------------------|
| ID String / VarChar (36)        |
| ManagerID String / VarChar (36) |
| OfficelD String / VarChar (36)  |
| ParentID String / VarChar (36)  |

- **Select or Cancel:** To select an entity and field, click on the **Select** button. To cancel the selection and exit the window, click on the **Cancel** button.
- **Field:** This space automatically displays the referenced field within the entity.
  - For a field to appear in this space, users must first make their selection.
- **Cascade (Master-Detail):** This is a checkbox which, if selected, allows changes made to the referencing entity to automatically apply to the referenced entity in a master-detail relationship.

✿ If this checkbox is selected, any changes made to the referencing entity (the “master” or “parent”) will be automatically applied to the referenced entity (the “detail” or “child”). This means that if you delete the referencing entity, all the referenced entities that are dependent on it will also be deleted.

#### ▪ **Cascade (Master-Detail): Example**

- To better understand the concept of cascading, consider the following

example:

- Assume that there are two entities: 'customer' and 'order'.
  - Each customer can have multiple orders, and the relationship between the customer and the order is one-to-many, established using a connecting entity.
- If the **Cascade (Master-Detail)** checkbox is selected for the reference attribute, deleting a customer entity will also result in the deletion of all the orders associated with that customer.
  - This is because an order without a customer is meaningless, and hence, it's logical to delete all the orders associated with a customer when the customer entity is removed.
- **Prevent Deletion of Master Entity:** This is a checkbox which, if selected, allows users to prevent the deletion of a master entity that has at least one child entity associated with it.



If this checkbox is selected, an attempt to delete a master entity that has at least one child entity associated with it will be automatically prevented by the system. This helps to maintain data integrity and prevent accidental deletion of important data.

- **Preventing Deletion of Master Entity: Example**
  - To better understand the concept of preventing the deletion of the master entity, consider the following example:
    - Assume that there are two entities: 'student' and 'course'.
      - Each student can take multiple courses, and the relationship between the student and the course is one-to-many, established using a connecting entity.
      - If a student is deleted, it may be undesirable to also delete the associated course.
    - If a user attempts to delete a student who is associated with at least one course, selecting the **Prevent Deletion of Master Entity** checkbox for the reference attribute will prevent the deletion of the student entity.
      - Instead, the system will throw an error to inform the user that the student entity cannot be deleted as it has at least one associated course.
- **Show as Column in Task List:** In this section, users can customize the display of attributes as columns in the task list of the web application.
  - **Hide:** This option allows users to remove the attribute from the list of columns.
  - **Hidden By Default:** This option allows users to hide the attribute from the list of columns by default.
    - Users can then choose whether to display the column by manually adding it.
  - **Show By Default:** This option allows users to display the attribute as a column by default in the list of columns.

- Users can then choose whether to hide the column by manually removing it.

✱ For more information on how to customize the columns in the task list of the **DBP** web application, [click here](#).

- **Read Only:** This is a checkbox which, if selected, allows users to ensure that the system will not update or overwrite any data stored in that attribute.

✱ This option is particularly useful when the value of the field is already defined in the database and should not be modified by users. For example, the value of the attribute may be automatically set by a trigger or an external tool.

- **Defines Field:** This is a checkbox which, if selected, allows users to designate the attribute as a defining or differentiating factor for the entity. This means that the attribute is considered a critical component of the entity and is necessary for its definition.
  - For example, in the case of a **Book** entity, the [Name](#) and [Author](#) attributes may be considered defining factors, and so both would have the **Defines Field** checkbox selected.

✱ When this checkbox is selected, the attribute will be displayed in the **Entity Definition** column of the [Audit Trail](#) form.

- **Track Changes:** This is a checkbox which, if selected, enables changes to be tracked for the attribute.
  - In the example, changes to the [Author](#) attribute of the **Book** entity could be tracked by selecting the **Track Changes** checkbox.

✱ When this checkbox is selected, any modifications made to the attribute will be recorded in the audit trail, with the old value and new value both displayed in the **Old Value** and **New Value** columns of the form.

✱ Selecting either the **Defines Field** or **Track Changes** checkbox will create an audit trail for the attribute. Each checkbox, however, impacts how the attribute is recorded in the **DBP** web application. For more information on the [Audit Trail](#) form, [click here](#).

! To finalize the configuration of the attribute, select the **Save Changes** option from the menu bar. To cancel, select the **Cancel Changes** option.

## 2. Attribute Menu

In this section of the tab, users can manage attributes.



The screenshot shows the 'Attribute' tab in the DBP Designer. On the left, there's a search bar and a tree view under 'New Gap Analysis' showing 'NewAttribute Id / UniqueIdentifier'. The main form has the following fields and options:

- Name:** NewAttribute
- Title:** (empty)
- Description:** (empty)
- Data field / Reference:** Radio buttons, with 'Data field' selected.
- Data Type:** UniqueIdentifier (Globally unique identifier)
- Size:** (empty)
- Precision:** (empty)
- Show as a column in the task list:** Radio buttons for 'Hide' (selected), 'Hidden by default', and 'Show by default'.
- Read Only:** (unchecked)
- Defines field:** (unchecked)
- Track changes:** (unchecked)
- Buttons:** '+ Add new Attribute', 'Copy Attribute', and '- Delete Attribute'.

A red line with a '2' in a circle highlights the bottom buttons.

- **Add New Attribute:** This button allows users to add a new attribute.
- **Copy Attribute:** This button allows users to copy the selected attribute.
  - To paste the attribute, users will need to select the **Paste Attribute** button in the [Entity](#) tab.
    - If selected, a new attribute with all the properties of the original will be automatically created.
    - Users can then modify the properties of the duplicate attribute as needed.
      - By default, however, the name of the duplicate attribute will follow this format: OriginalName\_1.

! Please ensure that the name of the duplicate attribute is unique and does not match any existing attribute name within the entity.

- **Delete Attribute:** This button allows users to delete the selected attribute.
  - If selected, users will be alerted of the action with a confirmation message.
    - To confirm the deletion, click on the **Yes** button. To cancel, click on the **No** button.

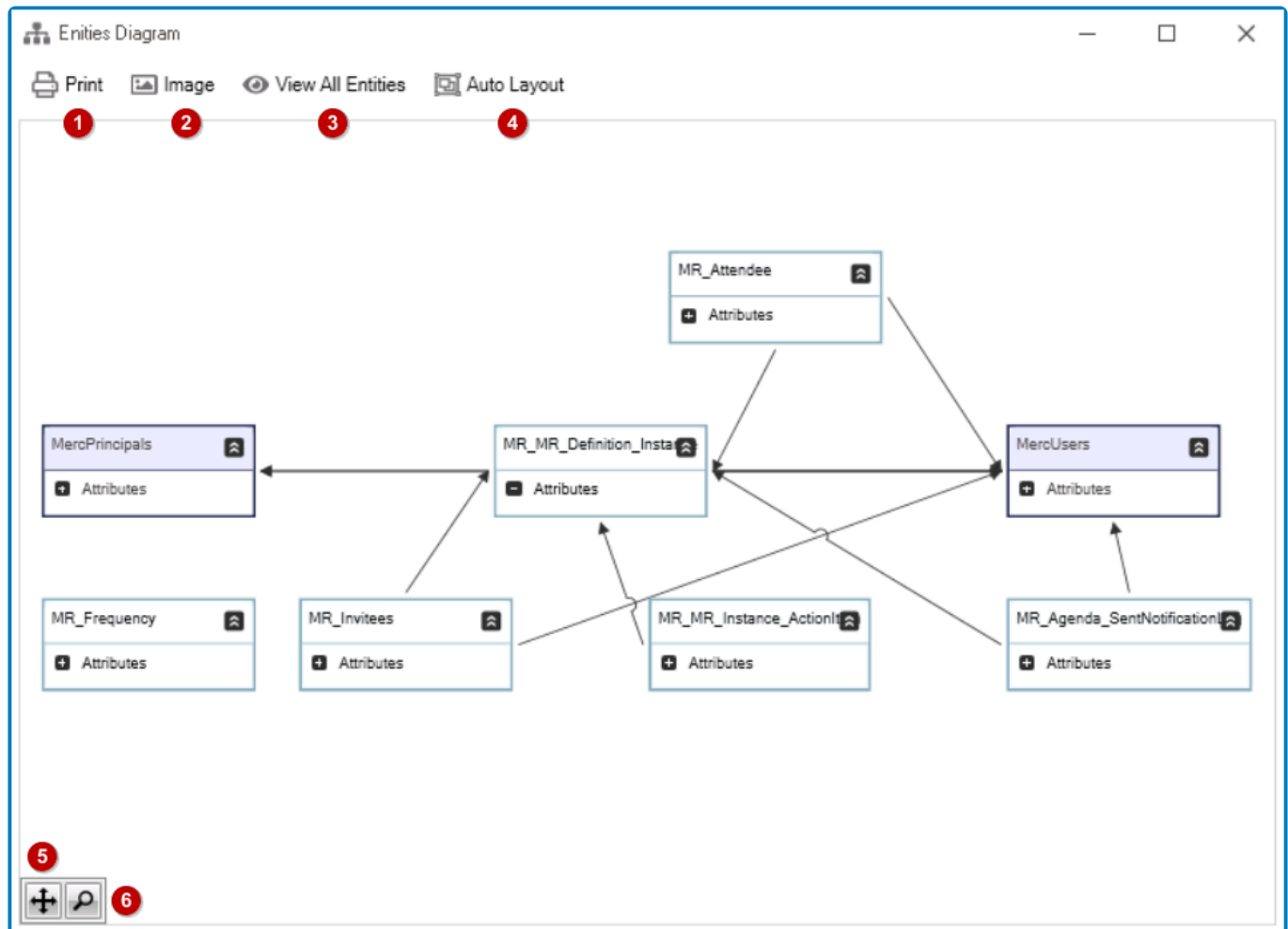
## 14.1.1.1.1. Data Type Per Form Control

Attributes are linked to form controls according to their data type. The table below provides a list of form controls and the compatible data types that can be used with them.

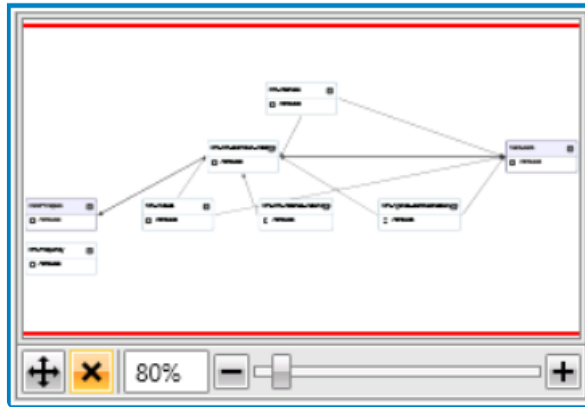
| Control  | Data Type  |
|--|--|
| <a href="#">Checkbox</a>   | Bit  |
| <a href="#">Comment</a>  | Unique Identifier (in case of communication with a second entity)  |
| <a href="#">Date/Time Picker</a>   | Date, DateTime, DateTime2  |
| <a href="#">File Attachment</a>  | NVarChar (size: 100, according to the number of attachable files), NVarChar_Max, VarChar, VarChar_Max  |
| <a href="#">Free Draw</a>  | VarBinary, VarBinary_Max.  |
| <a href="#">Gantt Chart</a>  | Unique Identifier (in case of communication with a second entity)  |
| <a href="#">Grid</a>   | Unique Identifier. It can be used to bind different fields together. The data type may or may not be same for each field in the list.                                    |
| <a href="#">Nested Form Link</a>   | Unique Identifier (in case of communication with a second entity)  |
| <a href="#">Numeric Textbox</a>  | Big Int, Decimal, Float, Int, Money, Numeric, Small Int, Small Money, Tiny Int.  |
| <a href="#">Radio-Button List</a>  | String Data Types (such as Char, NChar, NText, NVarChar, or Text), Numeric Data Types (such as Decimal, Float, Int, Numeric, or Real) , and the Bit (Boolean) data type. |
| <a href="#">Textbox (Single-Line, Multi-Line, Password) / Rich Textbox</a> | Char (Char or NChar), VarChar (VarChar or VarChar_Max), NVarChar (NVarChar or NVarChar_Max).   |
| <a href="#">Time Duration</a>  | DateTime   |
| <a href="#">Tree</a>   | Unique Identifier (in case of communication with a second entity)  |
| <a href="#">User Picture</a>   | VarChar (size: 36, in the case of communication with the MercUsers entity)   |
| <a href="#">User Signature</a>   | VarChar (size: 36, in the case of communication with the MercUsers entity)   |

## 14.1.2. Entity-Relationship Diagram

An Entity-Relationship (ER) diagram is a graphical representation of entities and their attributes, as well as the relationships between entities in an application.



1. **Print:** This button allows users to print the diagram.
  - If selected, the Print dialog box will appear, enabling users to specify the printer and other printing preferences.
2. **Image:** This button allows users to save the diagram as an image.
  - If selected, a Save As dialog will appear, enabling users to choose a location on their local system to save the file.
3. **View All Entities:** This button provides users with a comprehensive view of all entities within the diagram.
4. **Auto Layout:** This button allows users to automatically arrange the entities and their relationships in a visually appealing and organized manner, without the need for manual adjustment.
5. **Auto Fit:** This button allows users to automatically resize the diagram to fit within the available space, ensuring that all entities and relationships remain visible.
6. **Magnify:** This button allows users to zoom in or out on specific areas of the diagram.
  - If selected, a preview box will appear in the lower left corner of the screen, displaying a miniature view of the entire diagram along with a slider that users can use to adjust the zoom level.



- The slider displays the zoom level as a percentage, enabling users to precisely specify the level of magnification.

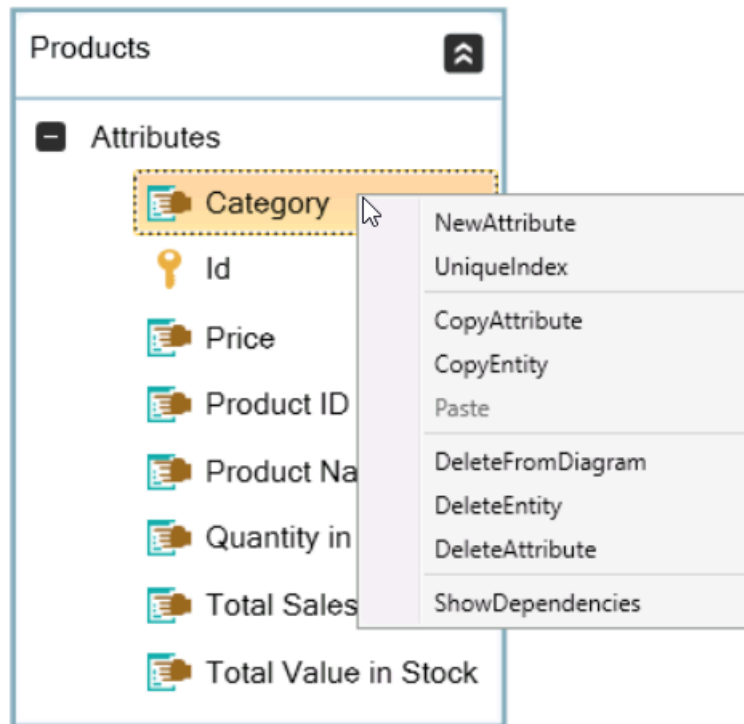
✿ The preview box includes a viewport or viewfinder, which is a smaller red box that serves as a window to the larger diagram. Users can drag the viewport around the preview box to focus on different areas of the diagram, and click on a particular section to quickly move to that area.

### Expand / Collapse Nodes

Users can expand and collapse the tree nodes to view or hide the associated attributes of each entity.

### Attribute Context Menu

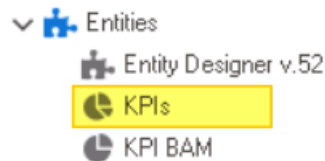
To access the attribute context menu, right-click on an attribute in the diagram. It will display the following options:



- **New Attribute:** This option allows users to add a new attribute under the entity.
  - If selected, it will display the [Attribute](#) window, in which users can define the properties of the attribute.
    - This window is largely a duplicate of the [Attribute](#) subtab in the [Entity Designer](#).
- **Unique Index:** This option allows users to create a unique index for the entity associated with the attribute.
- **Copy Attribute:** This option allows users to copy the attribute.
- **Copy Entity:** This option allows users to copy the entity associated with the attribute.
- **Paste:** This option allows users to paste the copied attribute.
  - To enable this option, users must first select an attribute and click on the **Copy Attribute** option.
- **Delete From Diagram:** This option allows users to remove the entity associated with the attribute from the diagram.
- **Delete Entity:** This option allows users to delete the entity associated with the attribute.
  - If selected, users will be alerted of the action with a confirmation message.
    - To confirm the deletion, click on the **Yes** button. To cancel, click on the **No** button.
- **Delete Attribute:** This option allows users to delete the attribute.
  - If selected, users will be alerted of the action with a confirmation message.
    - To confirm the deletion, click on the **Yes** button. To cancel, click on the **No** button.
- **Show Dependencies:** This option allows users to show the dependencies of the entity.

## 14.2. Key Point Indicators (KPIs)

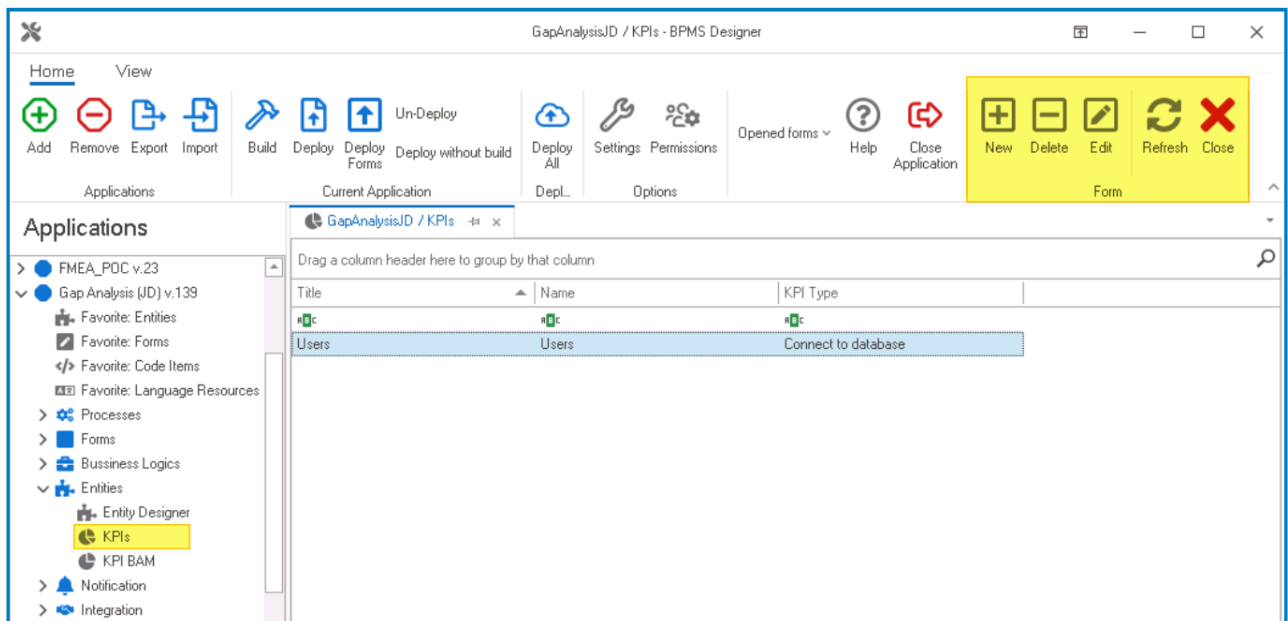
In the **KPIs** section of this module, users can define key performance indicators (KPIs) to measure the performance of their data sources.



- These KPIs rely on the selected data source(s) to calculate and display performance metrics, which can then be used to monitor and improve performance. Users can customize the KPIs to track specific metrics and create visual reports and dashboards using controls such as maps, charts, and graphs.

To manage KPIs in the application, take the following steps:

- From the [Application Tree](#), navigate to **Entities > KPIs** (double-click to open). It will display the **KPIs** tab and expand the menu bar to show additional options.



- In this tab, users can access a list of all defined KPIs in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



- **New:** This button allows users to define a new KPI.
  - If selected, it will display the [KPI Properties](#) window.
- **Delete:** This button allows users to delete an existing KPI.
  - To use this function, first select a KPI from the list.
- **Edit :** This button allows users to edit the properties of an existing KPI.
  - If selected, it will display the [KPI Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a KPI from the list.
- **Refresh :** This button allows users to refresh the [KPIs](#) tab.
- **Close:** This button allows users to close the [KPIs](#) tab.



A mixed KPI is a metric that combines different types of data to provide a more comprehensive view of performance. It can be made up of multiple individual indicators.

## 14.2.1. KPI Properties

---

In the KPI Properties window, users can create a new KPI for the application.

There are five tabs to consider in this window:

1. [Naming](#)
2. [DataSource](#)
3. [Execute Parameters](#)
4. [View and Insert Parameters](#)
5. [Timing](#)

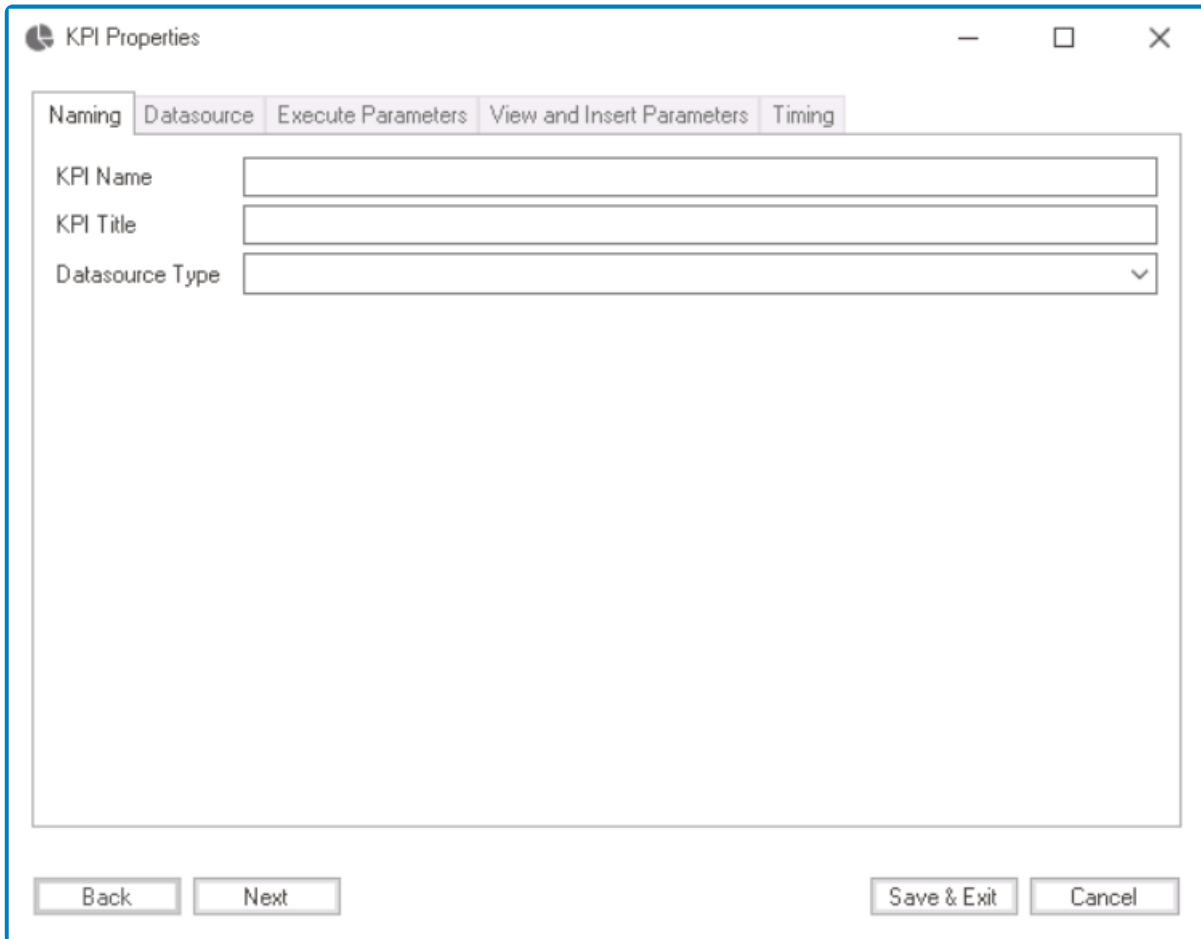


To save changes and exit the KPI Properties window, click on the **Save & Exit** button. To cancel changes and exit the window, click on the **Cancel** button or the **Exit (X)** button. To navigate to the previous or next tab, click on the **Back** or **Next** button, respectively.



## 14.2.1.1. Naming Tab

In the Naming tab, users can configure basic properties required for the KPI's identification and management.

The screenshot shows a window titled "KPI Properties" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there are five tabs: "Naming", "Datasource", "Execute Parameters", "View and Insert Parameters", and "Timing". The "Naming" tab is currently selected and active. It contains three input fields: "KPI Name" (a text box), "KPI Title" (a text box), and "Datasource Type" (a dropdown menu with a downward arrow). At the bottom of the dialog, there are four buttons: "Back", "Next", "Save & Exit", and "Cancel".

1. **KPI Name:** This is a text field that allows users to input the name of the KPI. It is a **mandatory** field.
  - There are the following rules determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot begin with a number.
    - A KPI must have a unique name. It cannot have the same name as another KPI in the application.
2. **KPI Title:** This is a text field that allows users to input the title of the KPI.
  - The title inputted in this field will be used as the display title for the KPI.
  - Unlike with names, there are no rules determining the validity of titles.
3. **Data Source Type:** This is a single-select dropdown field that displays the list of data source types. It is a **mandatory** field. It allows users to choose the appropriate data source type to associate with the KPI.
  - Data Source Types: Connect to DB, Manual, or Connect to WebService.
    - Connect to DB: This option allows users to connect the KPI to a specified database to retrieve performance data.
      - To set up the database connection, users can navigate to the [Database](#) subtab within the DataSource tab.

- Manual: This option enables users to manually input performance data into the KPI, making it useful when no specific data source is available.
  - To manually add columns and insert values into them, users can navigate to the [View and Insert Parameters](#) tab.
    - Once they have configured the required details in this tab, users can save their changes by clicking on the **Save & Exit** button.
- Connect to WebService: This option allows users to connect to an external web service to retrieve performance data for the KPI.
  - If this option is selected, it will enable the [WebService](#) subtab within the DataSource tab, which allows users to configure the web service connection settings.
    - Once they have configured the required details in this tab, users can save their changes by clicking on the **Save & Exit** button.



To enable the fields in the remaining tabs of the [KPI Properties](#) window, users must first select a data source type and save their changes. This will require them to exit the window and reopen it. To save and exit the window, users can click on the **Save & Exit** button.

## 14.2.1.2. DataSource Tab

In the DataSource tab, users can configure the connection details for their selected data source type.

### Database Tab

In this tab, users can specify the necessary information for connecting to a database.

1. **Connection:** This field allows users to choose the data source connection that will be used to retrieve data for the KPI.
  - By selecting a connection from this dropdown, users can establish a link between the query and the external data source.
2. **Select Table:** This field retrieves the list of tables (or entities) available in either the application or system. It allows users to choose the table that will be used to retrieve data for the KPI.

\* If created in the application, the tables displayed in this field will be retrieved from the Entity Designer section of the **Entities** module. If the tables already exist in the system, they will be retrieved from the database directly.

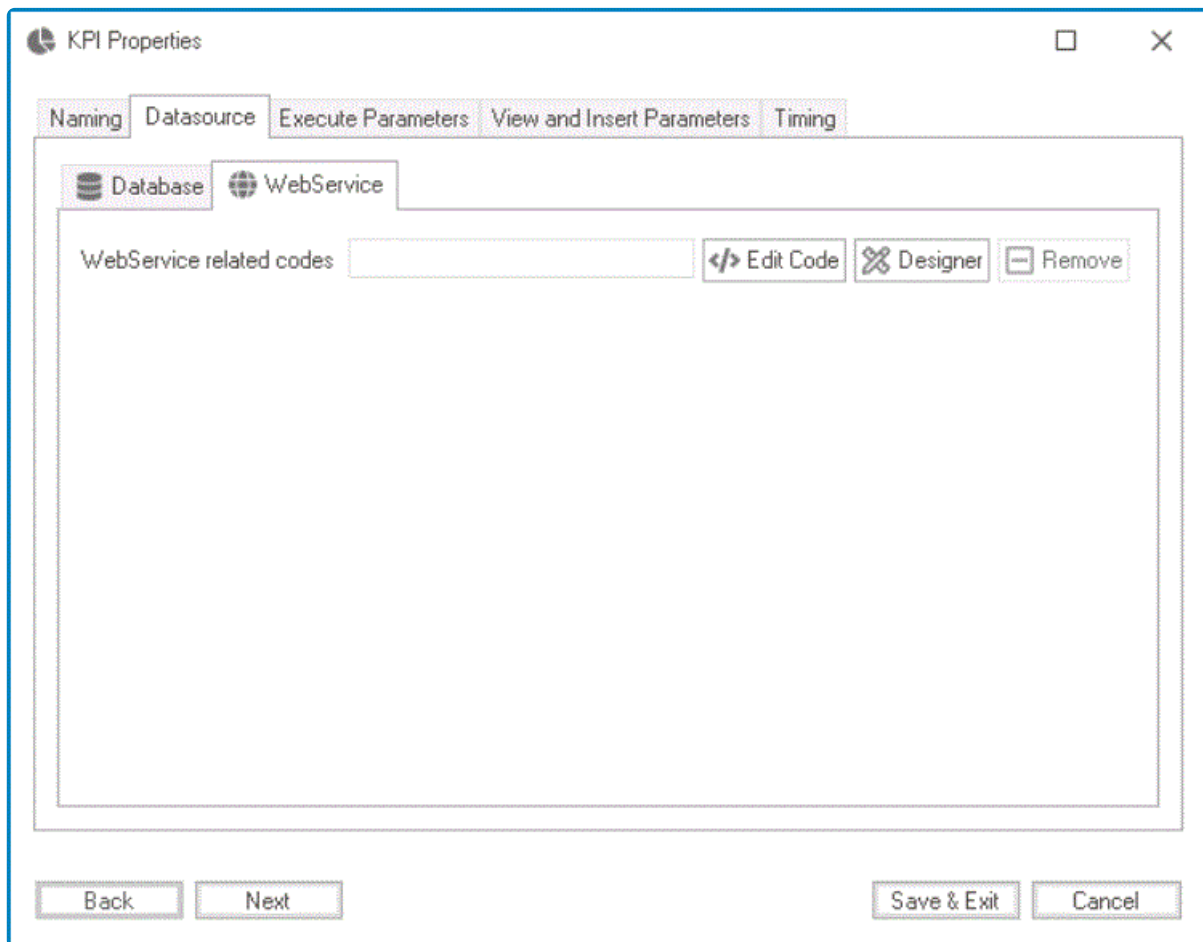
3. **SQL Query:** Once a table has been selected, this field will automatically be populated with a sample SQL query that specifies the table name and any necessary SQL syntax to retrieve the data for the KPI.
  - To use this sample, copy and paste it into the command box.

\* In addition to using the provided sample query, users can also use a parametric/dynamic query. To do so, enter the parameter name and its input data into the command box using the format 'ParameterName + \$'.

4. **Select Command:** In this tab, users can write SQL queries and specify custom functions or rules for manipulating the retrieved data.
  - **Edit Code:** This button allows users to launch the Code Designer, in which users can write code specifying custom functions or methods to manipulate the data retrieved by the SQL query.
    - For more details on how to customize SQL queries using the Code Designer, [click here](#).
  - **Designer:** This button allows users to launch the Rule Designer, in which they can define rules that specify the conditions and actions for the SQL query.
    - For more details on how to customize SQL queries using the Rule Designer, [click here](#).
  - **Remove:** This button allows users to remove the code.
5. **Count Command:** In this tab, users can write SQL queries that return a count of records meeting specific criteria.

## WebService Tab

In this tab, users can specify the necessary information for connecting to a web service.



1. **Edit Code:** This button allows users to launch the Code Designer, in which they can write custom code to manipulate the data retrieved by the web service connection.
  - For more details on how to customize web service-related code using the Code Designer, [click here](#).
2. **Designer:** This button allows users to launch the Rule Designer, in which users can define rules that specify the conditions and actions for the web service connection.
  - For more details on how to customize web service-related code using the Rule Designer, [click here](#).
3. **Remove:** This button allows users to remove the code.

## 14.2.1.3. Execute Parameters Tab

In the Execute Parameters tab, users can verify and execute the SQL query (entered in the [DataSource](#) tab).

KPI Properties

Naming DataSource **Execute Parameters** View and Insert Parameters Timing

Enter Parameter Name  
 SELECT \* FROM \${DB}.ray.Students where Id = '\${Id}'  
 1- \${DB}  
 2- \${Id}

Drag a column header here to group by that column

| Description | Value |
|-------------|-------|
| RBC         | RBC   |

Record 0 of 0

Execute

Back Next Save & Exit Cancel

\* For guidance on how to manipulate the data presented in this window, [click here](#).

In the table, there are two columns: Description and Value.

- If your query is parametric (i.e., contains parameter placeholders), enter the parameter names in the Description column and the corresponding values for the parameters in the Value column.
- Click on the **Execute** button.
- The program will execute the modified query with the parameter values you entered and display the results in the [View and Insert Parameters](#) tab.

\* If your query does not contain any parameter placeholders, you do not need to enter any values for the Description and Value parameters in the table. You can simply click on the **Execute** button to run the query and display the results in the next tab.

## 14.2.1.4. View and Insert Parameters Tab

In the View and Insert Parameters tab, users can view and manipulate the output of the SQL query (entered in the [DataSource](#) tab). In addition to viewing the query results, users can also modify the output by adding new columns that are not part of the original query and inserting values into them.

The screenshot shows the 'KPI Properties' dialog box with the 'View and Insert Parameters' tab selected. The dialog has five tabs: 'Naming', 'Datasource', 'Execute Parameters', 'View and Insert Parameters', and 'Timing'. The 'View and Insert Parameters' tab contains the following sections:

- Add Column:** A section with a 'Column Name' text input field, a 'Column Type' dropdown menu, and two buttons: a plus sign (+) to add a column and a minus sign (-) to remove a column.
- Unique Column:** A large text area for selecting a unique column from the query results.
- Set Data:** A large greyed-out area for setting data values.

At the bottom of the dialog, there are four buttons: 'Back', 'Next', 'Save & Exit', and 'Cancel'.

1. **Add Column:** In this section, users can add a new column to the table.
  - **Column Name:** This field allows users to input the name of the column. It is a **mandatory** field.
  - **Column Type:** This field allows users to choose the column type from a dropdown menu. It is a **mandatory** field.
  - : This button allows users to add the column to the table.
  - : This button allows users to remove a selected column from the table.
2. **Unique Column:** In this section, users can establish a link between a new column and an existing query column.
  - Each column in the table is listed in this field with a checkbox next to it, allowing users to select which column will serve as the unique identifier for the new column. By selecting a column as the unique identifier, the values in the new column will correspond to the values in the selected column.

