
GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

Visual SCIL Objects





Document ID: 1MRK 511 484-UEN
Issued: March 2021
Revision: A
Product version: 10.2

© 2021 Hitachi Power Grids. All rights reserved.

Table of contents

| | | |
|------------------|--------------------------------|-----------|
| Section 1 | Copyrights..... | 13 |
| Section 2 | Introduction..... | 15 |
| 2.1 | Use of symbols..... | 15 |
| 2.2 | Intended audience..... | 15 |
| 2.3 | Related documents..... | 15 |
| 2.4 | Document conventions..... | 15 |
| 2.5 | Document revisions..... | 16 |
| Section 3 | General properties..... | 17 |
| 3.1 | Overview..... | 17 |
| 3.1.1 | Predefined attributes..... | 17 |
| 3.1.2 | Predefined methods..... | 17 |
| 3.2 | Details..... | 17 |
| 3.2.1 | Predefined attributes..... | 17 |
| 3.2.2 | Predefined methods..... | 21 |
| Section 4 | VS_BOX..... | 25 |
| 4.1 | Overview..... | 25 |
| 4.1.1 | Action Attributes..... | 25 |
| 4.1.2 | Action Methods..... | 25 |
| 4.1.3 | Predefined Attributes..... | 25 |
| 4.1.4 | Predefined Methods..... | 26 |
| 4.2 | Details..... | 26 |
| 4.2.1 | Action Attributes..... | 26 |
| 4.2.2 | Action Methods..... | 27 |
| 4.2.3 | Predefined Attributes..... | 28 |
| 4.2.4 | Predefined Methods..... | 31 |
| Section 5 | VS_BUTTON..... | 33 |
| 5.1 | Overview..... | 33 |
| 5.1.1 | Action Attributes..... | 33 |
| 5.1.2 | Action Methods..... | 33 |
| 5.1.3 | Predefined Attributes..... | 33 |
| 5.1.4 | Predefined Methods..... | 34 |
| 5.2 | Details..... | 34 |
| 5.2.1 | Action Attributes..... | 34 |
| 5.2.2 | Action Methods..... | 35 |
| 5.2.3 | Predefined Attributes..... | 36 |
| 5.2.4 | Predefined Methods..... | 42 |
| Section 6 | VS_CHECK_BOX..... | 43 |
| 6.1 | Overview..... | 43 |

| | | |
|-------------------|------------------------------|-----------|
| 6.1.1 | Action Attributes..... | 43 |
| 6.1.2 | Action Methods..... | 43 |
| 6.1.3 | Predefined Attributes..... | 43 |
| 6.1.4 | Predefined Methods..... | 44 |
| 6.2 | Details..... | 44 |
| 6.2.1 | Action Attributes..... | 44 |
| 6.2.2 | Action Methods..... | 45 |
| 6.2.3 | Predefined Attributes..... | 46 |
| 6.2.4 | Predefined Methods..... | 49 |
| Section 7 | VS_COMBO..... | 51 |
| 7.1 | Overview..... | 51 |
| 7.1.1 | Action Attributes..... | 51 |
| 7.1.2 | Action Methods..... | 51 |
| 7.1.3 | Predefined Attributes..... | 51 |
| 7.1.4 | Predefined Methods..... | 52 |
| 7.2 | Details..... | 52 |
| 7.2.1 | Action Attributes..... | 52 |
| 7.2.2 | Action Methods..... | 53 |
| 7.2.3 | Predefined Attributes..... | 53 |
| 7.2.4 | Predefined Methods..... | 59 |
| Section 8 | VS_COMBO_POPDOWN..... | 63 |
| 8.1 | Overview..... | 63 |
| 8.1.1 | Action Attributes..... | 63 |
| 8.1.2 | Action Methods..... | 63 |
| 8.1.3 | Predefined Attributes..... | 63 |
| 8.1.4 | Predefined Methods..... | 64 |
| 8.2 | Details..... | 64 |
| 8.2.1 | Action Attributes..... | 64 |
| 8.2.2 | Action Methods..... | 65 |
| 8.2.3 | Predefined Attributes..... | 66 |
| 8.2.4 | Predefined Methods..... | 72 |
| Section 9 | VS_CONTAINER..... | 75 |
| 9.1 | Overview..... | 75 |
| 9.1.1 | Action Attributes..... | 75 |
| 9.1.2 | Action Methods..... | 75 |
| 9.1.3 | Predefined Attributes..... | 75 |
| 9.1.4 | Predefined Methods..... | 76 |
| 9.2 | Details..... | 76 |
| 9.2.1 | Action Attributes..... | 76 |
| 9.2.2 | Action Methods..... | 77 |
| 9.2.3 | Predefined Attributes..... | 78 |
| 9.2.4 | Predefined Methods..... | 84 |
| Section 10 | VS_DIALOG..... | 87 |

| | | |
|-------------------|-----------------------------|------------|
| 10.1 | Overview..... | 87 |
| 10.1.1 | Action Attributes..... | 87 |
| 10.1.2 | Action Methods..... | 87 |
| 10.1.3 | Predefined Attributes..... | 87 |
| 10.1.4 | Predefined Methods..... | 88 |
| 10.2 | Details..... | 89 |
| 10.2.1 | Action Attributes..... | 89 |
| 10.2.2 | Action Methods..... | 89 |
| 10.2.3 | Predefined Attributes..... | 90 |
| 10.2.4 | Predefined Methods..... | 96 |
| Section 11 | VS_FILE_CHOOSER..... | 103 |
| 11.1 | Overview..... | 103 |
| 11.1.1 | Action Attributes..... | 103 |
| 11.1.2 | Action Methods..... | 103 |
| 11.1.3 | Predefined Attributes..... | 103 |
| 11.1.4 | Predefined Methods..... | 104 |
| 11.2 | Details..... | 104 |
| 11.2.1 | Action Attributes..... | 104 |
| 11.2.2 | Action Methods..... | 105 |
| 11.2.3 | Predefined Attributes..... | 106 |
| 11.2.4 | Predefined Methods..... | 112 |
| Section 12 | VS_GRAPH..... | 119 |
| 12.1 | Overview..... | 119 |
| 12.1.1 | Action Attributes..... | 119 |
| 12.1.2 | Action Methods..... | 119 |
| 12.1.3 | Predefined Attributes..... | 119 |
| 12.1.4 | Predefined Methods..... | 120 |
| 12.2 | Details..... | 120 |
| 12.2.1 | Action Attributes..... | 120 |
| 12.2.2 | Action Methods..... | 121 |
| 12.2.3 | Predefined Attributes..... | 121 |
| 12.2.4 | Predefined Methods..... | 124 |
| Section 13 | VS_GRAPH_LEGEND..... | 133 |
| 13.1 | Overview..... | 133 |
| 13.1.1 | Action Attributes..... | 133 |
| 13.1.2 | Action Methods..... | 133 |
| 13.1.3 | Predefined Attributes..... | 133 |
| 13.1.4 | Predefined Methods..... | 134 |
| 13.2 | Details..... | 134 |
| 13.2.1 | Action Attributes..... | 134 |
| 13.2.2 | Action Methods..... | 134 |
| 13.2.3 | Predefined Attributes..... | 134 |
| 13.2.4 | Predefined Methods..... | 137 |

| | | |
|-------------------|-----------------------------|------------|
| Section 14 | VS_HEADER..... | 139 |
| 14.1 | Overview..... | 139 |
| 14.1.1 | Action Attributes..... | 139 |
| 14.1.2 | Action Methods..... | 139 |
| 14.1.3 | Predefined Attributes..... | 139 |
| 14.1.4 | Predefined Methods..... | 140 |
| 14.2 | Details..... | 140 |
| 14.2.1 | Action Attributes..... | 140 |
| 14.2.2 | Action Methods..... | 141 |
| 14.2.3 | Predefined Attributes..... | 142 |
| 14.2.4 | Predefined Methods..... | 146 |
| Section 15 | VS_HEADER_ITEM..... | 149 |
| 15.1 | Overview..... | 149 |
| 15.1.1 | Predefined Attributes..... | 149 |
| 15.2 | Details..... | 149 |
| 15.2.1 | Predefined Attributes..... | 149 |
| Section 16 | VS_ICON_VIEW..... | 153 |
| 16.1 | Overview..... | 153 |
| 16.1.1 | Action Attributes..... | 153 |
| 16.1.2 | Action Methods..... | 153 |
| 16.1.3 | Predefined Attributes..... | 153 |
| 16.1.4 | Predefined Methods..... | 154 |
| 16.2 | Details..... | 155 |
| 16.2.1 | Action Attributes..... | 155 |
| 16.2.2 | Action Methods..... | 155 |
| 16.2.3 | Predefined Attributes..... | 156 |
| 16.2.4 | Predefined Methods..... | 163 |
| Section 17 | VS_IMAGE..... | 167 |
| 17.1 | Overview..... | 167 |
| 17.1.1 | Predefined Attributes..... | 167 |
| 17.1.2 | Predefined Methods..... | 167 |
| 17.2 | Details..... | 167 |
| 17.2.1 | Predefined Attributes..... | 167 |
| 17.2.2 | Predefined Methods..... | 168 |
| Section 18 | VS_IMAGE_DOMAIN..... | 173 |
| 18.1 | Overview..... | 173 |
| 18.1.1 | Action Attributes..... | 173 |
| 18.1.2 | Action Methods..... | 173 |
| 18.1.3 | Predefined Attributes..... | 173 |
| 18.1.4 | Predefined Methods..... | 174 |
| 18.2 | Details..... | 175 |
| 18.2.1 | Action Attributes..... | 175 |
| 18.2.2 | Action Methods..... | 175 |

| | | |
|-------------------|-----------------------------|------------|
| 18.2.3 | Predefined Attributes..... | 176 |
| 18.2.4 | Predefined Methods..... | 183 |
| Section 19 | VS_IMAGE_EDITOR..... | 189 |
| 19.1 | Overview..... | 189 |
| 19.1.1 | Action Attributes..... | 189 |
| 19.1.2 | Action Methods..... | 189 |
| 19.1.3 | Predefined Attributes..... | 189 |
| 19.1.4 | Predefined Methods..... | 190 |
| 19.2 | Details..... | 190 |
| 19.2.1 | Action Attributes..... | 190 |
| 19.2.2 | Action Methods..... | 191 |
| 19.2.3 | Predefined Attributes..... | 191 |
| 19.2.4 | Predefined Methods..... | 198 |
| Section 20 | VS_INPUT_DIALOG..... | 203 |
| 20.1 | Overview..... | 203 |
| 20.1.1 | Action Attributes..... | 203 |
| 20.1.2 | Action Methods..... | 203 |
| 20.1.3 | Predefined Attributes..... | 203 |
| 20.1.4 | Predefined Methods..... | 204 |
| 20.2 | Details..... | 205 |
| 20.2.1 | Action Attributes..... | 205 |
| 20.2.2 | Action Methods..... | 205 |
| 20.2.3 | Predefined Attributes..... | 206 |
| 20.2.4 | Predefined Methods..... | 212 |
| Section 21 | VS_LABEL..... | 219 |
| 21.1 | Overview..... | 219 |
| 21.1.1 | Action Attributes..... | 219 |
| 21.1.2 | Action Methods..... | 219 |
| 21.1.3 | Predefined Attributes..... | 219 |
| 21.1.4 | Predefined Methods..... | 220 |
| 21.2 | Details..... | 220 |
| 21.2.1 | Action Attributes..... | 220 |
| 21.2.2 | Action Methods..... | 220 |
| 21.2.3 | Predefined Attributes..... | 220 |
| 21.2.4 | Predefined Methods..... | 222 |
| Section 22 | VS_LINE..... | 225 |
| 22.1 | Overview..... | 225 |
| 22.1.1 | Action Attributes..... | 225 |
| 22.1.2 | Action Methods..... | 225 |
| 22.1.3 | Predefined Attributes..... | 225 |
| 22.1.4 | Predefined Methods..... | 226 |
| 22.2 | Details..... | 226 |
| 22.2.1 | Action Attributes..... | 226 |

| | | |
|-------------------|----------------------------|------------|
| 22.2.2 | Action Methods..... | 226 |
| 22.2.3 | Predefined Attributes..... | 226 |
| 22.2.4 | Predefined Methods..... | 228 |
| Section 23 | VS_LIST..... | 229 |
| 23.1 | Overview..... | 229 |
| 23.1.1 | Action Attributes..... | 229 |
| 23.1.2 | Action Methods..... | 229 |
| 23.1.3 | Predefined Attributes..... | 229 |
| 23.1.4 | Predefined Methods..... | 230 |
| 23.2 | Details..... | 231 |
| 23.2.1 | Action Attributes..... | 231 |
| 23.2.2 | Action Methods..... | 232 |
| 23.2.3 | Predefined Attributes..... | 233 |
| 23.2.4 | Predefined Methods..... | 240 |
| Section 24 | VS_MAIN_DIALOG..... | 245 |
| 24.1 | Overview..... | 245 |
| 24.1.1 | Action Attributes..... | 245 |
| 24.1.2 | Action Methods..... | 245 |
| 24.1.3 | Predefined Attributes..... | 245 |
| 24.1.4 | Predefined Methods..... | 246 |
| 24.2 | Details..... | 247 |
| 24.2.1 | Action Attributes..... | 247 |
| 24.2.2 | Action Methods..... | 247 |
| 24.2.3 | Predefined Attributes..... | 248 |
| 24.2.4 | Predefined Methods..... | 255 |
| Section 25 | VS_MENU..... | 263 |
| 25.1 | Overview..... | 263 |
| 25.1.1 | Action Attributes..... | 263 |
| 25.1.2 | Action Methods..... | 263 |
| 25.1.3 | Predefined Attributes..... | 263 |
| 25.1.4 | Predefined Methods..... | 263 |
| 25.2 | Details..... | 264 |
| 25.2.1 | Action Attributes..... | 264 |
| 25.2.2 | Action Methods..... | 264 |
| 25.2.3 | Predefined Attributes..... | 264 |
| 25.2.4 | Predefined Methods..... | 265 |
| Section 26 | VS_MENU_ITEM..... | 269 |
| 26.1 | Overview..... | 269 |
| 26.1.1 | Action Attributes..... | 269 |
| 26.1.2 | Action Methods..... | 269 |
| 26.1.3 | Predefined Attributes..... | 269 |
| 26.1.4 | Predefined Methods..... | 270 |
| 26.2 | Details..... | 270 |

| | | |
|-------------------|------------------------------|------------|
| 26.2.1 | Action Attributes..... | 270 |
| 26.2.2 | Action Methods..... | 270 |
| 26.2.3 | Predefined Attributes..... | 270 |
| 26.2.4 | Predefined Methods..... | 272 |
| Section 27 | VS_MENUBAR..... | 273 |
| 27.1 | Overview..... | 273 |
| 27.1.1 | Action Attributes..... | 273 |
| 27.1.2 | Action Methods..... | 273 |
| 27.1.3 | Predefined Attributes..... | 273 |
| 27.1.4 | Predefined Methods..... | 274 |
| 27.2 | Details..... | 274 |
| 27.2.1 | Action Attributes..... | 274 |
| 27.2.2 | Action Methods..... | 274 |
| 27.2.3 | Predefined Attributes..... | 274 |
| 27.2.4 | Predefined Methods..... | 277 |
| Section 28 | VS_NOTEBOOK..... | 279 |
| 28.1 | Overview..... | 279 |
| 28.1.1 | Action Attributes..... | 279 |
| 28.1.2 | Action Methods..... | 279 |
| 28.1.3 | Predefined Attributes..... | 279 |
| 28.1.4 | Predefined Methods..... | 280 |
| 28.2 | Details..... | 280 |
| 28.2.1 | Action Attributes..... | 280 |
| 28.2.2 | Action Methods..... | 280 |
| 28.2.3 | Predefined Attributes..... | 281 |
| 28.2.4 | Predefined Methods..... | 284 |
| Section 29 | VS_NOTEBOOK_PAGE..... | 285 |
| 29.1 | Overview..... | 285 |
| 29.1.1 | Action Attributes..... | 285 |
| 29.1.2 | Action Methods..... | 285 |
| 29.1.3 | Predefined Attributes..... | 285 |
| 29.1.4 | Predefined Methods..... | 286 |
| 29.2 | Details..... | 286 |
| 29.2.1 | Action Attributes..... | 286 |
| 29.2.2 | Action Methods..... | 286 |
| 29.2.3 | Predefined Attributes..... | 287 |
| 29.2.4 | Predefined Methods..... | 291 |
| Section 30 | VS_NOTICE_DIALOG..... | 293 |
| 30.1 | Overview..... | 293 |
| 30.1.1 | Action Attributes..... | 293 |
| 30.1.2 | Action Methods..... | 293 |
| 30.1.3 | Predefined Attributes..... | 293 |
| 30.1.4 | Predefined Methods..... | 295 |

| | | |
|-------------------|----------------------------------|------------|
| 30.2 | Details..... | 295 |
| 30.2.1 | Action Attributes..... | 295 |
| 30.2.2 | Action Methods..... | 296 |
| 30.2.3 | Predefined Attributes..... | 298 |
| 30.2.4 | Predefined Methods..... | 307 |
| Section 31 | VS_NUMERIC_SPINNER..... | 313 |
| 31.1 | Overview..... | 313 |
| 31.1.1 | Action Attributes..... | 313 |
| 31.1.2 | Action Methods..... | 313 |
| 31.1.3 | Predefined Attributes..... | 313 |
| 31.1.4 | Predefined Methods..... | 314 |
| 31.2 | Details..... | 314 |
| 31.2.1 | Action Attributes..... | 314 |
| 31.2.2 | Action Methods..... | 315 |
| 31.2.3 | Predefined Attributes..... | 316 |
| 31.2.4 | Predefined Methods..... | 320 |
| Section 32 | VS_OPTION_BUTTON..... | 321 |
| 32.1 | Overview..... | 321 |
| 32.1.1 | Action Attributes..... | 321 |
| 32.1.2 | Action Methods..... | 321 |
| 32.1.3 | Predefined Attributes..... | 321 |
| 32.1.4 | Predefined Methods..... | 322 |
| 32.2 | Details..... | 322 |
| 32.2.1 | Action Attributes..... | 322 |
| 32.2.2 | Action Methods..... | 323 |
| 32.2.3 | Predefined Attributes..... | 324 |
| 32.2.4 | Predefined Methods..... | 327 |
| Section 33 | VS_PALETTE..... | 329 |
| 33.1 | Overview..... | 329 |
| 33.1.1 | Action Attributes..... | 329 |
| 33.1.2 | Action Methods..... | 329 |
| 33.1.3 | Predefined Attributes..... | 329 |
| 33.1.4 | Predefined Methods..... | 330 |
| 33.2 | Details..... | 330 |
| 33.2.1 | Action Attributes..... | 330 |
| 33.2.2 | Action Methods..... | 331 |
| 33.2.3 | Predefined Attributes..... | 332 |
| 33.2.4 | Predefined Methods..... | 338 |
| Section 34 | VS_PICTURE_CONTAINER..... | 341 |
| 34.1 | Overview..... | 341 |
| 34.1.1 | Action Attributes..... | 341 |
| 34.1.2 | Action Methods..... | 341 |
| 34.1.3 | Predefined Attributes..... | 341 |

| | | |
|-------------------|------------------------------------|------------|
| 34.1.4 | Predefined Methods..... | 342 |
| 34.2 | Details..... | 342 |
| 34.2.1 | Action Attributes..... | 342 |
| 34.2.2 | Action Methods..... | 343 |
| 34.2.3 | Predefined Attributes..... | 343 |
| 34.2.4 | Predefined Methods..... | 348 |
| Section 35 | VS_PRINT_OBJECT..... | 351 |
| 35.1 | Overview..... | 351 |
| 35.1.1 | Action Attributes..... | 351 |
| 35.1.2 | Action Methods..... | 351 |
| 35.1.3 | Predefined Attributes..... | 351 |
| 35.1.4 | Predefined Methods..... | 352 |
| 35.2 | Details..... | 352 |
| 35.2.1 | Action Attributes..... | 352 |
| 35.2.2 | Action Methods..... | 352 |
| 35.2.3 | Predefined Attributes..... | 353 |
| 35.2.4 | Predefined Methods..... | 355 |
| Section 36 | VS_PROGRAM_EDITOR..... | 361 |
| 36.1 | Overview..... | 361 |
| 36.1.1 | Action Attributes..... | 361 |
| 36.1.2 | Action Methods..... | 361 |
| 36.1.3 | Predefined Attributes..... | 362 |
| 36.1.4 | Predefined Methods..... | 362 |
| 36.2 | Details..... | 363 |
| 36.2.1 | Action Attributes..... | 363 |
| 36.2.2 | Action Methods..... | 364 |
| 36.2.3 | Predefined Attributes..... | 365 |
| 36.2.4 | Predefined Methods..... | 371 |
| Section 37 | VS_SCROLL_BAR..... | 377 |
| 37.1 | Overview..... | 377 |
| 37.1.1 | Action Attributes..... | 377 |
| 37.1.2 | Action Methods..... | 377 |
| 37.1.3 | Predefined Attributes..... | 377 |
| 37.1.4 | Predefined Methods..... | 378 |
| 37.2 | Details..... | 378 |
| 37.2.1 | Action Attributes..... | 378 |
| 37.2.2 | Action Methods..... | 378 |
| 37.2.3 | Predefined Attributes..... | 378 |
| 37.2.4 | Predefined Methods..... | 382 |
| Section 38 | VS_SEPARATOR_MENU_ITEM..... | 385 |
| 38.1 | Overview..... | 385 |
| 38.1.1 | Predefined Attributes..... | 385 |
| 38.1.2 | Predefined Methods..... | 385 |

| | | |
|-------------------|------------------------------|------------|
| 38.2 | Details..... | 385 |
| 38.2.1 | Predefined Attributes..... | 385 |
| 38.2.2 | Predefined Methods..... | 386 |
| Section 39 | VS_SLIDER..... | 387 |
| 39.1 | Overview..... | 387 |
| 39.1.1 | Action Attributes..... | 387 |
| 39.1.2 | Action Methods..... | 387 |
| 39.1.3 | Predefined Attributes..... | 387 |
| 39.1.4 | Predefined Methods..... | 388 |
| 39.2 | Details..... | 388 |
| 39.2.1 | Action Attributes..... | 388 |
| 39.2.2 | Action Methods..... | 388 |
| 39.2.3 | Predefined Attributes..... | 388 |
| 39.2.4 | Predefined Methods..... | 392 |
| Section 40 | VS_SPLIT_BAR..... | 395 |
| 40.1 | Overview..... | 395 |
| 40.1.1 | Action Attributes..... | 395 |
| 40.1.2 | Action Methods..... | 395 |
| 40.1.3 | Predefined Attributes..... | 395 |
| 40.1.4 | Predefined Methods..... | 396 |
| 40.2 | Details..... | 396 |
| 40.2.1 | Action Attributes..... | 396 |
| 40.2.2 | Action Methods..... | 396 |
| 40.2.3 | Predefined Attributes..... | 396 |
| 40.2.4 | Predefined Methods..... | 398 |
| Section 41 | VS_SUB_MENU_ITEM..... | 399 |
| 41.1 | Overview..... | 399 |
| 41.1.1 | Predefined Attributes..... | 399 |
| 41.1.2 | Predefined Methods..... | 399 |
| 41.2 | Details..... | 399 |
| 41.2.1 | Predefined Attributes..... | 399 |
| 41.2.2 | Predefined Methods..... | 401 |
| Section 42 | VS_TABLE..... | 403 |
| 42.1 | Overview..... | 403 |
| 42.1.1 | Action Attributes..... | 403 |
| 42.1.2 | Action Methods..... | 403 |
| 42.1.3 | Predefined Attributes..... | 404 |
| 42.1.4 | Predefined Methods..... | 405 |
| 42.2 | Details..... | 408 |
| 42.2.1 | Action Attributes..... | 408 |
| 42.2.2 | Action Methods..... | 409 |
| 42.2.3 | Predefined Attributes..... | 411 |
| 42.2.4 | Predefined Methods..... | 422 |

| | | |
|-------------------|---------------------------------|------------|
| Section 43 | VS_TEXT..... | 455 |
| 43.1 | Overview..... | 455 |
| 43.1.1 | Action Attributes..... | 455 |
| 43.1.2 | Action Methods..... | 455 |
| 43.1.3 | Predefined Attributes..... | 455 |
| 43.1.4 | Predefined Methods..... | 456 |
| 43.2 | Details..... | 457 |
| 43.2.1 | Action Attributes..... | 457 |
| 43.2.2 | Action Methods..... | 458 |
| 43.2.3 | Predefined Attributes..... | 460 |
| 43.2.4 | Predefined Methods..... | 467 |
| Section 44 | VS_TEXT_EDITOR..... | 471 |
| 44.1 | Overview..... | 471 |
| 44.1.1 | Action Attributes..... | 471 |
| 44.1.2 | Action Methods..... | 471 |
| 44.1.3 | Predefined Attributes..... | 471 |
| 44.1.4 | Predefined Methods..... | 472 |
| 44.2 | Details..... | 473 |
| 44.2.1 | Action Attributes..... | 473 |
| 44.2.2 | Action Methods..... | 474 |
| 44.2.3 | Predefined Attributes..... | 475 |
| 44.2.4 | Predefined Methods..... | 482 |
| Section 45 | VS_TEXT_SPINNER..... | 487 |
| 45.1 | Overview..... | 487 |
| 45.1.1 | Action Attributes..... | 487 |
| 45.1.2 | Action Methods..... | 487 |
| 45.1.3 | Predefined Attributes..... | 487 |
| 45.1.4 | Predefined Methods..... | 488 |
| 45.2 | Details..... | 488 |
| 45.2.1 | Action Attributes..... | 488 |
| 45.2.2 | Action Methods..... | 489 |
| 45.2.3 | Predefined Attributes..... | 490 |
| 45.2.4 | Predefined Methods..... | 494 |
| Section 46 | VS_TOGGLE_MENU_ITEM..... | 497 |
| 46.1 | Overview..... | 497 |
| 46.1.1 | Action Attributes..... | 497 |
| 46.1.2 | Action Methods..... | 497 |
| 46.1.3 | Predefined Attributes..... | 497 |
| 46.1.4 | Predefined Methods..... | 498 |
| 46.2 | Details..... | 498 |
| 46.2.1 | Action Attributes..... | 498 |
| 46.2.2 | Action Methods..... | 498 |
| 46.2.3 | Predefined Attributes..... | 498 |

| | | |
|-------------------|----------------------------|------------|
| 46.2.4 | Predefined Methods..... | 500 |
| Section 47 | VS_TREE..... | 501 |
| 47.1 | Overview..... | 501 |
| 47.1.1 | Action Attributes..... | 501 |
| 47.1.2 | Action Methods..... | 501 |
| 47.1.3 | Predefined Attributes..... | 501 |
| 47.1.4 | Predefined Methods..... | 502 |
| 47.2 | Details..... | 503 |
| 47.2.1 | Action Attributes..... | 503 |
| 47.2.2 | Action Methods..... | 504 |
| 47.2.3 | Predefined Attributes..... | 504 |
| 47.2.4 | Predefined Methods..... | 510 |
| Index..... | 515 | |

Section 1 Copyrights

The information in this document is subject to change without notice and should not be construed as a commitment by Hitachi Power Grids. Hitachi Power Grids assumes no responsibility for any errors that may appear in this document.

In no event shall Hitachi Power Grids be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall Hitachi Power Grids be liable for incidental or consequential damages arising from the use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from Hitachi Power Grids, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

© 2021 Hitachi Power Grids. All rights reserved.

Trademarks

ABB is a registered trademark of ABB Asea Brown Boveri Ltd. Manufactured by/for a Hitachi Power Grids company. All other brand or product names mentioned in this document may be trademarks or registered trademarks of their respective holders.

Guarantee

Please inquire about the terms of guarantee from your nearest Hitachi Power Grids representative.

Third Party Copyright Notices

List of Third Party Copyright notices are documented in "3rd party licenses.txt" and other locations mentioned in the file in SYS600 and DMS600 installation packages.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<https://www.openssl.org/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Section 2 Introduction

2.1 Use of symbols

This publication includes warning, caution and information symbols where appropriate to point out safety-related or other important information. It also includes tips to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:



Warning icon indicates the presence of a hazard which could result in personal injury.



Caution icon indicates important information or a warning related to the concept discussed in the text. It might indicate the presence of a hazard, which could result in corruption of software or damage to equipment/property.



Information icon alerts the reader to relevant factors and conditions.



Tip icon indicates advice on, for example, how to design a project or how to use a certain function.

Although warning hazards are related to personal injury, and caution hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all warnings and caution notices.

2.2 Intended audience

This manual is intended for installation personnel, administrators and skilled operators to support installation of the software.

2.3 Related documents

| Name of the Manual | Document ID |
|--|------------------|
| SYS600 10.2 Visual SCIL Application Design | 1MRK 511 483-UEN |

2.4 Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a dialog, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for file names.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. Although the Enter and Shift keys are not labeled they are written in capital letters, e.g. press ENTER.

- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key and so on.
- Press **CTRL+C** indicates that the user must hold down the CTRL key while pressing the C key (in this case, to copy a selected object).
- Press **ALT E C** indicates that the user presses and releases each key in sequence (in this case, to copy a selected object).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
 - The following convention is used for menu operations: **Menu Name/Menu Item/Cascaded Menu Item**. For example: select **File/Open/New Project**.
 - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, if the user enters a value that is out of range, the following message is displayed: **Entered value is not valid**.
The user may be told to enter the string MIF349 in a field. The string is shown as follows in the procedure: **MIF349**
- Variables are shown using lowercase letters: sequence name

2.5 Document revisions

| Revision | Version number | Date | History |
|----------|----------------|------------|---------------------------------|
| A | 10.2 | 31.03.2021 | New document for SYS600 10.2 |

Section 3 General properties

3.1 Overview

In this section, some general properties, such as attributes and methods, are described. The properties described here are not described again together with the object types only listed in the overview section if they exist.

3.1.1 Predefined attributes

| Name | Type |
|-------------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "BACKGROUND_HILITE_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "FILE_REVISION" | Text |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "MAX_HEIGHT" | Integer |
| "MAX_WIDTH" | Integer |
| "MIN_HEIGHT" | Integer |
| "MIN_WIDTH" | Integer |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "SCIL_GEOMETRY_SCIL_GEOMETRY" | List |
| "SOURCE_FILE_NAME" | Text |
| "VARIABLE_NAMES" | Text vector |

3.1.2 Predefined methods

| Name | Type |
|-----------------------|-----------------|
| "FLAG_FOR_EXECUTION" | No return value |
| "GET_HELP" | No return value |
| "QUEUE_FOR_EXECUTION" | No return value |
| "SET_MAX_SIZE" | No return value |
| "SET_MIN_SIZE" | No return value |

3.2 Details

3.2.1 Predefined attributes

_ATTRIBUTE_NAMES

Contains the names of all attributes known to the object.

| | |
|----------------|---|
| Description: | A list containing two text vector attributes, USER_DEFINED and PREDEFINED. The attribute USER_DEFINED is a text vector containing the names of user-defined attributes, i.e. the attributes created by .MODIFY command (or by .SET command in a picture container context). The attribute PREDEFINED is a text vector containing the names of the attributes defined by the object class. |
| Data type: | List |
| Default value: | Environment dependent |
| Access: | Read-only |
| Note: | The attributes are listed in no particular order. |

_BACKGROUND_COLOR

The background color of the object. This is the color that is used when drawing the background of the object. If the color is not specified, it is inherited from the parent object.

| | |
|----------------|---------------------------------|
| Description: | A color as represented in SCIL. |
| Data type: | Color |
| Default value: | Environment dependent |
| Access: | No limitations |

_BACKGROUND_HILITE_COLOR

The background highlight color of the object. This is the color that is used when drawing the highlight background of the object when the cursor is moved on the top of it.

| | |
|----------------|---------------------------------|
| Description: | A color as represented in SCIL. |
| Data type: | Color |
| Default value: | Environment dependent |
| Access: | No limitations |

_CHILD_OBJECTS

The names of the immediate children of the object listed in the order of creation.

| | |
|----------------|--|
| Description: | The names of the immediate children of the object. |
| Data type: | Text vector |
| Default value: | Environment dependent |
| Access: | Read-only |

_FILE_REVISION

The revision text of the file from which the object/picture was loaded.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Environment dependent |
| Access: | Read-only |

_FONT

The font used when displaying texts in an object. If the font is set on an object which contains other objects (dialogs, containers, etc.), the same font will be used in the contained object as well. If the font is specifically set on one of the contained objects, this will not change.

| | |
|----------------|-------------------------------|
| Description: | A font as represented in SCIL |
| Data type: | Font |
| Default value: | Environment dependent |
| Access: | No limitations |

_FOREGROUND_COLOR

The foreground color of the object. This is the color that is used on e.g. titles and other texts in an object. If the color is not specified it is inherited from the parent object.

| | |
|----------------|--------------------------------|
| Description: | A color as represented in SCIL |
| Data type: | Color |
| Default value: | Environment dependent |
| Access: | No limitations |

_GEOMETRY

The size and position of the object in VS coordinates. The position is always relative to the parent object. The position of dialogs is relative to the screen. The size of a dialog defines the content area of the dialog excluding borders, title bars, etc. In all other cases the size is defined by the bounding rectangle of the object including possible borders, scrollbars etc. This attribute cannot always be set freely due to restrictions based on possibly defined minimum or maximum sizes or automatic geometry management.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H, representing the position horizontally (X) and vertically (Y) as well as the width (W) and height (H), all according to the VS coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | No limitations |

_MAX_HEIGHT

Specifies the maximum height of the object. This restricts both user and programmatic manipulation of the object. The value can be set by means of the _SET_MAX_SIZE method.

| | |
|----------------|---------------------------------------|
| Description: | The maximum height in VS coordinates. |
| Data type: | Integer |
| Default value: | 32767 |
| Access: | Read-only |

_MAX_WIDTH

Specifies the maximum width of the object. This restricts both user and programmatic manipulation of the object. The value can be set by means of the _SET_MAX_SIZE method.

| | |
|----------------|--------------------------------------|
| Description: | The maximum width in VS coordinates. |
| Data type: | Integer |
| Default value: | 32767 |
| Access: | Read-only |

_MIN_HEIGHT

Specifies the minimum height of the object. This restricts both user and programmatic manipulation of the object. The value can be set by means of the _SET_MIN_SIZE method.

| | |
|----------------|---------------------------------------|
| Description: | The minimum height in VS coordinates. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_MIN_WIDTH

Specifies the minimum width of the object. This restricts both user and programmatic manipulation of the object. The value can be set by means of the _SET_MIN_SIZE method.

| | |
|----------------|--------------------------------------|
| Description: | The minimum width in VS coordinates. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_OBJECT_CLASS

Specifies the name of the class of the object. If the object is a window or picture function, the attribute has value "WINDOW" or "PICTURE_FUNCTION", respectively.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Object dependent |
| Access: | Read-only |

_OBJECT_NAME

Specifies the name of the object.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Object dependent |
| Access: | Read-only |

_SCIL_GEOMETRY

The size and position of the object in SCIL coordinates. The position is always relative to the parent object. The position of dialogs is relative to the screen. The size of a dialog defines the content area of the dialog excluding borders, title bars, etc. In all other cases the size is defined by the bounding rectangle of the object including possible borders, scrollbars etc. This attribute cannot always be set freely due to restrictions based on possibly defined minimum or maximum sizes or automatic geometry management.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H, representing the position horizontally (X) and vertically (Y) as well as the width (W) and height (H), all according to the SCIL coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | No limitations |

_SOURCE_FILE_NAME

The full operating system format name of the "vso" file where the object has been loaded from or the full name of the "pic" file where the window or picture function has been read from. Value is an empty string, if the object is created on-the-fly, i.e. using .CREATE, !WIN_NAME or !WIN_CREATE.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Environment dependent |
| Access: | Read-only |

_VARIABLE_NAMES

The names of SCIL variables defined in the SCIL context that the object belongs to.

| | |
|----------------|--|
| Description: | A text vector containing the variable names. |
| Data type: | Text vector |
| Default value: | Environment dependent |
| Access: | Read-only |
| Note: | The names are listed in no particular order. |

3.2.2 Predefined methods

_FLAG_FOR_EXECUTION

This method is used when the programmer wants to execute a SCIL program at a later point in time when the system is idle. An identifier for the flagged execution, as well as the program to be executed, are given as parameters in the method call. An empty text value is allowed for the identifier. There is no way to exactly know when in time the program is executed. However, a minimum time that has to expire before execution can be defined in seconds as an optional third parameter in the method call. If `_FLAG_FOR_EXECUTION` method of an object is called twice using the same name, the first flagged execution is cancelled if not yet done.

Below is an example that shows how to ensure that the blocking cursor of a dialog is switched off after a long program execution even if an error interrupting the normal program flow occurs.

```
...
my_dialog._flag_for_execution("unblock", ".set my_dialog._blocked =
false")
.set my_dialog._blocked = true
;long SCIL processing here
...

```

Return value data type: No return value

Parameter(s):

| | |
|---------|--|
| NAME | Text value used as the identifier of the flagged execution |
| PROGRAM | Text or Text vector, the program to be executed |
| DELAY | Real, the minimum delay in seconds |

_GET_HELP

Requests help for the object. Executing this method is equal to pressing the help button (F1 on MS-Windows look and feel) when the object is in focus.

Return value data type: No return value

_QUEUE_FOR_EXECUTION

This method is used when the programmer wants to execute a SCIL program at a later point in time when the system is idle. The program to be executed is given as a parameter in the method call. There is no way to exactly know when in time the program is executed. However, a minimum time that has to expire before execution can be defined in seconds as an optional second parameter in the method call. The maximum number of queued programs per object is 100. If this limit is exceeded, all queued executions are removed and a SCIL error is produced.

Below is an example that shows how to ensure that the blocking cursor of a dialog is switched off after a long program execution even if an error interrupting the normal program flow occurs.

```
...
my_dialog._queue_for_execution(".set my_dialog._blocked = false")
.set my_dialog._blocked = true
;long SCIL processing here

```

...

Return value data type: No return value

Parameter(s):

| | |
|---------|---|
| PROGRAM | Text or Text vector, the program to be executed |
| DELAY | Real, the minimum delay in seconds |

_SET_MAX_SIZE

Sets the maximum size of the object. This restricts both user and programmatic manipulation of the object size.

Return value data type: No return value

Parameter(s):

| | |
|--------|-----------------------------------|
| WIDTH | Integer, width in VS coordinates |
| HEIGHT | Integer, height in VS coordinates |

_SET_MIN_SIZE

Sets the minimum size of the object. This restricts both user and programmatic manipulation of the object size.

Return value data type: No return value

Parameter(s):

| | |
|--------|-----------------------------------|
| WIDTH | Integer, width in VS coordinates |
| HEIGHT | Integer, height in VS coordinates |

Section 4 VS_BOX

4.1 Overview

The VS_BOX item is a simple dialog item that does not contain any functionality. The object visualises a box that can be used to visually group related interface objects together. When a functional grouping of objects is desired, the container should be used instead of the box. The box can also be used to accommodate a VS_IMAGE. The appearance of the box depends on the current platform.