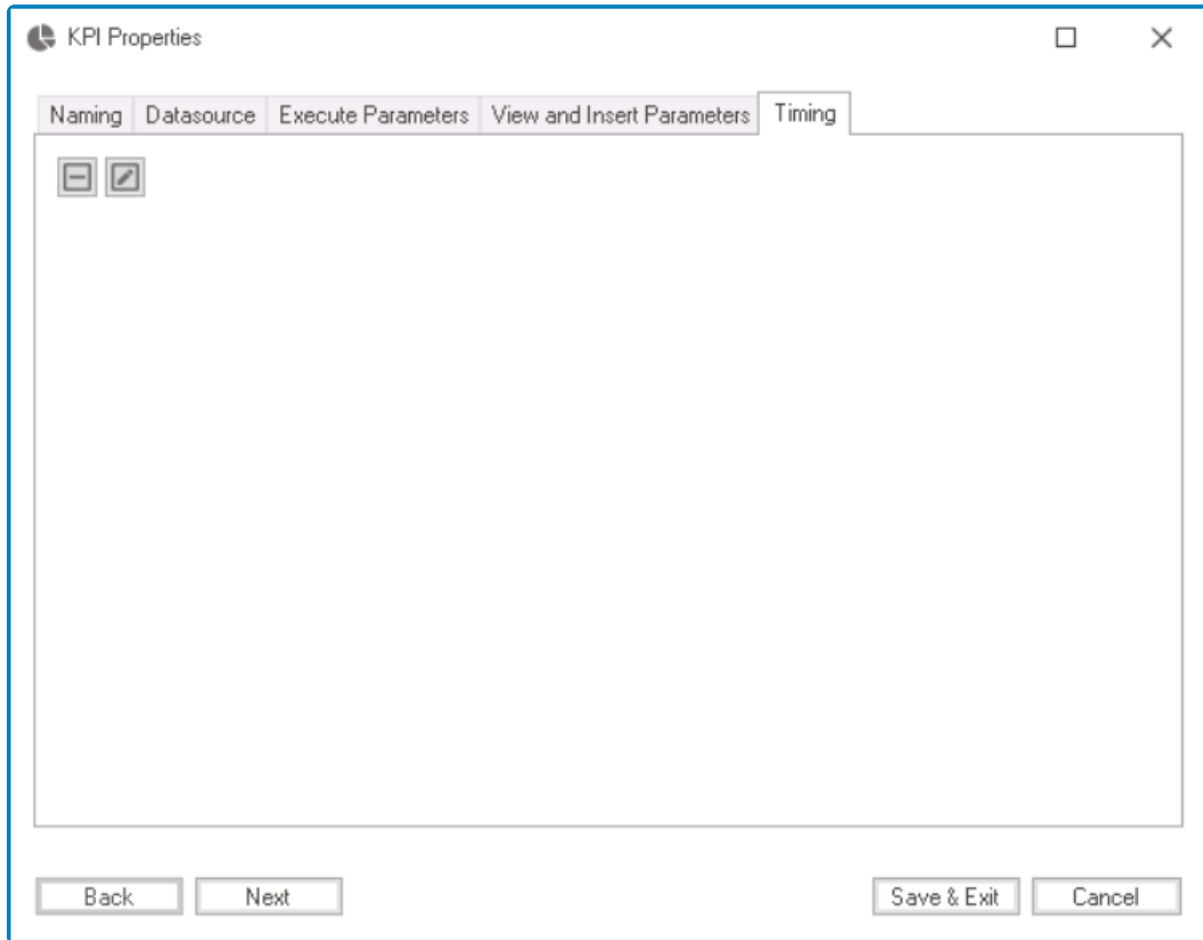



In order to save their custom data, users must select a unique column.

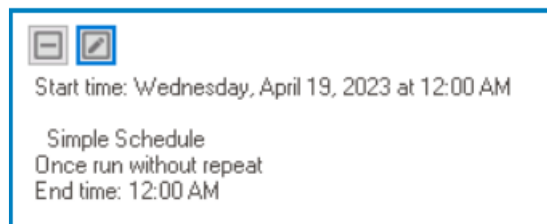
3. **Set Data:** In this section, users can view and manipulate the output of the SQL query.
  - To manipulate data inside the table, simply select the appropriate row and input the desired information.

## 14.2.1.5. Timing Tab

In the Timing tab, users can configure the KPI to run at a specific time or interval.



1.  : This button allows users to create a new schedule or modify an existing schedule for the KPI.
  - If selected, it will display the [Scheduler](#) window.
    - Once created and saved, the schedule will be displayed in the tab.

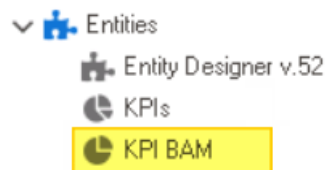


2.  : This button allows users to remove the schedule.



## 14.3. KPI BAM

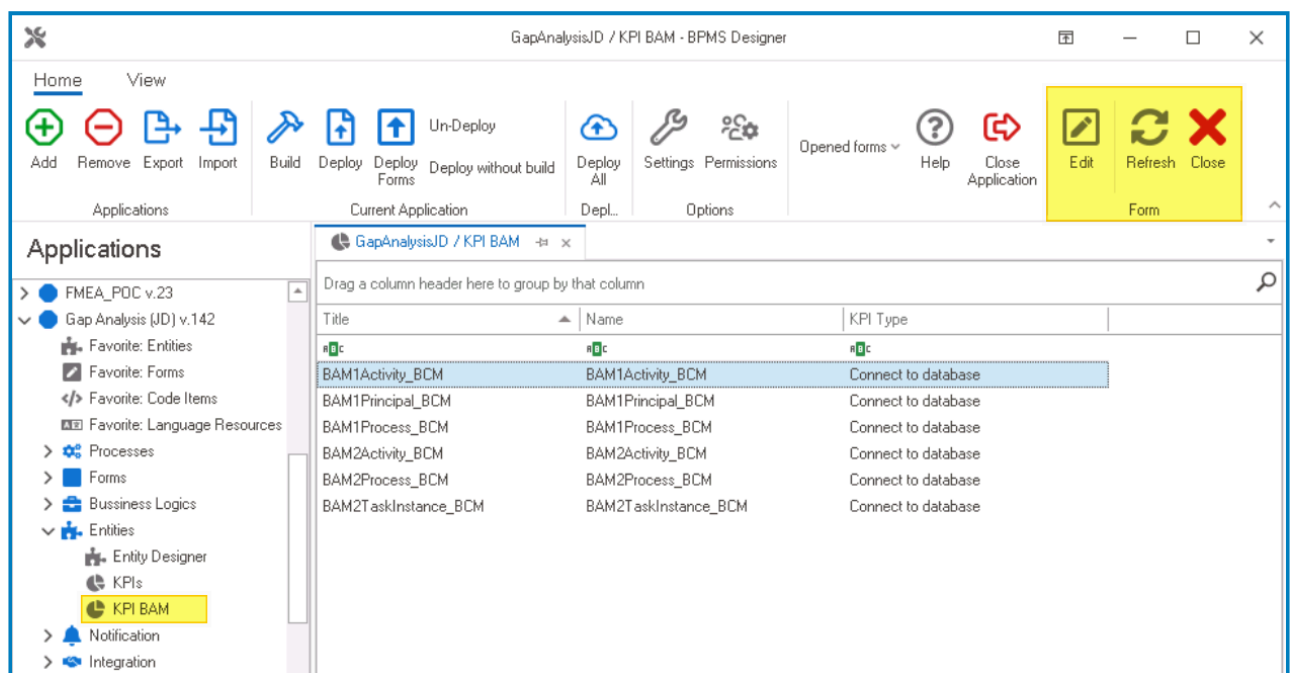
In the **KPI BAM** section of this module, users can view a list of key performance indicators for a specific business process or activity in the application.



\* **KPI = Key Performance Indicator / BAM = Business Activity Monitoring**

To manage KPIs in the application, take the following steps:

1. From the [Application Tree](#), navigate to **Entities > KPI BAM** (double-click to open). It will display the **KPI BAM** tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined KPIs in the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **Edit** : This button allows users to edit the properties of an existing KPI.
  - If selected, it will display the [KPI Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select a KPI from the list.
- **Refresh** : This button allows users to refresh the KPI BAM tab.
- **Close**: This button allows users to close the KPI BAM tab.

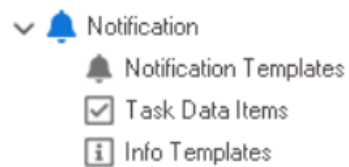
\* This list includes KPIs that are relevant to monitoring the performance of a process or activity. Users can edit the KPIs in the list to customize them to their specific needs.

# 15. Notification

---

The **Notification** module is a useful tool for customizing and managing notifications within the application workflow.

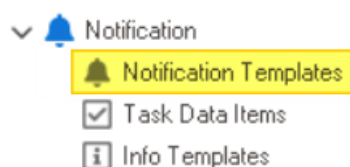
In the **Notification** module, users are provided access to the following elements:



1. [Notification Templates](#)
2. [Task Data Items](#)
3. [Info Templates](#)

## 15.1. Notification Templates

In the **Notification Templates** section of this module, users can create and customize notifications for the application.



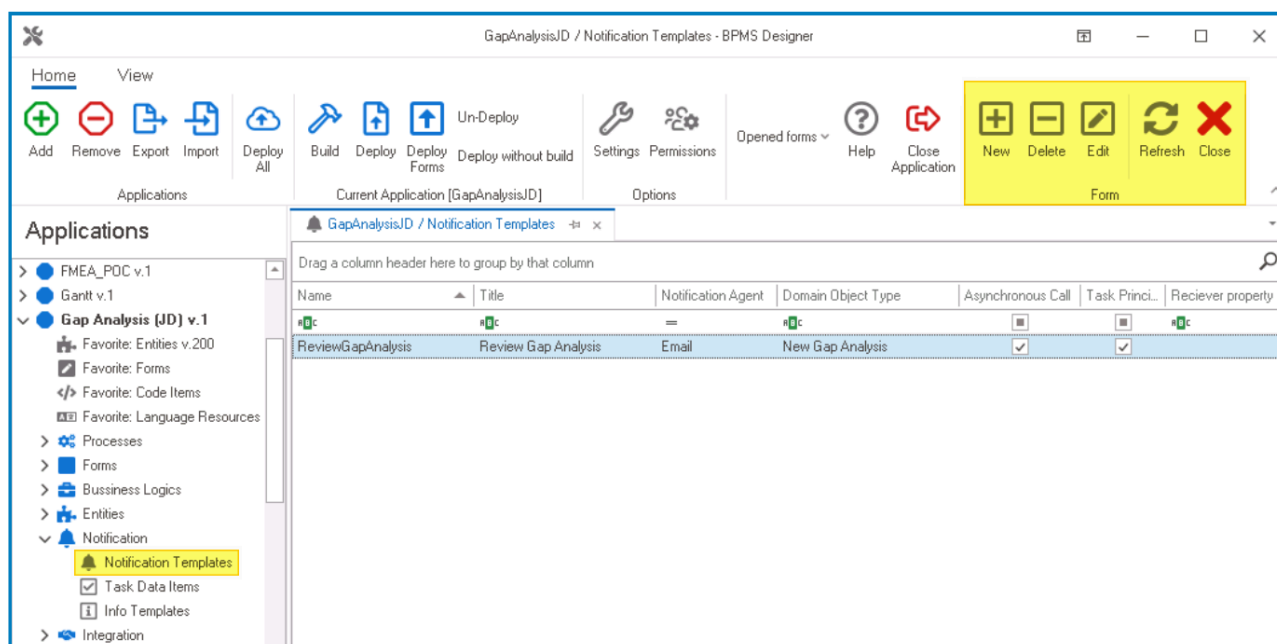
- ✿ Notification templates, or notification items, are predefined message formats used to alert users about specific situations within the application or workflow. They enable the system to send consistent and customized notifications via emails, text messages, or task instances based on predefined triggers or events.

There are two ways by which notification items can be defined, edited, and deleted in the program:

### 1. Menu Bar

To manage notification items using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Notifications > Notification Templates** (double-click to open). It will display the **Notification Templates** tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined notification items for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



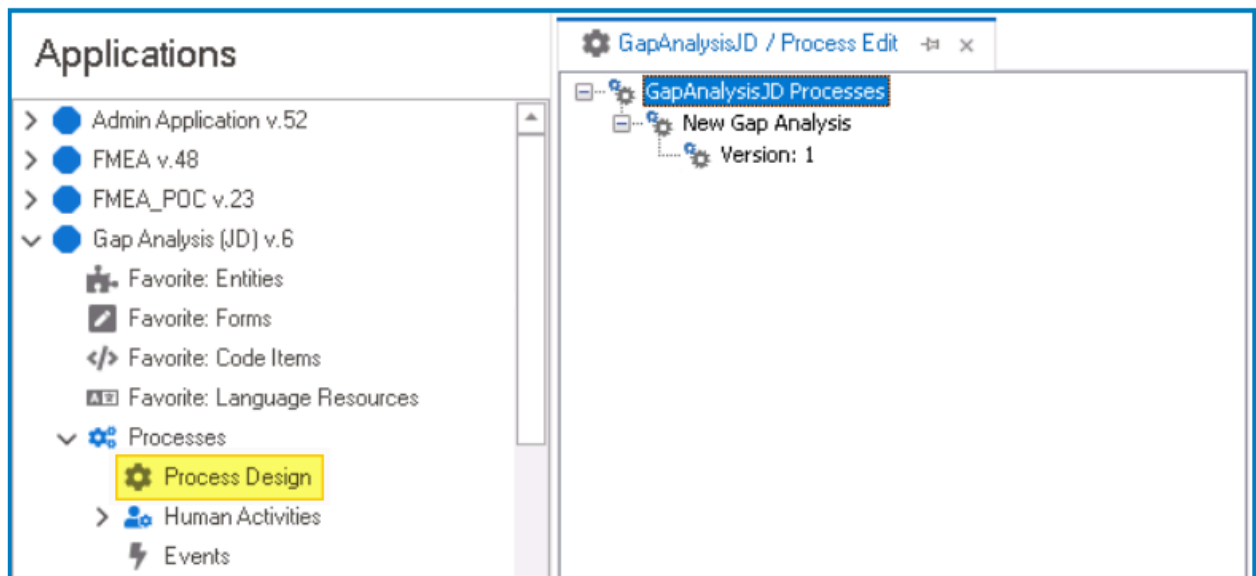
- **New:** This button allows users to define a new notification item.
  - If selected, it will display the [Notification Item Properties](#) window.
- **Delete:** This button allows users to delete an existing notification item.
  - To use this function, first select a notification item from the list.
- **Edit :** This button allows users to edit an existing notification item.
  - To use this function, first select a notification item from the list.
- **Refresh :** This button allows users to refresh the [Notification Templates](#) tab.
- **Close:** This button allows users to close the [Notification Templates](#) tab.

\* In order for a notification to be properly executed, it must be assigned to an activity in a process (as demonstrated in the set of steps below).

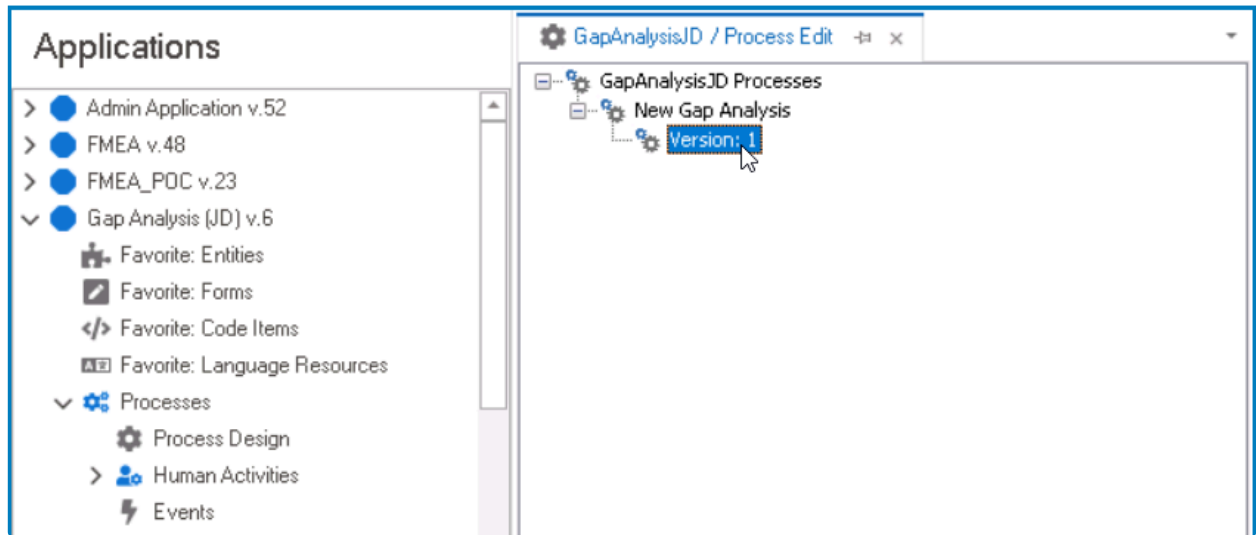
## 2. Object Properties

To manage notification items using the object properties, take the following steps:

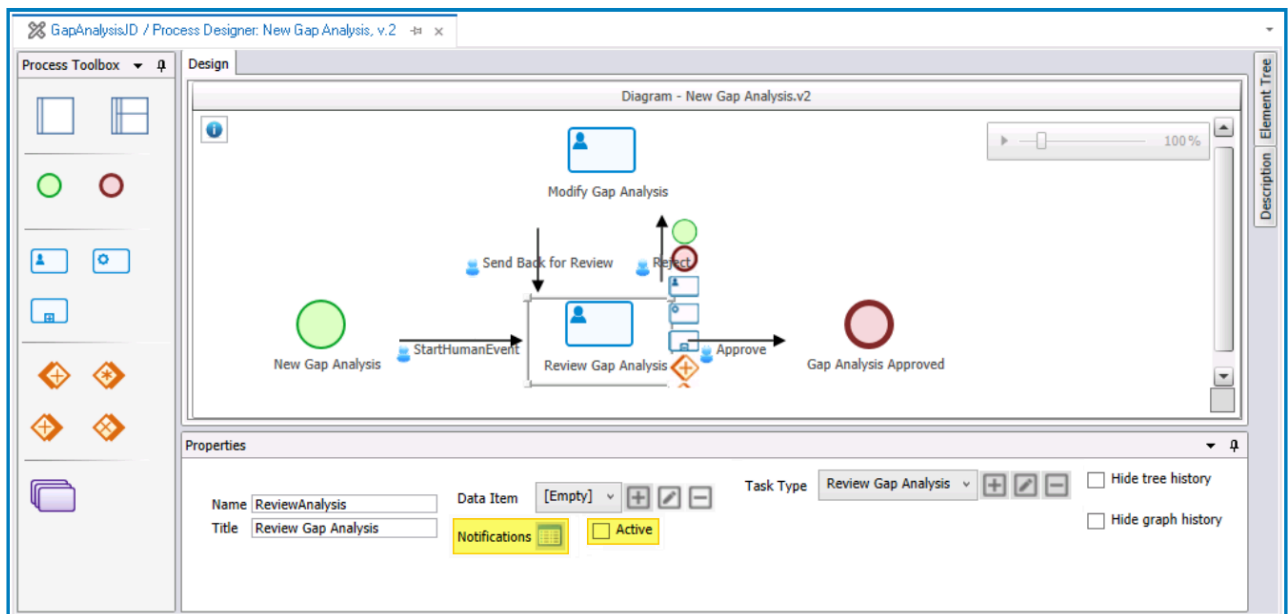
- From the [Application Tree](#), navigate to **Processes > Process Design** (double-click to open). It will display a new tab called [Process Edit](#).



- In the [Process Edit](#) tab, double-click on a process version. It will launch the [Process Designer](#) in a new tab.




- In the diagram of the Process Designer, select an applicable flow object (human state, system state, subprocess, or end event). It will display its associated properties in the Properties panel.



- From the Properties panel, first select the **Active** checkbox.



- Next, select the **Notifications**  icon. It will display the Activity Notifications window.

Activity Notifications

+ Add - Remove X Exit

Drag a column header here to group by that column

| Title | Notification Agent | Async                    | Principal                |
|-------|--------------------|--------------------------|--------------------------|
| ABC   | =                  | <input type="checkbox"/> | <input type="checkbox"/> |

Record 0 of 0

- **Add:** This button allows users to add a new notification.
  - Selecting this button will display the [Notifications](#) window, in which users can either define new notifications or select existing ones (if applicable) from the list.

Notifications

Home

+ New - Delete Edit Select Refresh Close

Form

Drag a column header here to group by that column

| Name              | Title               | Notification Agent | Domain Object Type | Asynchronous Call                   | Task Principi...                    | Reciever |
|-------------------|---------------------|--------------------|--------------------|-------------------------------------|-------------------------------------|----------|
| ReviewGapAnalysis | Review Gap Analysis | Email              | New Gap Analysis   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |          |

Record 1 of 1

- **New:** This button allows users to create a new notification item.
  - It will display the [Notification Item Properties](#) form.
- **Delete:** This button allows users to delete a notification item.
- **Edit:** This button allows users to edit an existing notification item.
- **Select:** This button allows users to select a notification item (to be triggered during the process).
  - Selecting this button will redirect users back to the [Activity Notifications](#)

window—the notification item now listed therein.

- Users can also select a notification item by double-clicking on it.
- **Refresh:** This button allows users to refresh the window.
- **Close:** This button allows users to close the Notifications window..
- **Remove:** This button allows users to remove a notification.
- **Exit:** This button allows users to exit the Activity Notifications window.



Once set up, these notifications will be assigned to the activity and executed at the applicable stage in the process.



## 15.1.1. Notification Item Properties

In the Notification Item Properties window, users can create a new notification item for the application.

1. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the notification item.
2. **Name:** This is a text field that allows users to input the name of the notification item. It is a **mandatory** field.
  - There are the following guidelines determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - A notification item must have a unique name. It cannot have the same name as another rule in the program.
3. **Title:** This is a text field that allows users to input the title of the notification item. It is a **mandatory** field.
  - Unlike with names, there are no guidelines determining the validity of titles.
4. **Subject Data Item:** This is a single-select dropdown field that retrieves the list of data items (from the [Task Data Item](#) section of the **Notification** module). It allows users to select the data item that will dynamically populate the subject or title of the notification.
  - To define a new data item, select the button.
    - It will display the [Data Item Properties](#) window.
  - To modify an existing data item, select the button.
  - To delete a data item, select the button.
5. **Body Data Item:** This is a single-select dropdown field that retrieves the list of data items (from

the [Task Data Item](#) section of the **Notification** module). It allows users to select the data item that will dynamically populate the body or text of the notification.

- As with the previous field, users can utilize the corresponding buttons to create, edit, and/or delete data items.

6. **Notification Agent:** This is a single-select dropdown field that displays the list of notification agents. It allows users to select the type of notification they want to use for the current notification item.

- **Notification Agents**

1. **Email:** This option allows users to send the notification as an email to the recipient's registered email address.
2. **SMS:** This option allows users to send the notification as a text message to the recipient's registered phone number.
3. **Task List:** This option allows users to send the notification as a new task instance to the assigned task type recipient.



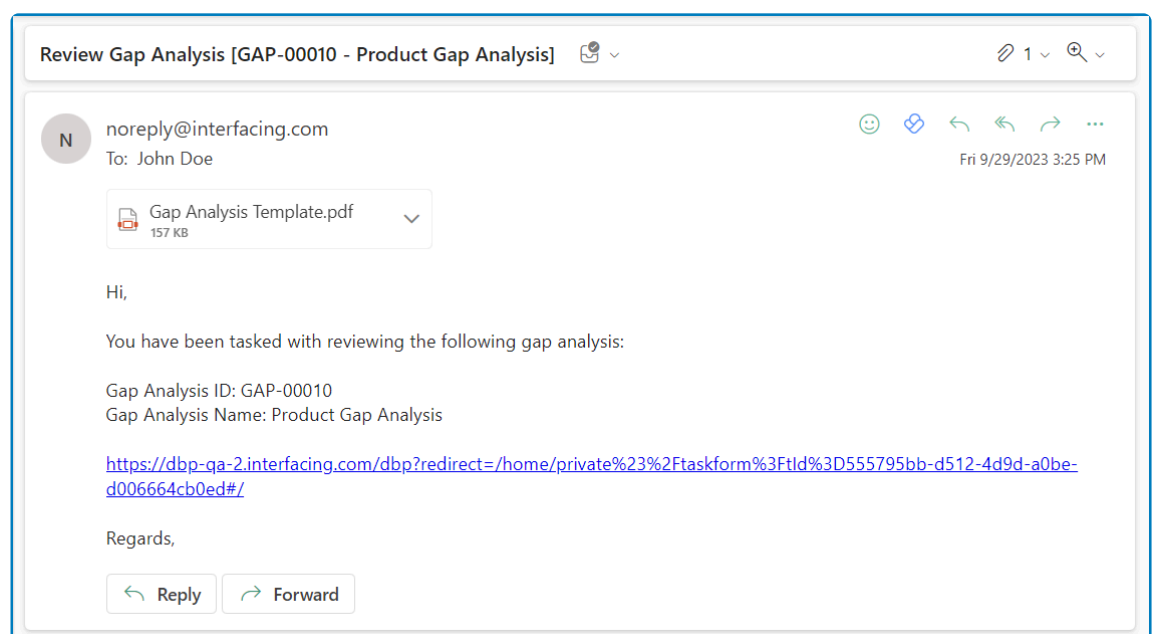
To configure the email and SMS settings, refer to the [Settings](#) of the Designer.

7. **Attach Exported File:** This is a checkbox which, if selected, allows users to attach the exported template file to their (email) notification.

- **Template Attachment – Workflow Implementation**

In a practical workflow, each task (e.g., [Task A](#), [Task B](#), [Task C](#)) is assigned its unique template and notification. Enabling the **Attach Exported File** checkbox for each task ensures that its corresponding template is attached to the email notification.

For example, if [Task A](#) has an assigned template, enabling this option will attach [Task A](#)'s template to the email, and the same applies to [Task B](#) and [Task C](#).




8. **Asynchronous Call:** This checkbox determines whether or not the notification call will be

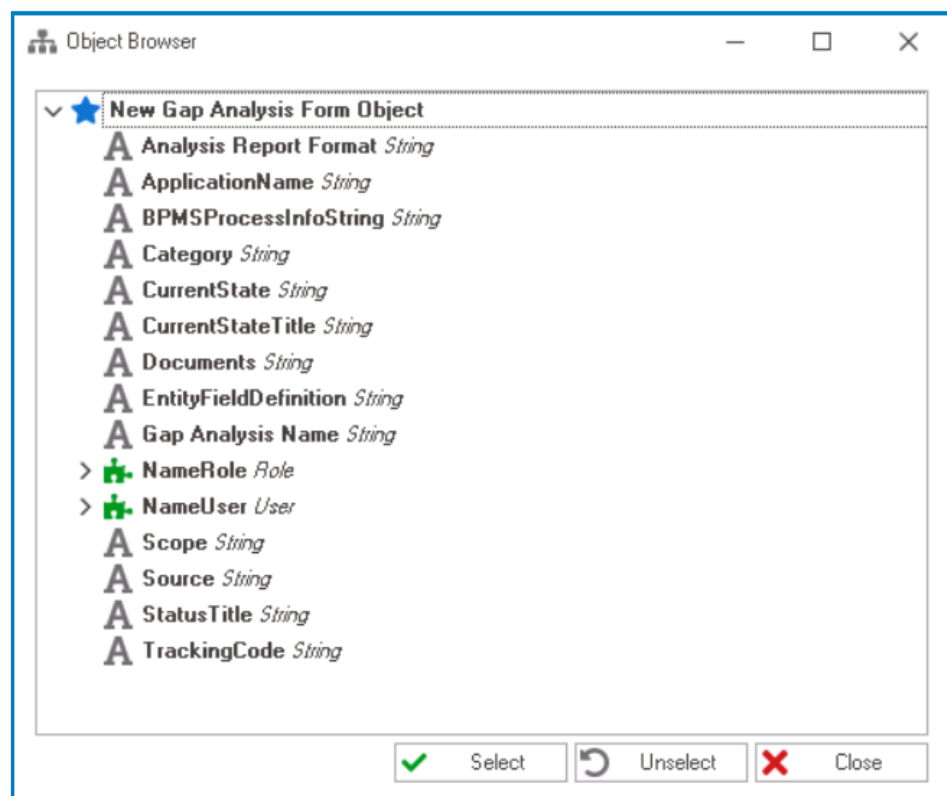
asynchronous.

- If selected, the system will process notifications in batches without any tagging-out pauses.

✳ This feature is useful for performance optimization, particularly when dealing with time-consuming notification operations.

9. **Recipients:** In this section, users can specify the recipients of the notification.

-  : This button allows users to select the object property or attribute that will define the recipient of the notification.
  - If selected, it will display the Object Browser window.




✳ In the Object Browser window, users can access a list of the associated entity's attributes. To select an attribute, either click on the \*Select button or double-click on the entry from the list.

! Please note that the chosen attribute from this list must be linked to a system table, specifically the user table.




- Once an attribute has been selected, it will populate the **Object Property** field.

Object Property   

-  : This button allows users to remove the selected object property.
- **Task Principals:** This is a checkbox which, if selected, allows users to send the notification

to the task's principals, i.e., the assignees of the task.

10. **Task Type:** This is a single-select dropdown field that retrieves the list of task types (from the [Task Types](#) section of the **Processes** module). It allows users to select the specific task type to which the notification will be associated.

- To define a new task type, select the  button.
  - It will display the [Task Type Properties](#) window.
- To modify an existing task type, select the  button.
- To delete a task type, select the  button.



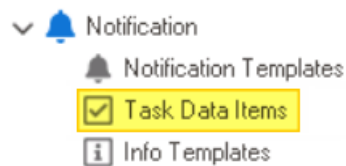
When the notification is linked to a task type, it will be sent to the assignees or recipients of that particular task type.



To save changes, click on the **Apply** button. To save changes and exit the [Notification Item Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 15.2. Task Data Items

In the **Task Data Items** section of this module, users can create and customize task data items for the application.



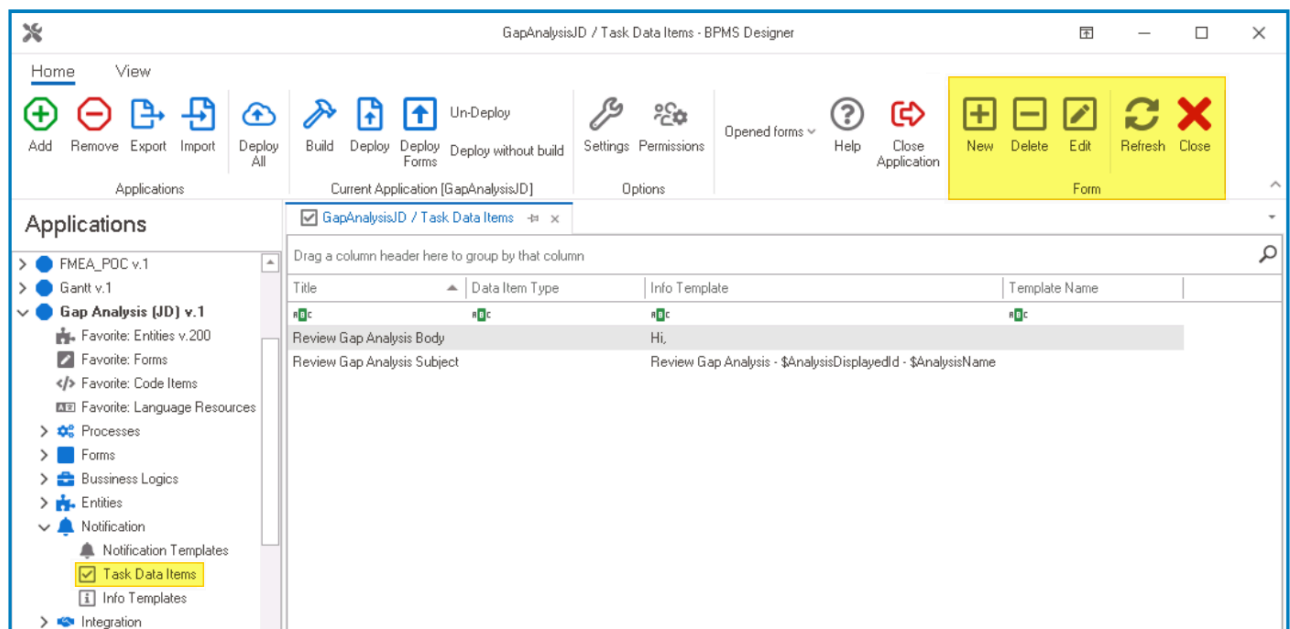
- ✿ Task data items are the content components of a notification, used to convey information to the recipients. They include the subject and body of the notification, acting as placeholders for dynamic details like the user's name, task specifics, or other relevant data.

There are three ways by which data items can be defined, edited, and deleted in the program:

### 1. Menu Bar

To manage data items using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Notifications > Task Data Items** (double-click to open). It will display the **Task Data Items** tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined data items for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



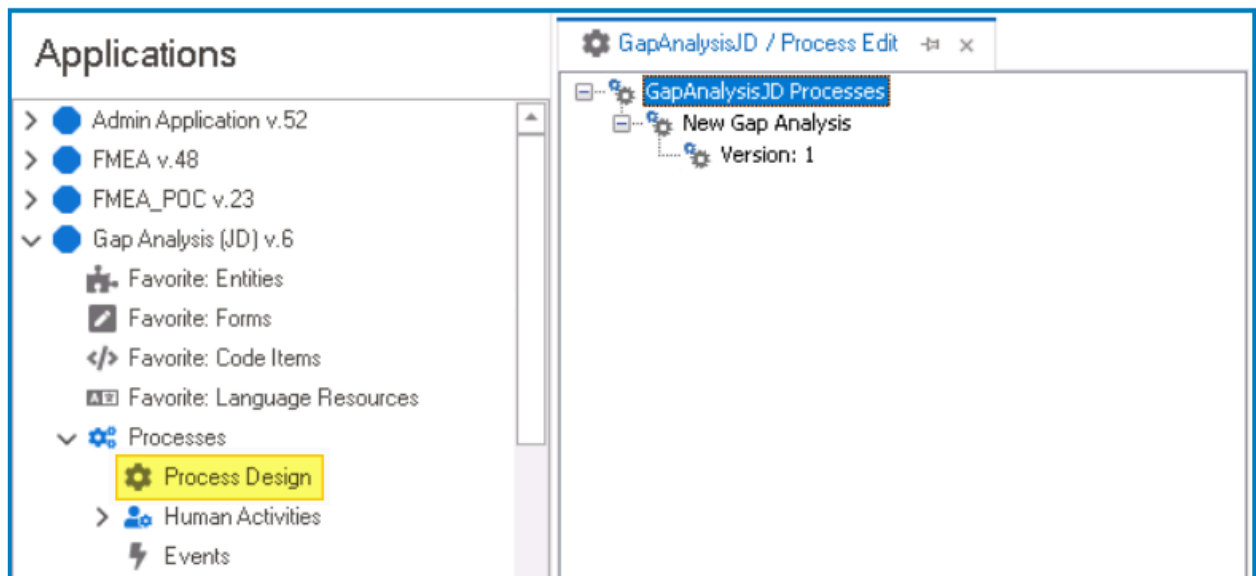
- **New:** This button allows users to define a new data item.
  - If selected, it will display the [Data Item Properties](#) window.
- **Delete:** This button allows users to delete an existing data item.
  - To use this function, first select a data item from the list.
- **Edit :** This button allows users to edit an existing data item.
  - To use this function, first select a data item from the list.
- **Refresh :** This button allows users to refresh the [Task Data Items](#) tab.
- **Close:** This button allows users to close the [Task Data Items](#) tab.

✿ To properly set up a task data item, users must link it to a notification (as demonstrated in the final set of steps listed below).