4.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

4.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

4.1.3 Predefined Attributes

| Name | Type |
|---------------------|-----------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_BORDER" | Boolean |
| "_HAS_IMAGE_MASK" | Boolean |
| "_IMAGE" | Image reference |
| "_OBJECT_CLASS" | Text |

Table continues on next page

| | |
|-----------------------------|-------------|
| " <u>OBJECT_NAME</u> " | Text |
| " <u>SCIL_GEOMETRY</u> " | List |
| " <u>SOURCE_FILE_NAME</u> " | Text |
| " <u>TITLE</u> " | Text |
| " <u>TOOLTIP_TEXT</u> " | Text |
| " <u>VARIABLE_NAMES</u> " | Text vector |
| " <u>VISIBLE</u> " | Boolean |

4.1.4 Predefined Methods

| Name | Type |
|--------------------------------|-----------------|
| " <u>FLAG_FOR_EXECUTION</u> " | No return value |
| " <u>GET_HELP</u> " | No return value |
| " <u>GET_STRING_WIDTH</u> " | Integer |
| " <u>QUEUE_FOR_EXECUTION</u> " | No return value |

4.2 Details

4.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

| | |
|------------|-------------|
| Data type: | Text vector |
|------------|-------------|

4.2.2 Action Methods

GET_TABLE_EDIT_TEXT

This action method is executed whenever the table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (i.e. just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

| | |
|---------------------------|------|
| Data type to be returned: | Text |
|---------------------------|------|

GET_TABLE_VIEW_TEXT

This action method is executed whenever the table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

| | |
|---------------------------|------|
| Data type to be returned: | Text |
|---------------------------|------|

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

| | |
|---------------------------|-----------------|
| Data type to be returned: | No return value |
|---------------------------|-----------------|

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object to the edit object value.

| | |
|---------------------------|-----------------|
| Data type to be returned: | No return value |
|---------------------------|-----------------|

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

| | |
|---------------------------|-----------------|
| Data type to be returned: | No return value |
|---------------------------|-----------------|

4.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FOREGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_BORDER

Determines whether the box item should be drawn with a border or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HAS_IMAGE_MASK

Specifies whether the image of the dialog item is drawn as a mask for the item foreground color or not. If this attribute is TRUE, the background of the dialog item is visible through the image. Note that not all dialog items can contain an image.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_IMAGE_MASK = TRUE
```

_IMAGE

Specifies an image that is associated with the object. The way that the image is displayed depends on the object. The image is given as an object reference to an image that has been loaded or created as an independent object.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...")
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of the dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is stopped.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if a dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

4.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---------|
| Return value data type: | Integer |
|-------------------------|---------|

Parameter(s)

| | |
|--------|---|
| STRING | Text |
| | A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 5 VS_BUTTON

5.1 Overview

A VS_BUTTON is a dialog item that can activate a command but have no value. In order to be useful, a button needs the action method NOTIFY that is executed when the user clicks the button. A button is activated when the user clicks the button. Buttons can also be activated by pressing certain keys in the keyboard that depend on the current look-and-feel. The button can be used to accommodate a VS_IMAGE.

5.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text Vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

5.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

5.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ALIGNMENT" | Text |
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |

Table continues on next page

| | |
|------------------------|-----------------|
| "_BACKGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_BORDER" | Boolean |
| "_HAS_FLAT_APPEARANCE" | Boolean |
| "_HAS_IMAGE_MASK" | Boolean |
| "_IMAGE" | Image reference |
| "_MNEMONIC" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_TITLE_VECTOR" | Text vector |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

5.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

5.2 Details

5.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

5.2.2 Action Methods

GET_TABLE_EDIT_TEXT

This action method is executed whenever the table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever the table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

The NOTIFY action method is executed when a button is clicked or otherwise activated.



When the button has a _POP_UP_MENU, the NOTIFY action method will not be called.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

5.2.3 Predefined Attributes

_ALIGNMENT

Specifies the multiline title alignment.

| | |
|----------------|---|
| Description: | A value of “LEFT” Or “RIGHT” Or “CENTER”. |
| Data type: | Text |
| Default value: | Center |
| Access: | No limitations |

Example:

```
.set ._ALIGNMENT="RIGHT"
```



_ALIGNMENT is applicable only for VS_BUTTON with multiline titles

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether the dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if the dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_BORDER

Determines whether the dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HAS_FLAT_APPEARANCE

Determines whether the VS_BUTTON should be drawn Flat or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_BUTTON._HAS_FLAT_APPEARANCE = TRUE
```



Flat appearance is applicable only for VS_BUTTON with image. The feature is implemented targeting Toolbar buttons. Flat Appearance is visible only for tools opened in Modern LAF.

_HAS_IMAGE_MASK

Specifies whether the image of the dialog item is drawn as a mask for the item foreground color or not. If this attribute is, the background of the dialog item is visible through the image. Note that not all dialog items can contain an image.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_IMAGE_MASK = TRUE
```

_IMAGE

Specifies an image that is associated with the object. The way that the image is displayed depends on the object. The image is given as an object reference to the image that has been loaded or created as an independent object.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...")
```

_MNEMONIC

Specifies the mnemonic character of the dialog item. The mnemonic may be visualized differently on different platforms.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._MNEMONIC = "A"
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies a pop-up menu that is associated with the object. This menu is activated by right-clicking the object.



When the _POP_UP_MENU is defined, the action method _NOTIFY won't be executed.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of the dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TITLE_VECTOR

Specifies a multiline title text of the button item.

| | |
|----------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Default value: | Empty vector |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE_VECTOR = vector("title line1","title line1","title line3")
```

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is brought over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if a dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

5.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

Return value data type: Integer

Parameter(s)

STRING

Text

A character string containing up to 255 characters.

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 6 VS_CHECK_BOX

6.1 Overview

A VS_CHECK_BOX object is an object that can be selected or cleared (deselected). The appearance of the object is dependent on the graphics platform. The attribute _VALUE contains the current state of the check box. The NOTIFY action method is called whenever the check box is clicked.

6.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

6.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

6.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |

Table continues on next page

| | |
|----------------------------|----------------|
| <u>"_MNEMONIC"</u> | Text |
| <u>"_OBJECT_CLASS"</u> | Text |
| <u>"_OBJECT_NAME"</u> | Text |
| <u>"_POP_UP_MENU"</u> | Menu reference |
| <u>"_SCIL_GEOMETRY"</u> | List |
| <u>"_SOURCE_FILE_NAME"</u> | Text |
| <u>"_TITLE"</u> | Text |
| <u>"_TOOLTIP_TEXT"</u> | Text |
| <u>"_VALUE"</u> | Integer |
| <u>"_VARIABLE_NAMES"</u> | Text vector |
| <u>"_VISIBLE"</u> | Boolean |

6.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| <u>"_FLAG_FOR_EXECUTION"</u> | No return value |
| <u>"_GET_HELP"</u> | No return value |
| <u>"_GET_STRING_WIDTH"</u> | Integer |
| <u>"_QUEUE_FOR_EXECUTION"</u> | No return value |

6.2 Details

6.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

6.2.2 Action Methods

GET_TABLE_EDIT_TEXT

This action method is executed whenever the table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever the table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of the dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and

COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

6.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|----------------|
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether the dialog item receives focus during normal keyboard traversal.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if the dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_MNEMONIC

Specifies the mnemonic character of the dialog item. The mnemonic may be visualized differently on different platforms.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._MNEMONIC = "A"
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with the object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of the dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is held over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE

Indicates the current value of the control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VALUE = 1
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

6.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

Return value data type: Integer

Parameter(s)

STRING Text

A character string containing up to 255 characters.

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 7 VS_COMBO

7.1 Overview

A VS_COMBO is a dialog item that enables the user to pick a selection from a list of choices. These choices can be defined by the application or entered by the user. A combination box has three associated behaviours: accept, select, and scroll situations. The behaviour of a combination box affects how the combination box maps user input (typing in the entry box or selecting from the choice box, for example) to a value in its list of values. Note that the default definition of these behaviours may vary according to look-and-feel. The indexing in combo box begins from 1.

7.1.1 Action Attributes

| Name | Type |
|--------------------|-------------|
| "ACCEPT" | Text vector |
| "SELECT" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

7.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "ACCEPT" | No return value |
| "SELECT" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

7.1.3 Predefined Attributes

| Name | Type |
|----------------------------|-------------|
| "ACCEPT_BEHAVIOR" | Text |
| "ATTRIBUTE_NAMES" | List |
| "AUTO_SORT_ENABLED" | Boolean |
| "BACKGROUND_COLOR" | Color |
| "CASE_SENSITIVITY_ENABLED" | Boolean |
| "CHILD_OBJECTS" | Text vector |
| "CLICK_FOCUSABLE" | Boolean |
| "ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FOCUSABLE" | Boolean |
| "FOCUSED" | Boolean |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "MODIFIABLE" | Boolean |

Table continues on next page

| | |
|---------------------|----------------|
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SCROLL_BEHAVIOR" | Text |
| "_SELECT_BEHAVIOR" | Text |
| "_SELECTED_INDEX" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VALUE_COUNT" | Integer |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |
| "_WRAP_ENABLED" | Boolean |

7.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_APPEND_TEXT" | No return value |
| "_DELETE_TEXT_AT" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_GET_TEXT_AT" | Text |
| "_MOVE_TEXT" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SELECT_TEXT_RANGE" | No return value |

7.2 Details

7.2.1 Action Attributes

ACCEPT

Setting this attribute specifies the action method ACCEPT.

Data type: Text vector

SELECT

Setting this attribute specifies the action method SELECT.

Data type: Text vector

TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

7.2.2 Action Methods

ACCEPT

The ACCEPT action method is executed whenever an accept situation has occurred.

Data type to be returned: No return value

SELECT

The SELECT action method is executed whenever a select situation has occurred.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

7.2.3 Predefined Attributes

_ACCEPT_BEHAVIOR

This attribute specifies how the combination box behaves in an accept situation. An accept situation is when the combination box makes the selected (or entered) value the current value. This occurs when the user presses Return or Enter, or when the focus leaves the combination box. Note the _ACCEPT_BEHAVIOR attribute concerns only to a modifiable combination box, also the action method ACCEPT is not executed if the combination box attribute _MODIFIABLE is set to FALSE and a selection is made from the combination box.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT. MATCH_OR_ADD: If any value is entered in the entry box and ENTER is pressed, the value will get added to the selection box. MATCH_OR_CLOSEST: If a value is entered in the entry box and ENTER is pressed, the closest value (Next number if it is numeric, closest string if it is alphabets) present is given. MATCH_OR_PARTIAL: If any value is entered in the entry box and ENTER is pressed, the partially matched value from the selection list is selected. MATCH_OR_FIRST: If any value is entered in the entry box and ENTER is pressed, the first value from the selection list is selected if no match is found. Otherwise, the match value is selected. MATCH_OR_NOTHING: If any value is entered in the entry box and ENTER is pressed but no match is found, no selection is made but the value will remain in the entry box (only). MATCH_OR_REVERT: If any value is entered in the entry box and ENTER is pressed, the previously selected value will be selected if no match is found. DEFAULT: If any value is entered in the entry box and ENTER is pressed, the partially matched value from the selection list is selected. |
| Data type: | Text |
| Default value: | "DEFAULT" |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SORT_ENABLED

Specifies whether the combo box list should be automatically sorted in alphabetical order.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO._AUTO_SORT_ENABLED = TRUE
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CASE_SENSITIVITY_ENABLED

If this attribute value is FALSE, the upper or lower case letters are ignored when comparing two values. If this value is TRUE, the case is significant.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether the dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSSED

Contains the value TRUE if the dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_MODIFIABLE

Specifies if the user is allowed to type into the entry box of the combo box. A change of appearance is also often combined with this attribute.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO._MODIFIABLE = FALSE
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with the object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SCROLL_BEHAVIOR

This attribute specifies how the combination box behaves in a scroll situation. A scroll situation is when the list in the choice box is scrolled. If the user types in the entry box, for every keystroke (except Return and ENTER), a scroll situation is triggered as the combination box tries to match a choice box value to the entry box value (depending on behavior).

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT. MATCH_OR_ADD: If any value is entered in the entry box and ENTER is pressed, it will scroll to that entry. MATCH_OR_CLOSEST: If value is entered in the entry box and ENTER is pressed, it will scroll to the closest value (next number if the value is numeric, closest string if the value is texts). MATCH_OR_PARTIAL: If any value is entered in the entry box and ENTER is pressed, the partially matched value from the selection list is scrolled to view. MATCH_OR_FIRST: If any value is entered in the entry box and ENTER is pressed, the first value from the selection list is scrolled to view if no match is found. MATCH_OR_NOTHING: If any value is entered in the entry box and ENTER is pressed, no scrolling is done if no match is found. MATCH_OR_REVERT: If any value is entered in the entry box and ENTER is pressed, the view scrolls to previously selected value if no match is found. DEFAULT: If any value is entered in the entry box and ENTER is pressed, the partially matched value from the selection list is scrolled to view. |
| Data type: | Text |
| Default value: | "DEFAULT" |
| Access: | No limitations |

_SELECT_BEHAVIOR

This attribute specifies how the combination box behaves in a select situation. A select situation is when the combination box tries to map user input to a value. This occurs when the

user selects a value from the choice box or enters text into the entry box. Every keystroke in the entry box (with the exception of Return and ENTER) triggers a select situation.

Description:

A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT.

MATCH_OR_ADD: If any value is entered in the entry box and ENTER is pressed, the new entry will be selected.

MATCH_OR_CLOSEST: If value is entered in the entry box and ENTER is pressed, the closest value will be selected (the next number if the value is numeric, the closest string if the value is text).

MATCH_OR_PARTIAL: If any value is entered in the entry box and ENTER is pressed, partially matched value is selected.

MATCH_OR_FIRST: If any value is entered in the entry box and ENTER is pressed, the first value from the selection list is selected if no match is found.

MATCH_OR_NOTHING: If any value is entered in the entry box and ENTER is pressed, the value is selected if no match is found.

MATCH_OR_REVERT: If any value is entered in the entry box and ENTER is pressed, the previously selected value will be selected if no match is found.

DEFAULT: If any value is entered in the entry box and ENTER is pressed, partially matched value is selected.

Data type:

Text

Default value:

"DEFAULT"

Access:

No limitations

_SELECTED_INDEX

Specifies the index of the selected value in the combo box.

Description:

An integer value.

Data type:

Integer

Default value:

0

Access:

No limitations

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT

Specifies the current text string in the input field of the VS_COMBO object.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

No limitations

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE_COUNT

The number of values in the list of values in the combo box.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if a dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_WRAP_ENABLED

Specifies whether the selection in the combo box list wraps around when the beginning or the end is reached using, for example, the arrow keys to traverse the list.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO._WRAP_ENABLED = FALSE
```

7.2.4 Predefined Methods

_APPEND_TEXT

Appends a text string to the list in the combo box. If the attribute _AUTO_SORT_ENABLED is set to TRUE, the list of texts is automatically resorted when a new text string is appended. Otherwise, the new text string is appended to the end of the list.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_DELETE_TEXT_AT

Deletes the text string at the specified index.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_GET_TEXT_AT

Returns the text string at the specified index.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_MOVE_TEXT

Moves a text string from an index to another in the combo box list of texts.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| FROM | Integer An integer value. |
| TO | Integer An integer value. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SELECT_TEXT_RANGE

Selects the range of text in the input field of a combo object or places the cursor in the text. When the start parameter is 1, the starting position for the selection is before the first character. When the start parameter is -1, the starting position for the selection is after the last character. Length parameter 0 indicates the placing of the cursor without selecting any characters. A positive length parameter defines the amount of selected characters to the right from starting point. A negative length parameter defines the amount of selected characters to the left from starting point. The length parameter is automatically truncated to the length of the text.

Return value data type: No return value

Parameter(s)

START Integer
An integer value.

LENGTH Integer
An integer value.

Section 8 VS_COMBO_POPDOWN

8.1 Overview

A VS_COMBO_POPDOWN box is a dialog item that enables the user to pick a selection from a list of choices. These choices can be defined by the application or entered by the user. A combination box has three associated behaviours: accept, select, and scroll situations. The defined behaviour of a combination box affects how the combination box maps user input (typing in the entry box or selecting from the choice box, for example) to a value in its list of values. Note that the default definition of these behaviours may vary according to look-and-feel. The indexing in combo popdown begins from 1.

8.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_ACCEPT" | Text vector |
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_SELECT" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

8.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "ACCEPT" | No return value |
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "SELECT" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

8.1.3 Predefined Attributes

| Name | Type |
|-----------------------------|-------------|
| "_ACCEPT_BEHAVIOR" | Text |
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_SORT_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CASE_SENSITIVITY_ENABLED" | Boolean |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |

Table continues on next page

| | |
|------------------------|----------------|
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_MODIFIABLE" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SCROLL_BEHAVIOR" | Text |
| "_SELECT_BEHAVIOR" | Text |
| "_SELECTED_INDEX" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VALUE_COUNT" | Integer |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |
| "_VISIBLE_VALUE_COUNT" | Integer |
| "_WRAP_ENABLED" | Boolean |

8.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_APPEND_TEXT" | No return value |
| "_DELETE_TEXT_AT" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_GET_TEXT_AT" | Text |
| "_MOVE_TEXT" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SELECT_TEXT_RANGE" | No return value |

8.2 Details

8.2.1 Action Attributes

_ACCEPT

Setting this attribute specifies the action method ACCEPT.

Data type: Text vector

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_SELECT

Setting this attribute specifies the action method SELECT.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

8.2.2 Action Methods

ACCEPT

The ACCEPT action method is executed whenever an accept situation has occurred.

Data type to be returned: No return value

GET_TABLE_EDIT_TEXT

This action method is executed whenever the table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever the table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

SELECT

The SELECT action method is executed whenever a select situation has occurred.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

8.2.3 Predefined Attributes

_ACCEPT_BEHAVIOR

This attribute specifies how the combination box behaves in an accept situation. An accept situation is when the combination box makes the selected (or entered) value the current value. This occurs when the user presses Return or Enter, or when the focus leaves the combination box.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT. |
| | MATCH_OR_ADD: If any value is entered in the entry box and ENTER is pressed, the value will get added to the selection box. |
| | MATCH_OR_CLOSEST: If value is entered in the entry box and ENTER is pressed, the closest value (next number if the value is numeric, closest string if the value is text) present is given. |
| | MATCH_OR_PARTIAL: If any value is entered in the entry box and ENTER is pressed, partially matched value from the selection list is selected. |
| | MATCH_OR_FIRST: If any value is entered in the entry box and ENTER is pressed, the first value from the selection list is selected if no match is found. Otherwise, the match value is selected. |
| | MATCH_OR_NOTHING: If any value is entered in the entry box and ENTER is pressed, no selection is made but the value will remain in the entry box (only) if no match is found. |
| | MATCH_OR_REVERT: If any value is entered in the entry box and ENTER is pressed, the previously selected value will be selected if no match is found. |
| | DEFAULT: If any value is entered in the entry box and ENTER is pressed, partially matched value from the selection list is selected. |
| Data type: | Text |
| Default value: | "DEFAULT" |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SORT_ENABLED

Specifies whether the combo box list should be automatically sorted in alphabetical order.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO_POPDOWN._AUTO_SORT_ENABLED = FALSE
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CASE_SENSITIVITY_ENABLED

If this attribute value is FALSE, the upper or lower case letters are ignored when comparing two values. If this value is TRUE, the case is significant.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether the dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if the dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_MODIFIABLE

Specifies if the user is allowed to type into the entry box of the combo box. A change of appearance is also often combined with this attribute.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO_POPDOWN._MODIFIABLE = FALSE
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with the object. This menu is activated by right-clicking the object.



When _MODIFIABLE==FALSE, the _POP_UP_MENU is disabled.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SCROLL_BEHAVIOR

This attribute specifies how the combination box behaves in a scroll situation. A scroll situation is when the list in the choice box is scrolled. If the user types in the entry box, for every keystroke (except Return and ENTER), a scroll situation is triggered as the combination box tries to match a choice box value to the entry box value (depending on behaviour).

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT. |
| Data type: | Text |
| Default value: | "DEFAULT" |
| Access: | No limitations |

_SELECT_BEHAVIOR

This attribute specifies how the combination box behaves in a select situation. A select situation is when the combination box tries to map user input to a value. This occurs when the user selects a value from the choice box or enters text into the entry box. Every keystroke in the entry box (with the exception of Return and ENTER) triggers a select situation.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: MATCH_OR_ADD, MATCH_OR_CLOSEST, MATCH_OR_PARTIAL, MATCH_OR_FIRST, MATCH_OR_NOTHING, MATCH_OR_REVERT and DEFAULT. |
| Data type: | Text |
| Default value: | "DEFAULT" |
| Access: | No limitations |

_SELECTED_INDEX

Specifies the index of the selected value in the combo popdown.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT

Specifies the current text string in the input field of the VS_COMBO_POPDOWN object.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE_COUNT

The number of values in the list of values in the combo box.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_VISIBLE_VALUE_COUNT

The number of simultaneously visible values in the list of the combo box.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 5 |
| Access: | No limitations |

Example:

```
.SET MY_COMBO_POPDOWN._VISIBLE_VALUE_COUNT = 4
```

_WRAP_ENABLED

Specifies whether the selection in the combo box list wraps around when the beginning or the end is reached using for example the arrow keys to traverse the list.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_COMBO_POPDOWN._WRAP_ENABLED = FALSE
```

8.2.4 Predefined Methods

_APPEND_TEXT

Appends a text string to the popdown list of the combo box. If the attribute _AUTO_SORT_ENABLED is set to TRUE, the list of texts is automatically resorted when the new text string is appended. Otherwise, the new text string is appended to the end of the list.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_DELETE_TEXT_AT

Deletes the text string at the specified index.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_GET_TEXT_AT

Returns the text string at the specified index.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_MOVE_TEXT

Moves a text string from an index to another in the combo box list of texts.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| FROM | Integer An integer value. |
| TO | Integer An integer value. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SELECT_TEXT_RANGE

Selects the range of text in the input field of a combo object or places the cursor in the text. When start parameter is 1, the starting position for the selection is before the first character. When start parameter is -1, the starting position for the selection is after the last character. Length parameter 0 indicates placing the cursor without selecting any characters. A positive length parameter defines the amount of selected characters to the right from starting point. A negative length parameter defines the amount of selected characters to the left from starting point. The length parameter is automatically truncated to the length of the text.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| START | Integer An integer value. |
| LENGTH | Integer An integer value. |

Section 9 VS_CONTAINER

9.1 Overview

A VS_CONTAINER is a dialog item that can contain other dialog items. The container itself must be placed within a VS_DIALOG or a VS_MAIN_DIALOG. Containers can also be placed in other containers to make up a hierarchy of containers as long as the top-level container is within a dialog. Usually the group of functionally related dialog items are placed inside the VS_CONTAINER.

9.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

9.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

9.1.3 Predefined Attributes

| Name | Type |
|-----------------------|-------------|
| "_ABANDON_ITEM" | Dialog Item |
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BOUNDS_GEOMETRY" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_CONFIRM_ITEM" | Dialog Item |
| "_CONTENT_GEOMETRY" | List |
| "_DEFAULT_FOCUS_ITEM" | Dialog Item |
| "_ENABLED" | Boolean |

Table continues on next page

| | |
|------------------------------------|-----------------|
| "_FILE_REVISION" | Text |
| "_FLAT_KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "_FOCUS_ITEM" | Dialog Item |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_IMAGE" | Image reference |
| "_ITEM_COUNT" | Integer |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

9.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_ITEM_INDEX" | Integer |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SET_ITEM_INDEX" | No return value |

9.2 Details

9.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

9.2.2 Action Methods**GET_TABLE_EDIT_TEXT**

This action method is executed whenever the table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever the table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of the dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

9.2.3 Predefined Attributes

_ABANDON_ITEM

Specifies which dialog item that is the abandon item. The NOTIFY action method of the abandon item is executed when the user presses the abandon key. The meaning of the abandon key depends on current look-and-feel. On Microsoft Windows, the abandon key is Escape (ESC).

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_BOUNDS_GEOMETRY

The BOUNDS_GEOMETRY is the entire virtual coordinate space of the container. Position (0,0) in the container coordinate system is at the lower left corner of the BOUNDS_GEOMETRY. The BOUNDS_GEOMETRY is interpreted in the same coordinate space as the GEOMETRY of the object. Each object in the container is placed relative to the lower left corner of the BOUNDS_GEOMETRY.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_CONFIRM_ITEM

Specifies which dialog item is the confirm item. The NOTIFY action method of the confirm item is executed when the user presses the confirm key. The meaning of the confirm key depends on current look-and-feel. On Microsoft Windows, the confirm key is Return (ENTER). The confirm item is usually visually indicated with a thicker border than other items.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

_CONTENT_GEOMETRY

The CONTENT_GEOMETRY is the portion of the coordinate space that is visible to the user at any given moment. If CONTENT_GEOMETRY is smaller than BOUNDS_GEOMETRY, scroll bars must be inserted for horizontal and vertical scrolling. The BOUNDS_GEOMETRY is clipped to CONTENT_GEOMETRY when the interior of the container is drawn.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | Read-only |

_DEFAULT_FOCUS_ITEM

Specified which dialog item should receive keyboard focus when the container receives keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._DEFAULT_FOCUS_ITEM = "MY_LIST"
```

_ENABLED

Specifies whether the dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FLAT_KEYBOARD_TRAVERSAL_ENABLED

Specifies that the container does not interfere with the traversal of the objects in the dialog (for example when using the TAB key to move from object to object). When _FLAT_KEYBOARD_TRAVERSAL_ENABLED is TRUE, the objects appear to be on the same level although they are within a container. The appearance of the container is also different depending on this attribute.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._FLAT_KEYBOARD_TRAVERSAL_ENABLED = TRUE
```

_FOCUS_ITEM

Changes the current keyboard focus of the dialog or container to the specified dialog item. The dialog item that has keyboard focus receives all normal keystrokes and certain function keys. If the object is open, the appearance may change to indicate which item has focus. For

some look-and-feels, the dashed line is drawn around the objects label. Note that the current focus item is set to the _FOCUS_ITEM when the dialog/container is opened, so any changes to a closed dialog/container may be lost when the dialog/container is opened.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._FOCUS_ITEM = "MY_OK_BUTTON"
```

_FOCUSABLE

Specifies whether the dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if the dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_BORDER

Determines whether the dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether the container has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._HORIZONTAL_SCROLL_BAR_ENABLED = TRUE
```

_IMAGE

Specifies an image that is associated with the object. The way that the image is displayed depends on the object. The image is given as an object reference to the image that has been loaded or created as an independent object.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...")
```

_ITEM_COUNT

Specifies the number of dialog items in the object. This attribute specifies the number of dialog items added directly to the object. Since certain dialog items may contain others (for example containers), the total number of dialog items in the object may be larger.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
@NUMBER_OF_DIALOG_ITEMS = MY_CONTAINER._ITEM_COUNT
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies a pop-up menu that is associated with the object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of the dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether the container has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._VERTICAL_SCROLL_BAR_ENABLED = TRUE
```

_VISIBLE

Specifies whether the dialog item is visible or not. Setting this attribute to FALSE may be useful if a dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

9.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_ITEM_INDEX

Returns the item index of the dialog item inside container. The item index specifies the tabbing and drawing order between dialog items inside a container.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| ITEM | <p>Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object.</p> |

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | <p>Text A character string containing up to 255 characters.</p> |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SET_ITEM_INDEX

Specifies the item index of the dialog item inside container. The item index specifies the tabbing and drawing order between dialog items inside a container.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ITEM | <p>Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object.</p> |
| INDEX | <p>Integer An integer value.</p> |

Section 10 VS_DIALOG

10.1 Overview

A VS_DIALOG is a top-level window that can contain dialog items. Dialogs can be created dynamically or by using the dialog editor. Dialog items are rectangular sub-regions of the dialog. Dialog items implement individual user interface objects, such as buttons, lists and other objects. The NOTIFY callback of the VS_DIALOG is run whenever a dialog is moved or resized.

10.1.1 Action Attributes

| Name | Type |
|----------------------|-------------|
| "_ARRANGE_NOTIFY" | Text vector |
| "_CLOSE_NOTIFY" | Text vector |
| "_GOT_FOCUS_NOTIFY" | Text vector |
| "_KEY_PRESSED" | Text vector |
| "_LOST_FOCUS_NOTIFY" | Text vector |
| "_NOTIFY" | Text vector |

10.1.2 Action Methods

| Name | Type |
|---------------------|-----------------|
| "ARRANGE_NOTIFY" | No return value |
| "CLOSE_NOTIFY" | Boolean |
| "GOT_FOCUS_NOTIFY" | No return value |
| "KEY_PRESSED" | Boolean |
| "LOST_FOCUS_NOTIFY" | No return value |
| "NOTIFY" | No return value |

10.1.3 Predefined Attributes

| Name | Type |
|----------------------------|-------------|
| "_ABANDON_ITEM" | Dialog Item |
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BACKGROUND_HILITE_COLOR" | Color |
| "_BUSY" | Boolean |
| "_CHILD_OBJECTS" | Text vector |
| "_CONFIRM_ITEM" | Dialog Item |
| "_DEFAULT_FOCUS_ITEM" | Dialog Item |
| "_FILE_REVISION" | Text |
| "_FOCUS_ITEM" | Dialog Item |

Table continues on next page

| | |
|-----------------------|------------------|
| "_FOCUSABLE" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_ICON" | Image reference |
| "_ICONIFIED" | Boolean |
| "_ITEM_COUNT" | Integer |
| "_LEADER_DIALOG" | Dialog reference |
| "_LEADER_DIALOG_ITEM" | Dialog item |
| "_MAX_HEIGHT" | Integer |
| "_MIN_WIDTH" | Integer |
| "_MAXIMIZED" | Boolean |
| "_MIN_HEIGHT" | Integer |
| "_MIN_WIDTH" | Integer |
| "_MODAL" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OPEN" | Boolean |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_STYLE" | Vector |
| "_TITLE" | Text |
| "_VARIABLE_NAMES" | Text vector |

10.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_ITEM_INDEX" | Integer |
| "_GET_POINTER_POS" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_PLACE" | No return value |
| "_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_SCIL_GEOMETRY" | No return value |
| "_PRINT" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_RAISE" | No return value |
| "_SET_ITEM_INDEX" | No return value |
| "_SET_MAX_SIZE" | No return value |
| "_SET_MIN_SIZE" | No return value |

10.2 Details

10.2.1 Action Attributes

_ARRANGE_NOTIFY

Setting this attribute specifies the action method ARRANGE_NOTIFY.

Data type: Text vector

_CLOSE_NOTIFY

Setting this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT.CLOSE_DIALOG", "")
```

_GOT_FOCUS_NOTIFY

Setting this attribute specifies the action method GOT_FOCUS_NOTIFY.

Data type: Text vector

_KEY_PRESSED

Setting this attribute specifies the action method KEY_PRESSED

Data type: Text vector

_LOST_FOCUS_NOTIFY

Setting this attribute specifies the action method LOST_FOCUS_NOTIFY.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

10.2.2 Action Methods

ARRANGE_NOTIFY

This action method is executed when the size of the dialog is changed with the mouse and button is released.

Data type to be returned: No return value

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user chose to close a dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed by the program. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

GOT_FOCUS_NOTIFY

This action method is executed immediately after the dialog has got input focus.

Data type to be returned: No return value

KEY_PRESSED

This action method is executed after a key or key combination is pressed in the dialog which has keyboard input focus. Four arguments are passed to the method: Key (text representation of the key), Alt_Enabled (boolean), Ctrl_Enabled (boolean), Shift_Enabled (boolean).



The following key combinations are system reserved and do not activate the action method: CTRL+ALT+DEL, ALT+TAB, ALT+ENTER, ALT+ESC, CTRL+ESC.

Data type to be returned: Boolean

LOST_FOCUS_NOTIFY

This action method is executed immediately after the dialog has lost input focus.

Data type to be returned: No return value

NOTIFY

Specifies the notification procedure of the object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

10.2.3 Predefined Attributes

_ABANDON_ITEM

Specifies which dialog item is the abandon item. The NOTIFY action method of the abandon item is executed when the user presses the abandon key. The meaning of the abandon key depends on current look-and-feel. On Microsoft Windows, the abandon key is Escape (ESC).

Description:

A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object.

Data type:

Dialog Item

Default value:

Empty text

Access:

No limitations

Example:

```
.SET MY_DIALOG._ABANDON_ITEM = "MY_CANCEL_BUTTON"
```

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BACKGROUND_HILITE_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CONFIRM_ITEM

Specifies which dialog item is the confirm item. The NOTIFY action method of the confirm item is executed when the user presses the confirm key. The meaning of the confirm key depends on current look-and-feel. On Microsoft Windows, the confirm key is Return (ENTER). The confirm item is usually visually indicated with a thicker border than other items.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._CONFIRM_ITEM = "MY_OK_BUTTON"
```

_DEFAULT_FOCUS_ITEM

Specifies which dialog item should receive keyboard focus when the dialog receives keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._DEFAULT_FOCUS_ITEM = "MY_TEXT"
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUS_ITEM

Specifies the dialog item that currently has keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._FOCUS_ITEM = "MY_TEXT"
```

_FOCUSABLE

Designates whether the window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system. This function only indicates that the window has use for key events. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_ICON

Specifies the icon used in the title of the window or when the window is iconified. Note that the _ICON attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether the window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ITEM_COUNT

Specifies the number of dialog items in the dialog. This attribute specifies the number of dialog items added directly to the dialog. Since certain dialog items may contain others (for example containers), the total number of dialog items in the dialog may be larger.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
@NUMBER_OF_DIALOG_ITEMS = MY_DIALOG._ITEM_COUNT
```

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side-effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialogs' reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like _LEADER_DIALOG attribute _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog holds the item specified. For more information about leader-follower dialog relationship, see description of the _LEADER_DIALOG attribute. Setting this attribute to empty text unsets the leader dialog for the given window.

This attribute can be used, for example, in a picture method, setting picture container as the leader dialog item. This is the only way to raise a new dialog from a picture context and set the

picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not the window should grab pointer and key input from other windows when it is opened. The effect when the _MODAL attribute is TRUE is that it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#)

_STYLE

The style flags determine what decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags has to always be set.
STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute, the window will contain the border along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows:

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags.

| | |
|----------------|--|
| Description: | Any combination of the following style flags given as a vector containing text strings. The flags are: STYLE_BACKGROUND - An opaque background STYLE_BORDER - A border around the content area STYLE_MOVABLE - A title bar or other draggable area STYLE_RESIZABLE - Resize handles or a grow box STYLE_ICONIFIABLE - An iconify/minimize box STYLE_MAXIMIZABLE - A maximize/zoom box STYLE_MAIN - The top-level, non-document window STYLE_DOCUMENT - Window contains a document STYLE_CONFIRM - Window is a confirmation dialog STYLE_UNCLOSABLE - No close box STYLE_PALETTE - Window is a Tool window |
| Data type: | Vector |
| Default value: | ("STYLE_BACKGROUND", "STYLE_BORDER", "STYLE_ICONIFIABLE", "STYLE_MAIN", "STYLE_MAXIMIZABLE", "STYLE_MOVABLE", "STYLE_RESIZABLE") |
| Access: | No limitations |

_TITLE

The title of the window. Note that the title may not be visible under certain circumstances, for example, if the window or dialog does not have a title bar.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

10.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_ITEM_INDEX

Returns the item index of the dialog item inside dialog. The item index specifies the tabbing and drawing order between dialog items inside dialog.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| ITEM | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current pointer position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions a dialogs relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog reference A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),
("PLACE_TOP", "PLACE_ALIGN"))

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER", ("PLACE_CENTER")))

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions dialogs relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM. Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100), ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself: @GEOM = YOUR_CONTAINER._GEOMETRY #MODIFY GEOM:V = LIST(X = 0, Y = 0)
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"), ("PLACE_CENTER"))

_PLACE_ACCORDING_TO_GEOMETRY

Places dialogs in relative to given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Placing MY_DIALOG at TOP/LEFT of RECT:
MY_DIALOG.PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))

PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions dialogs relative place compared to given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Placing MY_DIALOG at TOP/LEFT of RECT:
 MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOGRAPHY(LIST(X=200,Y=200,W=0,H=0),
 ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))

_PRINT

Prints the content of the dialog to the default printer. Note that any overlapping windows which are located inside the dialog are printed, too.

Return value data type: No return value

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_ITEM_INDEX

Specifies the item index of the dialog item inside dialog. The item index specifies the tabbing and drawing order between dialog items inside dialog.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ITEM | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| INDEX | Integer An integer value. |

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 11 VS_FILE_CHOOSER

11.1 Overview

A VS_FILE_CHOOSER is an object that lets the user search the file system for a specific file. It is used when the user specifies which file should be opened, or in which folder the file should be stored. When the action method APPLY_FILE is run, the path description to the selected file is passed in as an argument. The argument is read using the SCIL function ARGUMENT.

11.1.1 Action Attributes

| Name | Type |
|-----------------|-------------|
| "_APPLY_FILE" | Text vector |
| "_CLOSE_NOTIFY" | Text vector |
| "_HELP" | Text vector |
| "_NOTIFY" | Text vector |

11.1.2 Action Methods

| Name | Type |
|----------------|-----------------|
| "APPLY_FILE" | Boolean |
| "CLOSE_NOTIFY" | Boolean |
| "HELP" | Boolean |
| "NOTIFY" | No return value |

11.1.3 Predefined Attributes

| Name | Type |
|--------------------|------------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "BUSY" | Boolean |
| "CHILD_OBJECTS" | Text vector |
| "DIRECTORY" | Text |
| "DOCUMENT_NAME" | Text |
| "EXTENSION_FILTER" | Text |
| "FILE_REVISION" | Text |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "HAS_HELP" | Boolean |
| "ICON" | Image reference |
| "ICONIFIED" | Boolean |
| "LEADER_DIALOG" | Dialog reference |

Table continues on next page

| | |
|-----------------------|-------------|
| "_LEADER_DIALOG_ITEM" | Dialog item |
| "_MAX_HEIGHT" | Integer |
| "_MAX_HISTORY_COUNT" | Integer |
| "_MAX_WIDTH" | Integer |
| "_MAXIMIZED" | Boolean |
| "_MIN_HEIGHT" | Integer |
| "_MIN_WIDTH" | Integer |
| "_MODAL" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OPEN" | Boolean |
| "_OPTIONS" | Vector |
| "_PROMPT_TEXT" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_STYLE" | Vector |
| "_TITLE" | Text |
| "_TYPE" | Text |
| "_VARIABLE_NAMES" | Text vector |

11.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| "_ADD_FILTER_BY_REG_EXP" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_POINTER_POS" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_PLACE" | No return value |
| "_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_SCIL_GEOMETRY" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_RAISE" | No return value |
| "_SET_ACTIVE_FILTER" | No return value |
| "_SET_MAX_SIZE" | No return value |
| "_SET_MIN_SIZE" | No return value |

11.2 Details

11.2.1 Action Attributes

_APPLY_FILE

Setting this attribute specifies the action method APPLY_FILE.

Data type: Text vector

Example:

```
.SET MY_FILE_CHOOSER._APPLY_FILE = ("ROOT.SAVEFILE(ARGUMENT(1))", "..."
```

_CLOSE_NOTIFY

Setting this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT.CLOSE_DIALOG", "")
```

_HELP

Setting this attribute specifies the action method HELP.

Data type: Text vector

Example:

```
.SET MY_DIALOG._HELP = ("ROOT.HELP_DIALOG", "")
```

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

11.2.2 Action Methods

APPLY_FILE

This action method is executed when the user has chosen a file and pressed the **OK** or **Apply** button. The action method gets one input parameter that is the path, and a file that has been chosen in the file chooser. This action method should return TRUE if the file or directory is acceptable, in which case the file chooser is closed, or FALSE if the file or directory is not acceptable, in which case the file chooser is not closed.

Data type to be returned: Boolean

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user has chosen to close the dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed by the program. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

HELP

Specifies the action method that is executed when the **Help** button is pressed in the confirmation dialog.