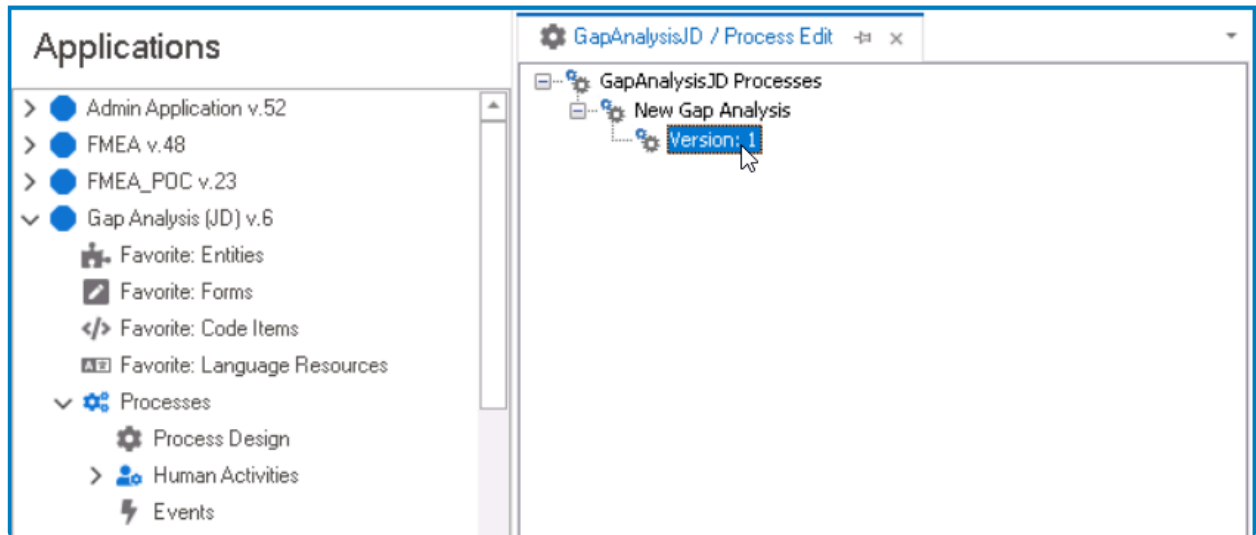
## 2. Object Properties

To manage data items using the object properties, take the following steps:

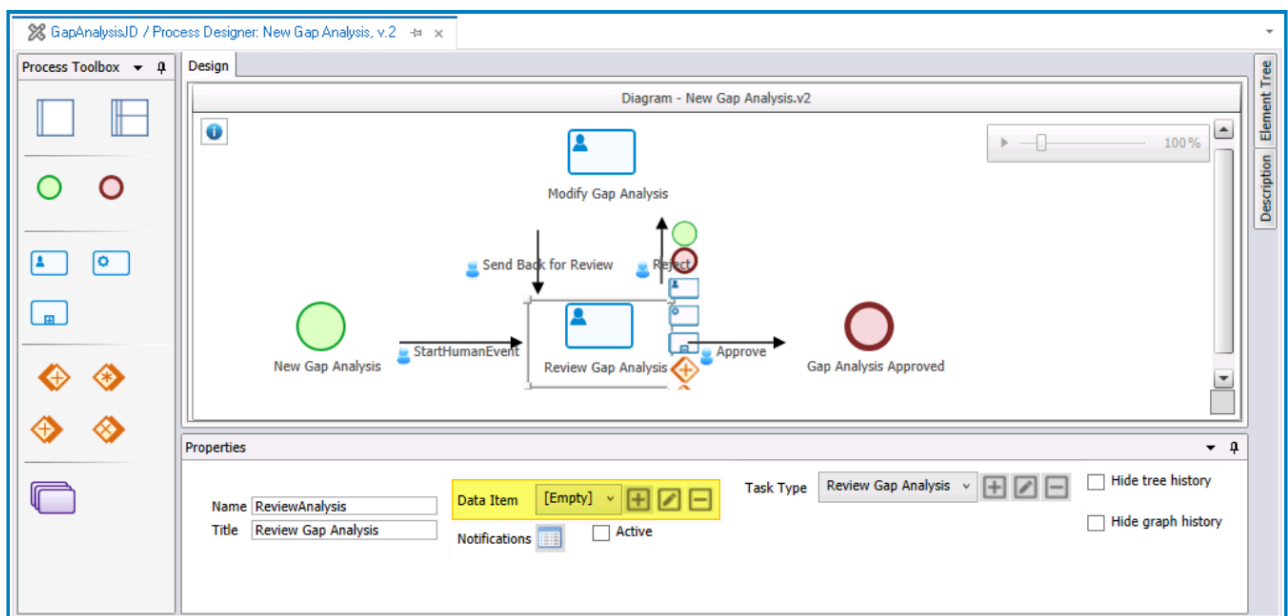
- From the [Application Tree](#), navigate to **Processes > Process Design** (double-click to open). It will display a new tab called [Process Edit](#).



- In the [Process Edit](#) tab, double-click on a process version. It will launch the [Process Designer](#) in a new tab.



- In the diagram of the Process Designer, select an applicable flow object (human state, system state, subprocess, or end event). It will display its associated properties in the Properties panel.



- From the Properties panel, select the desired option next to **Data Items**.

Data Item [Empty] + -

- To define a new data item, select the button.
  - It will display the [Data Item Properties](#) window.
- To modify an existing data item, select the button.
- To delete a data item, select the button.

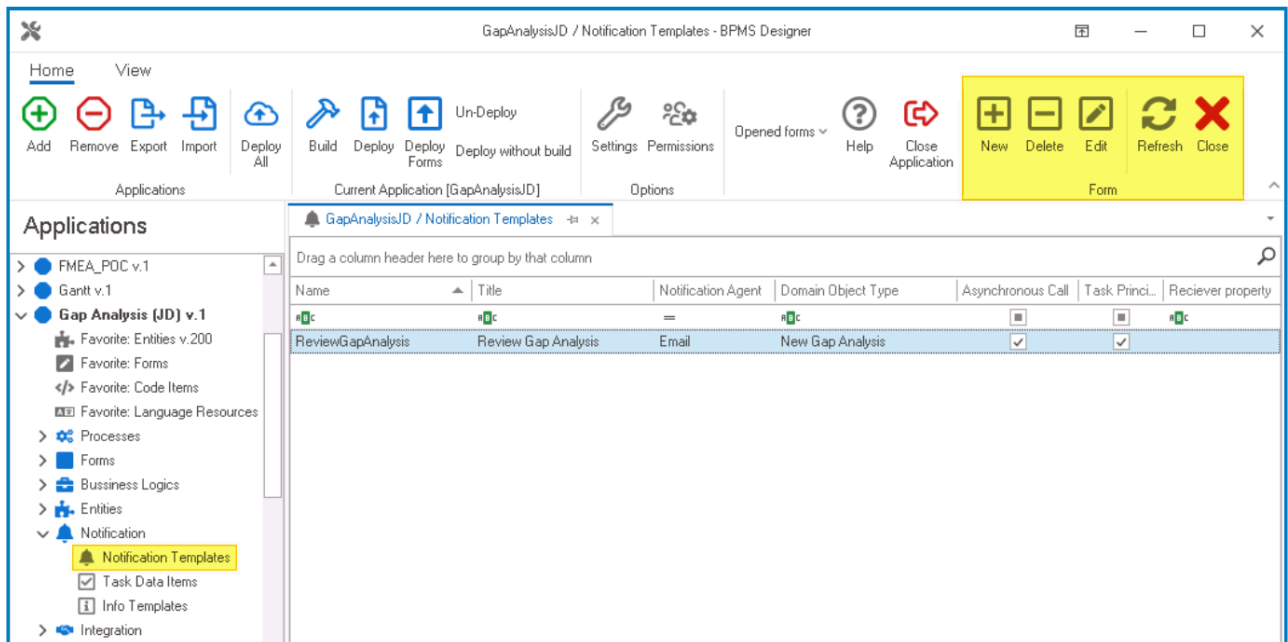


Once set up, this data item will be assigned to the activity and executed at the applicable stage in the process.

### 3. Notification Item Properties

To manage data items using the notification item properties, take the following steps:

- From the [Application Tree](#), navigate to **Notifications > Notification Templates** (double-click to open). It will display the [Notification Templates](#) tab and expand the menu bar to show additional options.



- From the menu bar, select the **New** button. It will display the [Notification Item Properties](#) window.




The screenshot shows the 'Notification Item Properties' dialog box. It has a title bar with a close button. Below the title bar are buttons for 'Apply', 'OK', and 'Cancel'. The dialog contains several fields and checkboxes:

- Entity:** A dropdown menu showing 'New Gap Analysis (New Gap Analysis)'.
- Name:** A text input field.
- Title:** A text input field.
- Subject Data Item:** A dropdown menu showing '[Empty]', with a yellow highlight and icons for adding, editing, and deleting.
- Body Data Item:** A dropdown menu showing '[Empty]', with a yellow highlight and icons for adding, editing, and deleting.
- Notification Agent:** A dropdown menu showing 'Default'.
- Asynchronous Call:** A checked checkbox.
- Recipients:** A section containing an 'Object Property' text input field, a 'Task Principals' checkbox, and a red minus button.
- Task Type:** A dropdown menu showing '[Empty]', with icons for adding, editing, and deleting.

- In this window, users can define the data item that will dynamically fill the subject or title of the notification, along with the data item that will dynamically populate the body or text of the notification.

This is a close-up of the 'Subject Data Item' and 'Body Data Item' fields from the previous screenshot. Both fields are highlighted in yellow and show '[Empty]' with icons for adding, editing, and deleting.



- To define a new data item, select the  button.
  - It will display the [Data Item Properties](#) window.
- To modify an existing data item, select the  button.
- To delete a data item, select the  button.

## 15.2.1. Data Item Properties

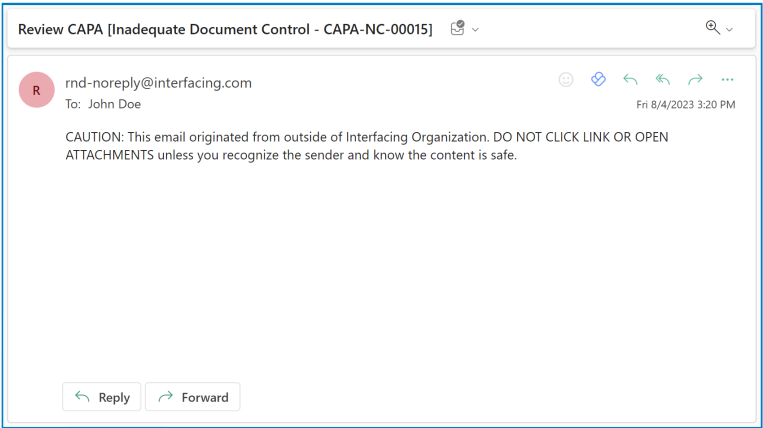
In the Data Item Properties window, users can create a new data item for the application.

1. **Title:** This is a text field that allows users to input the title of the data item. It is a **mandatory** field.
2. **Template Text:** In this section, users can define the template text that forms the content of the notification, serving as either its subject or body. This content can be tailored for different languages as required.
  - **Default:** This checkbox determines the default language and template text for the notification.
    - By default, this checkbox is selected for the English language tab. Users, however, can modify this selection based on their preferences.

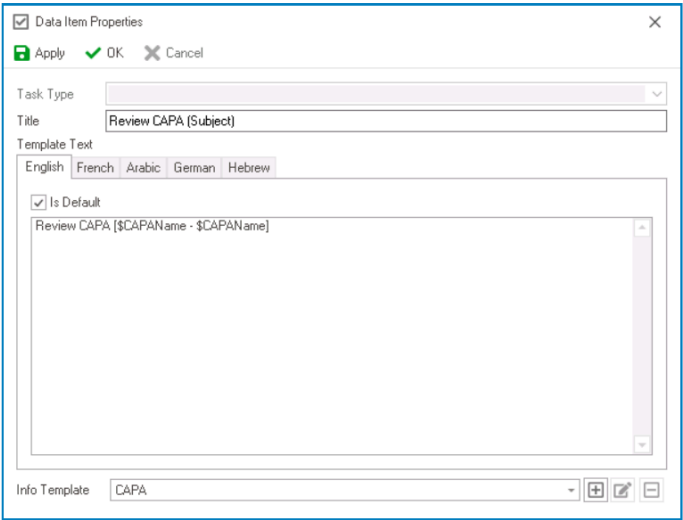


Only one language can be set as the default at a time.

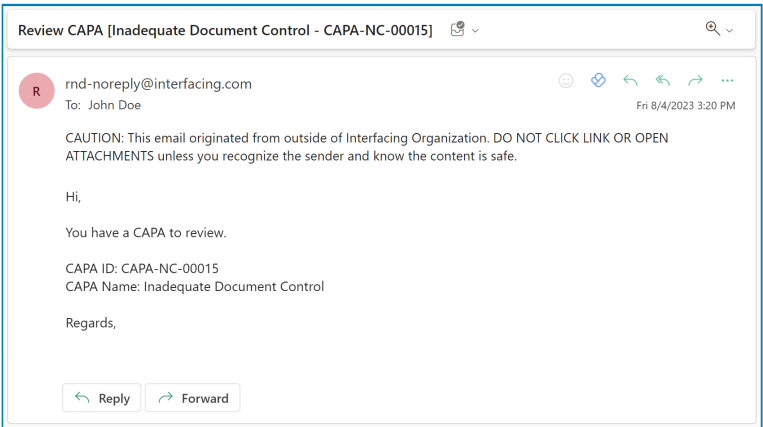
- Users can input static or dynamic text within these fields.
  - **Tips & Tricks for Incorporating Dynamic Template Text**
    - To incorporate dynamic text, users can utilize the \$ sign followed by the **name of the attribute**.
      - **Example: Notification Subject**
        - **Email:** Review CAPA [**Inadequate Document Control – CAPA-NC-00015**]



- Text: Review CAPA [**\$CAPAName** – **\$CAPAID**]



- **Example: Notification Body**
  - Email: CAPA ID: **CAPA-NC-00015**  
CAPA Name: **Inadequate Document Control**



- Text: CAPA ID: **\$CAPAID**  
CAPA Name: **\$CAPAName**

☒ Data Item Properties

Apply OK Cancel

Task Type

Title Review CAPA (Body)

Template Text

English French Arabic German Hebrew

☒ Is Default

Hi,  
You have a CAPA to review.  
CAPA ID: \$CAPAID  
CAPA Name: \$CAPAName  
Regards,

Info Template CAPA

- To include a link to the associated task, users can add %TaskLink%.
  - **Example: Task Link in Notification Body**
  - Email:

Review CAPA [Inadequate Document Control - CAPA-NC-00015]

rnd-noreply@interfacing.com  
To: John Doe  
Fri 8/4/2023 3:20 PM

CAUTION: This email originated from outside of Interfacing Organization. DO NOT CLICK LINK OR OPEN ATTACHMENTS unless you recognize the sender and know the content is safe.

Hi,  
You have a CAPA to review.  
CAPA ID: CAPA-NC-00015  
CAPA Name: Inadequate Document Control  
<https://dbp-dev-1.interfacing.com/dbp?redirect=/home/private%23%2Ftaskform%3Fid%3Dfc0b7453-1a35-4842-af45-602d9e017667#/>  
Regards,

Reply Forward

- Text: %TaskLink%

☒ Data Item Properties

Apply OK Cancel

Task Type

Title Review CAPA (Subject)

Template Text




English French Arabic German Hebrew

☒ Is Default

Review CAPA [\$CAPAName - \$CAPAName]

Info Template CAPA

- \* Users can determine which languages are available for use in the [Settings](#) of the Designer. Currently, however, only five languages are supported on the web server: English, French, German, Arabic, and Hebrew.

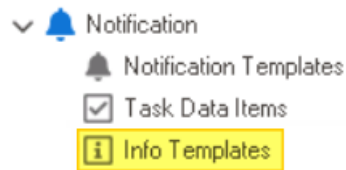
3. **Info Template:** This is a single-select dropdown field that retrieves the list of active info templates (from the [Info Templates](#) section of the **Notification** module). It allows users to select the info template associated with the data item.
- To define a new info template, select the  button.
    - It will display the [Info Template Properties](#) window.
  - To modify an existing info template, select the  button.
  - To delete an info template, select the  button.

- \* For effective integration of dynamic text into the data item, users must assign an info template.

- \* To save changes, click on the **Apply** button. To save changes and exit the [Data Item Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 15.3. Info Templates

In the **Info Templates** section of this module, users can create and customize info templates for the application.



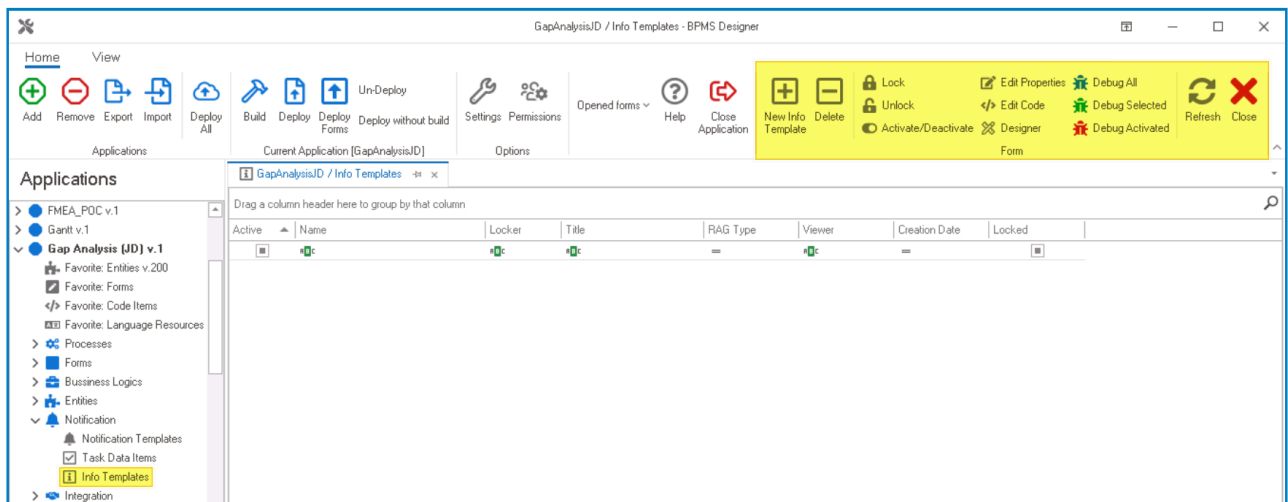
\* Info templates are dynamic parameters used to read process information and extract data from forms. They enable users to incorporate information directly from a particular form into the notification content.

There are three ways by which info templates can be defined, edited, and deleted in the program:

### 1. Menu Bar

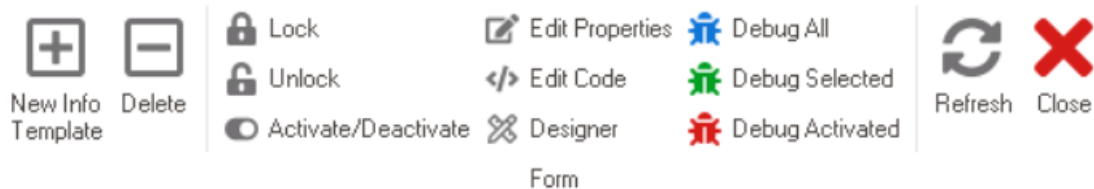
To manage info templates using the menu bar, take the following steps:

- From the [Application Tree](#), navigate to **Notifications > Info Templates** (double-click to open). It will display the **Info Templates** tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined info templates for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

- From the menu bar, select the desired option.



- **New:** This button allows users to define a new info template.
  - If selected, it will display the [Info Template Properties](#) window.
- **Delete:** This button allows users to delete an existing info template.
  - To use this function, first select an info template from the list.
- **Lock:** This button allows users to lock the info template, restricting access to it so that only one user can make changes to it at a time.
- **Unlock:** This button allows users to unlock the info template.
  - If selected, access restriction on the info template will be lifted.
- **Activate/Deactivate:** This is a toggle switch that allows users to choose between activating and deactivating the info template.

! Only active info templates are displayed in the **Info Template** dropdown of the [Data Item Properties](#) window.

- **Edit Properties:** This button allows users to edit the properties of an existing info template.
  - If selected, it will display the [Info Template Properties](#) window. In this version of the window, however, all fields are populated with previously inputted information.
  - To use this function, first select an info template from the list.
- **Edit Code:** This button allows users to launch the [Code Designer](#), in which users can write or edit code specifying the behavior of the info template.
  - For more details on how to define info template using the [Code Designer](#), [click here](#).
    - **Adding Code to Info Templates**

Info templates are a key tool for injecting dynamic data into notifications. By understanding the standard code structure, users can craft contextually relevant info templates.

#### • Standard Code Structure

```
parameters.Add("[Name of Attribute]", domainObject.[Attribute] ?? "");
```

##### ◦ Code Components

- **'parameters':** This container stores the dynamic data intended for the notification.
- **'Add("[Name of Attribute]", domainObject.[Attribute])':** This line incorporates the value of a specified attribute from a domain object into the **parameters** container.
  - Users should modify this line to include the desired attribute name and its location within

the domain object.

- '?? " ": This part ensures that if the attribute's value is null or empty, it will be replaced with an empty string.

- **Example: Adding a Title Attribute**

```
parameters.Add("Title", domainObject.Title ?? "");
```

- **Code Components**

- **'Title'**: Denotes the specific attribute being retrieved.
  - **'domainObject.Title'**: Points to the attribute within the domain object.
- **Designer**: This button allows users to launch the [Rule Designer](#), in which they can define rules that specify the conditions and actions for the info template.
    - **For more details on how to define info template using the Rule Designer, [click here](#).**
  - **Debug All**: This button allows users to debug all info templates in the list.
  - **Debug Selected**: This button allows users to debug only the selected info template in the list.
  - **Debug Activated**: This button allows users to debug only the info templates that are tagged as active.
  - **Refresh** : This button allows users to refresh the [Info Templates](#) tab.
  - **Close**: This button allows users to close the [Info Templates](#) tab.



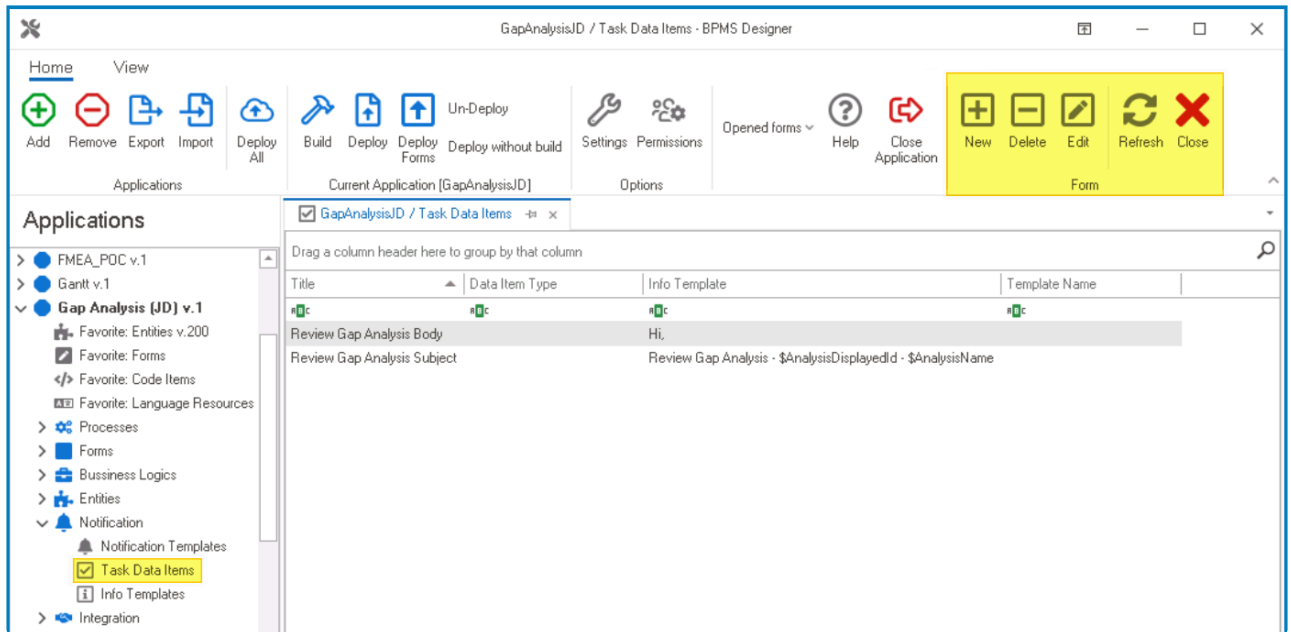
To properly set up an info template, users must link it to a data item (as demonstrated in the set of steps below).

## 2. Task Data Item Properties

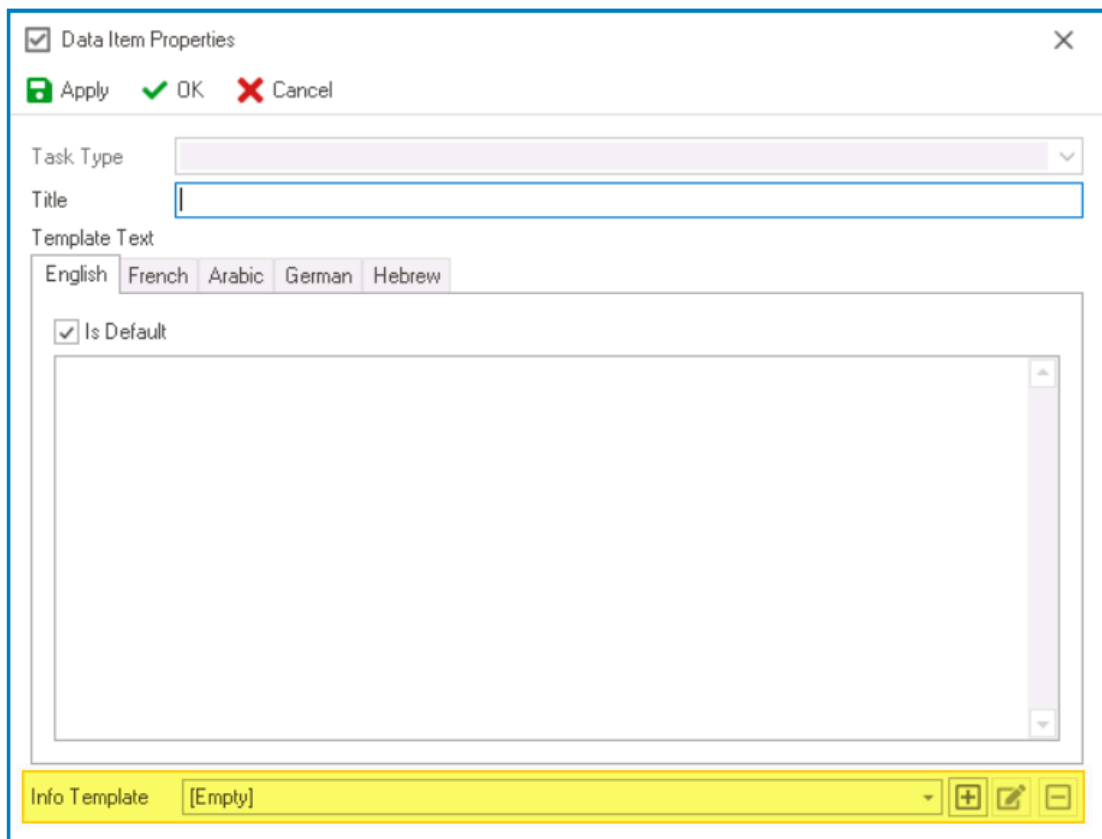
To manage info templates using the task data item properties, take the following steps:

- From the [Application Tree](#), navigate to **Notifications > Task Data Items** (double-click to open). It will display the [Task Data Items](#) tab and expand the menu bar to show additional options.








- From the menu bar, select the **New** button. It will display the [Data Item Properties](#) window.



- In this window, users can define the info template that links dynamic content from a specific form directly into the data item.

Info Template [Empty]   

- To define a new info template, select the  button.
  - It will display the [Info Template Properties](#) window.
- To modify an existing info template, select the  button.
- To delete an info template, select the  button.

## 15.3.1. Info Template Properties

In the [Info Template Properties](#) window, users can create a new info template for the application.

1. **Name:** This is a text field that allows users to input the name of the info template. It is a [mandatory](#) field.
  - There are the following guidelines determining the validity of names:
    - A valid name can contain letters, numbers, and the underscore symbol. It cannot, however, contain spaces or other special characters. It also cannot *begin* with a number.
    - An info template must have a unique name. It cannot have the same name as another info template in the program.
2. **Title:** This is a text field that allows users to input the title of the info template.
  - Unlike with names, there are no guidelines determining the validity of titles.
3. **Active:** This is a checkbox which, if selected, allows users to mark the info template out as active.



Only active info templates are displayed in the **Info Template** dropdown of the [Data Item Properties](#) window.

4. **Entity:** This is a single-select dropdown field that retrieves the list of entities (from the [Entity Designer](#) section of the **Entities** module). It allows users to select the entity associated with the info template.

### Adding Code to Info Templates

Info templates are a key tool for injecting dynamic data into notifications. By understanding the standard code structure, users can craft contextually relevant info templates.

- **Standard Code Structure**

```
parameters.Add("[Name of Attribute]", domainObject.[Attribute] ?? "");
```

- **Code Components**

- **'parameters':** This container stores the dynamic data intended for the notification.

- **'Add("[Name of Attribute]", domainObject.[Attribute]'**: This line incorporates the value of a specified attribute from a domain object into the **parameters** container.
  - Users should modify this line to include the desired attribute name and its location within the domain object.
- **'?? " "'**: This part ensures that if the attribute's value is null or empty, it will be replaced with an empty string.

- **Example: Adding a Title Attribute**

```
parameters.Add("Title", domainObject.Title ?? "");
```

- **Code Components**

- **'Title'**: Denotes the specific attribute being retrieved.
- **'domainObject.Title'**: Points to the attribute within the domain object.



To save changes, click on the **Apply** button. To save changes and exit the [Info Template Properties](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

# 16. Integration

---

The **Integration** module facilitates communication between the application and external services, APIs, databases, and other third-party tools. It also supports referencing and collaboration with other applications within the program

In the **Integration** module, users are provided access to the following elements:



1. [Resources](#)
2. [WebService Connectors](#)
3. [Connections](#)
4. [References](#)

## 16.1. Resources

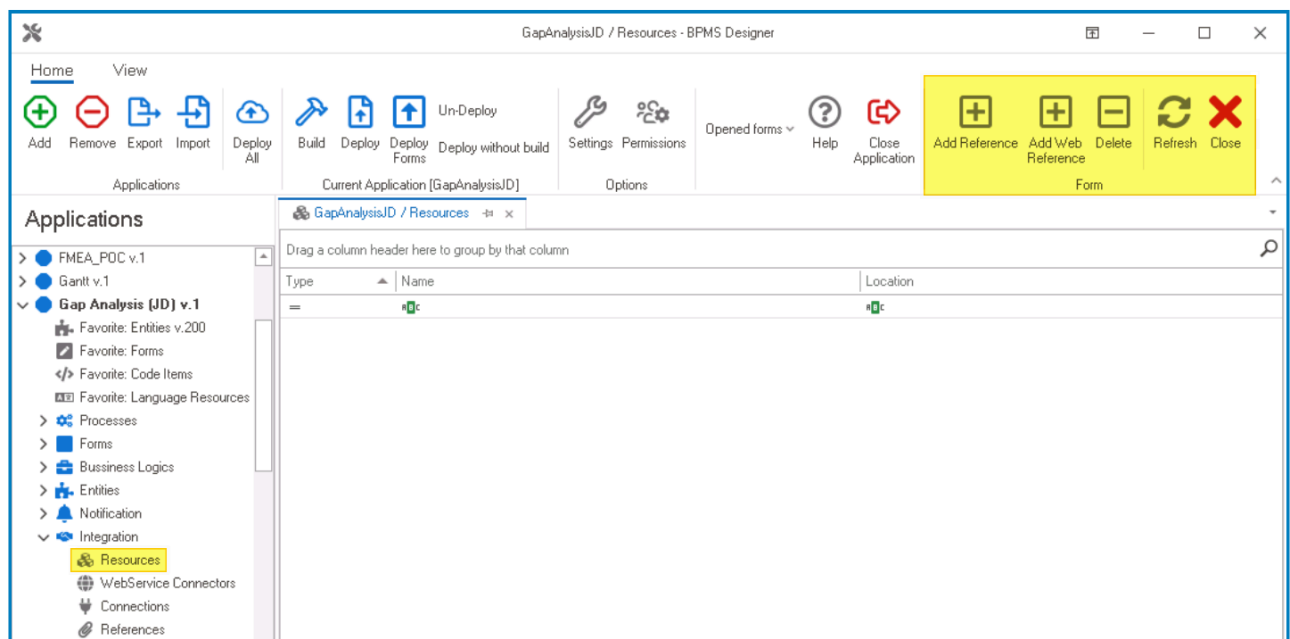
In the **Resources** section of this module, users can create resources for the application.



- ✿ These resources include assemblies, which enable integration of pre-built functionalities, and web references, which enable communication with external web services.

To manage resources in the program, take the following steps:

1. From the [Application Tree](#), navigate to **Integration > Resources** (double-click to open). It will display the **Resources** tab and expand the menu bar to show additional options.



- ✿ In this tab, users can access a list of all defined resources for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

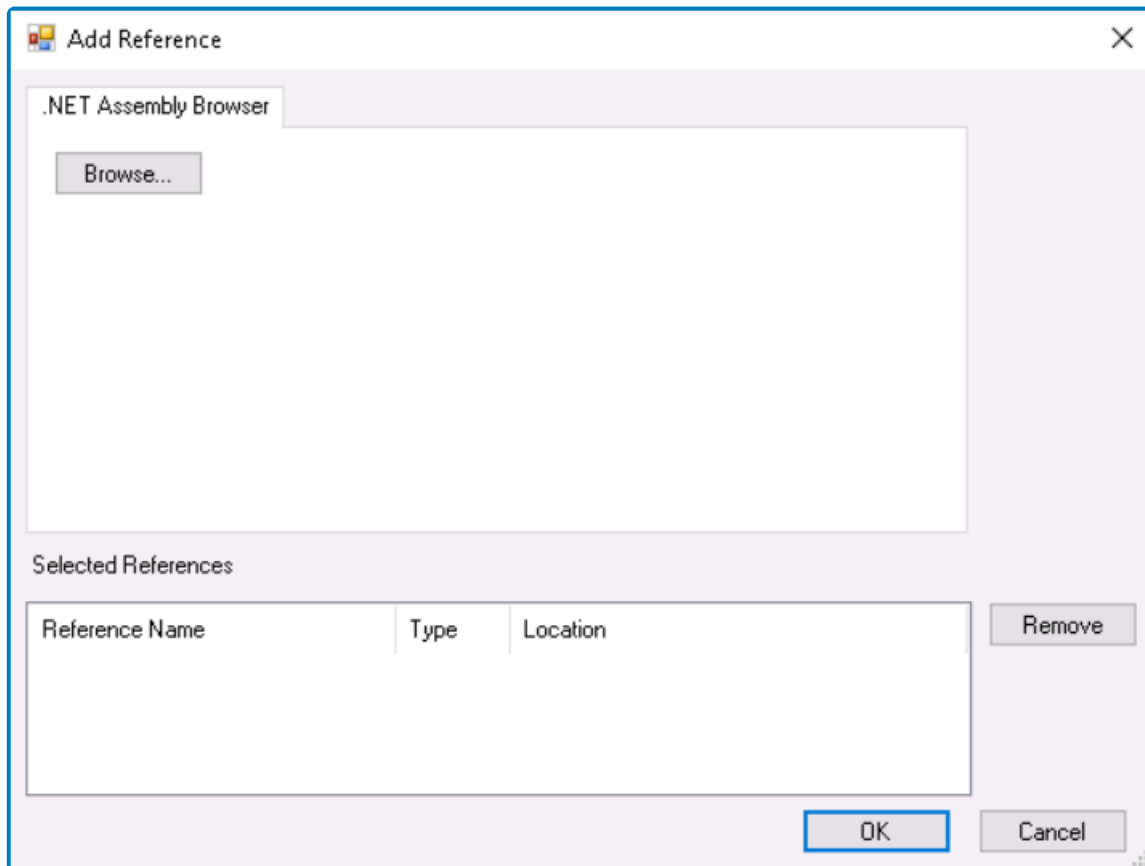
2. From the menu bar, select the desired option.



- **Add Reference:** This button allows users to add a new reference.
  - If selected, it will display the [Add Reference](#) window.
- **Add Web Reference:** This button allows users to add a new web reference.
  - If selected, it will display the [Add Web Reference](#) window.
- **Delete:** This button allows users to delete an existing resource.
  - To use this function, first select a resource from the list.
- **Refresh :** This button allows users to refresh the [Resources](#) tab.
- **Close:** This button allows users to close the [Resources](#) tab.

## 16.1.1. Add Reference

In the Add Reference window, users can create a new reference for the application.



1. **.Net Assembly Browser:** In this tab, users can browse and select assemblies—either executable (.exe) or dynamic link library (.dll) files—for integration into their application.
  - **Browse:** This button allows users to initiate the assembly selection process.
    - If selected, a file dialog will appear, enabling users to browse and choose the assemblies from their local system.
2. **Selected References:** This field displays all chosen assemblies, allowing users to view and manage the references they intend to integrate into their application.
  - **Remove:** This button allows users to remove a selected reference.
3. **OK:** This button allows users to save their changes and close the window.
4. **Cancel:** This button allows users to cancel their changes and close the window.

\* An **executable (EXE)** file is a type of file that contains a standalone program or application. It can be executed directly by the operating system, containing all necessary resources and instructions for the software to operate independently.

\* A **dynamic link library (DLL)** file is a type of file that programs can call upon to execute specific tasks. Unlike standalone executables, a **DLL** file doesn't operate independently; rather, it must be invoked by running code within an existing context.

## 16.1.2. Add Web Reference

In the Add Web Reference window, users can create a new web reference for the application.

\* These references enable communication between the **DBP** workflow management system and other systems in the code environment, facilitating the efficient transfer of information between interconnected systems.

1. **Address Bar:** This field allows users to input the address or URL of the desired web service.



- **Back:** This button allows users to navigate back to the previously visited web address.
  - **Forward:** This button allows users to move forward in their browsing history, if they have previously gone back.
  - **Refresh:** This button allows users to reload the current web page.
  - **Stop:** This button allows users to halt the loading process of the current web page.
  - **Go:** This button allows users to initiate the loading of the web page specified in the address bar.
2. **Type of Web Service:** In this section of the window, users can switch between different tabs to explore the contents of the chosen web service.
    - **WSDL:** If the web service adheres to the WSDL standard, users can access and explore its functionalities through this tab.

\* **WSDL = Web Services Description Language**

- **Available Web Services:** If the web service is accessible on the server, users can discover and access it from the list provided in this tab.
3. **Reference Name:** This is a text field that allows users to input a name for the web service reference, facilitating easy identification within the application.
  4. **Namespace:** This is a text field that allows users to define a namespace for the web service reference, aiding in categorization and organization.
    - **Example of Namespace Usage**  
For example, consider the incorporation of the line:

```
using System;
```

At the outset of a program, this allows the developer to employ the code:

```
Console.WriteLine("Hello, World!");
```



Instead of:

```
System.Console.WriteLine("Hello, World!");
```



Users can employ the designated name within this field to call upon the reference in the code environment.

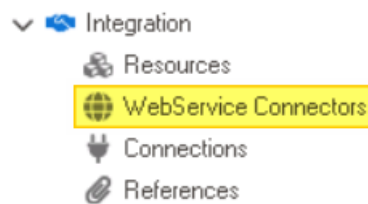


Namespaces are useful for logically grouping code and averting potential name conflicts, especially in scenarios where multiple libraries coexist within a codebase.

5. **Add:** This button allows users to integrate the selected web service reference into their application.
6. **Cancel:** This button allows users to cancel the process of adding the web service reference.

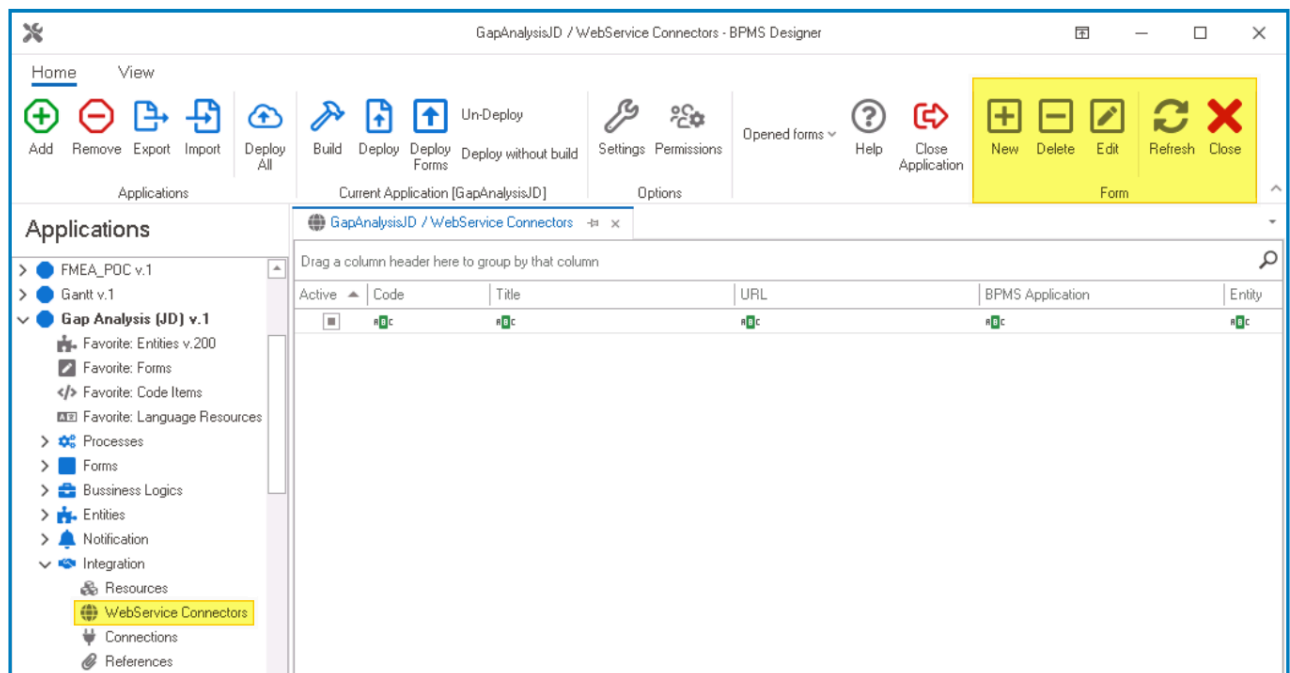
## 16.2. WebService Connectors

In the **WebService Connectors** section of this module, users can create connections between the application and external web services.



To manage web service connectors in the program, take the following steps:

1. From the [Application Tree](#), navigate to **Integration > WebService Connectors** (double-click to open). It will display the WebService Connectors tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined web service connectors for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new web service connector.
  - If selected, it will display the [WebService Connector](#) window.
- **Delete:** This button allows users to delete an existing web service connector.
  - To use this function, first select a web service connector from the list.
- **Edit :** This button allows users to edit an existing web service connector.
  - To use this function, first select a web service connector from the list.
- **Refresh :** This button allows users to refresh the [WebService Connectors](#) tab.
- **Close:** This button allows users to close the [WebService Connectors](#) tab.

## 16.2.1. WebService Connector

In the WebService Connector window, users can create a new connection to an external web service.

1. **Code:** This is a text field that allows users to input the code for the web service connection.
2. **Title:** This is a text field that allows users to input a title for the web service connection.
3. **Description:** This is a text field that allows users to provide a description of the web service connection.
4. **URL:** This is a text field that allows users to input the URL of the external web service to which the connection will be established.
5. **Application Name:** This field automatically displays the name of the application to which the web service connection is associated.
6. **Table Name:** This is a text field that allows users to specify the name of the table or entity associated with the web service connection.
7. **Event ID:** This is a text field that allows users to input the ID of the event associated with the web service connection.
8. **User ID:** This is a text field that allows users to input the ID of the user associated with the web service connection.
9. **Task Type ID:** This is a text field that allows users to input the ID of the task type associated with the web service connection.
10. **Active:** This is a checkbox which, if selected, allows users to mark the web service connection as active.
  - An active connection is available and operational within the application.
11. **By Tracking:** This is a checkbox which, if selected, allows users to categorize the web service connection by tracking criteria.
12. **By Task Type:** This is a checkbox which, if selected, allows users to categorize the web service

connection by task type.

- If this checkbox is selected, it will enable the **Task Type ID** field and disable the **By Tracking** field.



To save changes, click on the **Apply** button. To save changes and exit the [WebService Connector](#) window, click on the **OK** button. To cancel changes or exit the window, click on the **Cancel** button or the **Exit (X)** button.

## 16.3. Connections

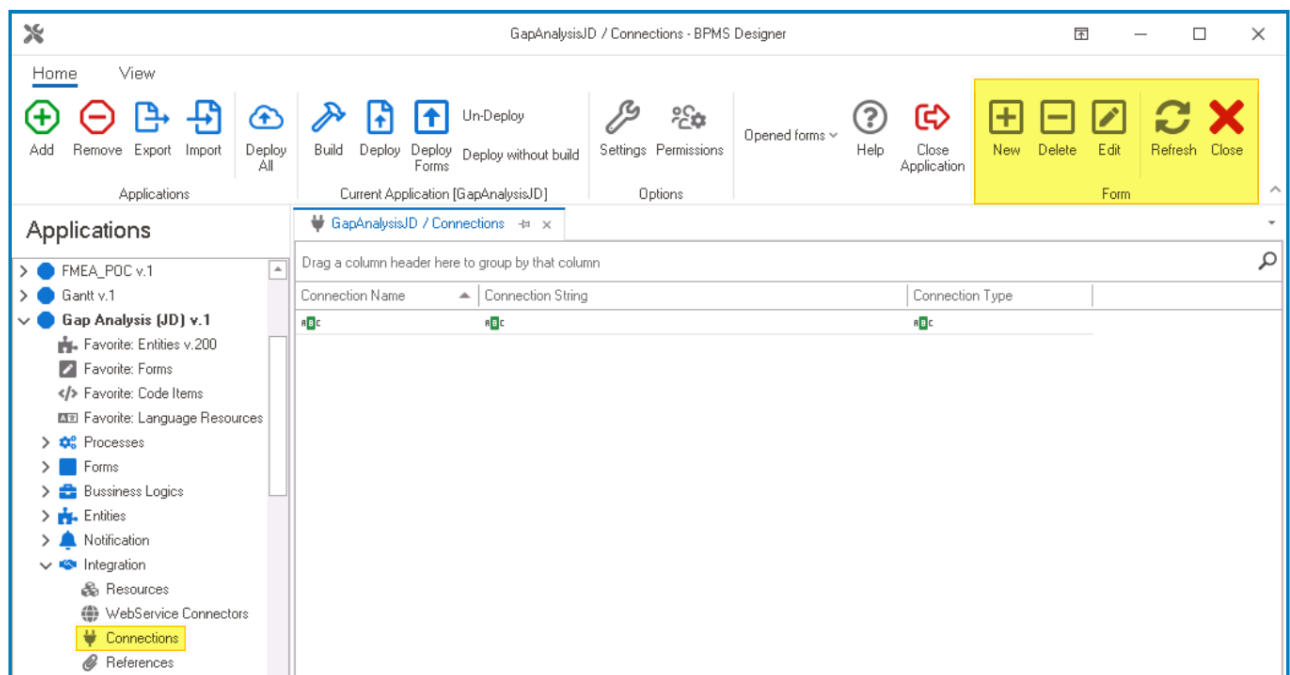
In the **Connections** section of this module, users can create connections between the application and external databases.



\* These database connections can be used to define [dropdown data sources](#) as well as [KPI data sources](#).

To manage database connections in the program, take the following steps:

1. From the [Application Tree](#), navigate to **Integration > Connections** (double-click to open). It will display the [Connections](#) tab and expand the menu bar to show additional options.



\* In this tab, users can access a list of all defined database connections for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

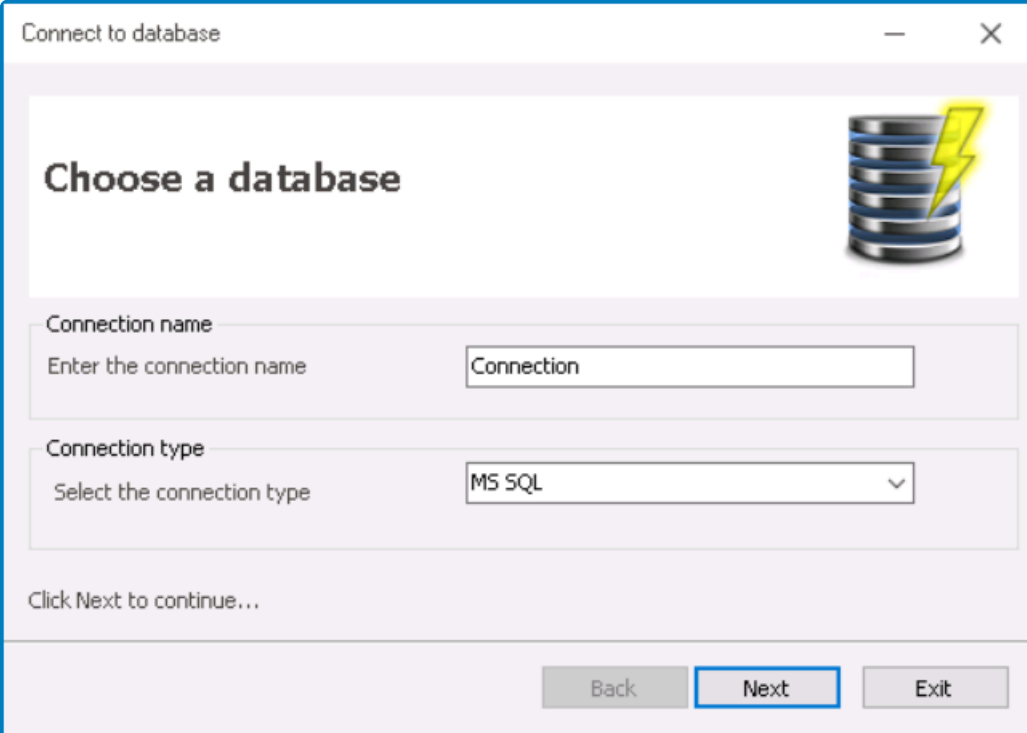
2. From the menu bar, select the desired option.



- **New:** This button allows users to define a new database connection.
  - If selected, it will display the [Connect to Database](#) window.
- **Delete:** This button allows users to delete an existing database connection.
  - To use this function, first select a database connection from the list.
- **Edit :** This button allows users to edit an existing database connection.
  - To use this function, first select a database connection from the list.
- **Refresh :** This button allows users to refresh the [Connections](#) tab.
- **Close:** This button allows users to close the [Connections](#) tab.

## 16.3.1. Connect to Database

In the Connect to Database window, users can configure connection settings and credentials for various database systems, including MySQL, OLE DB, ODBC, and Microsoft Access.



Connect to database

**Choose a database**

Connection name  
Enter the connection name

Connection

Connection type  
Select the connection type

MS SQL

Click Next to continue...

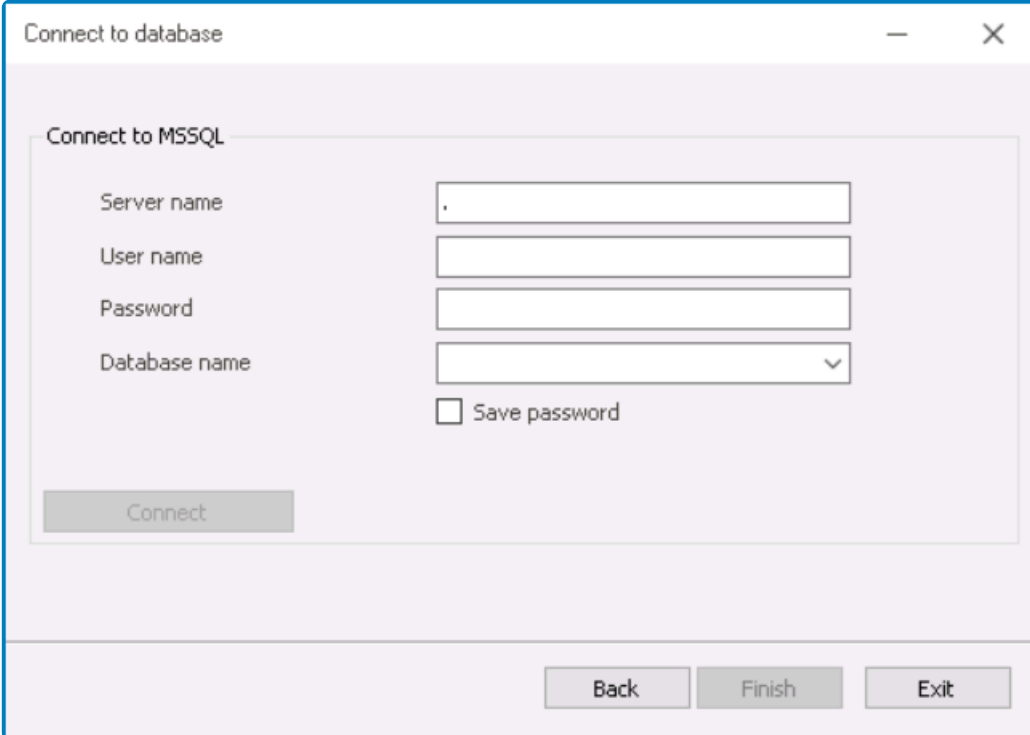
Back Next Exit

1. **Connection Name:** This is a text field that allows users to input the name of the database connection.
2. **Connection Type:** This field allows users to choose the connection type from a dropdown menu.
  - Depending on the connection type selected, the settings in the next step of the configuration process will adapt accordingly.
    - Connection Types:
      - [MySQL](#)
      - [OLE DB](#)
      - [ODBC](#)
      - [Access](#)
3. **Back:** This button allows users to navigate back to the previous step in the connection configuration process.
4. **Next:** This button allows users to proceed to the next step of the connection configuration process.
5. **Exit:** This button allows users to exit the configuration process and close the window.



## 16.3.1.1. MySQL

If MySQL is chosen as the connection type, the following configuration step will present options specifically tailored to MySQL settings.



1. **Server Name:** This is a text field that allows users to input the hostname of the MySQL server to which they want to establish a connection.
2. **User Name:** This is a text field that allows users to input their authentication username for accessing the specified server.
3. **Password:** This is a text field that allows users to input the password associated with the provided username.
  - Unlike a standard text field, this field masks the text input by displaying dots instead of actual characters.
4. **Database Name:** This is a single-select dropdown field that displays the list of databases associated with the specified server. It allows users to select the specific database to which they want to establish a connection.
5. **Save Password:** This is a checkbox which, if selected, allows users to save the entered password for future connections.
6. **Connect:** This button allows users to connect to the database, using the provided credentials and settings.

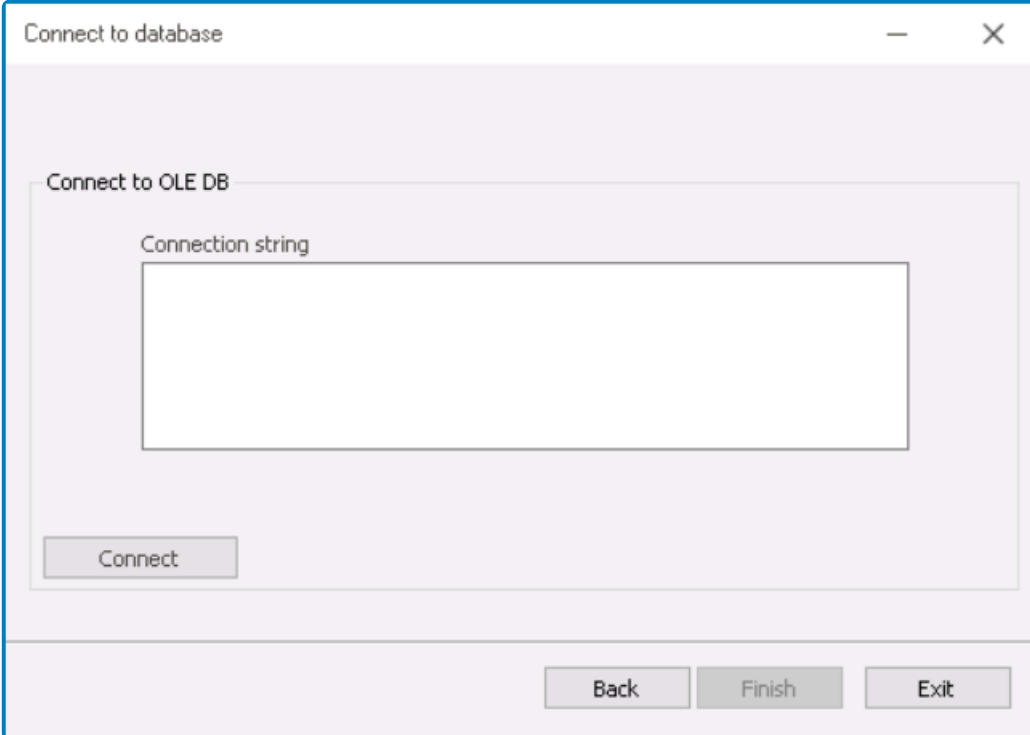
! Before a connection can be established, all provided information must undergo verification and confirmation. In case of any errors during this check, users will receive notifications through a dialog box.

7. **Back:** This button allows users to navigate back to the previous step in the connection configuration process.

- This action will return users to the [initial step](#) of the process.
8. **Finish:** This button allows users to finalize the connection configuration process.
    - If selected, the database connection will be created, saved, and recorded in the [Connections](#) tab.
  9. **Exit:** This button allows users to exit the configuration process and close the window.

## 16.3.1.2. OLE DB

If OLE DB is chosen as the connection type, the following configuration step will present options specifically tailored to OLE DB settings.



1. **Connection String:** This is a text field that allows users to input the connection string required to establish a connection to the database.

✿ Users should refer to the **MSDN Connection String Syntax** for the appropriate format and parameters.

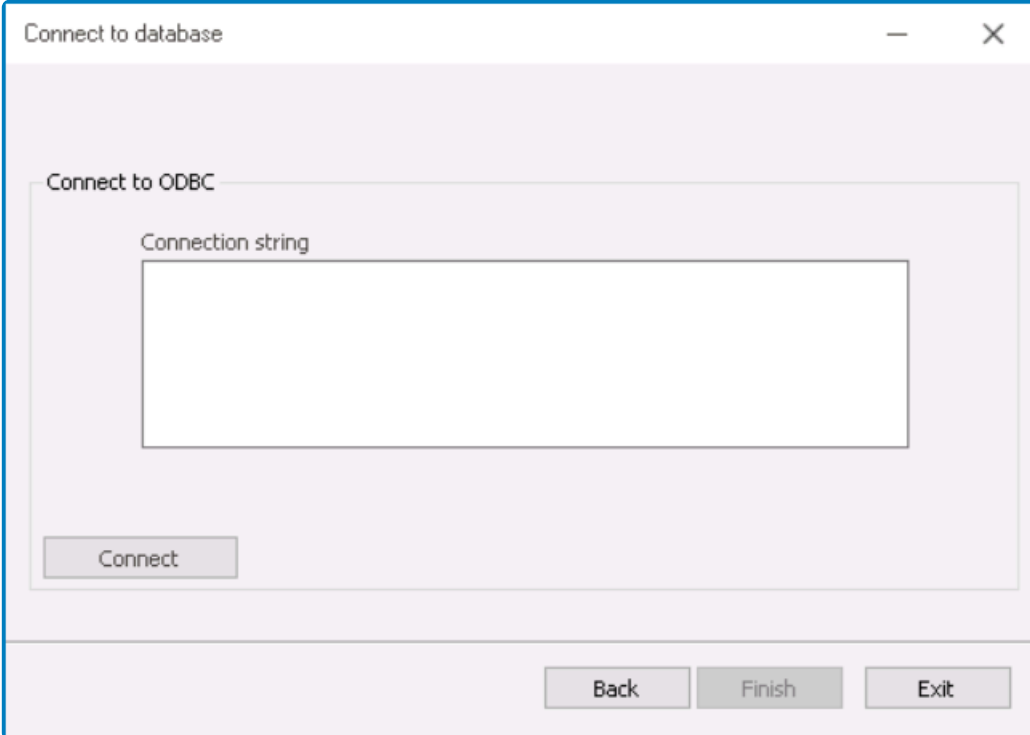
2. **Connect:** This button allows users to connect to the database, using the provided connection string.

! Before a connection can be established, all provided information must undergo verification and confirmation. In case of any errors during this check, users will receive notifications through a dialog box.

3. **Back:** This button allows users to navigate back to the previous step in the connection configuration process.
  - This action will return users to the [initial step](#) of the process.
4. **Finish:** This button allows users to finalize the connection configuration process.
  - If selected, the database connection will be created, saved, and recorded in the [Connections](#) tab.
5. **Exit:** This button allows users to exit the configuration process and close the window.

## 16.3.1.3. ODBC

If ODBC is chosen as the connection type, the following configuration step will present options specifically tailored to ODBC settings.



1. **Connection String:** This is a text field that allows users to input the connection string required to establish a connection to the database.

✿ For the correct format and parameters, users are advised to consult the documentation available at [www.connectionstrings.com](http://www.connectionstrings.com).

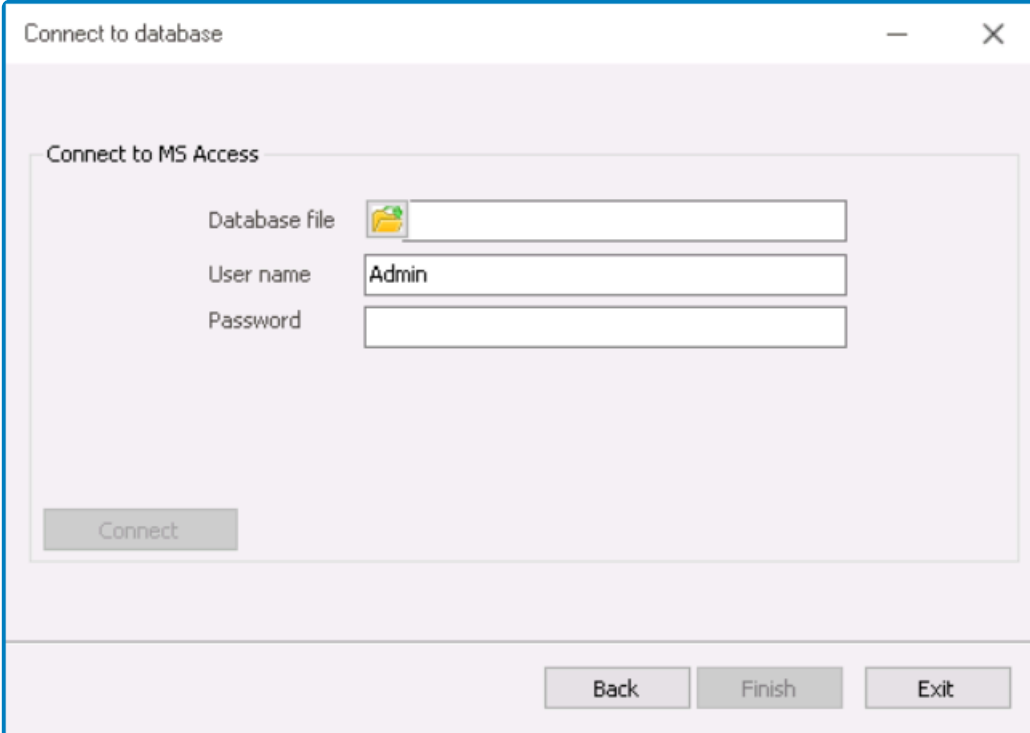
2. **Connect:** This button allows users to connect to the database, using the provided connection string.


! Before a connection can be established, all provided information must undergo verification and confirmation. In case of any errors during this check, users will receive notifications through a dialog box.

3. **Back:** This button allows users to navigate back to the previous step in the connection configuration process.
  - This action will return users to the [initial step](#) of the process.
4. **Finish:** This button allows users to finalize the connection configuration process.
  - If selected, the database connection will be created, saved, and recorded in the [Connections](#) tab.
5. **Exit:** This button allows users to exit the configuration process and close the window.

## 16.3.1.4. Access

If Access is chosen as the connection type, the following configuration step will present options specifically tailored to Microsoft Access settings.



1. **Database File:** This field allows users to specify the path to the database file.
  - If the  button is selected, a file dialog will appear, enabling users to browse and choose a file from their local system.
2. **User Name:** This is a text field that allows users to input their authentication username for accessing the database.
3. **Password:** This is a text field that allows users to input the password associated with the provided username.
  - Unlike a standard text field, this field masks the text input by displaying dots instead of actual characters.
4. **Connect:** This button allows users to connect to the database, using the provided credentials and settings.

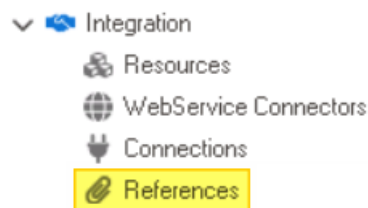
**!** Before a connection can be established, all provided information must undergo verification and confirmation. In case of any errors during this check, users will receive notifications through a dialog box.

5. **Back:** This button allows users to navigate back to the previous step in the connection configuration process.
  - This action will return users to the [initial step](#) of the process.
6. **Finish:** This button allows users to finalize the connection configuration process.
  - If selected, the database connection will be created, saved, and recorded in the [Connections](#) tab.

7. **Exit:** This button allows users to exit the configuration process and close the window.

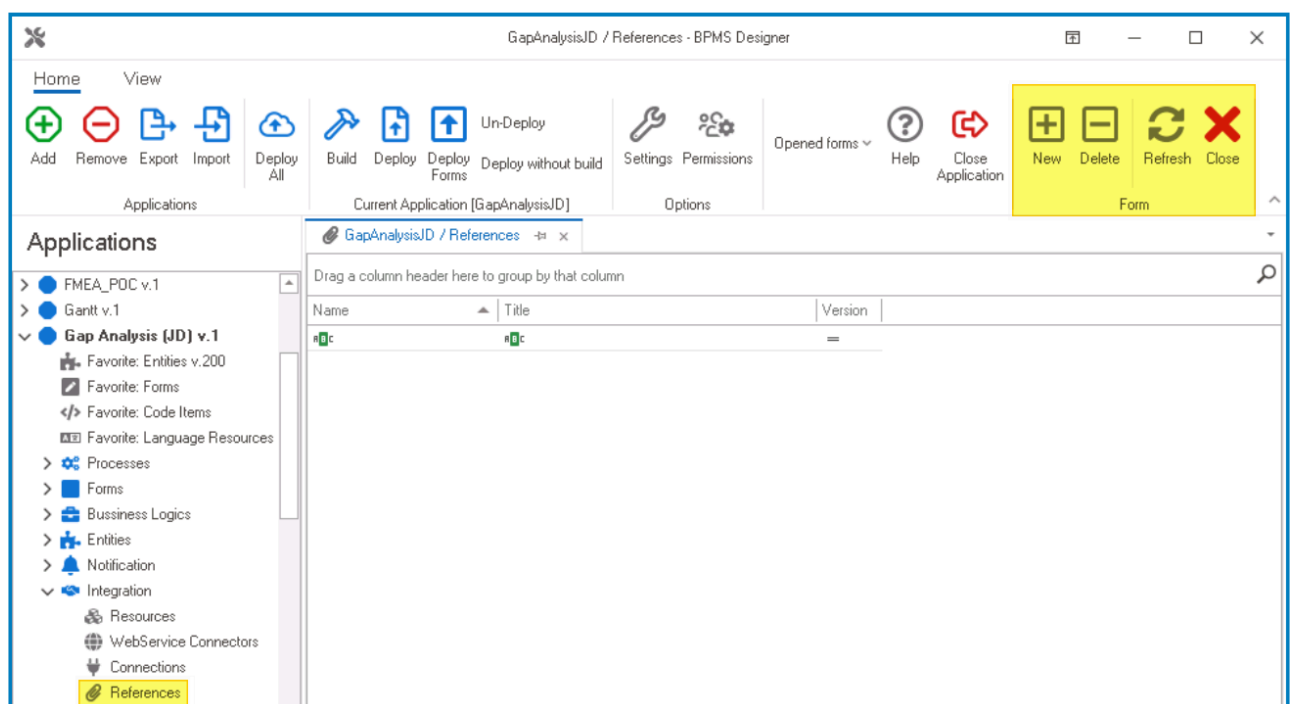
## 16.4. References

In the **References** section of this module, users can create references that connect the current application with other applications in the program.



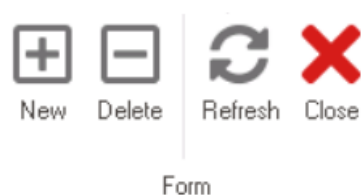
To manage references in the program, take the following steps:

1. From the [Application Tree](#), navigate to **Integration > References** (double-click to open). It will display the **References** tab and expand the menu bar to show additional options.



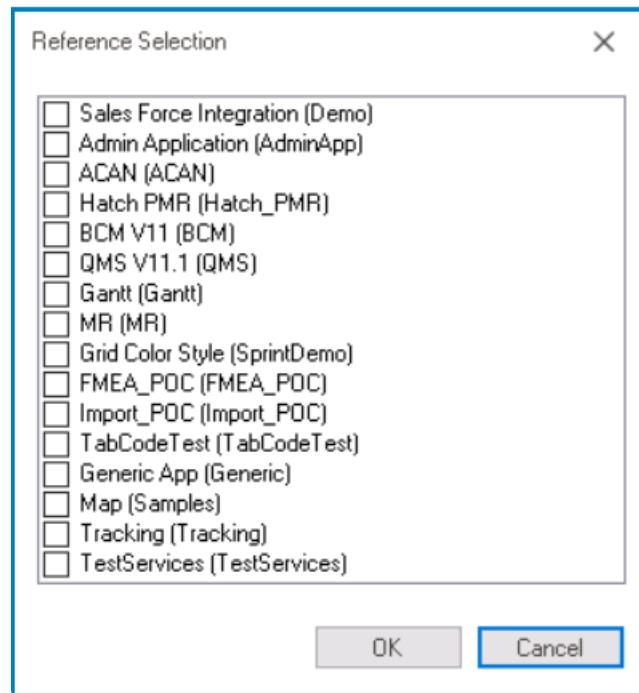
\* In this tab, users can access a list of all defined references for the application. For guidance on how to manipulate the data presented in this tab, [click here](#).

2. From the menu bar, select the desired option.

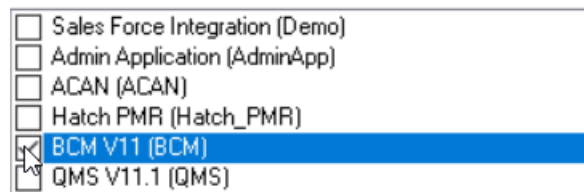


- **New:** This button allows users to define a new reference.

- If selected, it will display the Reference Selection window.



- In this window, users can navigate through the available applications and select the checkbox next to the application they want to reference.



- To confirm the selection, click on the **OK** button. To cancel, click on the **Cancel** button.
- **Delete:** This button allows users to delete an existing reference.
  - To use this function, first select a reference from the list.
- **Refresh :** This button allows users to refresh the References tab.
- **Close:** This button allows users to close the References tab.



# **17. Application Programming Interface (API)**

APIs are integrated simply and efficiently within the Digital Business Platform.

API insertion is possible in both “code” and “design” methods; insert types of commands in both design and code environments.

## 17.1. Define a Parameter (AddParameter)

---

The parameter definition can be done using both the “code” and the “design” method, which is given below with related explanations:

`parameters.Add(input1,input2);`

input1: String parameter name

input2: field name in entity

Output: Define the parameter with the first input name

## 17.2. Calling a Rule (CallRule)

---

Call the rule along with its parameters by using both the “code” and “design” methods below with related explanations:

`bpmAppService.BPMSServices.RuleService.CallRule(input1,input2,input3);`

input1: Rayvarz.BPMApp. [Applicant English name.] Rules.Rules. [English name rule]

input2: DomainObject Or Null

input3: [parameter 1], [parameter 2], ...