Data type to be returned: Boolean

NOTIFY

Specifies the notification procedure of the object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

11.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_DIRECTORY

Specifies the folder used when opening the file chooser.

| | |
|----------------|---|
| Description: | A text string representing the physical path. |
| Data type: | Text |
| Default value: | Environment dependent |
| Access: | Write-only |

_DOCUMENT_NAME

Specifies the default file name when opening the file chooser.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | Write-only |

_EXTENSION_FILTER

Specifies an extension filter for the file chooser. The value of the attribute should be a three letter text specifying the filename extension.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_FILE_CHOOSER._EXTENSION_FILTER = "TXT"
```

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_HAS_HELP

Specifies whether the confirmation dialog has a **Help** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_HELP = TRUE
```

_ICON

Specifies the icon used in the title of the window or when the window is iconified. Note that the **_ICON** attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether the window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialogs receive the same message as a side-effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is automatically deleted. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders, if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like _LEADER_DIALOG attribute, the _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one that holds the item specified. For more information about leader-follower dialogs relationship, see the description of the _LEADER_DIALOG attribute above. Setting this attribute to empty text unsets the leader dialog for the given window.

This attribute can be used, for example, in a picture methods, setting picture container as the leader dialog item. This is only way to raise a new dialog from a picture context and set the picture holding dialog as leader, because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
;  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_HISTORY_COUNT

Specifies how many values are stored in the history list of the file chooser. Note that this attribute may be irrelevant if using default file choosers on some specific platform.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 5 |
| Access: | No limitations |

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not the window should grab cursor and key input from other windows when it is opened. When the _MODAL attribute is TRUE, it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_OPTIONS

Sets option flags that specifies how the file chooser should behave. The visual effect setting a particular option is dependent on whether the operating system supports the option. The options are:

SHOW_READONLY=Show the "Read Only" toggle

DEFAULT_READONLY=This option is not in use

CONFIRM_OVERWRITE=Confirm file over writes

FILE_MUST_EXIST=Chosen item must already exist

PATH_MUST_EXIST=The path leading to the item must exist

HIDE_HIDDEN=Hide hidden items

ONLY_DIRS=Want to select a directory

MULTIPLE_ABANDON=If multiple selection, abandon apply calls if any one fails

HIDE_PATH_ENTRY=This option is not in use

READONLY=This option is not in use

OVERWRITE=This option is not in use

DIR=This option is not in use

| | |
|----------------|---|
| Description: | Any combination of the following style flags given as a vector containing text strings. The flags are: SHOW_READONLY, CONFIRM_OVERWRITE, FILE_MUST_EXIST, PATH_MUST_EXIST, HIDE_HIDDEN, ONLY_DIRS, MULTIPLE_ABANDON, DEFAULT_READONLY, HIDE_PATH_ENTRY, READONLY, OVERWRITE and DIR. |
| Data type: | Vector |
| Default value: | Empty vector |
| Access: | No limitations |

_PROMPT_TEXT

Specifies the prompt text used in file chooser.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | Write-only |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

SOURCE_FILE_NAME

For more information, see [Section 3](#).

STYLE

The style flags determine what decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags has to be set always.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute, the window will contain the border along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags.

Description:

Any combination of the following style flags given as a vector containing text strings. The flags are:
STYLE_BACKGROUND - An opaque background
STYLE_BORDER - A border around the content area
STYLE_MOVABLE - A title bar or other draggable area
STYLE_RESIZABLE - Resize handles or a grow box
STYLE_ICONIFIABLE - An iconify/minimize box
STYLE_MAXIMIZABLE - A maximize/zoom box
STYLE_MAIN - The top-level, non-document window
STYLE_DOCUMENT - Window contains a document
STYLE_CONFIRM - Window is a confirmation dialog
STYLE_UNCLOSABLE - No close box
STYLE_PALETTE - Window is a Tool window

Data type:

Vector

Default value:

("STYLE_BACKGROUND", "STYLE_BORDER",
"STYLE_ICONIFIABLE", "STYLE_MAIN",
"STYLE_MAXIMIZABLE", "STYLE_MOVABLE",
"STYLE_RESIZABLE")

Access:

No limitations

TITLE

The title of the window. Note that the title may not be visible under certain circumstances, for example, if the window or dialog does not have a title bar.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_TYPE

Sets the type for the file chooser. The type determines the behavior and some default texts of the file chooser. A value of "PUT" means that the user wants to save (or put) a file somewhere, and a value of "GET" means that the user wants to open (or get) a file. Note that the _TYPE attribute has to be set before any other attribute, otherwise attributes set before _TYPE can be overwritten when _TYPE is changed. Such attributes are for example _TITLE and _PROMPT_NAME.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: GET and PUT. |
| Data type: | Text |
| Default value: | "GET" |
| Access: | Write-only |

_VARIABLE_NAMES

For more information, see [Section 3](#).

11.2.4 Predefined Methods

_ADD_FILTER_BY_REG_EXP

Adds an advanced filtering function to the file chooser. The filter is built using regular expressions. Note that regular expression filtering is not supported by all kinds of file choosers. The most useful filtering function is often filename extension filtering, which is available on all kinds of file choosers through the attribute _EXTENSION_FILTER.

Return value data type: No return value

Parameter(s)

FILTER_ID Text

A text string containing up to 255 characters.

TITLE Text

A character string containing up to 255 characters.

PATTERN Text

A character string containing up to 255 characters.

MAKE_DEFAULT Boolean

A value of TRUE or FALSE.

Example:

The regular expression to show only files that start with uppercase "A" and have the extension "VSO" files would be :

"^A.+\. [vV] [sS] [oO]\$" This filter will show all files that start with uppercase "A" and have the extension VSO regardless of upper or lower case letters in the extension.

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current cursor position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions a dialogs relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog reference A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),  
("PLACE_TOP",  
"PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialogs relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM. Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER, the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),
 ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY
#MODIFY GEOM:V = LIST(X = 0, Y = 0)
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),
 ("PLACE_CENTER"))
```

PLACE_ACCORDING_TO_GEOMETRY

Positions a dialogs relative place compared to given rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER, the side specification can be left out because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

RECT

List

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions a dialogs relative place compared to given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

```
1: Placing MY_DIALOG at TOP/LEFT of RECT:
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_ACTIVE_FILTER

Specifies the active filter used in file chooser. Passed argument is the filter id specified by the _ADD_FILTER_BY_REG_EXP method.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| FILTER_ID | Text A text string containing up to 255 characters. |

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 12 VS_GRAPH

12.1 Overview

The VS_GRAPH is an object that through a simple SCIL interface is capable of visually displaying numerical data. The data that is displayed in the graph is called datasets. A graph can display several datasets using the same or different graph types. The individual value in the dataset is called a point.

12.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_AREA_SELECTED" | Text vector |
| "_HAIRLINE_MOVED" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |

12.1.2 Action Methods

| Name | Type |
|-------------------------------------|-----------------|
| "AREA_SELECTED" | No return value |
| "HAIRLINE_MOVED" | No return value |
| "TOOLTIP_IS_SHOWN TOOLTIP_IS_SHOWN" | No return value |

12.1.3 Predefined Attributes

| Name | Type |
|-------------------------|----------------|
| "_AREA_SELECT_ENABLED" | Boolean |
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DATA_SET_COUNT" | Integer |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_ORIENTATION_INVERTED" | Boolean |
| "_POP_UP_MENU" | Menu reference |

Table continues on next page

| | |
|-----------------------------|-------------|
| " <u>SCIL_GEOMETRY</u> " | List |
| " <u>SOURCE_FILE_NAME</u> " | Text |
| " <u>TOOLTIP_TEXT</u> " | Text |
| " <u>VARIABLE_NAMES</u> " | Text vector |
| " <u>VISIBLE</u> " | Boolean |

12.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| " <u>ADD_DATA_SET</u> " | No return value |
| " <u>ADD_HAIRLINE</u> " | No return value |
| " <u>ADD_LIMIT_VALUE</u> " | No return value |
| " <u>ADD_X_AXIS</u> " | No return value |
| " <u>ADD_Y_AXIS</u> " | No return value |
| " <u>FLAG_FOR_EXECUTION</u> " | No return value |
| " <u>GET_DATA_SET_VALUES</u> " | List |
| " <u>GET_HAIRLINE_INTERSECTION_POINT</u> " | List |
| " <u>GET_HELP</u> " | No return value |
| " <u>GET_STRING_WIDTH</u> " | Integer |
| " <u>QUEUE_FOR_EXECUTION</u> " | No return value |
| " <u>SET_DATA_SET_PROPERTIES</u> " | No return value |
| " <u>SET_DATA_SET_VALUES</u> " | No return value |
| " <u>SET_HAIRLINE_PROPERTIES</u> " | No return value |
| " <u>SET_X_AXIS_PROPERTIES</u> " | No return value |
| " <u>SET_Y_AXIS_PROPERTIES</u> " | No return value |
| " <u>ZOOM</u> " | No return value |

12.2 Details

12.2.1 Action Attributes

_AREA_SELECTED

Setting this attribute specifies the action method AREA_SELECTED.

Data type: Text vector

_HAIRLINE_MOVED

Setting this attribute specifies the action method HAIRLINE_MOVED.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

12.2.2 Action Methods

AREA_SELECTED

This action method is executed when an area has been selected with the cursor and the _AREA_SELECT_ENABLED attribute is TRUE. Four arguments are passed to this action method specifying the selected range within the graph drawing area. The ranges are given as a percentage of the entire unzoomed graph drawing area. The arguments are real values specifying HORIZONTAL_LOW, HORIZONTAL_HIGH, VERTICAL_LOW and VERTICAL_HIGH.

Data type to be returned: No return value

HAIRLINE_MOVED

This action method is executed whenever a hairline has been moved. The id of the hairline is passed as argument 1 to this action method.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

12.2.3 Predefined Attributes

_AREA_SELECT_ENABLED

Specifies whether the user can interactively select an area using the cursor. When an area within the graph drawing area has been selected the action method AREA_SELECTED is executed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DATA_SET_COUNT

Indicates how many data sets are inserted into a graph.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | Read-only |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

Description: A value of TRUE or FALSE.

Data type: Boolean

Default value: TRUE

Access: No limitations

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_ORIENTATION_INVERTED

Specifies whether the sense of the X-axis and Y-axis are inverted.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_POP_UP_MENU

Specifies the pop-up menu that is associated with the object. This menu is activated by right-clicking the object.

Description:

A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD.

Data type:

Menu reference

Default value:

Empty text

Access:

No limitations

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_TOOLTIP_TEXT

Specifies a tooltip text that appears over the object when the cursor is held over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

Description:

A value of TRUE or FALSE.

Data type:

Boolean

Default value:

TRUE

Access:

No limitations

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

12.2.4 Predefined Methods

_ADD_DATA_SET

Adds a data set to the graph. Both X-axis and Y-axis have to be added before the data set because the data set has to be bound to the axis.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| X_AXIS_ID | Integer An integer value. |
| Y_AXIS_ID | Integer An integer value. |

_ADD_HAIRLINE

Adds a hairline to the graph. A hairline can be used to visually track or analyse the Y-value of a dataset.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |

_ADD_LIMIT_VALUE

Adds a line specifying some kind of limit value to the graph. The line only has a visual appearance and no other functionality.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| AXIS_ID | Integer An integer value. |
| Y_VALUE | Real A real value. |
| LINE_COLOR | Color A color as referenced within SCIL. |
| LINE_STYLE | Text A text string containing one of the following keywords: SOLID, ONOFFDASH or DOUBLEDASH. |
| LINE_WIDTH | Integer An integer value. |

_ADD_X_AXIS

Adds an X-axis to the graph.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| TYPE | Text A text string containing one of the following keywords: INTEGER, REAL, DISCRETE or TIME. |

Table continues on next page

| | |
|------------|--|
| PLACEMENT | Text A text string containing one of the following keywords: TOP, BOTTOM, LEFT , RIGHT or RELATED_TO_OTHER. |
| RELATED_TO | Integer An integer value. |
| Y_POSITION | SCIL Value A SCIL value of the type INTEGER, REAL or TIME. |

_ADD_Y_AXIS

Adds an Y-axis to the graph.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| TYPE | Text A text string containing one of the following keywords: INTEGER, REAL, DISCRETE or TIME. |
| PLACEMENT | Text A text string containing one of the following keywords: TOP, BOTTOM, LEFT , RIGHT or RELATED_TO_OTHER. |
| RELATED_TO | Integer An integer value. |
| X_POSITION | SCIL Value A SCIL value of the type INTEGER, REAL or TIME. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_DATA_SET_VALUES

Reads the values of a specific data set from the VS_GRAPH. The values are returned as a list containing the attributes X_VALUES and Y_VALUES which are vectors containing the values. The types of the vector elements are dependent of respecive axis type.

| | |
|-------------------------|------------------------------|
| Return value data type: | List |
| Parameter(s) | |
| ID | Integer An integer value. |

_GET_HAIRLINE_INTERSECTION_POINT

Returns the cursor to where the specified hairline intersects with the specified data set. The Y-value is interpolated if SNAP_TO_CLOSEST is FALSE. The `snap_to_closestxd5 feature works only if the hairline is within the bounds of the data set. If the hairline is outside the bounds of the data set, the values returned are of data type NONE.

| | |
|-------------------------|--------------------------------------|
| Return value data type: | List |
| Parameter(s) | |
| ID | Integer An integer value. |
| DATASET | Integer An integer value. |
| SNAP_TO_CLOSEST | Boolean A value of TRUE or FALSE. |

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SET_DATA_SET_PROPERTIES

Sets the properties of the specified data set. Default value handling is not yet implemented, and therefore all properties except status colors have to be given. The properties are given as a list and are the following:

GRAPH_TYPE=Text. PLOT, AREA, BAR, STACKED_BAR or PIE

TITLE=Text. The text used as title for the data set.

LINE_STYLE=Text. Specifies the line style of the dataset, SOLID, ONOFFDASH or DOBLEDASH.

LINE_WIDTH=Integer. Specifies the line width of the dataset.

DEFAULT_COLOR=Color. The color that is used for the data display if status coloring is not used.

SHOW_LINE=Boolean. Specifies whether the line in a line graph is visible or not.

MARKER_SHAPE=Text. DOT, BOX, TRIANGLE or DIAMOND.

SHOW_MARKER=Boolean. Specifies whether the data point marker is visible or not.

SHOW_MARKER=Boolean. Specifies whether the data point marker is visible or not.

MARKER_COLOR=Color. The color that is used for the marker display if show_markerxd5 is true and 'graph_typexd5' is PLOT.

STATUS_0_COLOR=Color. The color that is used for the data display for the specified status of the value.

STATUS_1_COLOR=Color. The color that is used for the data display for the specified status of the value.

...

STATUS_9_COLOR=Color. The color that is used for the data display for the specified status of the value.

Return value data type:

No return value

Parameter(s)

ID

Integer

An integer value.

PROPERTY_LIST

List

A list containing the properties to be set.

_SET_DATA_SET_VALUES

Sets the values of a specific data set from the VS_GRAPH. The values are given as vectors. The type of the vector element is dependent of respective axis type. The last parameter, APPEND, specifies whether the 'x_valuesxd5 and 'y_valuesxd5 are appended to the current data. The X values of the appended data have to be larger than the previously existing data.

Return value data type:

No return value

Parameter(s)

ID

Integer

An integer value.

X_VALUES

List

A list containing the vectors x_values and y_values.

Y_VALUES

List

A list containing the vectors x_values and y_values.

APPEND

Boolean

A value of TRUE or FALSE.

_SET_HAIRLINE_PROPERTIES

Sets the properties of the specified hairline. Default value handling is not yet implemented, and therefore all properties have to be given. The properties are given as a list and are the following:

COLOR=Color. The color of the hairline.

ENABLED= Boolean. Specifies whether the hairline is in use or not. If this attribute is TRUE the action method _HAIRLINE_MOVED is executed whenever the hairline is moved.

LINE_STYLE=Text. Specifies the line style of the hairline, SOLID, ONOFFDASH or DOUBLEDASH.

LINE_WIDTH=Integer. Specifies the line width of the hairline.

POSITION=Real. Specifies where the hairline is located in the graph. The position is given in percentage (0.0-100.0).

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| PROPERTY_LIST | List A list containing the properties to be set. |

_SET_X_AXIS_PROPERTIES

Sets the properties of the specified hairline. Default value handling is not yet implemented so all properties have to be given. The properties are given as a list and are the following:

MIN=The minimum value that is displayed on the axis. A single SCIL value of the data type defined by the 'typexd5 of the axis.

MAX=The maximum value that is displayed on the axis. A single SCIL value of the data type defined by the 'typexd5 of the axis.

TIME_UNIT=Text. "", "SECONDS", "MINUTES", "HOURS", "DAYS", "WEEKS", "MONTHS" or "YEARS". This parameter is ignored if the axis type is not "TIME".

TIME_FORMAT=Text. Format string for formatting date/time output. According to the format of the ANSI C function strftime.

ANNOTATION_INCREMENT=Real. Specifies the spacing between the annotation labels. If the axis type is not real the value is rounded to the nearest whole integer value.

ANNOTATION_FONT=Font. The font of the annotation text.

ANNOTATION_COLOR=Color. The color of the annotation text.

TICK_INCREMENT=Real. Specifies the spacing between the tick marks. If the axis type is not real the value is rounded to the nearest whole integer value.

AXIS_COLOR=Color. The color of the axis.

AXIS_TITLE=Text. The title of the axis.

AXIS_TITLE_FONT=Font. The font of the axis title.

AXIS_TITLE_COLOR=Color. A color as defined in SCIL.

AXIS_TITLE_PLACEMENT=LEFT or RIGHT

DISPLAY_AXIS_TITLE_ENABLED=Boolean. Specifies whether the axis title is displayed or not.

GRID_MODE=Text. NONE, ANNOTATION or TICK

GRID_LINE_WIDTH=Integer. The line width of the grid line.

GRID_COLOR=Color. Specifies the color of the grid lines.

GRID_LINE_STYLE=Text. SOLID, ONOFFDASH or DOUBLEDASH.

LINE_STYLE=Text. SOLID, ONOFFDASH or DOUBLEDASH.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| PROPERTY_LIST | List A list containing the properties to be set. |

_SET_Y_AXIS_PROPERTIES

Sets the properties of the specified axis. Default value handling is not yet implemented, and therefore all properties have to be given. The properties are given as a list and are the following:

MIN=The minimum value that is displayed on the axis. A single SCIL value of the data type defined by the 'typexd5 of the axis.

MAX=The maximum value that is displayed on the axis. A single SCIL value of the data type defined by the 'typexd5 of the axis.

TIME_UNIT=Text. Empty string. Not relevant for the y-axis.

TIME_FORMAT=Text. Empty string. Not relevant for the y-axis.

ANNOTATION_INCREMENT=Real. Specifies the spacing between the annotation labels. If the axis type is not real the value is rounded to the nearest whole integer value.

ANNOTATION_FONT=Font. The font of the annotation text.

ANNOTATION_COLOR=Color. The color of the annotation text.

TICK_INCREMENT=Real. Specifies the spacing between the tick marks. If the axis type is not real the value is rounded to the nearest whole integer value.

AXIS_COLOR=Color. The color of the axis.

AXIS_TITLE=Text. The title of the axis.

AXIS_TITLE_FONT=Font. The font of the axis title.

AXIS_TITLE_COLOR=Color. A color as defined in SCIL.

AXIS_TITLE_PLACEMENT=TOP or BOTTOM

DISPLAY_AXIS_TITLE_ENABLED=Boolean. Specifies whether the axis title is displayed or not.

GRID_MODE=Text. NONE, ANNOTATION or TICK

GRID_LINE_WIDTH=Integer. The line width of the grid line.

GRID_COLOR=Color. Specifies the color of the grid lines.

GRID_LINE_STYLE=Text. SOLID, ONOFFDASH or DOUBLEDASH.