Output: Object

## 17.3. Calling Schedule Service (StartJob)

---

Call a scheduled event by using both the “code” and “design” methods below with related descriptions:

```
bpmAppService.BPMSServices.TimeEventService.StartJob(input1,input2);
```

input1: Rayvarz.BPMApp. [Applicant English name]. Events. [Event Name] .TimedEvents. [English Name Occurrence]

input2: DomainObject Or Null

Output: No

## 17.4. 11.4 Stop Scheduling (StopJob)

---

Stop a scheduled event using two code and design methods in the following manner:

bpmAppService.BPMSServices.TimeEventService.StopJob (input1, input2);

input1: Rayvarz.BPMApp. [Applicant English name]. Events. [Event Name] .TimedEvents. [English Name Occurrence]

input2: DomainObject Or Zero

Output: No

## 17.5. 11.5 Sum of two numbers (Arithmetic.Add)

---

Obtaining the sum of two numbers can be done using two methods, “code” and “design”, described below:

bpmAppService.HelperServices.Arithmetics.Add (input1, input2);

input1: first number

input2: second number

Output: Total sum

## 17.6. 11.6 Divide two numbers (Arithmetics.Division)

---

Obtain the result of dividing two numbers using the two methods of “code” and “design” described below:

```
bpmAppSelperServices.Arithmetics.Division(input1,input2);
```

input1: first number

input2: DOM number

Output: Divider divided by number

## 17.7. 11.7 Multiply two numbers (Arithmetics.Multiply)

---

Obtain the product of multiplying two numbers using the two methods “code” and “design” described below:

```
bpmAppService.HelperServices.Arithmetics.Multiply(input1, input2);
```

input1: first number

input2: second number

Output: Multiply du number



## 17.8. 11.8 Subtract two numbers (Arithmetics.Subtract)

---

Subtract two numbers using the two methods, “code” and “design”, described below:

```
bpmAppService.HelperServices.Arithmetics.Subtract(input1, input2);
```

input1: first number

input2: second number

Output: The result of subtracting two numbers

## **17.9. 11.9 Concatenate two Strings (Concat)**

Concatenate two strings using both the “code” and “design” methods described below:

```
bpmAppService.HelperServices.TextService.Concat ("input1", "input2");
```

input1: first text

input2: second text

Output: String

## 17.10. 11.10 Search Text (Contains)

---

Searching for a text or string in another text can be done using two methods, “code” and “design”, as described below:

bpmAppService.HelperServices.TextService.Contains (“input1”, “input2”);

input1: “first text”

input2: “second text”

Output: Boolean

## 17.11. 11.11 Add Date (AddDate)

---

Obtain the difference between the two Adds using two methods of “code” and “design”, as described below with examples:

bpmAppService.HelperServices.DateTimeService.AddDate (input1, input2, input3, input4);

input1: Date (Decimal)

input2: number of years

input3: number of months

input4: number of days

Output: Date (Decimal)

## 17.12. 11.12 Insert current date (CurrentDate) of the DateTime type

---

Insert the current date (with the DateTime type) using the two methods “code” and “design”, as described below with an example:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate;  
Output: Date (DateTime)
```

## 17.13. 11.13 Add time to date (CurrentDate.Add)

---

Add the time to the current date using the two methods “code” and “design”, given below with the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.Add (input1);
```

input1: first time

Output: DateTime

## 17.14. 11.14 Add Day to Date (CurrentDate.AddDays)

---

Add “x” amount of days to the current date using both the “code” and “design” methods, described below with the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.AddDays(input1);
```

input1: numerical value (Double)

Output: DateTime

## 17.15. 11.15 Add hours to date (CurrentDate.AddHours)

---

Add a certain number of hours to the current date using the two methods “code” and “design”, as described below:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.AddHours (input1);
```

input1: numeric value (double)

Output: DateTime



## 17.16. 11.16 Add milliseconds to date (CurrentDate.AddMilliseconds)

---

Add “x” number of milliseconds to the current date using the two methods “code” and “design”, described below with the related description:

bpmAppService.HelperServices.DateTimeService.CurrentDate.AddMilliseconds (input1);

input1: numeric value (double)

Output: DateTime

## 17.17. 11.17 Add Minute to Date (CurrentDate.AddMinutes)

---

Add “x” number of minutes to the current date using the two methods “code” and “design”, as described below:

bpmAppService.HelperServices.DateTimeService.CurrentDate.AddMinutes (input1);

input1: numeric value (double)

Output: DateTime

## 17.18. 11.18 Add months to date (CurrentDate.AddMonths)

---

Add a certain number of months to the current date using both the “code” and “design” methods, as described below with related explanations:

bpmAppService.HelperServices.DateTimeService.CurrentDate.AddMonths (input1);

input1: numeric value (int32)

Output: DateTime

## 17.19. 11.19 Add seconds to date (CurrentDate.AddSeconds)

---

Add “x” number of seconds to the current date using the two methods “code” and “design”, given below the related description:

bpmAppService.HelperServices.DateTimeService.CurrentDate.AddSeconds (input1);

input1: numeric value (double)

Output: DateTime

## 17.20. 11.20 Add tick to date (CurrentDate.AddTicks)

---

Add a tick to the current date using both the “code” and “design” methods, given below the related description:

bpmAppService.HelperServices.DateTimeService.CurrentDate.AddTicks (input1);

input1: numeric value (int64)

Output: DateTime

## 17.21. 11.21 Add year to date (CurrentDate.AddYears)

---

Add “x” number of years to the current date using both the “code” and “design” methods, given below with related explanations:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.AddYears (input1);
```

input1: numeric value (int32)

Output: DateTime

## 17.22. 11.22 Compare current date with date (CurrentDate.CompareTo)

---

A comparison of a date with the current date can be done using both the “code” and “design” methods, given below the related description:

`bpmAppService.HelperServices.DateTimeService.CurrentDate.CompareTo (input1);`

input1: DateTime Date or Object

Output: int32

## 17.23. 11.23 Take a Date From (CurrentDate.Date)

---

Drawing history from the current date can be done using two methods, “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.Date;
```

Output: DateTime



## 17.24. 11.24 Getting the day out of the date (CurrentDate.Day)

---

It is possible to take a few months from the current date using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.Day;
```

Output: DateTime

## 17.25. 11.25 Take a day of week (CurrentDate.DayOfWeek)

---

It is possible to take a few days from the current date using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.DayOfWeek;
```

Output: String

## 17.26. 11.26 Take a few days from the date (CurrentDate.DayOfYear)

---

It is possible to take a few days from the current date using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.DayOfYear;
```

Output: int

## 17.27. 11.27 Getting the time out of the current date (CurrentDate.Hour)

---

Retrieve the clock from the current date using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentDate.Hour;
```

Output: int

## 17.28. 11.28 Insert current date (CurrentMiladiDate) in decimal format

---

Insert current date (decimal type) using two methods: “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentMiladiDate;
```

Output: Date (Decimal)

## 17.29. 11.29 Insert Current Time (CurrentTime)

---

Insert the current time using two methods of “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.CurrentTime;
```

Output: Date (TimeSpan)

## 17.30. 11.30 Difference between two dates (DateDiffGregorian)

---

Obtain the difference between the two dates using two methods of “code” and “design”, given below with related explanations:

bpmAppService.HelperServices.DateTimeService.DateDiffGregorian (input1, input2);

input1: start date

input2: end date

Output: Date (TimeSpan)

## 17.31. 11.31 Insert Day of the Week (DayOfWeek)

---

Insert the day of the week into the code using both the “code” and “design” method, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.DayOfWeek;
```

Output: String



## 17.32. 11.32 Convert date decimal to datetime (GetDate)

---

Transform the date from the decimal to the Date time using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.GetDate (input1);
```

input1: [Date in Decimal format]

Output: Date time

## 17.33. 11.33 Day of the week (GetDayOfWeekByDate)

---

Get the day of the week from the date (with the DateTime type) using the two methods “code” and “design”, given below the related description:

```
bpmAppService.HelperServices.DateTimeService.GetDayOfWeekByDate(input1);
```

input1: Date with DateTime format

Output: String

## 17.34. 11.34 The day of the week (GetDayOfWeekByNumeric)

---

Obtain the day of the week from the date (Numeric type) using the two methods “code” and “design”, given below the related description:

bpmAppService.HelperServices.DateTimeService.GetDayOfWeekByNumeric (input1, input2);

input1: [date with DateTime format]

input2: True or False (meaning True or True)

Output: String

## **17.35. 11.35 Sum of 2 times (TimeAdd)**

---

The sum of two times can be done using the two methods “code” and “design”, given below the related description:

bpmAppService.HelperServices.DateTimeService.TimeAdd (input1, input2);

input1: first time

input2: second time

Output: TimeSpan

## **17.36. 11.36 Subtraction of 2 time (TimeDiff)**

The two-time subtraction can be done using both the “code” and the “design” method, which is given below with related explanations:

bpmAppService.HelperServices.DateTimeService.TimeDiff (input1, input2);

input1:, first time

input2: second time

Output: TimeSpan

## **17.37. 11.37 Receive Groups (GetAllGroups)**

Obtain a list of the identifier of the groups to which the user or the party is affiliated with using both the “code” and the “design” method, given below the related description:

bpmAppService.BPMSServices.MercuryService.GetAllGroups (input1);

input1: ID Staff or ID User (for example: bpmAppService.BPMSServices.MercuryService.CurrentStaff.ID)

Output: List

## 17.38. 11.38 Password Comparison (CheckPassword)

---

Compare the password entered with the password in the Merc list using two methods, “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.MercuryService.CheckPassword (input1, input2);

input1: object from the user

input2: Password

Output: Boolean

## 17.39. 11.39 Assignment to User (AssignStaffToUser)

---

One-way assignment to the user can be done using both the “code” and the “design” method, which is given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.AssignStaffToUser(input1,input2,input3);
```

input1: ID Staff

input2: ID User

input3: True Or False (True= Catalog Bpms)

Output: Void



## 17.40. 11.40 Delete User Assignment (DeleteUserAssignment)

---

Remove a one-to-one assignment to the user using both the “code” and the “design” method, which is given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.DeleteUserAssignment(input1,input2,input3);
```

input1: ID Staff

input2: ID User

input3: True Or False (True= Catalog Bpms)

Output: Void

## 17.41. 11.41 Update Unit (UpdateDepartment)

---

An organizational unit update can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.MercuryService.UpdateDepartment(input1,input2);

input1: ID Department

input2: True Or False (True= Catalog Bpms)

Output: Void

## **17.42. 11.42 Update Group (UpdateGroup)**

The group update can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.UpdateGroup(input1,input2);
```

input1: ID Group

input2: True Or False (True= Catalog Bpms)

Output: Void

## 17.43. 11.43 Update Staff (UpdateStaff)

---

The side update can be done using both the “code” and “design” methods, which are given below the related description.

```
bpmAppService.BPMSServices.MercuryService.UpdateStaff(input1,input2);
```

input1: ID Staff

input2: True Or False (True= Catalog Bpms)

Output: Void

## 17.44. 11.44 Update User (UpdateUser)

---

The side update can be done using both the “code” and “design” methods, which are given below the related description.

```
bpmAppService.BPMSServices.MercuryService.UpdateUser(input1,input2);
```

input1: ID User

input2: True Or False (True= Catalog Bpms)

Output: Void

## 17.45. 11.45 Insert member in the Group (AddMemberToGroup)

---

Inserting a member or user in a group can be done using both the “code” and “design” method, which is given below the related description.

```
bpmAppService.BPMSServices.MercuryService.AddMemberToGroup(input1,input2,input3);
```

input1: ID Group

input2: ID User or ID Staff

input3: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

Output: Void

## 17.46. 11.46 Remove member from the Group (RemoveMemberofGroup)

---

Removing a member or user from a group can be done using both the “code” and the “design” method, which is given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.RemoveMemberOfGroup(input1,input2,input3);
```

input1: ID Group

input2: ID User or ID Staff

input3: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

Output: Void

## 17.47. 11.47 Remove User from Group (RemoveMemberOfGroup)

---

Deleting a particular user from a specific group can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.RemoveMemberOfGroup(input1,input2,input3);
```

input1: ID Group

input2: ID User

input3: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

Output: Void



## 17.48. 11.48 Create or Update User (SaveUser)

---

Creating or updating a user can be done using both the “code” and “design” methods, which are given below the related description.

```
bpmAppService.BPMSServices.MercuryService.SaveUser(input1,input2,input3,input4,input5);
```

input1: Object Of User

input2: ID Group

input3: ID Permission

input4: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

input5: True or False ( True= Save , False= Update)

Output: Void

## 17.49. 11.49 Create or Update an Organizational Unit (saveDepartment)

---

Creating or updating an organizational unit can be done using both “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.saveDepartment(input1,input2,input3);
```

input1: Department

input2: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

input3: True or False ( True= Save , False= Update)

Output: Void

## 17.50. 11.50 Create or Update Group (saveGroup)

---

Creating or updating a group can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.saveGroup(input1,input2,input3);
```

input1: Group

input2: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

input3: True or False ( True= Save , False= Update)

Output: Void

## 17.51. 11.51 Create or Update a Staff (saveStaff)

---

Creating or updating a side can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.MercuryService.saveStaff(input1,input2,input3); input1: Group input2: True or False ( True= Catalog Bpms , False= Catalo Eoffice) input3: True or False ( True= Save , False= Update)  
Output: Void

## 17.52. 11.52 Update User (UpdateUserWithoutPassword)

---

A user's update without regard to the password can be done using both the "code" and the "design" method, which is given below the related description.

bpmAppService.BPMSServices.MercuryService.UpdateUserWithoutPassword(input1,input2);

input2: ID User

input2: True or False ( True= Catalog Bpms , False= Catalo Eoffice)

Output: Void

## 17.53. 11.53 Get current Manager's ID (ManagersIdOfCurrentStaff)

---

Obtaining the current side admins identifier can be done using the two methods “code” and “design”, which is given below the related description:

```
bpmAppService.BPMSServices.MercuryService.ManagersIdOfCurrentStaff();
```

Output: String

First, select “Organization Chart Service” in the “Process” section, and then select the “Get Current Identity Manager (ManagersIdOfCurrentStaff)” method.

## **17.54. 11.54 Get a list of current Managers (GetManagersOfCurrentStaff)**

---

Obtaining a list of current state admins identifiers can be done using both the “code” and “design” methods, which are given below the related description:

```
bpmAppService.BPMSServices.MercuryService.GetManagersOfCurrentStaff();
```

Output: List

## 17.55. 11.55 Get personally identifiable information (GetPrincipal)

---

Obtaining an object or identifier properties in the Principal table can be done using both the “code” and “design” methods, which are given below the related description:

bpmAppService.BPMSServices.MercuryService.GetPrincipal (input1);

input1: ID Staff or ID User or ID Department (for example:

bpmAppService.BPMSServices.MercuryService.CurrentStaff.ID)

Output: Rayvarz.Profiles.Principal



## 17.56. 11.56 Get admin ID (GetManager)

---

Obtaining the selected identifier manager can be done using both the “code” and “design” methods, which are given below the related description:

`bpmAppService.BPMSServices.MercuryService.GetManager (input1);`

input1: ID Staff (for example: `bpmAppService.BPMSServices.MercuryService.CurrentStaff.ID`)

Output: `Rayvarz.Profiles.Principal`

## 17.57. 11.57 Get started User Information (MercuryService.StarterUser)

---

Calling and receiving user-initiated process information can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.StarterUser();
```

Output: Rayvarz.Profiles.User

## **17.58. 11.58 Get current User information**

---

Calling and receiving current user information can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.CurrentUser();
```

Output: Rayvarz.Profiles.User

## **17.59. 11.59 Get the header side manager information (MercuryService.ManagerOfStarterStaff)**

---

Calling and receiving information from the processor initiator side can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.ManagerOfStarterStaff();
```

Output: Rayvarz.Profiles.Principal

## 17.60. 11.60 Get Started Information (MercuryService.StarterStaff)

---

Calling and receiving information about the process initiator can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.StarterStaff();
```

Output: Rayvarz.Profiles.Staff

## 17.61. 11.61 Getting Starter Side Manager Information (MercuryService.ManagerOfCurrentStaff)

---

Calling and receiving information from the processor initiator side can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.ManagerOfCurrentStaff();
```

Output: Rayvarz.Profiles.Staff

## 17.62. 11.62 Get Current Staff Information (MercuryService.CurrentStaff)

---

Calling and receiving current side information can be done using both the “code” and “design” methods, which are given below with related explanations:

```
bpmAppService.BPMSServices.MercuryService.CurrentStaff();
```

Output: Rayvarz.Profiles.Staff

## 17.63. 11.63 Get Managers for User Positions (MercuryService.GetUserManagerByUser)

Calling all managers of different positions of a user can be done using two methods of “code” and “design”, which is given below the related description:

```
bpmAppService.BPMSServices.Mervice.GetUserManagerByUser(input1;
```

input1: ID User

Output: List



## **17.64. 11.64 Get Users admin side (MercuryService.GetUserManagerByStaff)**

---

Calling all users who are on the other side of the manager can be done using both the “code” and the “design” method, which is given below the related description.

```
bpmAppService.BPMSServices.MercuryService.GetUserManagerByStaff(input1);
```

input1: ID Staff

Output: List

## **17.65. 11.65 Get Group Members (MercuryService.GetUsersByGroupCode)**

---

Calling all users of a group can be done using both the “code” and “design” methods, which are given below the related description.

```
bpmAppService.BPMSServices.MercuryService.GetUsersByGroupCode(input1);
```

input1: ID Group

Output: List(User)

## **17.66. 11.66 Receive the Organizational Unit (MercuryService.GetDepartmentByStaff)**

Getting an organizational unit on one side can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.GetDepartmentByStaff(input1);
```

input1: ID Staff

Output: Department

## 17.67. 11.67 Obtaining Organizational Units for a single User (MercuryService.GetDepartmentByUser)

---

Getting the list of organizational units for a user's positions is done using two methods, "code" and "design", which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.GetDepartmentByUser(input1);
```

input1: ID User

Output: List(Department)

## 17.68. 11.68 Get the User's positions (MercuryService.GetUserStaffs)

---

Getting the positions of a user can be done using both the “code” and “design” methods, which are given below the related description.

```
bpmAppService.BPMSServices.MercuryService.GetUserStaffs(input1);
```

input1: ID User

Output: IDictionary

## 17.69. 11.69 Get the position of the top Manager of one side (MercurySide.GetManagerOfBeforeCEODepartment)

Receiving the position of the top-level manager (one category before the CEO), one side, can be done using both the “code” and the “design” method, which is given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.GetManagerOfBeforeCEODepartment(input1);
```

input1: Staff

Output: Staff

## 17.70. 11.70 Get the post of a higher-level unit manager (MercuryService.GetManagerOfUpperDepartmentmen

Getting the position of the higher-level manager of a side on the organizational chart can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.GetManagerOfUpperDepartment(input1);
```

input1: Staff

Output: Staff

## **17.71. 11.71 Get current User units (MercuryService.CurrentUserDepartment)**

---

Acquiring current user organizational units can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.CurrentUserDepartment();
```

Output: List Department



## **17.72. 11.72 Get current manager User list (MercuryService.CurrentUserManager)**

---

A list of current user manager positions can be made using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.CurrentUserManager();
```

Output: List Principal

## 17.73. 11.73 Get current User positions (MercuryService.CurrentUserStaffs)

---

Receiving current user positions can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.CurrentUserStaffs();
```

Output: IDictionary

## 17.74. 11.74 Get Current User (MercuryService.CurrentStaff)

---

Acquiring the active user current in the system can be done using both the “code” and the “design” method, which is given below the related description.

```
bpmAppService.BPMSServices.MercuryService.CurrentStaff();
```

Output: IDictionary

## **17.75. 11.75 Receive admin-side admin ID (MercuryService.GetStaffMangerByStaff)**

Getting a side-by-side manager's id can be done using both the "code" and "design" methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.MercuryService.GetStaffManagerByStaff(input1);
```

input1: Staff

Output: ID Staff

## 17.76. 11.76 Fetch information from the database (ExecuteQuery)

---

Fetch information from the database using two methods, “code” and “design”, which is given below the related description:

bpmAppService.BPMSServices.DataSourceService.ExecuteQuery (input1, “input2”);

input1: Rayvarz.BPMApp. [Applicant English name]. Connections.Connections. [Name of database connection]

input2: “Requested Query”

Output: List

## 17.77. 11.77 Fetch information from the current database (ExecuteHqlQuery)

---

Fetching information from the current database can be done using both the “code” and “design” methods, which are given below the related description:

bpmAppService.BPMSServices.HQLService.ExecuteHqlQuery (input1);

input1: String “Required Query” (string q = string.Format (“FROM BPMAPP\_HGRequest WHERE Shop IS NULL”))

Output: List (Query result)

## 17.78. 11.78 Get Objects (GetAll)

---

Obtaining all objects of one type, or in other words all instances of one type (for example, all instances of a request, which is an object in the database), can be done using both the “code” and “design” methods, which are in Below is the description given to it

bpmAppService.BPMSServices.HQLService.GetAll (input1);

input1: Typeof (type desired) – (typeof (Rayvarz.BPMApp. [Application name] .Module. [Table name]));

Output: List

## 17.79. 11.79 Get the complete object domain (GetDomainObject)

---

Obtaining a domain object completely can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.HQLService.GetDomainObject (input1, input2);

input1: Typeof (type desired) – (typeof (Rayvarz.BPMApp. [Application name] .Module. [Table name]));

input2: new Guid ()

Output: Full DomainObject



## 17.80. 11.80 Fetch Database Information (ExecuteSqlQuery)

---

Fetching information from the database along with the variable during the work can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.SqlService.ExecuteSqlQuery (input1);

input1: String Query (string qry = string.Format (“select [field name] FROM [table name] where id == ‘0’,” domainObject.id))

Output: List (Query result)

## 17.81. 11.81 Get all database information (ExecuteSqlQueryFromDB)

---

Database data can be accessed using two methods, “code” and “design”, which is given below with related explanations.

```
bpmAppService.BPMSServices.SqlService.ExecuteSqlQueryFromDB(input1);
```

Output: List

## 17.82. 11.82 General code tracking system (GenerateTracking)

---

The production of the system code tracking can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.TrackingService.GenerateTracking (input1);

input1: DomainObject (entity)

Output: TrackingCode

## 17.83. 11.83 Get Tracking Code (GetTrackingNumber)

---

Getting the tracking code can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.TrackingService.GetTrackingNumber();
```

Output: int64

## 17.84. 11.84 Send Notification to User (SendByNotification)

---

Sending a notification of the type of SMS and email to the selected user can be done using both the “code” and “design” methods, which are given below the related explanations.

bpmAppService.BPMSServices.NotificationService. SendByNotification (input1, input2, input3);

input1: Defined notification(see 11.3)

input2: DomainObject

input3: User

Output: No

## 17.85. 11.85 Send notification to (SendByNotification)

---

Sending notifications of text and e-mail to a selected destination can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.NotificationService. SendByNotification (input1, input2, input3);

input1: Defined notification(see 11.3)

input2: DomainObject

input3: User

Output: No

## 17.86. 11.86 Send notification to recipient list (SendByNotification)

---

Sending a notification of SMS and e-mail to a list of recipients can be done using both the “code” and “design” methods, which are given below the related description.

bpmAppService.BPMSServices.NotificationService. SendByNotification (input1, input2, input3);

input1: Defined notification(see 11.3)

input2: DomainObject

input3: List of recipients

Output: No

## 17.87. 11.87 Send notification to recipient email list (SendByNotification)

---

Sending SMS and e-mail notifications to recipients' emails can be done using both the "code" and "design" methods, which are given below with related explanations.

bpmAppService.BPMSServices.NotificationService. SendByNotification (input1, input2, input3);

input1: Defined notification(see 11.3)

input2: DomainObject

input3: Email list

Output: No



## 17.88. 11.88 Send notification to an email (SendByNotification)

---

Sending an SMS and e-mail notification to an e-mail from the receiver can be done using both the “code” and the “design” method, which is given below the related description.

bpmAppService.BPMSServices.NotificationService. SendByNotification (input1, input2, input3);

input1: Defined declaration

input2: DomainObject

input3: Receiver

Output: No

## 17.89. 11.89 Create notifications and send to recipients (Send)

---

Making a notification and sending it to the receiver can be done using two methods of “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.NotificationService. Send (input1, input2, input3, input4, input5, input6);

input1: string (Notification Title)

input2: string (Notification Text)

input3: NotificationAgent (Notification Type)

input4: Boolean (True or False)

input5: string (receiver)

input6: SortedList (Attached and Optional)

Output: No

## 17.90. 11.90 Create notifications and send to receiver types (Send)

---

Creating a notification and sending it to the receiver types can be done using the two methods “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.NotificationService. Send (input1, input2, input3, input4, input5, input6);

input1: string (notification title)

input2: string (notification text)

input3: NotificationAgent (notification type)

input4: Boolean (True or False)

input5: string (receiver)

input6: SortedList (Attached and Optional)

Output: No

## 17.91. 11.91 Create notifications and send to recipients list (Send)

---

Creating a notification and sending it to the recipient's list can be done using both the “code” and the “design” method, which is given below with related explanations.

bpmAppService.BPMSServices.NotificationService. Send (input1, input2, input3, input4, input5, input6);

input1: string (notification title)

input2: string (notification text)

input3: NotificationAgent (notification type)

input4: Boolean (True or False)

input5: list (recipients)

input6: SortedList (Attached and Optional)

Output: does not have

## 17.92. 11.92 Create a notification and send to recipient email list (Send)

---

Making a notification and sending it to the recipient's e-mail list can be done using both the “code” and the “design” method, which is given below the related description.

bpmAppService.BPMSServices.NotificationService. Send (input1, input2, input3, input4, input5, input6);

input1: string (notification title)

input2: string (notification text)

input3: NotificationAgent (notification type)

input4: Boolean (True or False)

input5: list (recipient email)

input6: SortedList (Attached and Optional)

Output: No

## 17.93. 11.93 Create notification and send to recipient's email (Send)

---

Making a notification and sending it to the recipient's e-mail can be done using both the "code" and the "design" method, which is given below the related description.

bpmAppService.BPMSServices.NotificationService.Send (input1, input2, input3, input4, input5, input6);

input1: string (notification title)

input2: string (notification text)

input3: NotificationAgent (notification type)

input4: Boolean (True or False)

input5: string (receiver email)

input6: SortedList (Attached and Optional)

Output: No

## 17.94. 11.94 Send Transaction Type Notice (SendCopy)

---

Sending a copy of the transcript type can be done using both the “code” and “design” methods, which are given below the related description.

bpmAppService.BPMSServices.NotificationService. SendCopy (input1, input2, input3);

input1: Defined declaration (See 11.3)

input2: DomainObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input3: event

Output: No

## 17.95. 11.95 Event call (RaiseEvent)

---

Calling an event from a process can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.ProcessService.RaiseEvent (input1, input2);

input1: event

input2: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

Output: GuardResult



## 17.96. 11.96 Call event with specified user (RaiseEvent)

---

Calling an event of a process with a specific user can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.ProcessService.RaiseEvent (input1, input2, input3);

input1: event

input2: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input3: user

Output: GuardResult

## **17.97. 11.97 Start the process (StartProcess)**

Starting a process can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.ProcessService.StartProcess (input1, input2, input3);

input1: DomainObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input2: True Or False

input3: Rayvarz.BPMApp. [Application name]. Processes.Processes. [Issue No.]. [Process Name]

Output: DomainObjectBase

## 17.98. 11.98 Start process with the user and the Starter (StartProcess)

---

The beginning of a process with a user and a specific starting point can be accomplished using two methods, “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.ProcessService.StartProcess (input1, input2, input3, input4, input5);

input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input2: True Or False

input3: Starter User

input4: Starter Staff

input5: Rayvarz.BPMApp. [Application name]. Processes.Processes. [Issue No.]. [Process Name]

Output: DomainObjectBase

## 17.99. 11.99 Start the process automatically (StartProcessOnPersistentObject)

---

Starting a process that does not start automatically can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.ProcessService.StartProcessOnPersistentObject (input1, input2, input3, input4);

input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input2: Starter User

input3: Rayvarz.BPMApp. [Application name]. Processes.Processes. [Issue No.]. [Process Name]

input4: True Or False

Output: DomainObjectBase

## 17.100. 11.100 Get an object from an entity (GetDomainObject)

---

Obtaining an object from a particular entity can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.GetDomainObject (input1, input2);

input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

input2: ID DomainObject

Output: DomainObject

## 17.101. 11.101 Create an object from an entity (CreateDomainObject)

---

Creating an object from a specific entity can be done using two methods, “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject (input1);  
input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])  
Output: DomainObject

## 17.102. 11.102 Create an object from an entity with an identifier (CreateDomainObject)

---

Creating an object from a specific entity with its identifier can be done using two methods, “code” and “design”, which is given below the related description.

```
bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject (input1, input2);
```

```
input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])
```

```
input2: DominObject.id
```

```
Output: DomainObject
```

## 17.103. 11.103 Delete an object from an entity (DeleteDomainObject)

---

Deleting an object from a specific entity can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.DeleteDomainObject (input1);  
input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])  
Output: Void



## 17.104. 11.104 Save an object from an entity (SaveDomainObject)

---

The storage of an object from a specific entity can be done using two methods, “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.DomainObjectService.SaveDomainObject (input1);  
input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])  
Output: Void

## 17.105. 11.105 Save an object from an object (SaveObject)

---

It is possible to store an object from a particular object using two methods, “code” and “design”, which is given below the related description.

```
bpmAppService.BPMSServices.DomainObjectService.SaveObject(input1);
```

input1: Object

Output: void

## 17.106. 11.106 Get objects of an entity (GetAll)

---

Obtaining all objects of a particular entity can be done using both the “code” and the “design” method, which is given below the related description.

bpmAppService.BPMSServices.DomainObjectService.GetAll (input1);

input1: DominObject (Rayvarz.BPMApp. [Application name] .Module [entity name])

Output: List

## 17.107. 11.107 Get Parents Object (GetParentObject)

---

Getting the current parent's object can be done using both the "code" and "design" methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.DomainObjectService.GetParentObject();
```

Output: Parent current DomainObject

## **17.108. 11.108 Changing information in the database (CommitTransaction)**

---

Applying changes to information in a database can be done using both the “code” and “design” methods, which are given below with related explanations.

```
unitOfWork.CommitTransaction();
```

Output: Void

## **17.109. 11.109 Get the file from the attachment control (GetFileFromControl)**

---

Getting the file from the attachment control and using it in other parts can be done using both the “code” and the “design” method, which is given below the related description.

bpmAppService.BPMSServices.DmsService.GetFileFromControl (input1);

input1: attribute associated with attach control (domainObject. [adjective name])

Output: Dms

## 17.110. 11.110 Get the file from the attachment control (GetFile)

---

Getting the file from the attachment control and using it in other parts can be done using both the “code” and the “design” method, which is given below the related description.

bpmAppService.BPMSServices.DmsService.GetFile (input1);

input1: attribute associated with attach control (domainObject. [adjective name])

Output: String

## 17.111. 11.111 Search the folder named (GetFolderDefaultForAttachment)

---

Searching for a folder with a name can be done using both the “code” and “design” method, which is given below the related description.

```
bpmAppService.BPMSServices.DmsService.GetFolderDefaultForAttachment(input1);
```

input1: Folder Name

Output: String



## 17.112. 11.112 Search the folder with the name and id (GetFolderByNameAndParentId)

---

A folder with a parent folder name and id can be searched using two methods, “code” and “design”, which is given below the related description,

```
bpmAppService.BPMSServices.DmsService.GetFolderByNameAndParentId(input1,input2);
```

input1: ID Parent Folder

input1: Folder Name

Output: String

## 17.113. 11.113 Create folder with name and id (CreateFolderSerialize)

---

Creating a folder with a parent folder name and identifier can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.DmsService.CreateFolderSerialize(input1,input2);
```

input1: ID Parent Folder

input1: Folder Name

Output: String

## 17.114. 11.114 Create folder with name and id (CreateFolderSerialize)

---

Creating a folder with a parent folder name and identifier can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.DmsService.CreateFolderSerialize(input1,input2);
```

input1: ID of Parent Folder

input1: Folder Name

Output: Dms

## 17.115. 11.115 Get the children of a node (GetAllTreeNodeFromChildObject)

---

Getting all the children of a node can be done using the two “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.GetAllTreeNodeFromChildObject (input1, input2, input3, input4, input5);

input1: Child object

input2: Foreign Key to the root with the type String

input3: Parent Foreign Key

input4: Foreign Key to itself with type List

input5: Foreign Key to itself

Output: List Node

## 17.116. 11.116 Get the children of root (GetAllTreeNodeFromRootObject)

---

Getting all the children of the root of the tree can be done using two methods, “code” and “design”, which is given below the related description.

bpmAppService.BPMSServices.DomainObjectService.GetAllTreeNodeFromRootObject (input1, input2, input3);

input1: Child object

input2: Parent Foreign Key

input3: Foreign Key to itself

Output: List Node

## 17.117. 11.117 Get Tree Roots (GetParentsNode)

---

Getting all the child's parents and inserting it in the third parameter input can be done using both the “code” and “design” method, which is given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.GetParentsNode (input1, input2, input3);

input1: Object entity that is attached to the tree

input2: Foreign Key to itself

input3: The parameter to which the output is placed

Output: List Node

## 17.118. 11.118 Get root tree with child object (GetRootObject)

---

Getting all the child's parents and inserting it in the third parameter input can be done using both the "code" and "design" method, which is given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.GetRootObject (input1, input2, input3);

input1: The object of the child's Entity

input2: Foreign Key to itself

input3: Parent Foreign Key

Output: List Node

## 17.119. 11.119 Read from excel (ReadOfExcel)

---

Getting an Excel file for reading is done using the two “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.DomainObjectService.ReadOfExcel (input1, input2, input3);

input1: file name

input2: name of the sheet

input3: counter

Output: Table



## 17.120. 11.120 Open File or Print (GetWordFormat)

---

Getting a file to open it or print it directly can be done using both the “code” and “design” method, which is given below the related description.

```
bpmAppService.BPMSServices.DmsService.GetWordFormat (input1, input2, input3, input4);
```

input1: file name

input2: application name

input3: True or false

input3: True or false

Output: String

## 17.121. 11.121 Call Web Service Connection (InvokeBPMSService)

---

The call to the Web service connection can be done using both the “code” and the “design” method, which is given below with related explanations.

bpmAppService.BPMSServices.WebConnectorService. InvokeBPMSService (input1, input2, input3);

input1: Code WebConnector

input2: Dictionary (field list)

input3: Id Object

Output: Void

## **17.122. 11.122 Calling the Web Service Connection with XMM (InvokeBPMSService)**

---

The call to the Web service connection can be done using both the “code” and the “design” method, which is given below with related explanations.

bpmAppService.BPMSServices.WebConnectorService. InvokeBPMSService (input1, input2, input3);  
input1: Code WebConnector  
input2: Dictionary (list of fields)  
input3: Id Object  
Output: Void

## **17.123. 11.123 Calling the Web Service with the User and the User (InvokeBPMSService)**

A call to a Web service or user-to-side input can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.WebConnectorService. InvokeBPMSService (input1, input2, input3);

input1: Code WebConnector

input2: Dictionary (field list)

input3: Id Object

input3: Id Starter User Or (user who starts the event)

input3: Id Starter Staff Or (the beginning of the event)

Output: Void

## **17.124. 11.124 Calling the Web service with the user and the XML (InvokeBPMSService)**

A call to a Web service or user-to-side input can be done using both the “code” and “design” methods, which are given below with related explanations.

bpmAppService.BPMSServices.WebConnectorService. InvokeBPMSService (input1, input2, input3);

input1: Code WebConnector

input2: Dictionary (field list)

input3: Id Object

input4: Id Starter User Or (user who starts the event)

input5: Id Starter Staff Or (the beginning of the event)

Output: Void

## 17.125. 11.125 X-MEL split into distinct attributes (XmlToProperties)

---

The X-MEL split into individual attributes, along with its values, can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.WebConnectorService.XmlToProperties(input1,);
```

input1: Xml

Output: Void

## 17.126. 11.126 Convert Dictionary to XMM Format (PropertiesToXml)

---

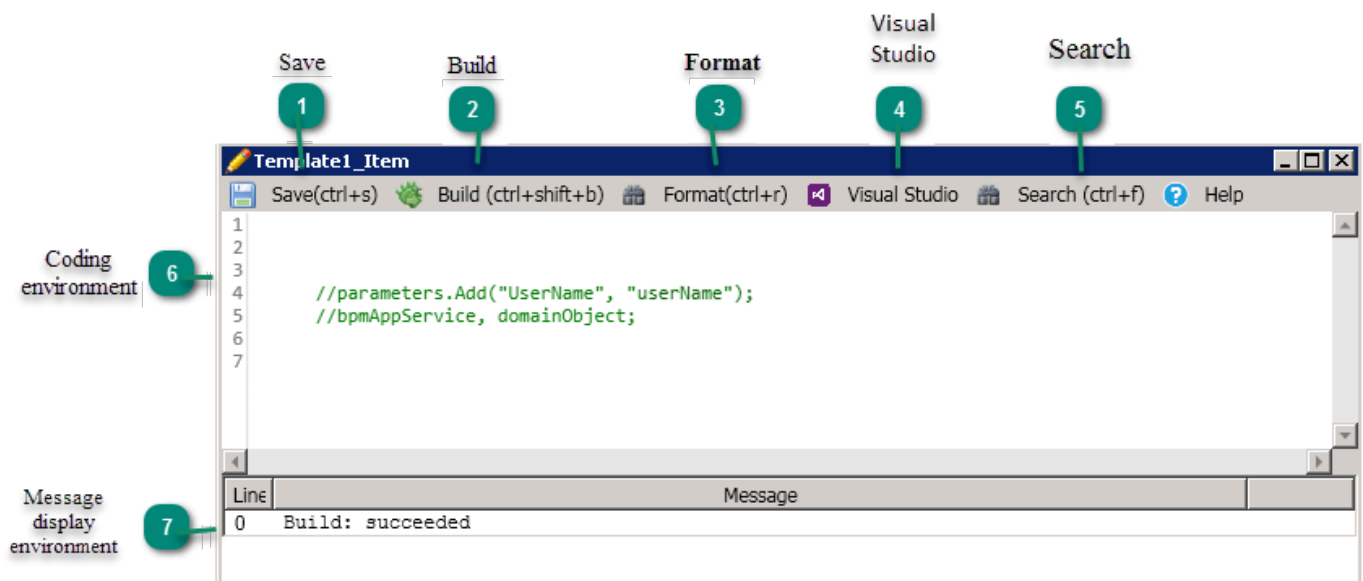
The conversion of the dictionary's list into XMML can be done using both the “code” and “design” methods, which are given below with related explanations.

```
bpmAppService.BPMSServices.WebConnectorService. PropertiesToXml(input1,);
```

input1: Dictionary

Output: Void

# 17.127. 11.127 Code Environment

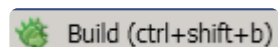


## 1. Save:



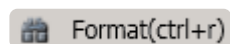
By typing this button, the typed script is saved.

## 2. Build:



By choosing this button, the script is typed in terms of the grammatical structure and the result of the review is displayed in the “message” section.

## 3. Format:



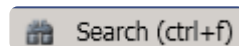
By selecting this button, the textual structure of the inserted code is set.

## 4. Visual Studio:



By choosing this button, Visual Studio software opens to enter the desired code.

## 5. Search



By choosing this button, the search form is displayed to search or replace the word.

## 6. Coding Environment:





This box is where the code is to be inserted.4.

**Note:** Depending on which version of the script opens, it can be used in this API framework.

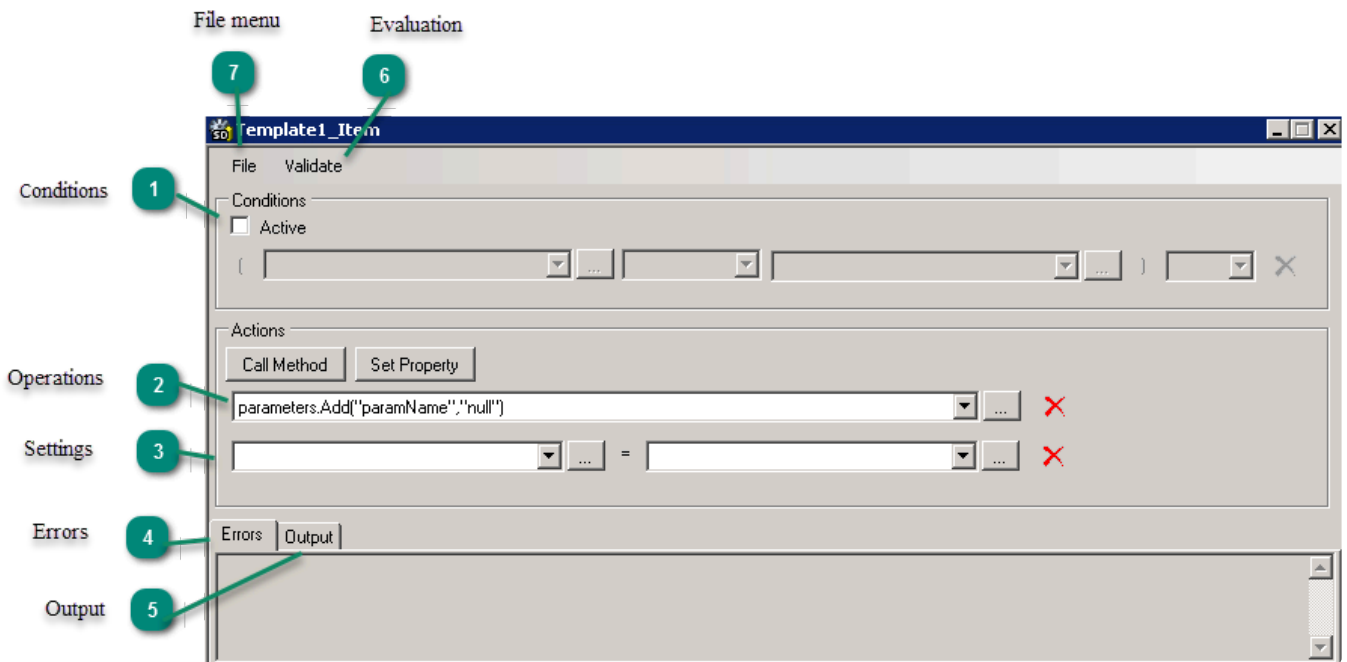
#### 7. Message display Environment:

| Line |                  |
|------|------------------|
| 0    | Build: succeeded |

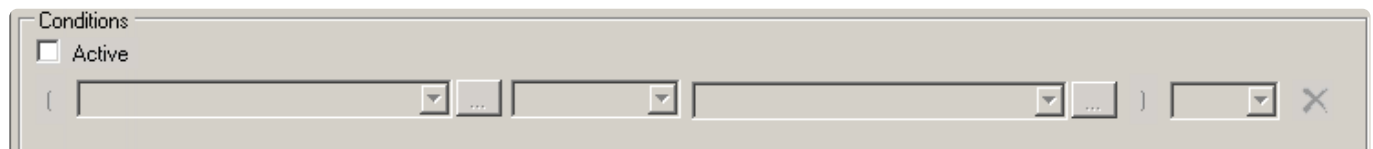
After checking the code using the build button, errors or messages from this review are displayed in this box.

# 17.128. 11.128 Design Environment



If you need to define a script without writing code associated with it, the design environment is used.



## 1. Conditions:



If the desired code has conditions, then you can select the conditions after selecting “Enabled” in the enabled boxes.


To insert a bet, first select the active button  and select the desired field using existing services, then select or type the value in the next box by selecting the button .

Note: To add a second bet, in the last box, select one of the words “and” or “or”.

To remove the bet, select the button  next to each bet.

## 2. Operations:






Each time you select the “Operation” button in the behavior box, an operation box appears. To insert an operation, first select the active button  and then select the desired operation using the available services.

To remove behavior, select the button  next to each behavior.

## 3. Settings:



Each time you select the Set Property” button in the Actions box, a setting box appears. To enter the

initialization, first select the active button  and select the desired field using existing services, then select or type the value in the next box by selecting the button . To remove behavior, select the button  next to each behavior.

#### 4. Errors:

In the case of design evaluation, the design errors are displayed in this heading.

#### 5. Output:

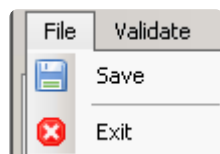
The results of the design review shown in this heading are displayed.

#### 6. Evaluation:



By selecting this button, the logic of selected behaviors is examined and the result of the check is displayed in the output tab or errors.

#### 7. File Menu:



By choosing the file button for storage options, to save the design and exit, you will be able to access the design form to exit.

# 18. Business Logic API

---

## Easy access to the code (Snippet Code)

In the code environment, type the word reservation and press Ctrl+Tab+Tab or Ctrl+Tab, the code will be automatically inserted in the coding environment.

## 18.1. Reservation Word

| code  | Reservation Word                             |
|---|--|
| code for each service For example, the<br>bpmAppService.BPMSServices.RuleService                  | Word in parentheses<br>For example, the rule |
| if(System.Diagnostics.Debugger.IsAttached)<br>System.Diagnostics.Debugger.Break();                | Debug  |
| foreach (var VARIABLE in COLLECTION ){}   | foreach                                      |
| bpmAppService.BPMSServices.DomainObjectService.CreateDomainObject(typeof(<br>DOMAINOBJECTTYPE )); | Create                                       |
| bpmAppService.BPMSServices.ProcessService.RaiseEvent( EVENT ,<br>domainObject);                   | Raise  |
| bpmAppService.BPMSServices.ProcessService.StartProcess  | Start  |
| bpmAppService.BPMSServices.SqlService.ExecuteSqlQuery<<br>DOMAINOBJECTTYPE >( QUERY );            | runSql                                       |
| bpmAppService.BPMSServices.HQLService.ExecuteHqlQuery( QUERY );                                   | runHql                                       |

## 18.2. HQL

```
bpmAppService.BPMSServices.HQLService(hql) Data retrieval from the database.
```

| Name              | Type of Return  | Input Parameter  |
|-------------------|---|--|
| ExecuteHqlQuery   | System.Collections.IList                                  | string query   |
| ExecuteHqlQuery   | System.Collections.IList                                  | string query,<br>System.Collections.Generic.Dictionary<string, object><br>parameters |
| GetAllByCriterion | System.Collections.IList                                  | System.Type type   |
| GetAll            | System.Collections.IList                                  | System.Type type   |
| GetDomainObject   | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase | System.Type type, System.Guid<br>typeof(Rayvarz.BPMApp.[application]<br>Entity))     |

## 18.3. SQL

---

bpmAppService.BPMSServices.SqlService(sql) Data retrieval from the database.

| Name            | Type of Return           | Input Parameter                          | Description                      |
|-----------------|--------------------------|--|----------------------------------|
| ExecuteSqlQuery | System.Collections.IList | string query                             | Data retrieval from the database |
| ExecuteSqlQuery | System.Collections.IList | string query, System.Type<br>returnClass | Data retrieval from the database |

## 18.4. User

bpmAppService.BPMSServices.MercuryService(user) User data.

| Name                            | Type of Return                  | Input Parameter                 | Description                                    |
|---------------------------------|---------------------------------|---------------------------------|--|
| GetStaffManagerByUser           | System.Collections.Generic.List | string currentUser              | The Manager calls b                            |
| GetStaffManagerByUser           | Rayvarz.Profiles.Principal      | string currentStaff             | Calling the position m                         |
| GetUserManagerByUser            | System.Collections.Generic.List | string currentUser              | User call manager ba                           |
| GetUserManagerByStaff           | System.Collections.Generic.List | string currentStaff             | User call manager Ba                           |
| GetUsersByGroupCode             | System.Collections.Generic.List | string groupCode                | Call users based on                            |
| GetDepartmentByStaff            | Rayvarz.Profiles.Department     | string currentStaff             | Call Department bas                            |
| GetDepartmentByUser             | System.Collections.Generic.List | string currentUser              | Call Department bas                            |
| GetUserStaffs                   | System.Collections.IDictionary  | string currentUser              | User call                                      |
| GetManagerOfBeforeCEODepartment | Rayvarz.Profiles.Staff          | Rayvarz.Profiles.Staff<br>staff | Calling Manager befo                           |
| GetManagerOfUpperDepartment     | Rayvarz.Profiles.Staff          | Rayvarz.Profiles.Staff<br>staff | Calling manager abo                            |
| CurrentUserDepartment           | System.Collections.Generic.List |                                 | Calling the current us                         |
| CurrentUserManager              | System.Collections.Generic.List |                                 | Calling current user m                         |
| CurrentUserStaffs               | System.Collections.IDictionary  |                                 | Call the current User                          |
| CurrentStaff                    | Rayvarz.Profiles.Staff          |                                 | Call current current S                         |
| ManagerOfCurrentStaff           | Rayvarz.Profiles.Principal      |                                 | Call the manager of c                          |
| StarterStaff                    | Rayvarz.Profiles.Staff          |                                 | Initialization call                            |
| ManagerOfStarterStaff           | Rayvarz.Profiles.Principal      |                                 | Returns the manager                            |
| CurrentUser                     | Rayvarz.Profiles.User           |                                 | Call the current user                          |
| StarterUser                     | Rayvarz.Profiles.User           |                                 | Starter User Call                              |
| CurrentDepartment               | Rayvarz.Profiles.Department     |                                 | Calling the current D                          |
| GetManagerId                    | string                          | string userStaffId              | Calling the Manager                            |
| GetManager                      | Rayvarz.Profiles.Principal      | string userStaffId              | Returns a manager c                            |
| GetPrincipal                    | Rayvarz.Profiles.Principal      | string principalId              | Principal call based c                         |
| GetManagersOfCurrentStaff       | System.Collections.Generic.List |                                 | Returns all of the ma                          |
| ManagersIdOfCurrentStaff        | string                          |                                 | A string of characteri<br>managers output = 'D |



|              |                                 |                    |                                       |
|--------------|---------------------------------|--------------------|---------------------------------------|
|              |                                 |                    | B532-8BDAB377547<br>'BE44F4B8-8A13-4A |
| GetAllGroups | System.Collections.Generic.List | string principalId | Calling all user group                |

# 18.5. Rule

bpmAppService.BPMSServices.RuleService(rule) Call of Law and Action.

| Name       | Type of Return | Input Parameter  | Description   |
|------------|----------------|--|---|
| CallRule   | object         | Rayvarz.BPMApp.[application].Rules.Rules baseRule,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase domainObject, params object[] parameters   | Calling the law The parameter<br>two parameters are passed to t<br>bpmAppService.BPMSServices<br>domainObject, new Object[]{"ru<br>rule var args = parameters <sup>0</sup> var<br>Rayvarz.BPMApp.BPMSTest.M |
| CallAction | object         | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase domainObject,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase eventObject, action, string applicationName action:<br>Rayvarz.BPMApp.BPMSTest.Actions.Actions action:<br>Rayvarz.BPMApp.BPMSTest.Actions.PostFormActions action:<br>Rayvarz.BPMApp.BPMSTest.Actions.PostStarterFormActions action: Rayvarz.BPMApp.BPMSTest.Actions.PreFormActions action:<br>Rayvarz.BPMApp.BPMSTest.Actions.PreStarterFormActions | Call action   |

# 18.6. Process

bpmAppService.BPMSServices.ProcessService(process) Passing the status of a process.

| Name                           | Description   | Input Parameter   |
|--------------------------------|---|---|
| RaiseEvent                     | Rayvarz.BPMS.RuleModel.GuardResult,<br>Rayvarz.BPMS.RuleModel.GuardResult result;=><br>result.Messagestring,result.Resultbool   | Rayvarz.BPMApp<br>Rayvarz.BPMS.In   |
| RaiseEvent                     | Rayvarz.BPMS.RuleModel.GuardResult<br>Rayvarz.BPMS.RuleModel.GuardResult result;=><br>result.Messagestring,result.Resultbool  | Rayvarz.BPMApp<br>Rayvarz.BPMS.In<br>Rayvarz.Profiles.  |
| StartProcess                   | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase   | Rayvarz.BPMS.In<br>tracking, params l   |
| StartProcess                   | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase   | Rayvarz.BPMS.In<br>tracking, User use   |
| StartProcessOnPersistentObject | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>persistentObject, User user, ProcessVersionInfo<br>processVersionInfo | Start the process<br>Rayvarz.BPMApp<br>sample : bpmApp<br>ProcessService.S<br>bpmAppService.E<br>Rayvarz.BPMApp |

## 18.7. Domain

bpmAppService.BPMSServices.DomainObjectService(domain) Calling and creating and deleting an entity.

| Name               | Type of Return  | Input Parameter   |
|--------------------|---|---|
| GetDomainObject    | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase | System.Type domainObjectT   |
| CreateDomainObject | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase | System.Type domainObjectT<br>(Rayvarz.BPMApp. [Applicati  |
| DeleteDomainObject |   | Rayvarz.BPMS.Infrastructure<br>domainObject   |
| SaveDomainObject   |   | Rayvarz.BPMS.Infrastructure<br>domainObject   |
| GetAll             | System.Collections.IList                                  | System.Type type<br>typeof(Rayvarz.BPMApp.[app  |
| GetParentObject    | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase | Rayvarz.BPMS.Infrastructure<br>childDomainObject Child Exis<br>hierarchicalPropertyName Ex<br>parentPropertyName is the p |
| GetParentObject    | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase |   |

# 18.8. Task

bpmAppService.BPMSServices.TaskService(task) Job service.

| Name             | Type of Return | Input Parameter                                | Description   |
|------------------|----------------|--|---|
| SendToCompleted  |                | DomainObjectBase<br>domainObject,<br>User user | Dashboard and fill out a job of work has been done<br>related entities user: User doing work domainObject: Do not make the object of entity     |
| SendToTerminated |                | DomainObjectBase<br>domainObject,<br>User user | Do not do things out of the dashboard and annulment of<br>related entities user: User doing work domainObject: Do not make the object of entity |

# 18.9. TimeEvent

bpmAppService.BPMSServices.TimeEventService(timeevent) Event information.

| Name     | Type of Return | Input Parameter   | Description                 |
|----------|----------------|---|-----------------------------|
| StartJob |                | Rayvarz.BPMApp.[application].Events baseEvent,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject | Start the<br>event<br>Timed |

## 18.10. Notification

bpmAppService.BPMSServices.NotificationService(notification) Notification information.

| Name               | Type of Return | Input Parameter  | Description        |
|--------------------|----------------|--|--------------------|
| SendByTaskType     |                | Rayvarz.BPMApp.[application].Tasks.TaskList.TaskTypes<br>baseTaskType,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject  | Send notifications |
| SendByNotification |                | Rayvarz.BPMApp.[application].Notifications.Notifications<br>baseNotification,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject, Rayvarz.Profiles.Principal principal       | Send notifications |
| SendByNotification |                | Rayvarz.BPMApp.[application].Notifications.Notifications<br>baseNotification,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject, Rayvarz.Profiles.User user                 | Send notifications |
| SendByNotification |                | Rayvarz.BPMApp.[application].Notifications.Notifications<br>baseNotification,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject, System.Collections.Generic.IList users     | Send notifications |
| SendByNotification |                | Rayvarz.BPMApp.[application].Notifications.Notifications<br>baseNotification,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject, System.Collections.Generic.IList addresses | Send notifications |
| SendByNotification |                | Rayvarz.BPMApp.[application].Notifications.Notifications<br>baseNotification,<br>Rayvarz.BPMS.Infrastructure.DomainObject.DomainObjectBase<br>domainObject, string address                             | Send notifications |
| Send               |                | string subject, string body, NotificationAgent<br>notificationAgent, bool isAsync, Rayvarz.Profiles.Principal<br>principal NotificationAgent : DefaultAgent, Email, Messenger,<br>SMS, eOffice         | Send notifications |
| Send               |                | string subject, string body, NotificationAgent notificationAgent,<br>bool isAsync, Rayvarz.Profiles.User user  | Send notifications |
| Send               |                | string subject, string body, NotificationAgent notificationAgent,<br>bool isAsync, System.Collections.Generic.IList users  | Send notifications |
| Send               |                | string subject, string body, NotificationAgent notificationAgent,<br>bool isAsync, System.Collections.Generic.IList addresses  | Send notifications |

|                |  |  |  |
|----------------|--|--|--|
| Send           |  | string subject, string body, NotificationAgent notificationAgent, bool isAsync, string address | Send notifications   |
| SendByActivity |  | ProcessVersionInfo processVersionInfo, string activityName, DomainObjectBase domainObject      | Send notifications of a status (activityName = Name status)) |
| SendCopy       |  | BaseTemplate notification, DomainObjectBase domainObject, Event eEvent                         | submit Transcript  |



# 18.11. Tracking

bpmAppService.BPMSServices.TrackingService(tracking) Tracking information .

| Name                 | Type of Return                  | Input Parameter   |
|----------------------|---------------------------------|---|
| GenerateTrackingCode | string                          | Rayvarz.BPMS.Infrastructure.DomainObject.DomainObject<br>domainObject |
| GetTrackingNumber    | long                            |   |
| GetTrackingInfoList  | System.Collections.Generic.List |   |

# 18.12. Data Source

bpmAppService.Services.DataSourceServices(datasource) Data retrieval from the database.

| Name                | Type of Return        | Input Parameter                                    | Description  |
|---------------------|-----------------------|--|--|
| ExecuteQuery        | System.Data.DataTable | BaseTemplate<br>connection, string<br>query        | Fetch information from the database connection<br>the database defined in the communication / dat<br>Rayvarz.BPMApp.BPMSTest.Connections.Conn<br>query: Required Query   |
| GenerateSelectQuery | string                | DataTable<br>dataTable, List<<br>schemadeclaration | Convert DataTable to Query for use in cascading<br>schemadeclaration.Add (new KeyValuePair ("Id"<br>schemadeclaration.Add (new KeyValuePair ("Na<br>schemadeclaration.Add (new KeyValuePair ("Ag<br>mAppService.BPMSServices. DataSourceServic<br>(dataTable, schemadeclaration); The output stru<br>field type that matches the data schema. Schem<br>structure is the field name and field type that ma<br>Schemadeclaration |

## 18.13. DateTime

bpmAppService.HelperServices.DateTimeService(datetime) Date and time information.

| Name                     | Type of Return | Input Parameter                               | Description   |
|--------------------------|----------------|---|---|
| DateDiffGregorian        | TimeSpan       | DateTime startDate, DateTime endDate          | Difference between two dates                                    |
| AddDate                  | decimal        | decimal date, int years, int months, int days | Add day, month, year, year to date (solar history)              |
| DateDiffPersian          | TimeSpan       | decimal startDate, decimal endDate            | Difference between two dates (solar)                            |
| GetGregorianCalendarDate | DateTime       | decimal persianDate                           | Convert date format from Persian Calendar to Gregorian Calendar |
| GetDate                  | DateTime       | decimal gregorianCalendarDate                 | Return Date Type of a numeric date (yyyymmdd)                   |
| TimeDiff                 | TimeSpan       | TimeSpan startTime, TimeSpan endTime          | Difference of two times   |
| TimeAdd                  | TimeSpan       | TimeSpan time1, TimeSpan time2                | Sum two times   |
| CurrentDate              | DateTime       |   | Today's date  |
| CurrentTime              | TimeSpan       |   | Time now  |
| DayOfWeek                | string         |   | Day of the week   |

# 18.14. Math

bpmAppService.HelperServices.Arithmetics(math) numeric operators.

| Name     | Type of Return | Input Parameter                  | Description    |
|----------|----------------|----------------------------------|----------------|
| Add      | double         | double operand1, double operand2 | Sum            |
| Subtract | double         | double operand1, double operand2 | Subtract       |
| Multiply | double         | double operand1, double operand2 | Multiplication |
| Division | double         | double operand1, double operand2 | Division       |

# 18.15. Text

bpmAppService.HelperServices.TextService(text) Text Operators.

| Name     | Type of Return | Input Parameter                     | Description   |
|----------|----------------|-------------------------------------|---|
| Concat   | string         | string operand1,<br>string operand2 | Combine two texts   |
| Contains | bool           | string operand1,<br>string operand2 | Returns a value indicating whether a specified substring(operand2) occurs within this string(operand1). |

## 18.16. Point

---

(Scripting) Rayvarz.BPMApp.BPMSTest.FormControls .

Example:

### **Name Control**

```
string name = formControl.RayNumericTextBox1.Id;
```

### **Enable and disable control**

```
formControl.RayNumericTextBox1.Enabled = false;
```

```
formControl.RayNumericTextBox1.Enabled = true;
```

### **Visible control**

```
formControl.RayNumericTextBox1.Visible = false;
```

```
formControl.RayNumericTextBox1.Visible = true;
```

### **Mandatory data control**

```
SetRequiredFieldValidate(string message, string validationGroup)
```

```
formControl.RayNumericTextBox1.SetRequiredFieldValidate("Enter data", validationGroup);
```

### **Data range control**

```
SetRangeValidate(string message, string validationGroup, double minValue, double maxValue)
```

```
formControl.RayNumericTextBox1.SetRangeValidate("The data is between 0 and 10 ", validationGroup,  
0, 10);
```

### **Setting an entity ID (DomainObject)**

If you want to store an entity id with a certain value in the database, we will write the assignment script in action before the start form.