LINE_STYLE=Text. SOLID, ONOFFDASH or DOUBLEDASH.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| PROPERTY_LIST | List A list containing the properties to be set. |

_ZOOM

Zooms the graph according to given parameters. The ranges are given as a percentage (0.0-100.0) of the total graph drawing area. When zooming a graph, the axis are also changed so that they are on their original places in the zoomed graph if they were placed on LEFT, RIGHT, TOP or BOTTOM. The ‘annotation_incrementxd5’ and ‘tick_incrementxd5’ for the entire axis are automatically adjusted so that about the same amount of annotations and ticks are visible in the zoomed graph. If an axis type is TIME, the ‘time_unitxd5’ may change if relevant.

| | |
|-------------------------|-----------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| HORIZONTAL_LOW | Real A real value. |
| HORIZONTAL_HIGH | Real A real value. |
| VERTICAL_LOW | Real A real value. |
| VERTICAL_HIGH | Real A real value. |

Section 13 VS_GRAPH_LEGEND

13.1 Overview

This object is used together with the VS_GRAPH object to display the legend for a VS_GRAPH object. A legend displays the title, line style and marker style of the different datasets in the graph.

13.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_TOOLTIP_IS_SHOWN" | Text vector |

13.1.2 Action Methods

| Name | Type |
|-------------------------------------|-----------------|
| "TOOLTIP_IS_SHOWN TOOLTIP_IS_SHOWN" | No return value |

13.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRAPH" | Dialog Item |
| "_HAS_BORDER" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

13.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

13.2 Details

13.2.1 Action Attributes

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

13.2.2 Action Methods

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

13.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FOREGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRAPH

Specifies which VS_GRAPH object a legend is connected to.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

13.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of a given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 14 VS_HEADER

14.1 Overview

The VS_HEADER object can be used as a header for any kind of column based data. The usage of this object is based on both the VS_HEADER and VS_HEADER_ITEM. A VS_HEADER is the main object that can contain any number of VS_HEADER_ITEMS. The user interacts with the header items and the programmer gets a notification of the actions through the action methods of VS_HEADER. The VS_HEADER can be inserted in a dialog using the Dialog Editor but the VS_HEADER_ITEM objects are inserted by the program.

Example: (the INIT method of a VS_HEADER object)

```
.CREATE ITEM1=VS_HEADER_ITEM(_WIDTH=50, _TITLE="Item 1")
.CREATE ITEM2=VS_HEADER_ITEM(_WIDTH=100, _TITLE="Item 2")
```

14.1.1 Action Attributes

| Name | Type |
|-----------------------|-------------|
| "_HEADER_RESIZED" | Text vector |
| "ITEM_DOUBLE_CLICKED" | Text vector |
| "ITEM_MOVED" | Text vector |
| "ITEM_RESIZED" | Text vector |
| "ITEM_SELECTED" | Text vector |
| "NOTIFY" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

14.1.2 Action Methods

| Name | Type |
|-------------------------------------|-----------------|
| "HEADER_RESIZED" | No return value |
| "ITEM_DOUBLE_CLICKED" | No return value |
| "ITEM_MOVED" | No return value |
| "ITEM_RESIZED" | No return value |
| "ITEM_SELECTED" | No return value |
| "TOOLTIP_IS_SHOWN TOOLTIP_IS_SHOWN" | No return value |

14.1.3 Predefined Attributes

| Name | Type |
|--------------------|-------------|
| "ARRANGE_ENABLED" | Boolean |
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |

Table continues on next page

| | |
|---------------------|-----------------|
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_BORDER" | Boolean |
| "_HAS_IMAGE_MASK" | Boolean |
| "_IMAGE" | Image reference |
| "_ITEM_COUNT" | Integer |
| "_ITEM_SPACING" | Integer |
| "_MNEMONIC" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SCROLL_OFFSET" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

14.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

14.2 Details

14.2.1 Action Attributes

_HEADER_RESIZED

Setting this attribute specifies the action method HEADER_RESIZED.

Data type: Text vector

_ITEM_DOUBLE_CLICKED

Setting this attribute specifies the action method ITEM_DOUBLE_CLICKED.

Data type: Text vector

_ITEM_MOVED

Setting this attribute specifies the action method ITEM_MOVED.

Data type: Text vector

_ITEM_RESIZED

Setting this attribute specifies the action method ITEM_RESIZED.

Data type: Text vector

_ITEM_SELECTED

Setting this attribute specifies the action method ITEM_SELECTED.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

14.2.2 Action Methods

HEADER_RESIZED

Executed whenever the size of the whole VS_HEADER has been changed.

Data type to be returned: No return value

ITEM_DOUBLE_CLICKED

Executed whenever an item has been double-clicked by the user. The position number of the VS_HEADER_ITEM that was double-clicked is passed as argument 1 to this action method.

Data type to be returned: No return value

ITEM_MOVED

Executed whenever the user has moved an object. Two arguments are passed to this action method, the position FROM which the item was moved and the position TO which it was moved.

Data type to be returned: No return value

ITEM_RESIZED

Executed whenever an item has been resized by the user. Two arguments are passed to this action method, the position of the item that was resized and the new width of the item.

Data type to be returned: No return value

ITEM_SELECTED

Executed whenever an item has been selected (single-clicked) by the user. The position number of the selected item is passed to this action method as argument 1.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over the dialog item.

Data type to be returned: No return value

14.2.3 Predefined Attributes

_ARRANGE_ENABLED

Specifies whether the user is allowed to change the order of the items in the header. If the user moves an item, the action method ITEM_MOVED is executed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FOREGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HAS_IMAGE_MASK

Specifies whether the image of a dialog item is drawn as a mask for the item foreground color or not. If this attribute is TRUE, the background of the dialog item is visible through the image. Note that not all dialog items can contain an image.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_IMAGE_MASK = TRUE
```

_IMAGE

Specifies an image that is associated with the object. The way that the image is displayed depends on the object. The image is given as an object reference to the image that has been loaded or created as an independent object.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...")
```

_ITEM_COUNT

Readable attribute that contains the number of items in the VS_HEADER.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_ITEM_SPACING

The spacing between the items in the header.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 2 |
| Access: | No limitations |

_MNEMONIC

Specifies the mnemonic character of the dialog item. The mnemonic may be visualized differently on different platforms.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._MNEMONIC = "A"
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SCROLL_OFFSET

The amount of times the items in the header has been scrolled. This is useful for example when the VS_HEADER has to follow another object that can be scrolled.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of the dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies the tooltip text that appears over the object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

14.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 15 VS_HEADER_ITEM

15.1 Overview

The VS_HEADER_ITEM object is used together with the VS_HEADER object. The usage of this object is based on both the VS_HEADER and VS_HEADER_ITEM. A VS_HEADER is the main object that can contain any number of VS_HEADER_ITEMS. The user interacts with the header items and the programmer receives a notification of the actions through the action methods of VS_HEADER. The VS_HEADER can be inserted in a dialog using the Dialog Editor but the VS_HEADER_ITEM objects are inserted by the program.

Example: (the INIT method of a VS_HEADER object)

```
.CREATE ITEM1=VS_HEADER_ITEM(_WIDTH=50, _TITLE="Item 1")
.CREATE ITEM2=VS_HEADER_ITEM(_WIDTH=100, _TITLE="Item 2")
```

15.1.1 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_MAX_WIDTH" | Integer |
| "_MIN_WIDTH" | Integer |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OFFSET" | Integer |
| "_RESIZE_ENABLED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_WIDTH" | Integer |

15.2 Details

15.2.1 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_MAX_WIDTH

Specifies the maximum width of an item. The user cannot enlarge the item beyond this size.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_MIN_WIDTH

Specifies the minimum width of an item. The user cannot make the item smaller than this size.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OFFSET

A read only attribute that indicates where the left side of an item is located in relation to the VS_HEADER.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | Read-only |

_RESIZE_ENABLED

Specifies whether the user can resize an item or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

The visible title of the item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_WIDTH

The current width of the item.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

Section 16 VS_ICON_VIEW

16.1 Overview

A VS_ICON_VIEW object can hold an arbitrary number of icons. The object has a predefined method _IS_ICON_SELECTED that can be used to check the state of the icons, for example, as a response to the DOUBLE_CLICK action method. VS_IMAGE objects can be attached to VS_ICON_VIEW.

16.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_DOUBLE_CLICK" | Text vector |
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

16.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "DOUBLE_CLICK" | No return value |
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

16.1.3 Predefined Attributes

| Name | Type |
|------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_SCROLL_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DRAG_OUTLINE" | Text |
| "_DRAG_POLICY" | Text |
| "_DRAG_PROHIBITED" | Boolean |
| "_DRAG_THRESHOLD" | Integer |

Table continues on next page

| | |
|---------------------------------|----------------|
| "_ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FOCUSABLE" | Boolean |
| "FOCUSED" | Boolean |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "GRABBING_HAND_ENABLED" | Boolean |
| "HAS_BORDER" | Boolean |
| "HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "HORIZONTAL_SCROLL_LOCKED" | Boolean |
| "ICON_COUNT" | Integer |
| "LIVE_SELECTION_ENABLED" | Boolean |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "POP_UP_MENU" | Menu reference |
| "SCIL_GEOMETRY" | List |
| "SELECT_ENCLOSURE_ENABLED" | Boolean |
| "SINGLE_SELECTION_ENABLED" | Boolean |
| "SNAP_TO_GRID_ENABLED" | Boolean |
| "SOURCE_FILE_NAME" | Text |
| "TOOLTIP_TEXT" | Text |
| "VARIABLE_NAMES" | Text vector |
| "VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "VERTICAL_SCROLL_LOCKED" | Boolean |
| "VIEW_BY" | Text |
| "VISIBLE" | Boolean |

16.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "APPEND_ICON" | Integer |
| "ARRANGE" | No return value |
| "FLAG_FOR_EXECUTION" | No return value |
| "GET_HELP" | No return value |
| "GET_ICON_NAME" | Text |
| "GET_STRING_WIDTH" | Integer |
| "IS_ICON_SELECTED" | Boolean |
| "QUEUE_FOR_EXECUTION" | No return value |
| "REMOVE_ICON" | No return value |
| "SET_ICON_IMAGE_LARGE" | No return value |
| "SET_ICON_IMAGE_SMALL" | No return value |
| "SET_ICON_NAME" | No return value |

16.2 Details

16.2.1 Action Attributes

_DOUBLE_CLICK

Setting this attribute specifies the action method DOUBLE_CLICK.

Data type: Text vector

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

16.2.2 Action Methods

DOUBLE_CLICK

The DOUBLE_CLICK program is run whenever the user double-clicks an item in the icon view. When a double-click occurs, the user can find out which icon was clicked with the method _IS_ICON_SELECTED.

Data type to be returned: No return value

GET_TABLE_EDIT_TEXT

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert an internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert the edit object value of the cell into the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object into the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of a cell to the current value of a view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

16.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SCROLL_ENABLED

Designates whether or not the view is automatically scrolled when an object is moved outside of the view area of the object.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_DOMAIN._AUTO_SCROLL_ENABLED = TRUE
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DRAG_OUTLINE

Specifies the outlining method when dragging objects within the view.

| | |
|--------------|--|
| Description: | A text string containing one of the following keywords: OUTLINE_ALL, OUTLINE_TWO, OUTLINE_LIMITRECT and OUTLINE_NONE. OUTLINE_ALL: When the icon is dragged, the entire icon will be outlined. OUTLINE_TWO: When the icon is dragged, the entire icon will be outlined by double lines. OUTLINE_LIMITRECT: When the icon is dragged, the entire icon will be outlined by rectangle. OUTLINE_NONE: When the icon is dragged, the icon will not be outlined. |
|--------------|--|



_Drag_Prohibited = FALSE should be set for this to work.

| | |
|----------------|----------------|
| Data type: | Text |
| Default value: | "OUTLINE_ALL" |
| Access: | No limitations |

_DRAG_POLICY

Specifies the dragging policy to be used when dragging objects.

| | |
|--------------|---|
| Description: | A text string containing one of the following keywords: DRAG_NORMAL, DRAG_AUTOSCROLL and EXTERNAL_DRAG. DRAG_NORMAL : Normal drag without any scroll effect. DRAG_AUTOSCROLL : While the Icon is dragged the view gets scrolled automatically. EXTERNAL_DRAG : Icon can be dragged to a wide area. |
|--------------|---|



_Drag_Prohibited = FALSE should be set for this to work.

| | |
|----------------|-------------------|
| Data type: | Text |
| Default value: | "DRAG_AUTOSCROLL" |
| Access: | No limitations |

_DRAG_PROHIBITED

Designates whether or not dragging is prohibited for all objects within the view.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_DRAG_THRESHOLD

Specifies the drag threshold to be used when dragging objects within the view. The drag threshold is the amount of movement that is required initially (in pixels) to begin dragging the object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Designates whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if the user has grabbed it, so that the scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HORIZONTAL_SCROLL_LOCKED

Specifies whether the horizontal scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ICON_COUNT

Specifies the number of icons in the icon view.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_LIVE_SELECTION_ENABLED

Specifies how selected objects in the view become highlighted. When this attribute is set to TRUE, live selection actively selects and unselects objects as tracking with the marquee occurs, rather than selecting after the mouse button has been released.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECT_ENCLOSED_ENABLED

Specifies how objects in the view become selected. Setting this attribute to TRUE means that the objects in the view are selected only when completely enclosed within the selection marquee. When attribute is set to FALSE, the objects become selected if any part of the object lies within the selection marquee.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SINGLE_SELECTION_ENABLED

Specifies whether single selection is used in domain view or not. When _SINGLE_SELECTION_ENABLED attribute is set to TRUE, only one image can be selected at a time.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SNAP_TO_GRID_ENABLED

Specifies whether Snap to grid is used or not. Snap to grid means that objects that are moved to a new location in the domain are aligned according to a grid.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies a tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VERTICAL_SCROLL_LOCKED

Specifies whether the vertical scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VIEW_BY

Specifies whether the icons in view are displayed as large or small icons. The large and small icons can be set using methods _SET_ICON_IMAGE_LARGE and _SET_ICON_IMAGE_SMALL. Note that if neither large nor small icon has not been defined, the default icon is used.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: BY_LARGE_ICON and BY_SMALL_ICON. |
| Data type: | Text |
| Default value: | "BY_LARGE_ICON" |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

16.2.4 Predefined Methods**_APPEND_ICON**

Appends an icon into the icon view. The number returned by this function must be used in all subsequent references to the icon. The image of the icon is set using methods _SET_ICON_IMAGE_LARGE and _SET_ICON_IMAGE_SMALL.

Return value data type: Integer

_ARRANGE

Arranges the icon view so that the icons are displayed as non-overlapping.

Return value data type: No return value

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_ICON_NAME

Returns the name of the specified icon.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_IS_ICON_SELECTED

The method returns TRUE if the specified icon is currently selected.

| | |
|-------------------------|------------------------------|
| Return value data type: | Boolean |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_REMOVE_ICON

Removes the specified icon from the icon view.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |

_SET_ICON_IMAGE_LARGE

Sets the large image of the specified icon. The image can be loaded from a resource file or created at runtime.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_SET_ICON_IMAGE_SMALL

Sets the small image of the specified icon. The image can be loaded from a resource file or created at runtime.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_SET_ICON_NAME

Sets the name of the specified icon.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ICON_ID | Integer An integer value. |
| NAME | Text A character string containing up to 255 characters. |

Section 17 VS_IMAGE

17.1 Overview

A VS_IMAGE is a bitmap image that can be created dynamically or loaded using the .LOAD command. The VS_IMAGE can be programmatically manipulated on pixel level, or by using the SCIL drawing functions. The VS_IMAGE also has methods for importing and exporting external bitmap files.

17.1.1 Predefined Attributes

| Name | Type |
|------------------|---------|
| "_COLOR_COUNT" | Integer |
| "_HEIGHT" | Integer |
| "_IS_TRUE_COLOR" | Boolean |
| "_WIDTH" | Integer |

17.1.2 Predefined Methods

| Name | Type |
|-------------------------|-----------------|
| "_EXPORT_DIB" | No return value |
| "_FILL_PIXMAP" | No return value |
| "_GET_PIXMAP_MASK_AT" | Integer |
| "_GET_PIXMAP_VALUE_AT" | Integer |
| "_IMPORT_DIB" | No return value |
| "_SET_PALETTE_COLOR_AT" | No return value |
| "_SET_PIXMAP_MASK_AT" | No return value |
| "_SET_PIXMAP_VALUE_AT" | No return value |
| "_SET_SIZE" | No return value |
| "_SET_SIZE_PRESERVE" | No return value |

17.2 Details

17.2.1 Predefined Attributes

_COLOR_COUNT

Specifies the number of colors used in the palette of a certain image.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 2 |
| Access: | No limitations |

_HEIGHT

Specifies the height of an image in pixels.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 10 |
| Access: | Read-only |

_IS_TRUE_COLOR

Specifies whether an image is true color. True color images do not have a color palette, but contain 24-bit RGB pixmap values between 1 and 16777216 (2^{24}). When _IS_TRUE_COLOR changes its value, all pixmap values are automatically converted.

The attribute _COLOR_COUNT cannot be changed if _IS_TRUE_COLOR is TRUE.



Large true color images require sufficient operating system memory resources to be edited and displayed. Below are listed the memory requirements for large true color images to be loaded:
640x480 - 921,600 bytes
800x600 - 1,440,000 bytes
960x720 - 2,073,600 bytes
1280x960 - 3,686,400 bytes

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_WIDTH

Specifies the width of an image in pixels.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 10 |
| Access: | Read-only |

17.2.2 Predefined Methods

_EXPORT_DIB

Exports a Windows bitmap (Device Independent Bitmap) to the file specified by PATH.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| PATH | Text A text string representing the physical path. |

_FILL_PIXMAP

Fills a pixmap (a two dimensional array of pixels) with the specified value.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| VALUE | <p>Integer An integer value representing a reference to a particular color in the pixmap color palette.</p> |

_GET_PIXMAP_MASK_AT

Returns the pixmap mask at specified location with a specified value.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| X | <p>Integer An integer value.</p> |
| Y | <p>Integer An integer value.</p> |
| VERT_HORIZ | <p>Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL.</p> |
| VALUE | <p>Integer An integer value representing a reference to a particular color in the pixmap color palette.</p> |

_GET_PIXMAP_VALUE_AT

Returns the pixmap (a two dimensional array of pixels) value in the specified location.

| | |
|-------------------------|--------------------------------------|
| Return value data type: | Integer |
| Parameter(s) | |
| X | <p>Integer An integer value.</p> |
| Y | <p>Integer An integer value.</p> |

_IMPORT_DIB

Imports a Windows bitmap (Device Independent Bitmap) from the a specified by PATH. The files generally have the extension .BMP, although some DIBs can be found with the extension .DIB. The import function only changes the image part, the programs, if any, are preserved.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| PATH | <p>Text A text string representing the physical path.</p> |

_SET_PALETTE_COLOR_AT

Specifies the used color in image palette with specified index.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |
| COLOR | Color A color as referenced within SCIL. |

_SET_PIXMAP_MASK_AT

Sets the current pixmap value in a location masked with specified mask according to the value.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| X | Integer An integer value. |
| Y | Integer An integer value. |
| VERT_HORIZ | Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL. |
| MASK | Integer An integer value. |
| VALUE | Integer An integer value representing a reference to a particular color in the pixmap color palette. |

_SET_PIXMAP_VALUE_AT

Sets the pixmap (a two dimensional array of pixels) location with a specified value.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| X | Integer An integer value. |
| Y | Integer An integer value. |
| VALUE | Integer An integer value representing a reference to a particular color in the pixmap color palette. |

_SET_SIZE

Sets the size of an image. The content of the image is not preserved.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| W | Integer An integer value. |
| H | Integer An integer value. |

_SET_SIZE_PRESERVE

Sets the size of an image. The content of the image is preserved according to the specified flags.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| W | Integer An integer value. |
| H | Integer An integer value. |
| PRESERVE | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PRESERVE_HORIZ_LEFT , PRESERVE_HORIZ_CENTER , PRESERVE_HORIZ_RIGHT , PRESERVE_VERT_TOP , PRESERVE_VERT_CENTER , PRESERVE_VERT_BOTTOM and PRESERVE_CLEAR_EXPOSED . PRESERVE_HORIZ_LEFT : Will preserve the Horizontal Left portion of image in the space provided for image, usually the bottom left corner portion. PRESERVE_HORIZ_CENTER : Will preserve the Horizontally center portion of image in the space provided for image, usually the bottom center portion. PRESERVE_HORIZ_RIGHT : Will preserve the Horizontally right portion of image in the space provided for image, usually the bottom right corner portion. PRESERVE_VERT_TOP : Will preserve the vertical Top portion of the image in the space provided for image, usually the left top corner portion. PRESERVE_VERT_CENTER : Will preserve the vertical center portion of the image in the space provided for image, usually the left center portion. PRESERVE_VERT_BOTTOM : Will preserver the vertical bottom portion of the image in the space provided for image, usually the left bottom portion PRESERVE_CLEAR_EXPOSED : Will not draw the image. |



The view mainly depend on the size of the image provided in this function, e.g :
`_set_size_preserve(w, h, preserve, value)` :
`root\myimage._set_size_preserve(40, 40, "PRESERVE_HORIZ_CENTER", 40)`
 Here w = 40,h = 40 and value = 40

| | |
|-------|--|
| VALUE | Integer An integer value representing a reference to a particular color in the image color palette. |
|-------|--|

Section 18 VS_IMAGE_DOMAIN

18.1 Overview

The VS_IMAGE_DOMAIN is an object that can contain an arbitrary number of different sized images. VS_IMAGE objects can be attached into VS_IMAGE_DOMAIN.