```
domainObject.Id = new Guid("80E67399-606F-49D4-AFBC-EEE8067B55D2")
```

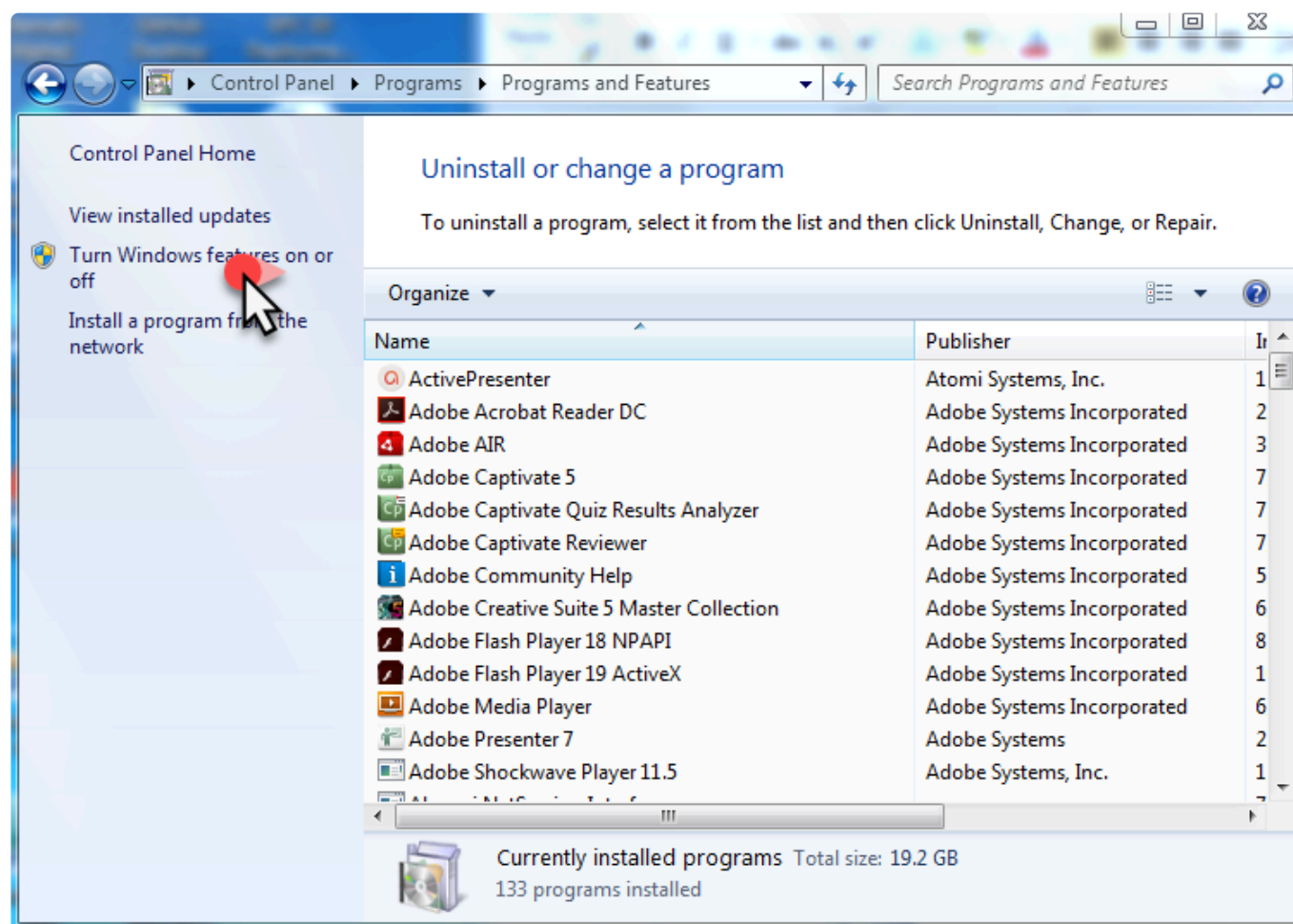
# 19. Attachments

---

## 19.1. 13.1 Microsoft IIS 6 SSL Compatibility Components

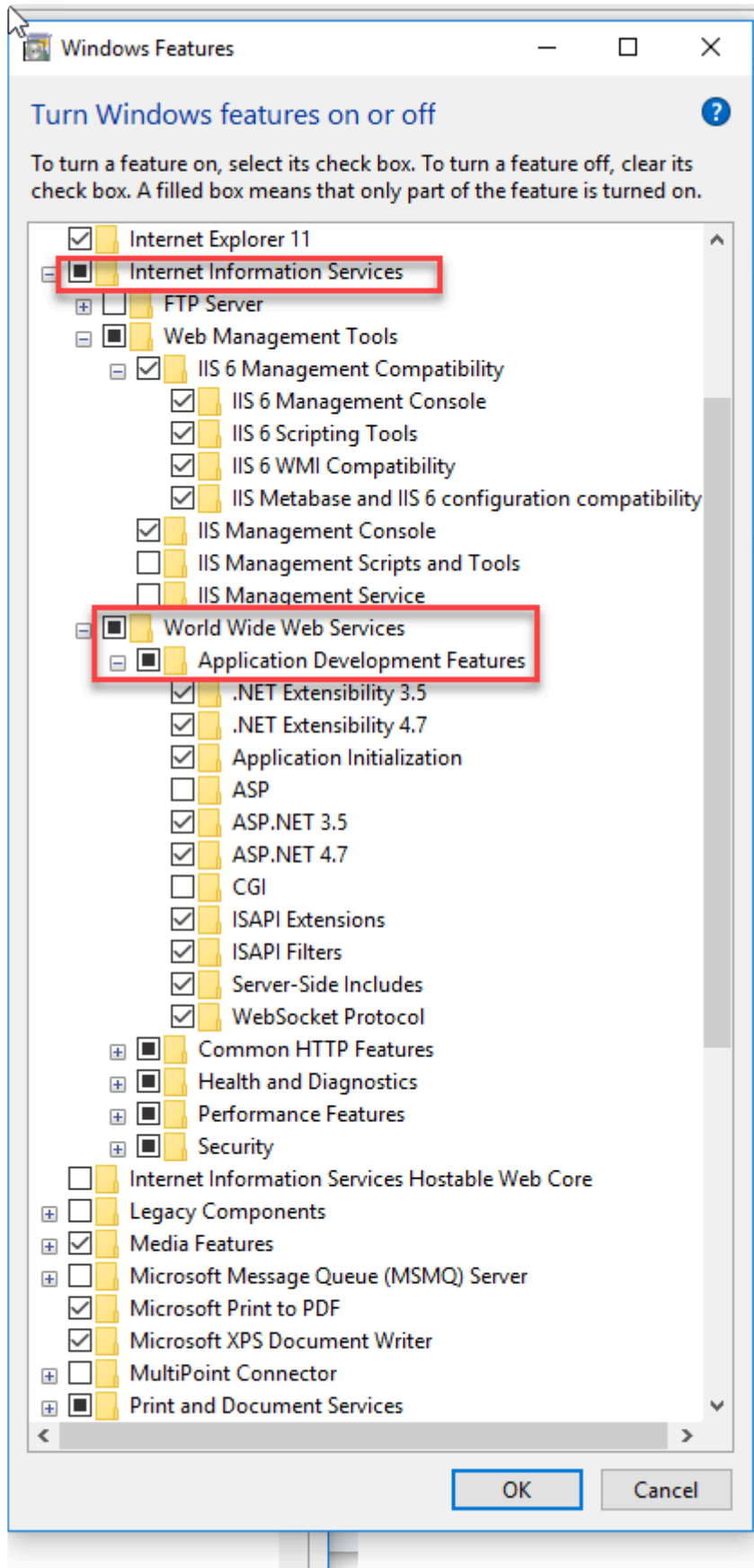
Install the IIS 6.0 Management Compatibility Components in Windows 7 or in Windows Vista from Control Panel

1. Click Start, click Control Panel, click Programs and Features, and then click Turn Windows features on or off.



2. Open Internet Information Services.
3. Open Web Management Tools.
4. Open IIS 6.0 Management Compatibility.
5. Click to select the IIS 6 Metabase and IIS 6 configuration compatibility, IIS 6 WMI Compatibility, IIS 6 Scripting Tools, and IIS 6 Management Console check boxes.





**NOTE:** For Windows 8/8.1 in IIS the following features should be selected as well (switched off by default)

World Wide Web Services ->Common HTTP Features -> Static Content.

6. Select World Wide Web Services Select Application Development Features Check all the fields except ASP and CGI

7. Click OK.

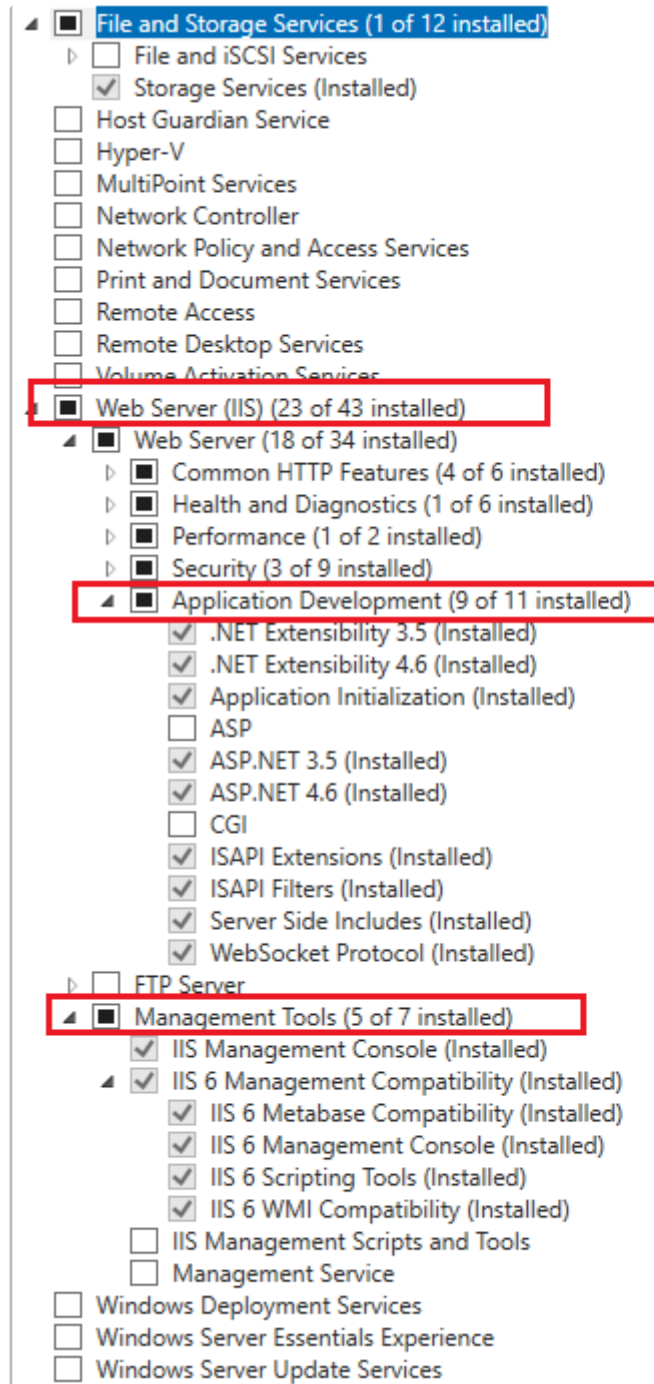
### **Instructions for Windows Server 2008.**

Install the IIS 6.0 Management Compatibility Components in Windows Server 2008 R2 or in Windows Server by using the Server Manager tool

1. Click Start, click Administrative Tools, and then click Server Manager.
2. In the navigation pane, expand Roles, right-click Web Server (IIS), and then click Add Role Services.
3. In the Select Role Services pane, scroll down to IIS 6 Management Compatibility.
4. Click to select the IIS 6 Metabase Compatibility, IIS 6 WMI Compatibility, and IIS 6 Management Console check boxes.
5. In the Select Role Services pane, click Next.
6. In the Confirm Installations Selections pane, click Install.
7. Click Close to exit the Add Role Services wizard.

### **Instructions for Windows Server 2016**

1. Open Server Manager and click Manage > Add Roles and Features. Click Next.
2. Select Role-based or feature-based installation and click Next.
3. Select the appropriate server. The local server is selected by default. Click Next.
4. On the Select role services dialog box, verify that the web server components listed below are enabled. Click Next

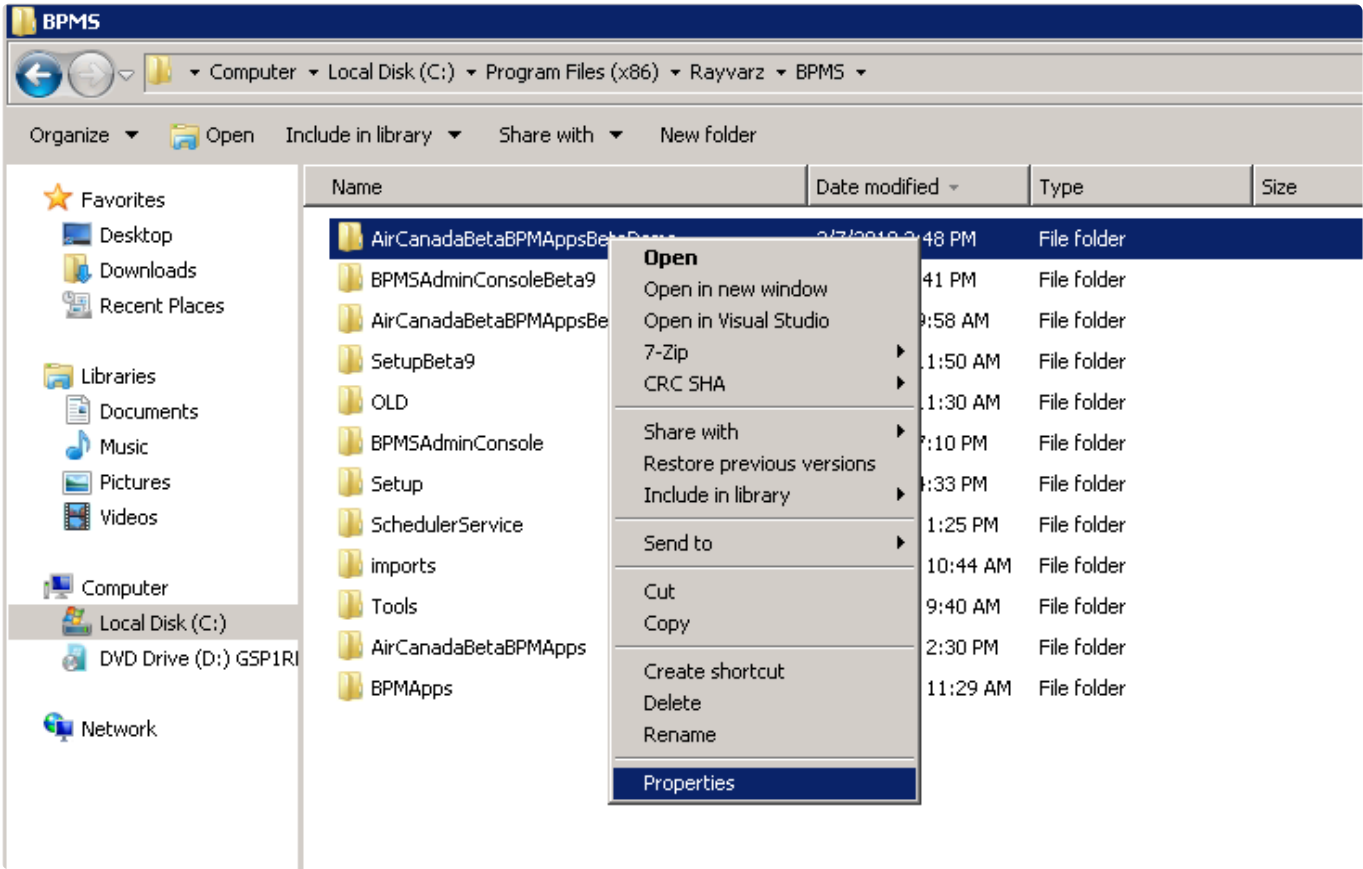


5. Verify that your settings are correct and click Install.

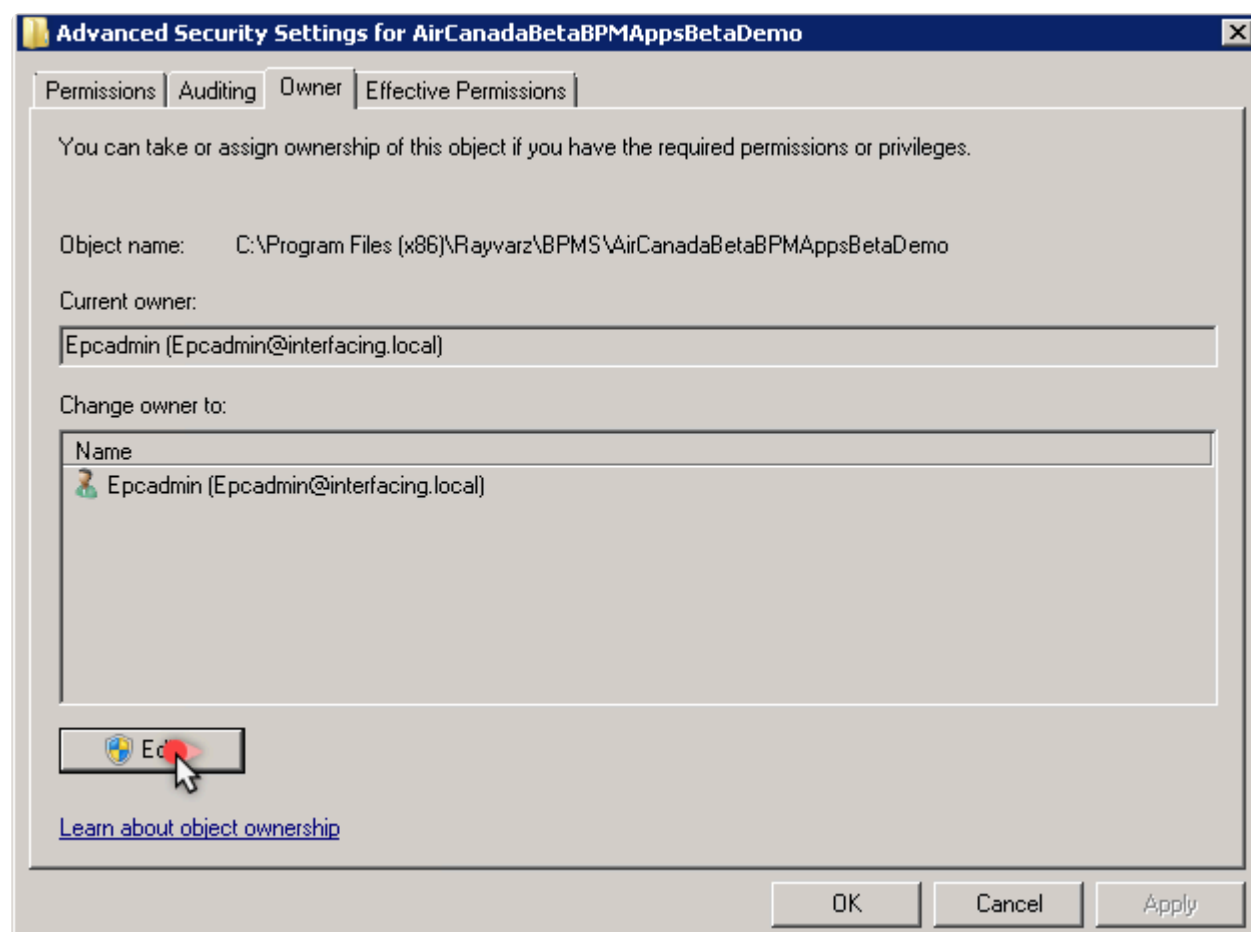
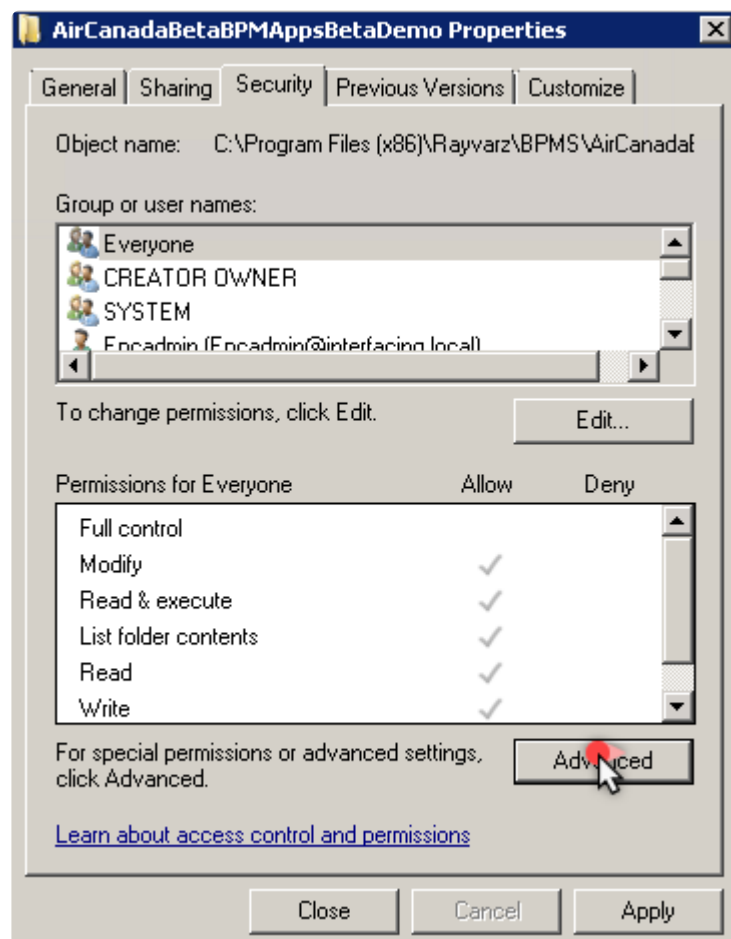
6. When the installation completes, click Close to exit the wizard.

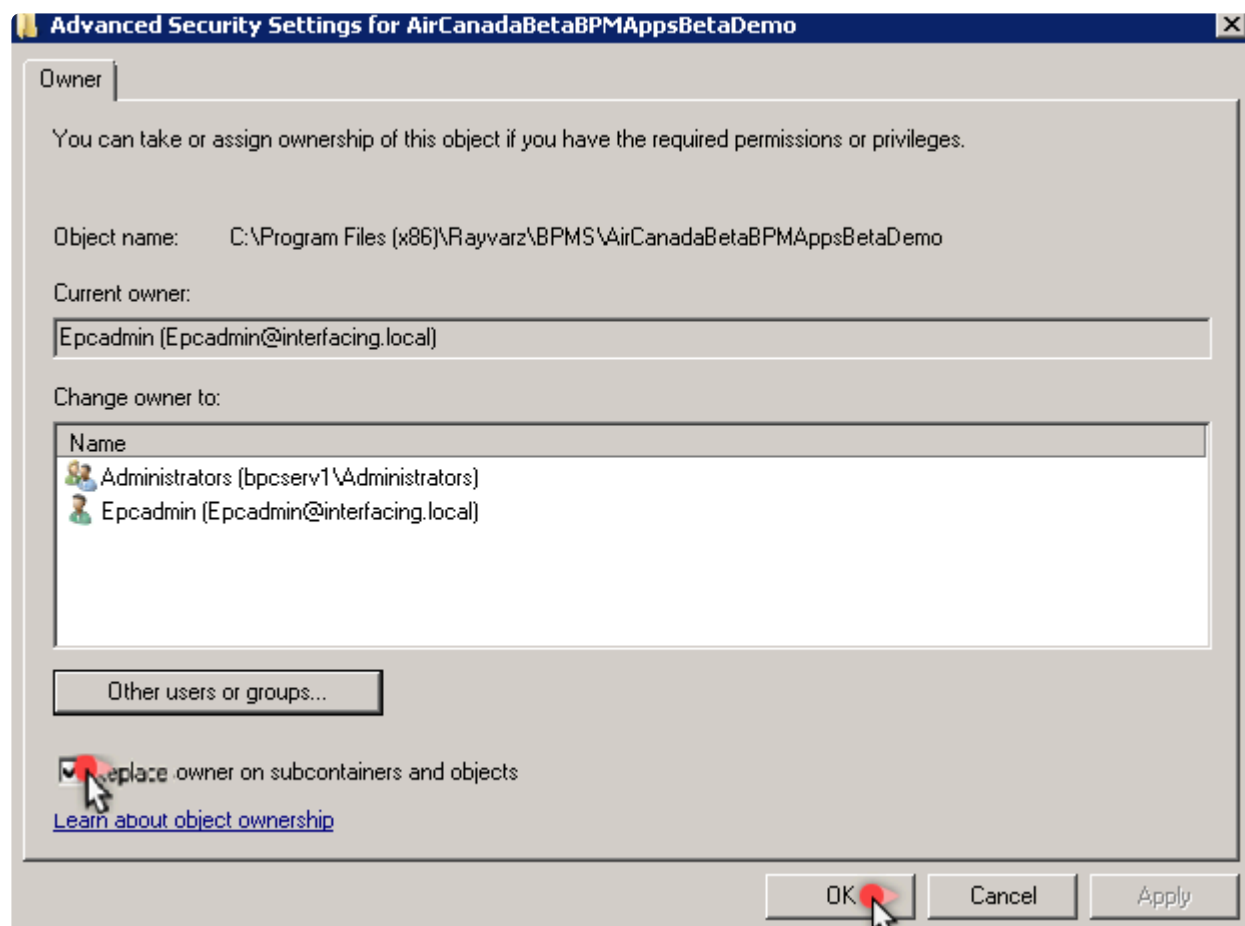
# 19.2. 13.2 Setting Permission For sharing BPMApp Folder.

Setting Permission For shared BPMApp Folder in multi designer environment.

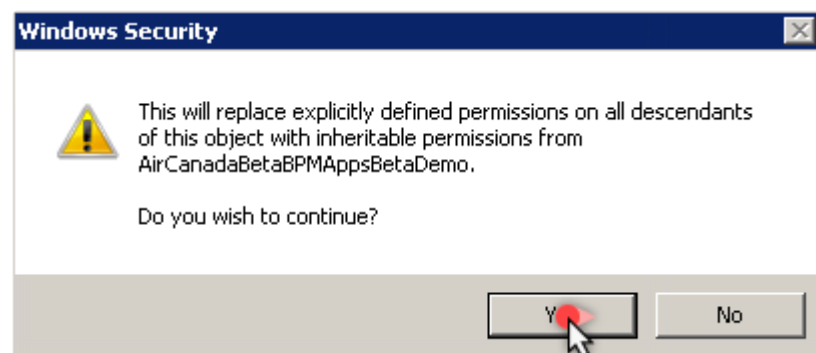
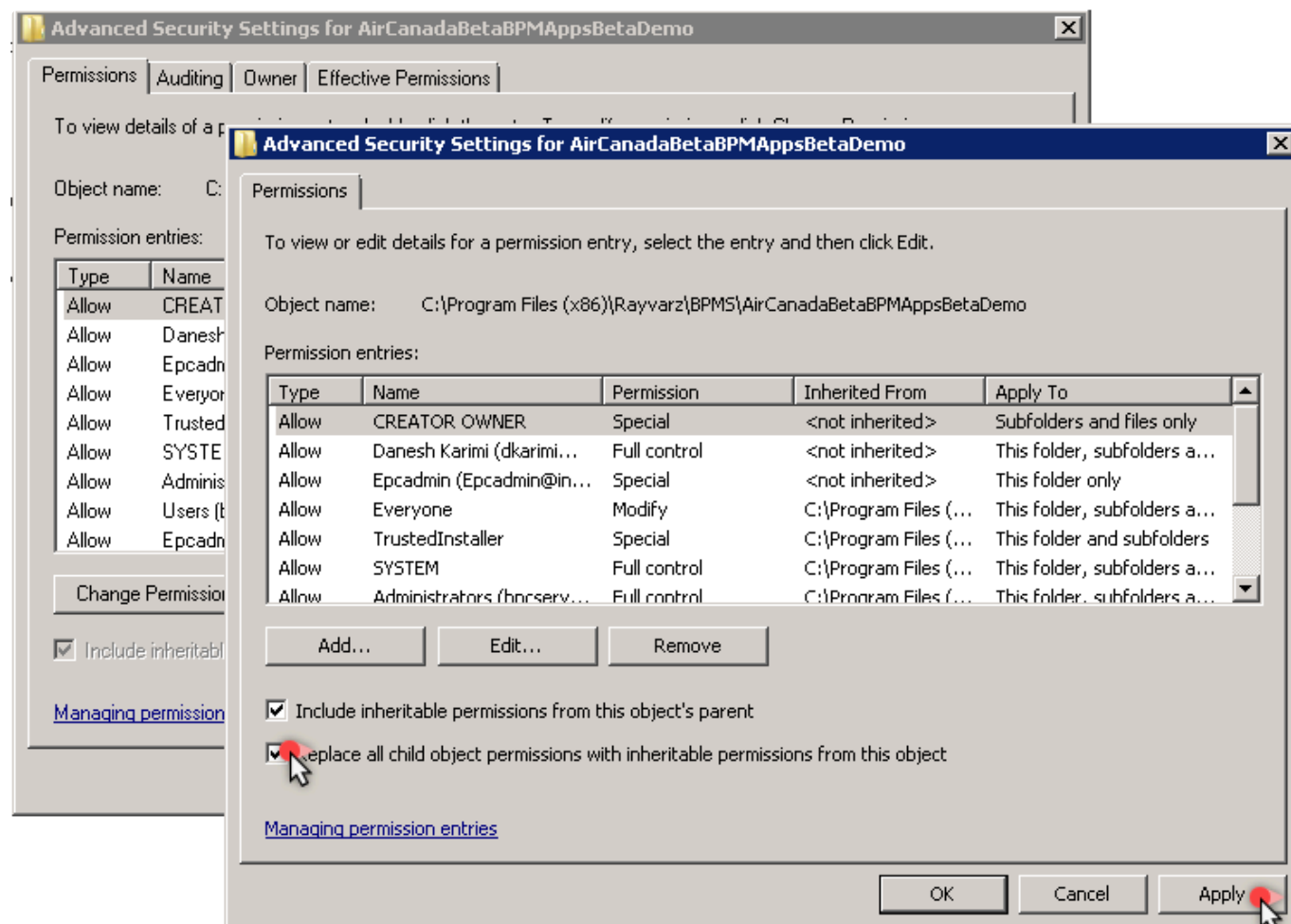


1. Take ownership for BPMS and all Children

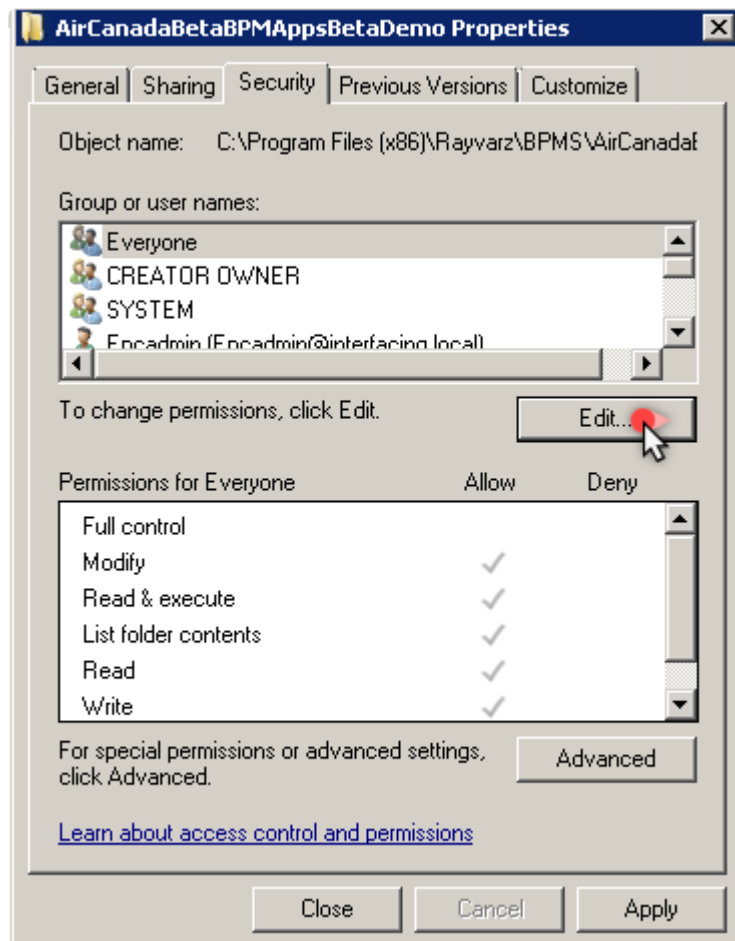




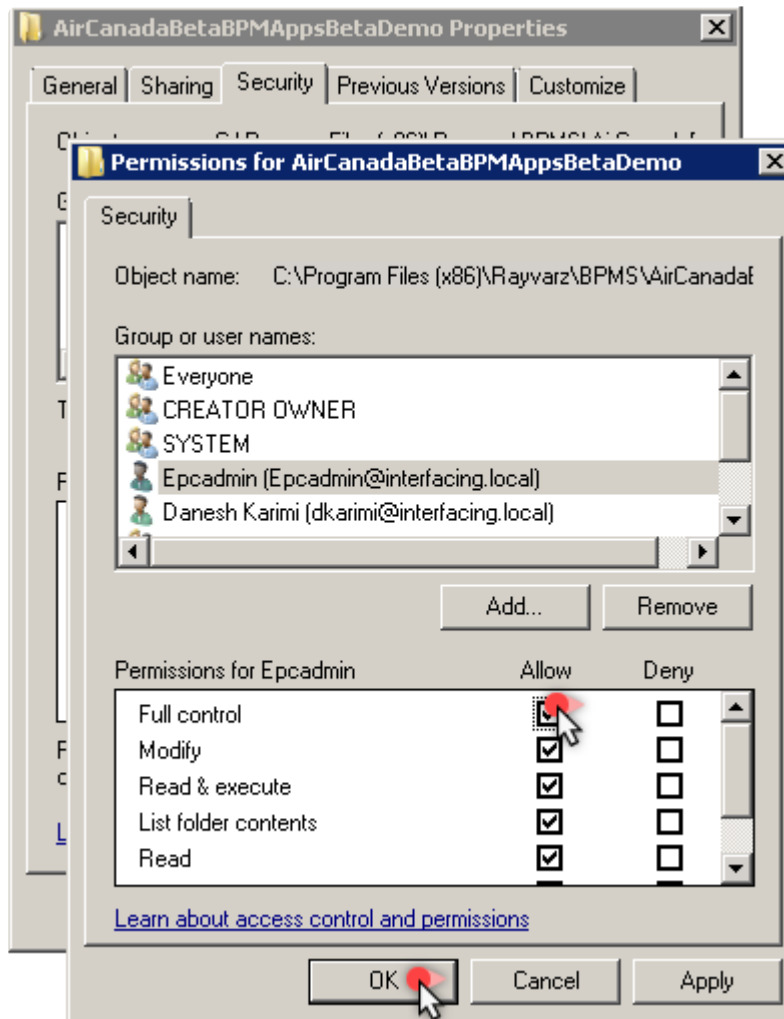
## 2. Override permissions (replace all children permissions)



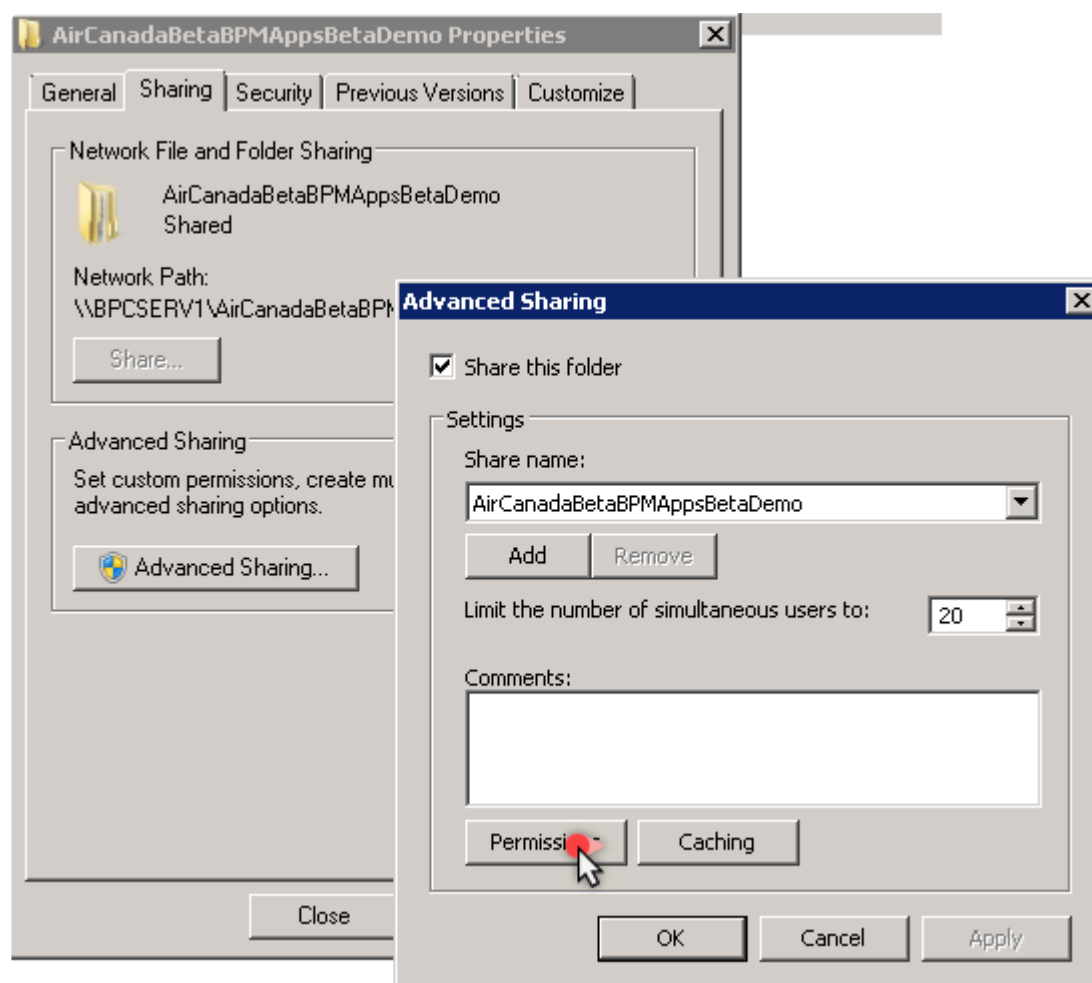
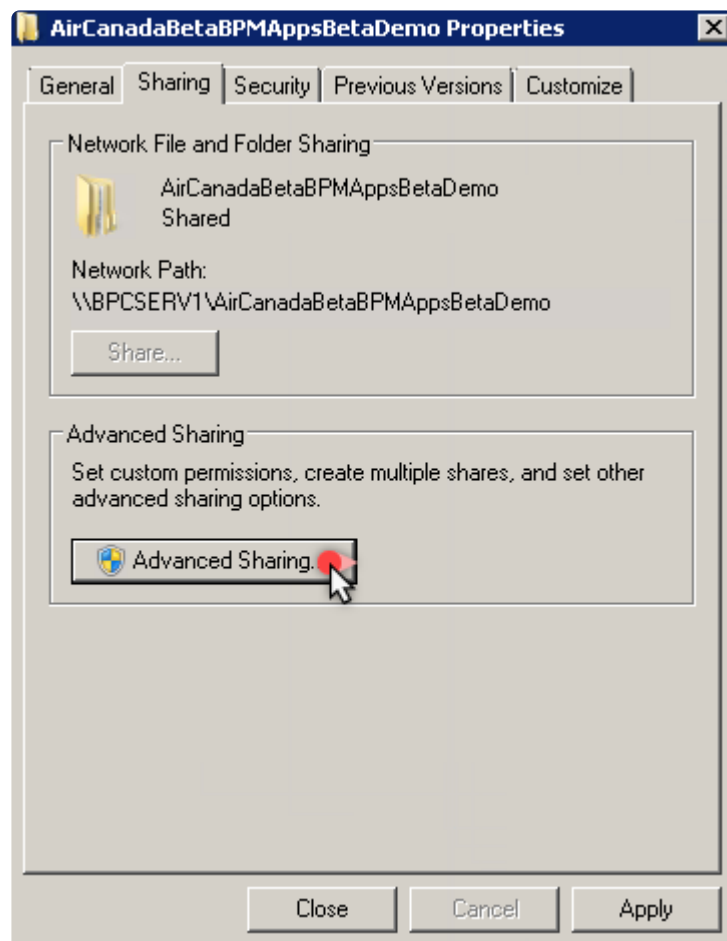
3. Add security permission to that user-> Designer

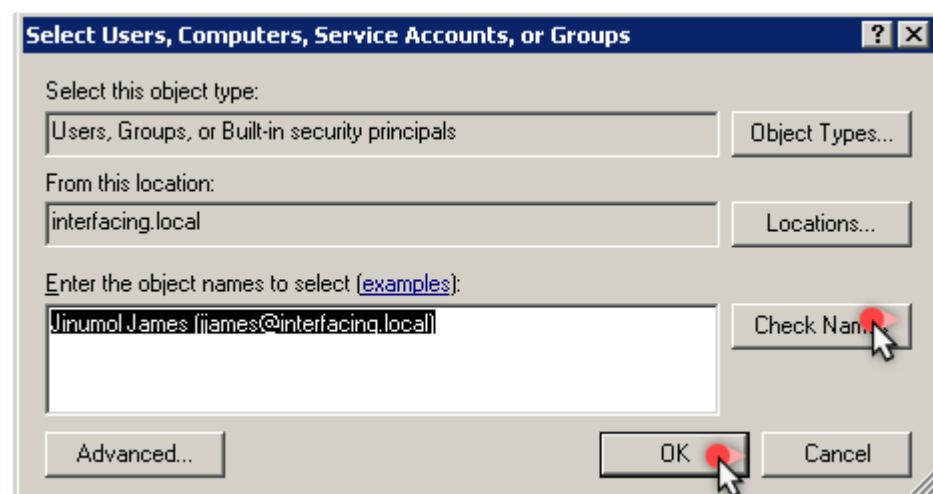
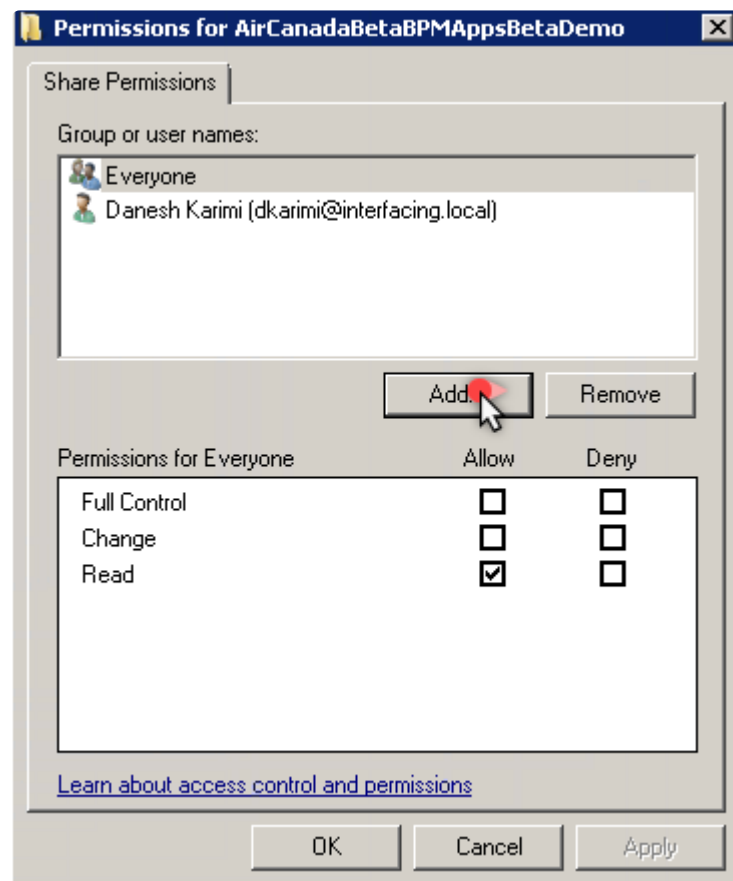


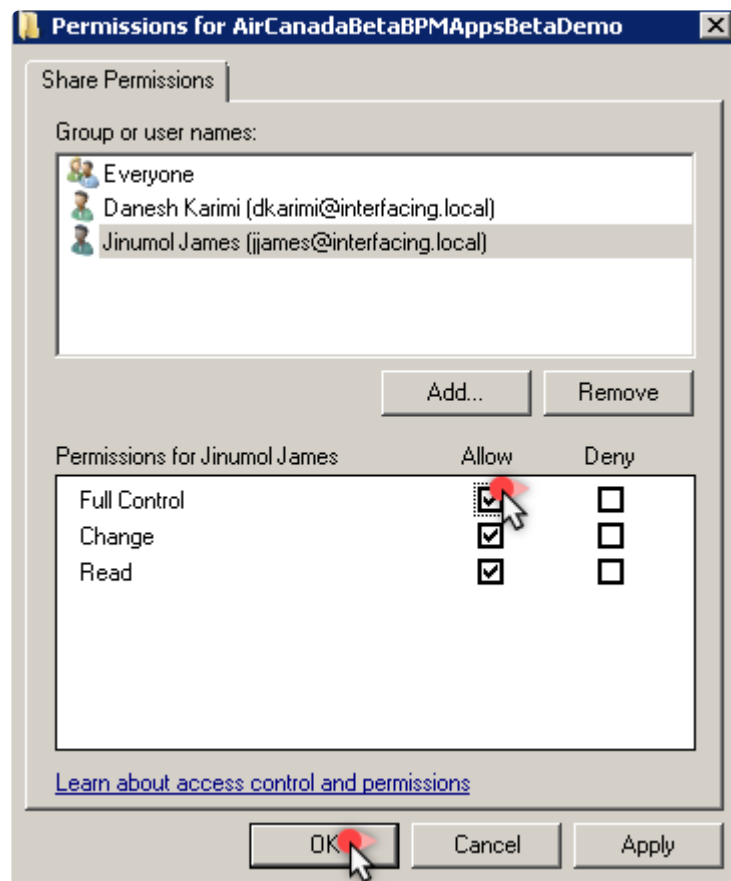




#### 4. Add share permissions







## **19.3. 13.3 MSDTC Services**

---

## 19.3.1. 13.3.1 Introduction

---

The **Microsoft Distributed Transaction Coordinator (MSDTC)** service is a component of modern versions of Microsoft Windows that is responsible for coordinating transactions that span multiple resource managers, such as databases, message queues, and file systems.

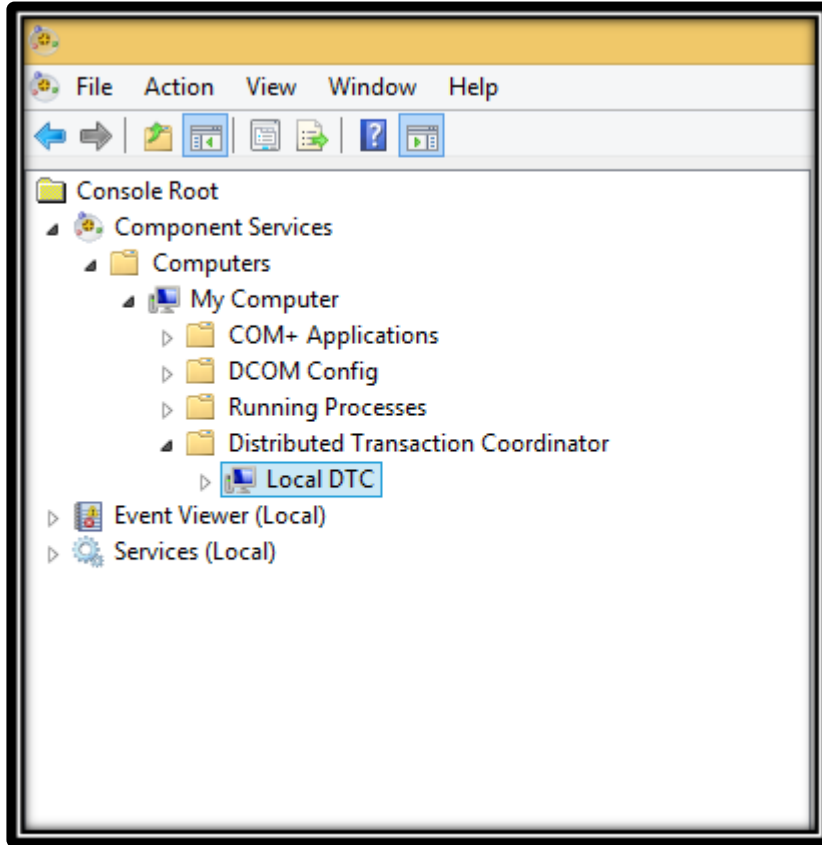
MSDTC performs the transaction coordination role for components, usually with COM and .NET architectures.

In case of BPMS application, MSDTC Service is required only on server.

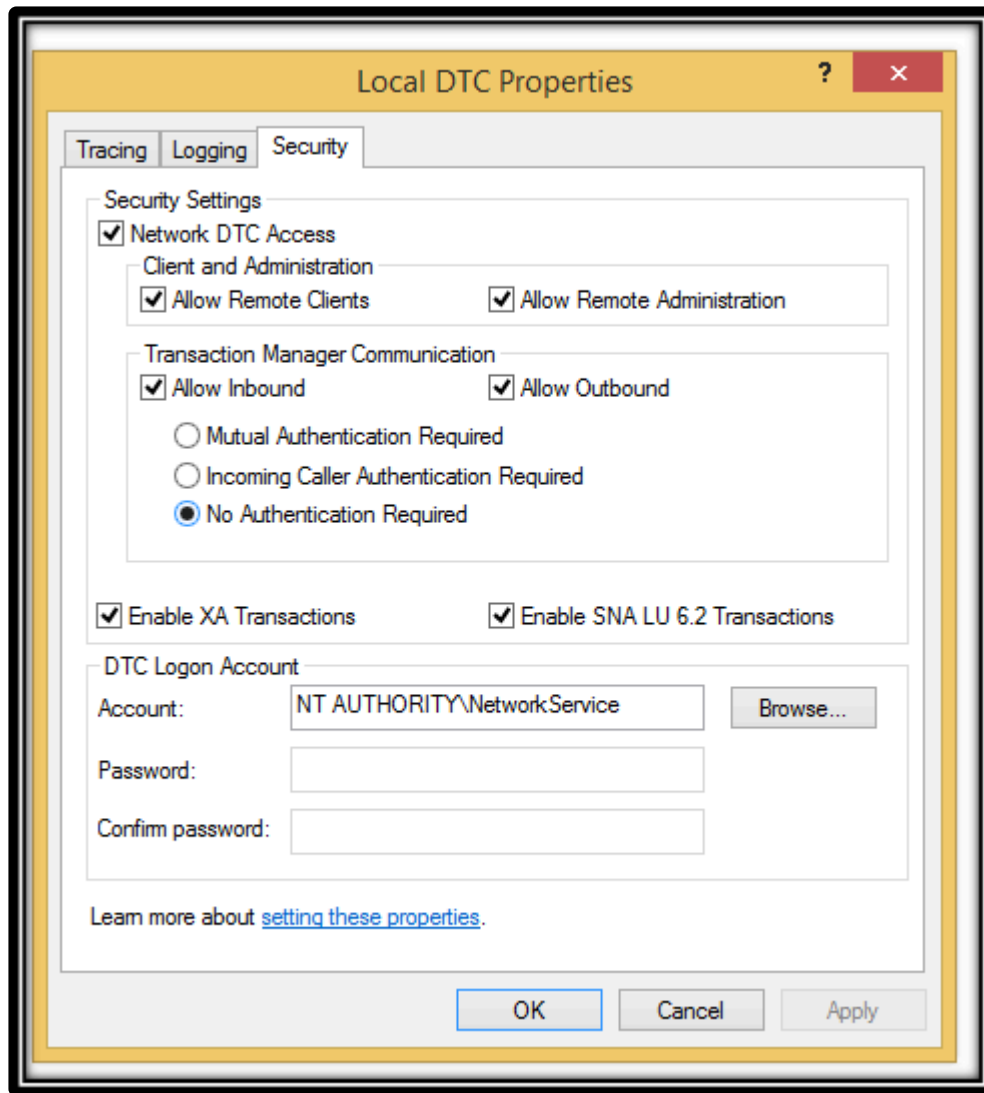
## 19.3.2. 13.3.2 Steps to Configure MSDTC Services

---

1. From **Start**, search for **dcomcnfg** and press **Enter** on your keyboard.



2. Expand the nodes in the **Console** pane to locate the DTC (e.g. Local DTC).
3. Right-click the DTC then click **Properties**.
4. Click the **Security** tab.



5. Under **Security Settings**, select the **Network DTC Access**
6. Under **Transaction Manager Communication**, select the **Allow Inbound** and **Allow Outbound** checkboxes.
7. Under **Transaction Manager Communication**, click the **No Authentication Required** radio button.
8. Select the **Enable XA Transactions** checkbox.
9. Click **OK**.

**Note:** In Windows 8 search for dcomcnfg in Start. And for Windows 10 search for Component Services in Start.

<https://support.resolver.com/hc/en-ca/articles/207161116-Configure-Microsoft-Distributed-Transaction-Coordinator-MSDTC->



## 19.4. 13.4 Externalizing the web form

If it's necessary to externalize the form, the user:

In the URL instead of "private#" need to put private?co=t#

And if it's anonymous instead of home we need to put home?co=t/#

### Normal Form:

| Flag                     | Priori... | Subject   | Created at          | Response deadline | Done at | Sender       |
|--------------------------|-----------|---|---------------------|-------------------|---------|--------------|
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Waiting for the confirmation of attendees test1 | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | <b>Approve Document Change:</b>                 | <b>2 months ago</b> |                   |         | <b>admin</b> |

### Externalized Web Form

| Flag                     | Priori... | Subject   | Created at          | Response deadline | Done at | Sender       |
|--------------------------|-----------|---|---------------------|-------------------|---------|--------------|
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Waiting for the confirmation of attendees test1 | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | Approve Document Change:                        | 2 months ago        |                   |         | admin        |
| <input type="checkbox"/> |           | <b>Approve Document Change:</b>                 | <b>2 months ago</b> |                   |         | <b>admin</b> |

## 20. Troubleshooting

---

# 20.1. This configuration section cannot be used at this path

**Issue:** When we try to open web for the first time after installation we may get this error:

HTTP Error 500.19 – Internal Server Error  
The requested page cannot be accessed because the related configuration data for the page is invalid.

Detailed Error Information:  
Module IIS Web Core  
Notification BeginRequest  
Handler Not yet determined  
Error Code 0x80070021  
Config Error This configuration section cannot be used at this path. This happens when the section is locked at a parent level. Locking is either by default (overrideModeDefault="Deny"), or set explicitly by a location tag with overrideMode="Deny" or the legacy allowOverride="false"

**HTTP Error 500.19 - Internal Server Error**  
The requested page cannot be accessed because the related configuration data for the page is invalid.

Detailed Error Information:

|              |   |               |                                |
|--------------|---|---------------|--------------------------------|
| Module       | IIS Web Core  | Requested URL | http://localhost:80/MyBPMS     |
| Notification | BeginRequest  | Physical Path | C:\inetpub\wwwroot\RayvarzBPMS |
| Handler      | Not yet determined  | Logon Method  | Not yet determined             |
| Error Code   | 0x80070021  | Logon User    | Not yet determined             |
| Config Error | This configuration section cannot be used at this path. This happens when the section is locked at a parent level. Locking is either by default (overrideModeDefault="Deny"), or set explicitly by a location tag with overrideMode="Deny" or the legacy allowOverride="false". |               |                                |
| Config File  | \\?\C:\inetpub\wwwroot\RayvarzBPMS\web.config   |               |                                |

Config Source:

```
133: </staticContent-->
134: <handlers>
135: <remove name="ChartImageHandler" />
```

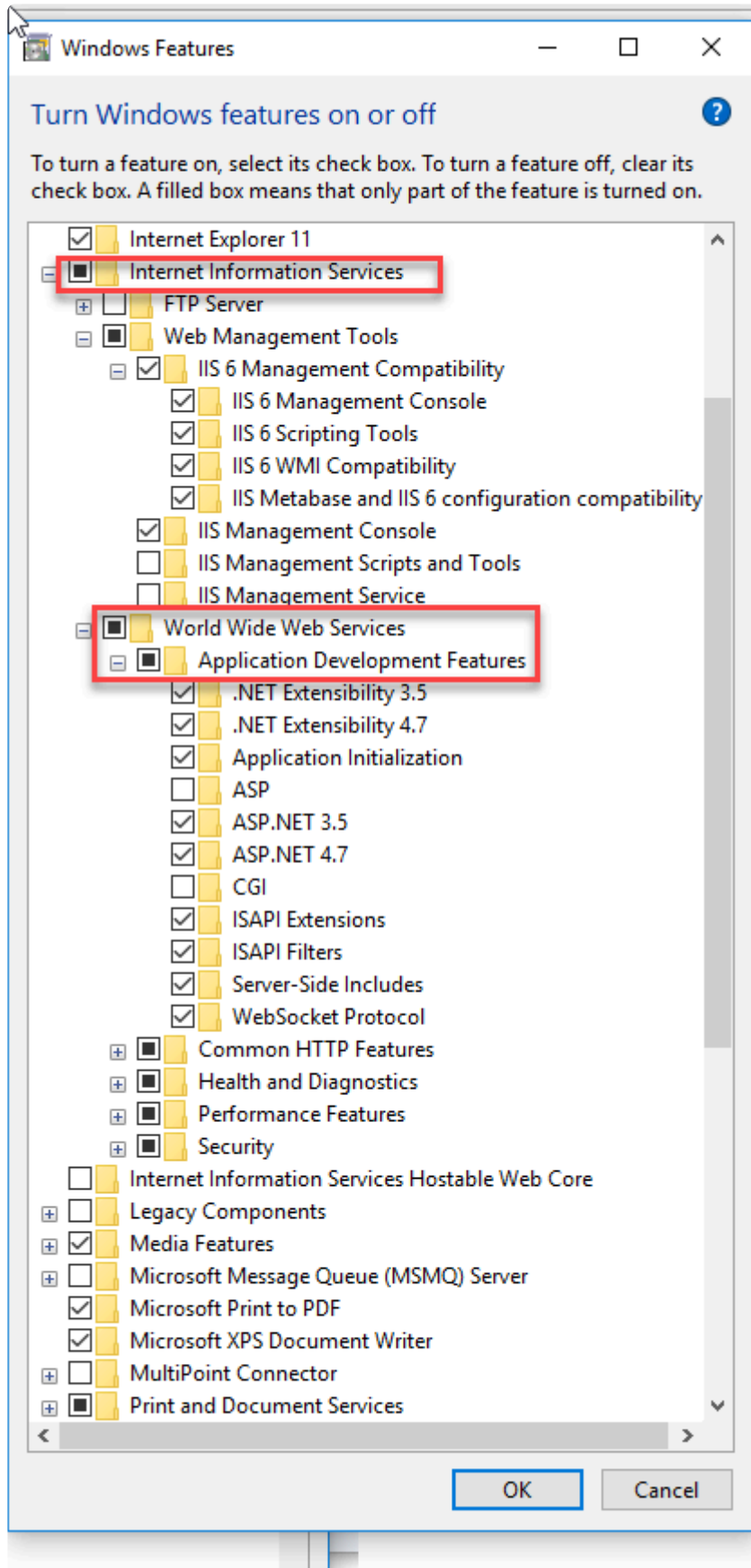
More Information:

This error occurs when there is a problem reading the configuration file for the Web server or Web application. In some cases, the event logs may contain more information about what caused this error.

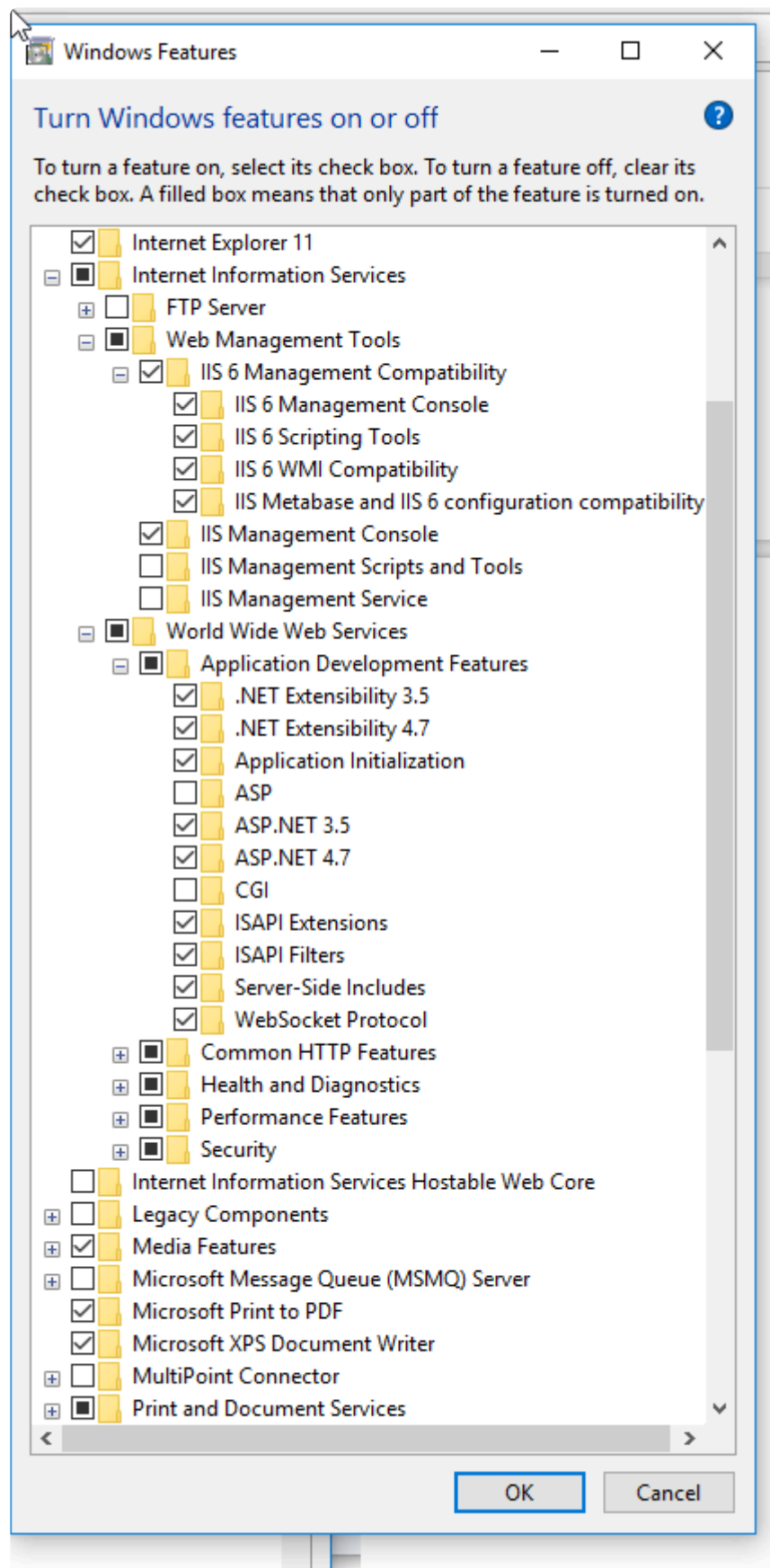
[View more information.](#)

## Solution

We need to install all of the application development features of IIS.



Note that CGI and ASP (classic) is not needed.



## 20.2. Access to the path 'Rayvarz\BPMS\Temp' is denied

---

**Issue:** Error in opening a web form that includes a rule (any type of rule ex: pre form action, action, etc.) in the debug mode.

### Solution

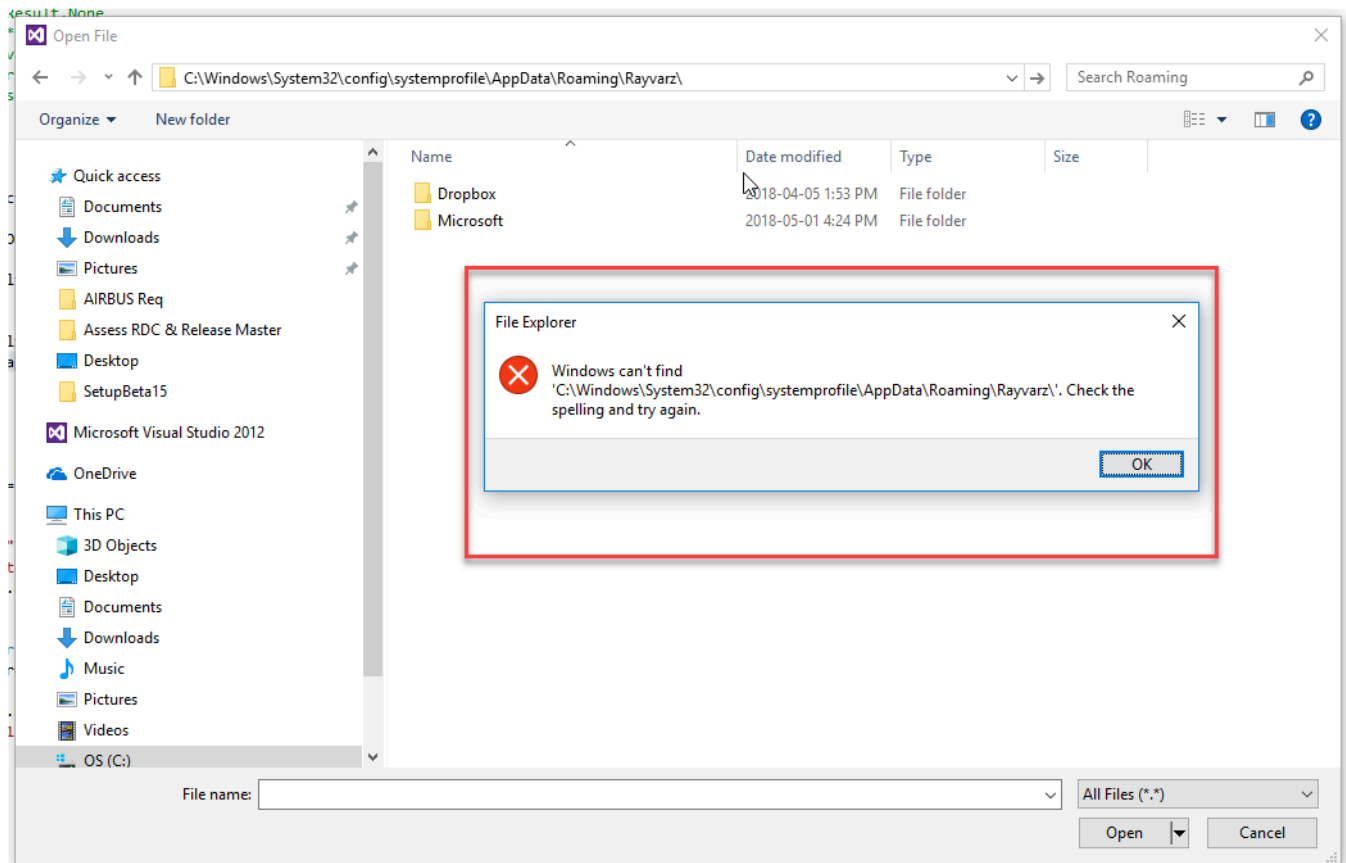
- 1) Give NETWORKSERVICE full access to C:\Windows\System32\config\systemprofile\AppData\Roaming
- 2) Remove ReadOnly from C:\Windows\System32\config\systemprofile\AppData\Roaming
- 3) Reset web service

## 20.3. Not able to access temp files by visual studio

**Issue:** In order to be able to debug rules in the web we need to attach our Visual Studio to the web process.

Sometimes we could find the right destination during debug which is:

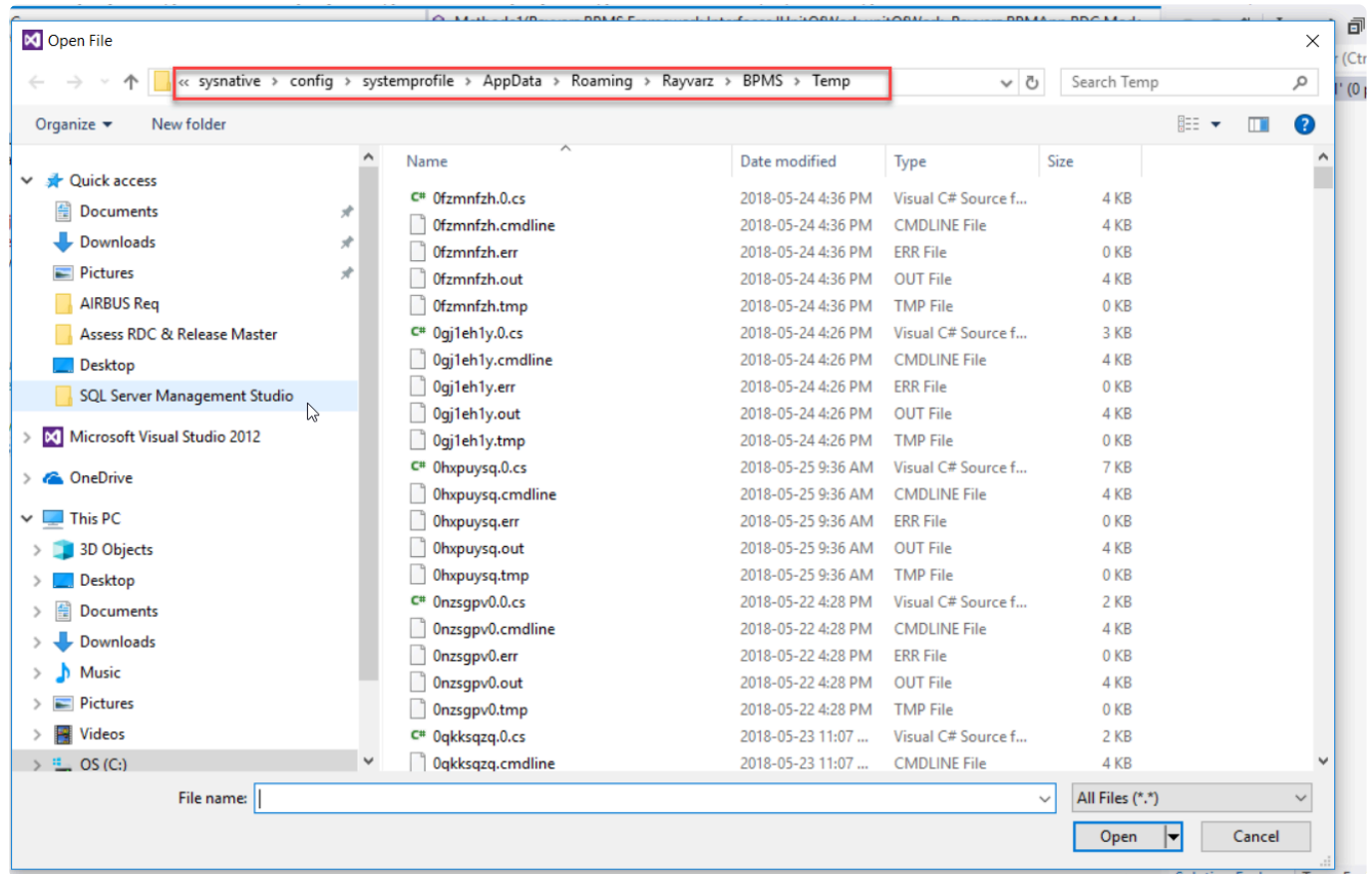
C:\Windows\System32\config\systemprofile\AppData\Roaming\Company



### Solution

Visual Studio is a 32-bit process. On a 64-bit OS it is subject to the File System Redirector. A workaround is to navigate to c:\windows\sysnative instead. So we need to open this URI with Visual Studio:

C:\Windows\sysnative\config\systemprofile\AppData\Roaming\Rayvarz\BPMS\Temp



Remember that you can't open this folder directly with windows explorer. It's only visible to 32 apps in this case Visual Studio.



## 20.4. Error in saving web form when there is an attachment

---

**Issue:** When we try to save a webform with attachment we have this error:

Level (0) Message: Transaction not successfully started

Level (0) StackTrace: at NHibernate.Transaction.AdoTransaction.CheckBegun() in p:\nhibernate-core\src\NHibernate\Transaction\AdoTransaction.cs:line 402 at NHibernate.Transaction.AdoTransaction.Rollback() in p:\nhibernate-core\src\NHibernate\Transaction\AdoTransaction.cs:line 241 at Rayvarz.BPMS.Framework.UnitOfWork.CommitTransaction(Boolean clearSession) in D:\Tfs\Main\Source\Rayvarz.BPMS.Framework\UnitOfWork.cs:line 236

Please note that we are using BPMS attachment feature and not separated DMS module.

### Solution

The reason of the error is that you haven't installed DMS (attachment module of BPMS) correctly.

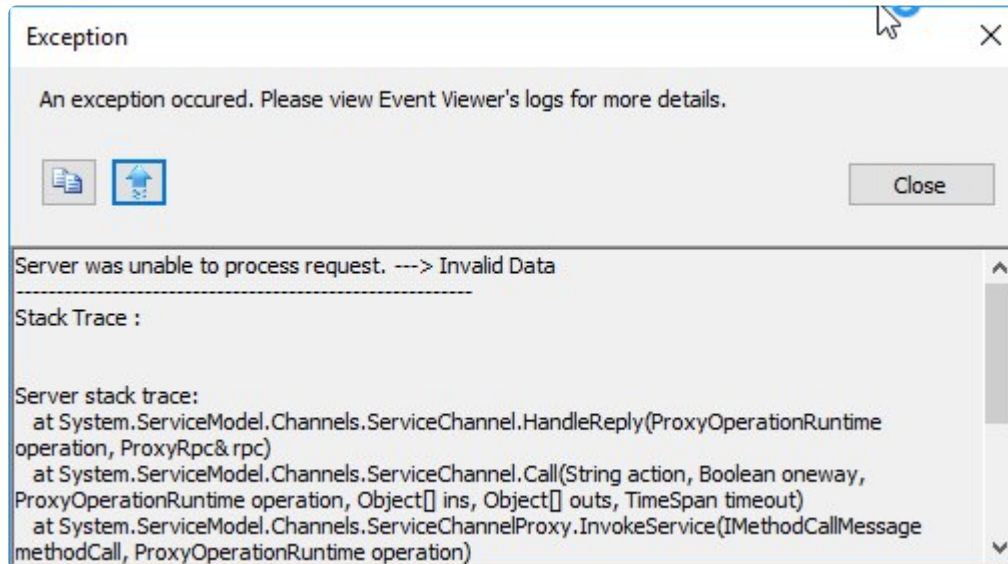
You need to check these:

1. Open the BPMS setup and make sure that DMS is installed.
2. Check RaySysSpc table.
  - a. DMSPath should have a value
  - b. UseDMS should be false
3. Check the Path which is the value of DMSPath in RaySysSpc on the server. You have to have a Root folder and a folder with GUID name inside the root folder. If you don't have that it means that DMS is not configured via setup. You need to use setup to reconfigure DMS.

## 20.5. “Invalid Data” Error on deploy any BPM Application

**Issue:** When you try to deploy any BPM Application via Admin Console you'll see this error:

Server was unable to process request. ---> Invalid Data



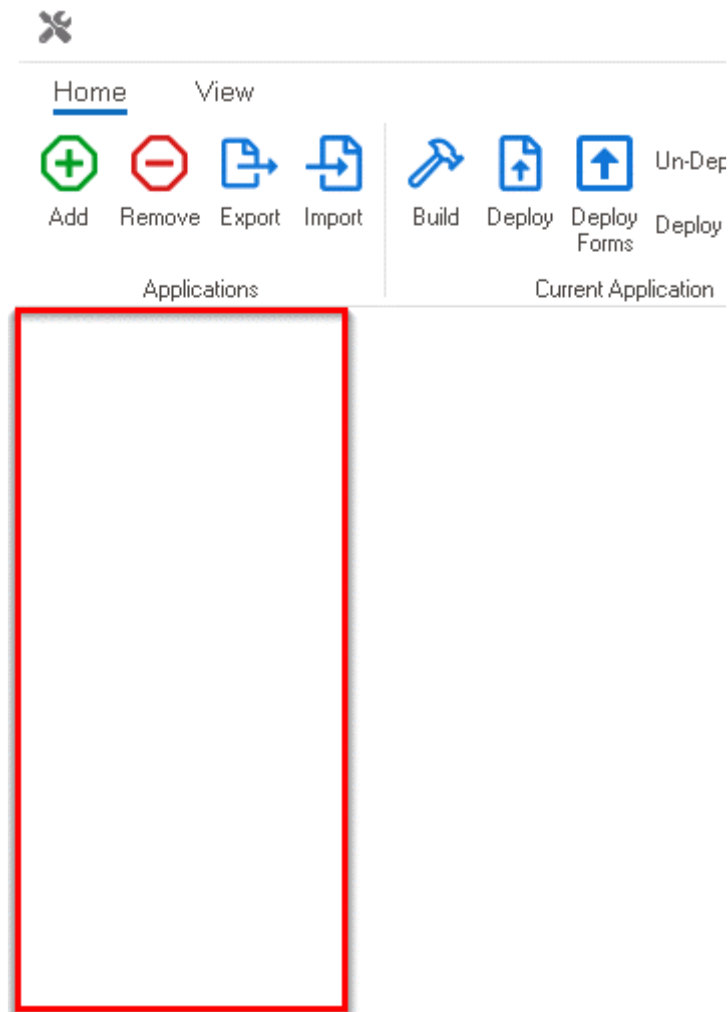
### Solution

The reason of this error is the time of the server (which runs the web application) and the client (which run the admin console) are not the same.

You need to sync the time in both design environment and server to resolve this issue.

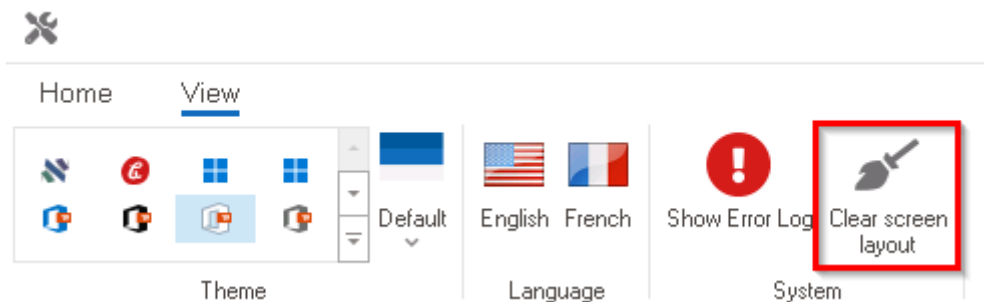
## 20.6. Application Tree Not Visible

**Issue:** The Application Tree is not visible when the Designer is opened.



**Solution:** To resolve this issue, follow these steps:

1. Open the View tab in the menu bar of the Designer.
2. Click on **Clean Screen Layout** option.



3. Restart the Designer.

After performing these steps, the Application Tree should be visible in the Designer.

## 20.7. Application Icons Not Visible in Web

**Issue:** After performing a 'clean' import and deploying the application, the icons are not visible on the **DBP** web interface, showing only blank spaces.

**Solution:** To resolve this issue, follow these steps:

1. Go to the [Home Setup](#) form of the **DBP** web application.
  - This form can be accessed from the **Facilities** menu.
2. Locate your menu item in the setup.
3. Verify that the image associated with the home item is visible.
  - Do not make any changes to the image.
4. Save the menu item.
5. Repeat this process for all items that are missing images.
6. Return to the Designer and redeploy the application.

By following these steps, the icons should now be visible on the web interface.



This issue should be completely resolved in version 14.1 of the Designer program.

## 20.8. Form Access Blocked

**Issue:** When clicking on a folder in the **DBP** web application, an error message appears stating “You don’t have permission to access the form.”

**Solution:** To resolve this issue, follow these steps:

1. Go to the [Home Setup](#) form of the **DBP** web application.
  - This form can be accessed from the **Facilities** menu.
2. Locate the menu item associated with the folder.
3. Verify that the **Is Container Menu** checkbox is selected

### Menu item settings

☒ Show tile on Home page

☐ Show in first level

Shown name

French name

☐ Is container menu

☒ Hide in top menu

Grid SaveAndNew Feature

4. Save the menu item.

By selecting the **Is Container Menu** checkbox, you should now have the necessary permissions to access the form within the folder.

## 20.9. Visual Studio – Error When Building Solution

---

**Issue:** When attempting to build the solution, an error message is encountered: “Could not load file or assembly ‘DevExpress.XtraEditors.vxx.x.dll’ or one of its dependencies...”

**Solution:** To resolve this issue, follow the steps below:

1. Install [DevExpress](#).
  - Download and install the DevExpress components required by the solution.
2. Ensure proper references.
  - Once DevExpress is installed, ensure that the project references are correctly pointing to the DevExpress assemblies. Open the solution in Visual Studio, right-click on the project, select **Manage NuGet Packages**, and verify that the DevExpress packages are installed and up-to-date. If any references are missing or outdated, update them accordingly.
3. Rebuild the solution.
  - After verifying the DevExpress installation and references, rebuild the solution. Right-click on the project and select **Rebuild** to compile the code. The error should no longer occur if the DevExpress components are properly installed and referenced.

By installing DevExpress and ensuring the correct references, you should be able to build the solution without encountering the mentioned error related to the ‘DevExpress.XtraEditors.vxx.x.dll’ assembly or its dependencies.