18.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_DOUBLE_CLICK" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |

18.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "DOUBLE_CLICK" | No return value |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

18.1.3 Predefined Attributes

| Name | Type |
|--------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_SCROLL_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_BOUNDS_GEOMETRY" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DRAG_OUTLINE" | Text |
| "_DRAG_POLICY" | Text |
| "_DRAG_PROHIBITED" | Boolean |
| "_DRAG_THRESHOLD" | Integer |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_GRID" | Integer |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_HORIZONTAL_SCROLL_LOCKED" | Boolean |
| "_LIVE_SELECTION_ENABLED" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCALE_X" | Integer |
| "_SCALE_Y" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SELECT_ENCLOSED_ENABLED" | Boolean |
| "_SELECTED_OBJECTS" | Integer |
| "_SHOW_IMAGE_BORDER_ENABLED" | Boolean |
| "_SINGLE_SELECTION_ENABLED" | Boolean |
| "_SNAP_TO_GRID_ENABLED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_GRID" | Integer |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VERTICAL_SCROLL_LOCKED" | Boolean |
| "_VISIBLE" | Boolean |

18.1.4 Predefined Methods

| Name | Type |
|-------------------------------------|-----------------|
| "_APPEND_IMAGE" | Integer |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_IMAGE_LABEL" | Text |
| "_GET_IMAGE_LABEL_HEIGHT" | Integer |
| "_GET_IMAGE_LABEL_WIDTH" | Integer |
| "_GET_IMAGE_X" | Integer |
| "_GET_IMAGE_Y" | Integer |
| "_GET_SELECTED_IMAGE" | Integer |
| "_GET_STRING_WIDTH" | Integer |
| "_INVALIDATE_IMAGE" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_REMOVE_IMAGE" | No return value |
| "_REPLACE_IMAGE" | No return value |
| "_SET_IMAGE_ANCHORED" | No return value |
| "_SET_IMAGE_LABEL" | No return value |
| "_SET_IMAGE_LABEL_BACKGROUND_COLOR" | No return value |

Table continues on next page

| | |
|--|-----------------|
| <u>"_SET_IMAGE_LABEL_FOREGROUND_COLOR"</u> | No return value |
| <u>"_SET_IMAGE_SELECTABLE"</u> | No return value |
| <u>"_SET_IMAGE_X"</u> | No return value |
| <u>"_SET_IMAGE_Y"</u> | No return value |
| <u>"_SET_SELECTED_IMAGE"</u> | No return value |

18.2 Details

18.2.1 Action Attributes

_DOUBLE_CLICK

Setting this attribute specifies the action method DOUBLE_CLICK.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

18.2.2 Action Methods

DOUBLE_CLICK

The DOUBLE_CLICK program is run whenever the user double-clicks an image in the image domain. When a double-click occurs, the user can find out which image was clicked with the method _GET_SELECTED_IMAGE.

Data type to be returned: No return value

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

18.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SCROLL_ENABLED

Designates whether or not the view is automatically scrolled when an object is moved outside of the view area of the object.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_DOMAIN._AUTO_SCROLL_ENABLED = TRUE
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_BOUNDS_GEOMETRY

The _BOUNDS_GEOMETRY is the entire virtual coordinate space of the image domain. Position (0,0) in the image domain coordinate system is at the lower left corner of the _BOUNDS_GEOMETRY. The _BOUNDS_GEOMETRY is interpreted in the same coordinate space as the _GEOMETRY of the object. Each object in the image domain is placed relative to the lower left corner of the _BOUNDS_GEOMETRY.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | (X=0,Y=0,W=100,H=100) |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3s](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DRAG_OUTLINE

Specifies the outlining method when dragging objects within the view.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: OUTLINE_ALL, OUTLINE_TWO, OUTLINE_LIMITRECT and OUTLINE_NONE. |
| Data type: | Text |
| Default value: | Object dependent |
| Access: | No limitations |

_DRAG_POLICY

Specifies the dragging policy to be used when dragging objects.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: DRAG_NORMAL, DRAG_AUTOSCROLL and EXTERNAL_DRAG. |
| Data type: | Text |
| Default value: | Object dependent |
| Access: | No limitations |

_DRAG_PROHIBITED

Specifies whether or not dragging is prohibited for all objects within the view.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_DRAG_THRESHOLD

Specifies the drag threshold to be used when dragging objects within the view. The drag threshold is the amount of movement that is required initially (in pixels) to begin dragging the object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Object dependent |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Specifies whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if the user grabbed it, so that scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_GRID

Specifies whether the horizontal grid is used in image domain or not. When this attribute is set to TRUE, the horizontal line is drawn in every row.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_HORIZONTAL_SCROLL_LOCKED

Specifies whether horizontal scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_LIVE_SELECTION_ENABLED

Specifies how selected objects in the view become highlighted. When this attribute is set to TRUE, live selection actively selects and unselects objects as tracking with the marquee occurs, rather than selecting after the mouse button has been released.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCALE_X

Specifies the width amount to scale the image in domain.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 100 |
| Access: | No limitations |

_SCALE_Y

Specifies the height amount to scale the image in domain.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 100 |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECT_ENCLOSED_ENABLED

Specifies how objects in the view become selected. Setting this attribute to TRUE means that objects in the view are selected only when completely enclosed within the selection marquee. When the attribute is set to FALSE, the objects become selected if any part of the object lies within the selection marquee.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_SELECTED_OBJECTS

Specifies the selected images in domain view. When the attribute _SINGLE_SELECTION_ENABLED is set to TRUE, only one image can be selected at a time.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Object dependent |
| Access: | Read-only |

_SHOW_IMAGE_BORDER_ENABLED

Specifies whether the image border is shown in the image domain.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SINGLE_SELECTION_ENABLED

Specifies whether single selection is used in domain view or not. When _SINGLE_SELECTION_ENABLED attribute is set to TRUE, only one image can be selected at a time.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_SNAP_TO_GRID_ENABLED

Specifies whether Snap to grid is used or not. Snap to grid means that objects that are moved to a new location in the domain are aligned according to a grid.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_GRID

Specifies whether a vertical grid is used in image domain or not. When this attribute is set to TRUE, the vertical line is drawn in every column.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_VERTICAL_SCROLL_LOCKED

Specifies whether vertical scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

18.2.4 Predefined Methods

_APPEND_IMAGE

Appends an image to the image domain.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| X | Integer An integer value. |
| Y | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_IMAGE_LABEL

Returns the label of a specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_GET_IMAGE_LABEL_HEIGHT

Returns the label height of a specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | Integer |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_GET_IMAGE_LABEL_WIDTH

Returns the label width of a specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | Integer |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_GET_IMAGE_X

Returns the x-coordinate of a specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | Integer |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_GET_IMAGE_Y

Returns the y-coordinate of a specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | Integer |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_GET_SELECTED_IMAGE

Returns the selected image identification.

Return value data type: Integer

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_INVALIDATE_IMAGE

Invalidate the specified image.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_REMOVE_IMAGE

Removes a specified image from the image domain.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

_REPLACE_IMAGE

Replaces a specified image with another image.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_SET_IMAGE_ANCHORED

Specifies whether an image is anchored or not. Anchored image can be selected but cannot be moved.

| | |
|-------------------------|--------------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |
| ANCORED | Boolean A value of TRUE or FALSE. |

_SET_IMAGE_LABEL

Specifies an image label.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

Table continues on next page

| | |
|------------------|--|
| LABEL | Text A character string containing up to 255 characters. |
| LABEL_POSITION_X | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. PLACE_NEAR: Completely outside with a small margin PLACE_ADJACENT: Completely outside with no margin PLACE_STRADDLE: Half inside, half outside PLACE_ALIGN: Completely inside with no margin PLACE_INSET: Completely inside with a small margin PLACE_HALF: Opposite side touches centre of base dialog PLACE_CENTER: Centered inside |
| LABEL_POSITION_Y | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. PLACE_NEAR: Completely outside with a small margin PLACE_ADJACENT: Completely outside with no margin PLACE_STRADDLE: Half inside, half outside PLACE_ALIGN: Completely inside with no margin PLACE_INSET: Completely inside with a small margin PLACE_HALF: Opposite side touches centre of base dialog PLACE_CENTER: Centered inside |

_SET_IMAGE_LABEL_BACKGROUND_COLOR

Specifies the background of an image label.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| COLOR | Color A color as referenced within SCIL. |

_SET_IMAGE_LABEL_FOREGROUND_COLOR

Specifies the foreground of an image label.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ID | Integer An integer value. |
| COLOR | Color A color as referenced within SCIL. |

_SET_IMAGE_SELECTABLE

Specifies whether an image can be selected or not.

| | |
|-------------------------|--------------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |
| SELECTABLE | Boolean A value of TRUE or FALSE. |

_SET_IMAGE_X

Specifies the X coordinate of an image.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |
| X | Integer An integer value. |

_SET_IMAGE_Y

Specifies the Y coordinate of an image.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |
| Y | Integer An integer value. |

_SET_SELECTED_IMAGE

Specifies an image as selected.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| IMAGE_ID | Integer An integer value. |

Section 19 VS_IMAGE_EDITOR

19.1 Overview

The VS_IMAGE_EDITOR is an editor with which the user can edit, import and export bitmap graphics. It is also possible to create transparent bitmaps.

19.1.1 Action Attributes

| Name | Type |
|-----------------|-------------|
| "_APPLY_IMAGE" | Text vector |
| "_CLOSE_NOTIFY" | Text vector |
| "_NOTIFY" | Text vector |

19.1.2 Action Methods

| Name | Type |
|----------------|-----------------|
| "APPLY_IMAGE" | No return value |
| "CLOSE_NOTIFY" | Boolean |
| "NOTIFY" | No return value |

19.1.3 Predefined Attributes

| Name | Type |
|-----------------------|------------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BIT_SIZE" | Integer |
| "_BUSY" | Boolean |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRID_ENABLED" | Boolean |
| "_HAS_SELECTION" | Boolean |
| "_ICON" | Image reference |
| "_ICONIFIED" | Boolean |
| "_IMAGE" | Image reference |
| "_IS_MODIFIED" | Boolean |
| "_LEADER_DIALOG" | Dialog reference |
| "_LEADER_DIALOG_ITEM" | Dialog item |

Table continues on next page

| | |
|----------------------------------|-------------------|
| " <u>MAX_HEIGHT</u> " | Integer |
| " <u>MAX_WIDTH</u> " | Integer |
| " <u>MAXIMIZED</u> " | Boolean |
| " <u>MIN_HEIGHT</u> " | Integer |
| " <u>MIN_WIDTH</u> " | Integer |
| " <u>OBJECT_CLASS</u> " | Text |
| " <u>OBJECT_NAME</u> " | Text |
| " <u>OPEN</u> " | Boolean |
| " <u>SCIL_GEOMETRY</u> " | List |
| " <u>SHOW_LOCATION_ENABLED</u> " | Boolean |
| " <u>SOURCE_FILE_NAME</u> " | Text |
| " <u>STYLE</u> " | Vector |
| " <u>TITLE</u> " | Text |
| " <u>USER_MENUBAR</u> " | Menubar reference |
| " <u>VARIABLE_NAMES</u> " | Text vector |

19.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| " <u>DROP_SELECTION</u> " | No return value |
| " <u>FLAG_FOR_EXECUTION</u> " | No return value |
| " <u>GET_HELP</u> " | No return value |
| " <u>GET_POINTER_POS</u> " | No return value |
| " <u>GET_STRING_WIDTH</u> " | Integer |
| " <u>PLACE</u> " | No return value |
| " <u>PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY</u> " | No return value |
| " <u>PLACE_ACCORDING_TO_GEOMETRY</u> " | No return value |
| " <u>PLACE_ACCORDING_TO_SCIL_GEOMETRY</u> " | No return value |
| " <u>QUEUE_FOR_EXECUTION</u> " | No return value |
| " <u>RAISE</u> " | No return value |
| " <u>SET_MAX_SIZE</u> " | No return value |
| " <u>SET_MIN_SIZE</u> " | No return value |

19.2 Details

19.2.1 Action Attributes

APPLY_IMAGE

Setting this attribute specifies the action method APPLY_IMAGE.

Data type: Text vector

CLOSE_NOTIFY

Setting this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT CLOSE_DIALOG", "")
```

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

19.2.2 Action Methods

APPLY_IMAGE

This action method is executed when the user closes the Image Editor using the menu **File > Close**.

Data type to be returned: No return value

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user closes a dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed by the program. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

NOTIFY

Specifies the notification procedure of the object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

19.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3s](#).

_BIT_SIZE

Specifies the enlargement scale when the Image Editor displays the image that is being edited. A value of 1 specifies 100%.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 5 |
| Access: | No limitations |

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3s](#).

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Designates whether a window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system, and this function only indicates that the window is useful for key events. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_GRID_ENABLED

Specifies whether the Image Editor shows a grid.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_SELECTION

Specifies whether an image has been selected in Image Editor or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Read-only |

_ICON

Specifies the icon used in the title of a window or when a window is iconified. Note that the _ICON attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether a window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_IMAGE

Specifies the image reference to the image being edited in Image Editor.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

_IS_MODIFIED

This read-only attribute shows whether the image was modified in the image editor or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Read-only |

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side-effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like the _LEADER_DIALOG attribute, the _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one that holds the item specified. Setting this attribute to empty text unsets the leader dialog for the given window. For more information about leader-follower dialogs relationship, see the description of the _LEADER_DIALOG attribute.

This attribute can be used, for example, in picture methods, setting picture container as the leader dialog item. This is the only way to raise a new dialog from a picture context and set the picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
;  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the entire display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SHOW_LOCATION_ENABLED

Specifies whether the Image Editor continuously displays the position of the cursor when editing an image.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_STYLE

The style flags determine which decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags has to always be set.
STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute the window will contain a border, along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows:

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent. Not all platforms can distinguish all combinations of style flags.

Description:

Any combination of the following style flags given as a vector containing text strings. The flags are:
STYLE_BACKGROUND - An opaque background
STYLE_BORDER - A border around the content area
STYLE_MOVABLE - A title bar or other draggable area
STYLE_RESIZABLE - Resize handles or a grow box
STYLE_ICONIFIABLE - An iconify/minimize box
STYLE_MAXIMIZABLE - A maximize/zoom box
STYLE_MAIN - The top-level, non-document window
STYLE_DOCUMENT - Window contains a document
STYLE_CONFIRM - Window is a confirmation dialog
STYLE_UNCLOSABLE - No close box
STYLE_PALETTE - Window is a Tool window

Data type:

Vector

Default value:

```
("STYLE_BACKGROUND", "STYLE_BORDER",
"STYLE_ICONIFIABLE", "STYLE_MAIN",
"STYLE_MAXIMIZABLE", "STYLE_MOVABLE",
"STYLE_RESIZABLE")
```

Access:

No limitations

_TITLE

The title of a window. Note that the title may not be visible under certain circumstances, for example if the window or dialog does not have a title bar.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

No limitations

_USER_MENU BAR

Setting the attribute _USER_MENU BAR to have a text reference to a VS_MENU BAR replaces the original menubar of the VS_IMAGE_EDITOR. To gain control over the image editor the menu items need to conform to the command names of the image editor. The commands are as follows:

| | |
|--------------------|--|
| ApplyEditor | Commits the changes made in an image editor to the original image and calls the image editor's apply function. |
| Clear | Clears the current selection in an image editor. |
| CloseEditor | Closes an image editor dialog. |
| Copy | Copies the current selection in an image editor to the scrap. |
| CreateTransparency | Creates a transparency for the image being edited in an image editor. |
| CropSelection | Crops the image being edited to the current selection in an image editor. |
| Cut | Copies the current selection in an image editor to the scrap and clears the current selection. |
| DeleteTransparency | Deletes the transparency associated with the image being edited by an image editor. |
| DrawCenter | Toggles the draw centre option of an image editor. |
| EditPalette | Opens the color palette editor to edit the color palette associated with the image being edited. |
| Export | Opens the export file chooser to allow the user to export the image being edited in an image editor. |
| FatBits | Toggles the magnification of bits in an image editor. |
| FlipHor | Flips the selected bits in an image editor horizontally. |
| FlipVer | Flips the selected bits in an image editor vertically. |
| Import | Opens the import file chooser to allow the user to import an image to edit in an image editor. |
| MaskSelection | Creates an initial transparency mask reflecting the current selection in an image editor. |
| Paste | Pastes the contents of the scrap into the image being edited by an image editor. |
| Preview | Toggles the visible state of the preview window for an image editor. |
| Redo | Selects the redo of the last action available in the undo log for an image editor. |
| Revert | Reverts the image being edited in an image editor to the last saved state. |
| Rotate | Rotates the current selection in an image editor 90 degrees. |
| SetFont | Opens the font chooser to allow the user to change the default font of an image editor. |
| SetSize | Opens the set size confirmation dialog to change the size of the image being edited in an image editor. |
| ShowGrid | Toggles the grid option for an image editor. |
| ShowPosition | |



This can be done only once for an object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENUBAR object. A VS_MENUBAR can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menubar reference |
| Default value: | Empty text |
| Access: | Write-only |

_VARIABLE_NAMES

For more information, see [Section 3](#).

19.2.4 Predefined Methods

_DROP_SELECTION

Drops the selected image in Image Editor.

Return value data type: No return value

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current cursor position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions a dialog's relative placement compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER** the side specification can be left out, because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

BASE

Dialog reference

A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Example:

Example 1: Placing **MY_DIALOG** at TOP/LEFT of **YOUR_DIALOG**:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),
("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering **MY_DIALOG** relative to **YOUR_DIALOG**:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialog's relative placement compared to the given rectangle in the coordinate system of a dialog item. The **BASE** and **RECT** parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters **X_FLAGS** and **Y_FLAGS** specify the relation to **BASE** on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: **PLACE_LEFT**, **PLACE_RIGHT**, **PLACE_TOP** and **PLACE_BOTTOM**.

Usually, **BASE** specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and **RECT** is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as **BASE**, and any rectangle as **RECT**.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out, because both sides are automatically used when centering.

Return value data type:

No return value

Parameter(s)

BASE

Dialog Item

A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object.

RECT

List

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP.

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),  
("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_GEOMETRY

Positions a dialog's relative placement compared to given rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER** the side specification can be left out, because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM , PLACE_LEFT , PLACE_RIGHT , PLACE_NEAR , PLACE_ADJACENT , PLACE_STRADDLE , PLACE_ALIGN , PLACE_INSET , PLACE_HALF , PLACE_CENTER and PLACE_TOP . |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM , PLACE_LEFT , PLACE_RIGHT , PLACE_NEAR , PLACE_ADJACENT , PLACE_STRADDLE , PLACE_ALIGN , PLACE_INSET , PLACE_HALF , PLACE_CENTER and PLACE_TOP . |

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),
 ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions a dialog's relative placement compared to a given SCIL rectangle. The **RECT** parameter specifies the coordinates to be used as base for the positioning. The parameters **X_FLAGS** and **Y_FLAGS** specify the relation to **RECT** on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: **PLACE_LEFT**, **PLACE_RIGHT**, **PLACE_TOP** and **PLACE_BOTTOM**.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside
PLACE_ALIGN = Completely inside with no margin
PLACE_INSET = Completely inside with a small margin
PLACE_HALF = Opposite side touches centre of given rectangle
PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out, because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

```
1: Placing MY_DIALOG at TOP/LEFT of RECT:  
  
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_MAX_SIZE

For more information, see [Section 3](#)

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 20 VS_INPUT_DIALOG

20.1 Overview

A VS_INPUT_DIALOG is used to ask for an input string from the user. The attribute _PROMPT_TEXT specifies the text to be displayed and _INPUT_TEXT is used to set a default string, as well as read the string given by the user. The _PROMPT_TEXT has to be set before the dialog is opened, otherwise the dialog is not automatically resized according to the amount of text. The string, which should be read, can be placed in the APPLY action method. The APPLY action method is executed when the user clicks the **OK** or **Apply** button.

20.1.1 Action Attributes

| Name | Type |
|-----------------|-------------|
| "_APPLY" | Text vector |
| "_CLOSE_NOTIFY" | Text vector |
| "_DEFAULTS" | Text vector |
| "_HELP" | Text vector |
| "_NOTIFY" | Text vector |

20.1.2 Action Methods

| Name | Type |
|----------------|-----------------|
| "APPLY" | Boolean |
| "CLOSE_NOTIFY" | Boolean |
| "DEFAULTS" | Boolean |
| "HELP" | Boolean |
| "NOTIFY" | No return value |

20.1.3 Predefined Attributes

| Name | Type |
|-----------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BUSY" | Boolean |
| "_BUTTONS_VERTICALLY" | Boolean |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_APPLY" | Boolean |

Table continues on next page

| | |
|-----------------------|------------------|
| " HAS_CANCEL" | Boolean |
| " HAS_DEFAULTS" | Boolean |
| " HAS_HELP" | Boolean |
| " HAS_OK" | Boolean |
| " ICON" | Image reference |
| " ICONIFIED" | Boolean |
| " INPUT_TEXT" | Text |
| " LEADER_DIALOG" | Dialog reference |
| " LEADER_DIALOG_ITEM" | Dialog item |
| " MAX_HEIGHT" | Integer |
| " MAX_WIDTH" | Integer |
| " MAXIMIZED" | Boolean |
| " MIN_HEIGHT" | Integer |
| " MIN_WIDTH" | Integer |
| " MODAL" | Boolean |
| " OBJECT_CLASS" | Text |
| " OBJECT_NAME" | Text |
| " OPEN" | Boolean |
| " PROMPT_TEXT" | Text |
| " SCIL_GEOMETRY" | List |
| " SOURCE_FILE_NAME" | Text |
| " STYLE" | Vector |
| " TITLE" | Text |
| " VARIABLE_NAMES" | Text vector |

20.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| " FLAG_FOR_EXECUTION" | No return value |
| " GET_HELP" | No return value |
| " GET_POINTER_POS" | No return value |
| " GET_STRING_WIDTH" | Integer |
| " PLACE" | No return value |
| " PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| " PLACE_ACCORDING_TO_GEOMETRY" | No return value |
| " PLACE_ACCORDING_TO_SCIL_GEOMETRY" | No return value |
| " QUEUE_FOR_EXECUTION" | No return value |
| " RAISE" | No return value |
| " SET_MAX_SIZE" | No return value |
| " SET_MIN_SIZE" | No return value |

20.2 Details

20.2.1 Action Attributes

APPLY

Setting this attribute specifies the action method APPLY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._APPLY = ("ROOT.APPLY_DIALOG", "")
```

CLOSE_NOTIFY

Setting this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT.CLOSE_DIALOG", "")
```

DEFAULTS

Setting this attribute specifies the action method DEFAULTS.

Data type: Text vector

Example:

```
.SET MY_DIALOG._DEFAULTS = ("ROOT.DEFAULTS_DIALOG", "")
```

HELP

Setting this attribute specifies the action method HELP.

Data type: Text vector

Example:

```
.SET MY_DIALOG._HELP = ("ROOT.HELP_DIALOG", "")
```

NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

20.2.2 Action Methods

APPLY

Specifies the action method that is executed when the **Apply** button is clicked in the confirmation dialog. If this action method is not set, the dialog is closed by default when **Apply** is clicked. The dialog is automatically closed if the user returns TRUE from this action method, when the **Apply** is clicked.

Data type to be returned: Boolean

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user chose to close the dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed by the program. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

DEFAULTS

Specifies the action method that is executed when the **Defaults** button is pressed in the confirmation dialog.

Data type to be returned: Boolean

HELP

Specifies the action method that is executed when the **Help** button is pressed in the confirmation dialog.

Data type to be returned: Boolean

NOTIFY

Specifies the notification procedure for an object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

20.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_BUTTONS_VERTICALLY

Specifies that the buttons within the confirmation dialog should be arranged vertically instead of horizontally. The default is horizontal arrangement.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._BUTTONS_VERTICALLY = TRUE
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Designates whether the window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system, and this function only indicates that the window has use for key events. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_HAS_APPLY

Specifies whether the confirmation dialog has an **Apply** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_APPLY = FALSE
```

_HAS_CANCEL

Specifies whether the confirmation dialog has a **Cancel** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_CANCEL = TRUE
```

_HAS_DEFAULTS

Specifies whether the confirmation dialog has a **Defaults** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_DEFAULTS = FALSE
```

_HAS_HELP

Specifies whether the confirmation dialog has a **Help** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_HELP = TRUE
```

_HAS_OK

Specifies whether the confirmation dialog has an **OK** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_OK = TRUE
```

_ICON

Specifies the icon used in the title of a window or when a window is iconified. Note that the _ICON attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether the window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_INPUT_TEXT

The text that the user has written into the text field. This attribute is also used to set a default value when opening input dialog.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side-effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like the _LEADER_DIALOG attribute, the _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one that holds the item specified. Setting this attribute to empty text unsets the leader dialog for the given window. For more information about leader-follower dialogs relationship, see description of the _LEADER_DIALOG attribute.

This attribute can be used, for example, in a picture methods, setting picture container as the leader dialog item. This is only way to raise a new dialog from a picture context and set the picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
;  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the entire display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not the window should grab cursor and key input from other windows when it is opened. When the _MODAL attribute is TRUE, it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_PROMPT_TEXT

Specifies the text telling the user what to write into an input text field. The text is automatically formatted to fit into the dialog. The user explicitly sets line breaks by inserting the ASCII(10) character into the text. The _PROMPT_TEXT has to be set before the dialog is opened, otherwise the dialog is not resized properly.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | Write-only |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_STYLE

The style flags determine what decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags have to be set always. STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute, the window will contain a border along with header information and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent. Not all platforms can distinguish all combinations of style flags.

Description:

Any combination of the following style flags given as a vector containing text strings. The flags are:
STYLE_BACKGROUND - An opaque background
STYLE_BORDER - A border around the content area
STYLE_MOVABLE - A title bar or other draggable area
STYLE_RESIZABLE - Resize handles or a grow box
STYLE_ICONIFIABLE - An iconify/minimize box
STYLE_MAXIMIZABLE - A maximize/zoom box
STYLE_MAIN - The top-level, non-document window
STYLE_DOCUMENT - Window contains a document
STYLE_CONFIRM - Window is a confirmation dialog
STYLE_UNCLOSABLE - No close box
STYLE_PALETTE - Window is a Tool window

Data type:

Vector

Default value:

("STYLE_BACKGROUND","STYLE_BORDER",
"STYLE_CONFIRM","STYLE_MOVABLE")

Access:

No limitations

_TITLE

The title of a window. Note that the title may not be visible under certain circumstances, for example if the window or dialog does not have a title bar.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

No limitations

_VARIABLE_NAMES

For more information, see [Section 3](#).

20.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current pointer position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions dialogs relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog reference A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),  
("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions dialogs relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

Table continues on next page

| | |
|---------|---|
| RECT | List |
| X_FLAGS | Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),  
("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_GEOMETRY

Positions dialogs relative place compared to given rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions dialogs relative place compared to given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 21 VS_LABEL

21.1 Overview

The VS_LABEL object is a dialog item that displays a text label. It does not contain any other functionality.

21.1.1 Action Attributes

| Name | Type |
|------------------------------------|-------------|
| "TOOLTIP_IS_SHOWN" | Text vector |

21.1.2 Action Methods

| Name | Type |
|------------------------------------|-----------------|
| "TOOLTIP_IS_SHOWN" | No return value |

21.1.3 Predefined Attributes

| Name | Type |
|-------------------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

21.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

21.2 Details

21.2.1 Action Attributes

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

21.2.2 Action Methods

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

21.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies the tooltip text appearing over the object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

21.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 22 VS_LINE

22.1 Overview

The VS_LINE object is a dialog item that displays a line. It does not contain any other functionality. The appearance of the line depends of the platform on which it is displayed. The line will be drawn horizontally or vertically depending on which one of the attributes _WIDTH and _HEIGHT in the _GEOMETRY is larger. The line is drawn along the left or bottom side of the specified GEOMETRY rectangle.

22.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_TOOLTIP_IS_SHOWN" | Text vector |

22.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "TOOLTIP_IS_SHOWN" | No return value |

22.1.3 Predefined Attributes

| Name | Type |
|------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BOTTOM_SHADOW_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_LINE_STYLE" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_TOP_SHADOW_COLOR" | Color |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

22.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

22.2 Details

22.2.1 Action Attributes

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

22.2.2 Action Methods

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

22.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BOTTOM_SHADOW_COLOR

The color used to draw the darker side of a line, if it is in 3D mode (_LINE_STYLE is STYLE_INSET or STYLE_OUTSET).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_FOREGROUND_COLOR

Specifies the foreground color for a dialog item. The foreground color is, for example, the color which is used when drawing the tile of an object.

Example:

```
.SET MY_OBJECT._FOREGROUND_COLOR=(65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_LINE_STYLE

One of STYLE_INSET, STYLE_OUTSET, STYLE_SOLID or STYLE_DASHED. Defines the line drawing style.

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_TOP_SHADOW_COLOR

The color used to draw the lighter side of a line, if it is in 3D mode (_LINE_STYLE is STYLE_INSET or STYLE_OUTSET).

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

22.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 23 VS_LIST

23.1 Overview

The VS_LIST object is a list that can contain an arbitrary amount of rows and columns. The NOTIFY method of the VS_LIST is run whenever the selected cell of the list has changed. The content of a cell or cell areas can be set or read by using predefined methods.

23.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_DOUBLE_CLICK" | Text vector |
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

23.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "DOUBLE_CLICK" | No return value |
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

23.1.3 Predefined Attributes

| Name | Type |
|------------------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_SCROLL_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DRAWING_SELECTION_ENABLED" | Boolean |
| "_ENABLED" | Boolean |
| "_EXTEND_COLUMN_SELECTION_ENABLED" | Boolean |
| "_EXTEND_ROW_SELECTION_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_FIRST_VISIBLE_COLUMN" | Integer |
| "_FIRST_VISIBLE_ROW" | Integer |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_GRID_ENABLED" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_KEYBOARD_SELECTION_ENABLED" | Boolean |
| "_KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_ROW_COUNT" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_CELLS" | Integer |
| "_SELECTED_ROW" | Integer |
| "_SELECTION_POLICY" | Text |
| "_SELECTION_SENSE_ENABLED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_TRAVERSAL_WRAP_ENABLED" | Boolean |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_GRID_ENABLED" | Boolean |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

23.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_CELL_TEXT" | Text |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SCROLL_POSITION_INTO_VIEW" | No return value |
| "_SCROLL_SELECTION_INTO_VIEW" | No return value |
| "_SET_CELL_TEXT" | No return value |
| "_SET_CELL_TEXT_RANGE" | No return value |
| "_SET_COLUMN_WIDTH" | No return value |
| "_SET_ROW_HEIGHT" | No return value |

23.2 Details

23.2.1 Action Attributes

_DOUBLE_CLICK

Setting this attribute specifies the action method DOUBLE_CLICK.

Data type: Text vector

Example:

```
.SET MY_LIST._DOUBLE_CLICK = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

23.2.2 Action Methods

DOUBLE_CLICK

The DOUBLE_CLICK program is run whenever the user double-clicks an item in a list. When a double-click occurs, the user can find out which row was clicked with the _SELECTED_ROW attribute.

Data type to be returned: No return value

GET_TABLE_EDIT_TEXT

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for a view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of a cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

23.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SCROLL_ENABLED

Specifies whether or not a list automatically scrolls when the user drags the mouse outside of the area currently visible in the list.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DRAWING_SELECTION_ENABLED

Specifies whether the drawing selection is used or not. If _DRAWING_SELECTION_ENABLED is set to TRUE, all the list cells that are under the cursor when dragging the mouse become selected. If _DRAWING_SELECTION_ENABLED is set to FALSE, the selected area is rectangular.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_LIST._DRAWING_SELECTION_ENABLED = TRUE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_EXTEND_COLUMN_SELECTION_ENABLED

When this attribute is set to TRUE, the entire column will be selected when any of the cells in the column is selected. The _SELECTION_POLICY attribute works as before, except that it operates on columns instead of single cells when this attribute is set to TRUE. Only one column can be selected when _SELECTION_POLICY is ONE_ONLY_SELECTION.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_LIST._EXTEND_COLUMN_SELECTION_ENABLED = TRUE
```

_EXTEND_ROW_SELECTION_ENABLED

When this attribute is set to TRUE, the entire row will be selected when any of the cells in the row is selected. The SELECTION_POLICY attribute works as before except, that it operates on rows instead of single cells when this attribute is set to TRUE. Only one row can be selected when SELECTION_POLICY is ONE_ONLY_SELECTION.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_LIST._EXTEND_ROW_SELECTION_ENABLED = TRUE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FIRST_VISIBLE_COLUMN

Contains the index of the leftmost visible column.

| | |
|----------------|---|
| Description: | An integer from 1 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | Read-only |

_FIRST_VISIBLE_ROW

Contains the index of the topmost visible row.

| | |
|----------------|---|
| Description: | An integer from 1 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | Read-only |

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Designates whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if the user grabbed it, so that the scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_GRID_ENABLED

Specifies whether the horizontal grid is enabled or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_KEYBOARD_SELECTION_ENABLED

If set to TRUE, this attribute specifies whether an object allows keyboard selection through the keyboard focus. This means, for example, that when a list has focus and a key is pressed, the selected cell of the list changes to the first one that starts with the character pressed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_KEYBOARD_TRAVERSAL_ENABLED

Specifies if keyboard traversal is used in list elements or not. If the value is TRUE, the object responds to keyboard commands. In this case, the user can make selections or change focused cell with keyboard arrow keys. If value is FALSE, the object will only handle the mouse events. In this case, the user can make selections or change focused cell only with mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the pop-up menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_ROW_COUNT

Specifies the number of rows in a list. Note that setting this attribute to zero clears the list.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_LIST._ROW_COUNT = 0
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_CELLS

Specifies the selected cells in a list. Note that if attribute _SELECTION_POLICY is set to ONE_ONLY_SELECTION, only one cell can be selected at a time. Note also, that setting this attribute does not give a SCIL error, if one or several of the specified cells are out of range, but the values that are out of range are ignored.

| | |
|----------------|---|
| Description: | A vector containing zero or more items. The items are vectors containing two integer items. |
| Data type: | Integer |
| Default value: | Empty vector |
| Access: | No limitations |

Example:

```
.SET MY_LIST._SELECTED_CELLS = ((1, 1), (1, 2), (1, 3))
```

_SELECTED_ROW

Specifies the number of selected row.

| | |
|----------------|---|
| Description: | An integer from 0 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
@ROW = MY_LIST._SELECTED_ROW
```

_SELECTION_POLICY

Specifies the selection policy in a list.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: FREE_SELECTION, ONE_ONLY_SELECTION, RECTANGULAR_SELECTION and NO_SELECTION. |
| Data type: | Text |
| Default value: | "ONE_ONLY_SELECTION" |
| Access: | No limitations |

_SELECTION_SENSE_ENABLED

If _SELECTION_SENSE_ENABLED is TRUE and the user adjust-clicks on selected cells, the cells are removed from the selection. When _SELECTION_SENSE_ENABLED is FALSE , clicking with the mouse only adds cells to the selection. Note that the implementation of adjust-click may vary between platforms. On the Windows platform, adjust-click is implemented as CTRL+left mouse button.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_TRAVERSAL_WRAP_ENABLED

Specifies whether the content of the list wraps around or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_GRID_ENABLED

Specifies whether the vertical grid is enabled or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

23.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_CELL_TEXT

Gets the text from a given cell in the list.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

Example:

```
@TEXT = MY_LIST._GET_CELL_TEXT(1, 1)
```

_GET_CELL_TEXT_RANGE

Returns the text vector ranging from the given cell in the list. The text area can be fetched horizontally or vertically.

| Parameter(s) | |
|--------------|--|
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| DIRECTION | Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL. |
| LENGTH | Integer An integer value. |

Example:

```
@VECTOR = MY_LIST._GET_CELL_TEXT_RANGE(1, 1, "HORIZONTAL", 10)
```

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SCROLL_POSITION_INTO_VIEW

Scrolls a list so that the specified cell is at least partly visible.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_SCROLL_SELECTION_INTO_VIEW

Scrolls a list so that the selection is displayed in the view.

Return value data type: No return value

_SET_CELL_TEXT

Sets the text of a given cell in the list. The list is automatically expanded if the text is set into a row or a column that does not exist.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| TEXT | Text A character string containing up to 255 characters. |

Example:

```
MY_LIST._SET_CELL_TEXT(1, 1, "Inserted Text")
```

_SET_CELL_TEXT_RANGE

Sets the text area ranging from the given cell in the list. The text area can be set horizontally or vertically. The list is automatically expanded if the text area is set into a row or a column that does not exist. The list can be cleared by setting the attribute _ROW_COUNT to zero.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| DIRECTION | Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL. |

Example:

```
MY_LIST._SET_CELL_TEXT_RANGE(1, 1, "HORIZONTAL", %VECTOR)
```

_SET_COLUMN_WIDTH

Sets the width of a specified column.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| WIDTH | Integer An integer value. |

Example:

```
MY_LIST._SET_COLUMN_WIDTH(1, 80)
```

_SET_ROW_HEIGHT

Sets the height of a specified row.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| HEIGHT | Integer An integer value. |

Example:

```
MY_LIST._SET_ROW_HEIGHT(1, 25)
```


Section 24 VS_MAIN_DIALOG

24.1 Overview

The main dialog should be used as the root object for independent dialogs or tools that must not interfere with the SCIL context of the parent object. A VS_MAIN_DIALOG works like a VS_DIALOG except that it is used as a barrier object between the SCIL context of the containing dialog/picture and the VS_MAIN_DIALOG with all its contents. It also works as a wall in the object visibility space.

24.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_ARRANGE_NOTIFY" | Text vector |
| "CLOSE_NOTIFY" | Text vector |
| "GOT_FOCUS_NOTIFY" | Text vector |
| "KEY_PRESSED" | Text vector |
| "LOST_FOCUS_NOTIFY" | Text vector |
| "NOTIFY" | Text vector |

24.1.2 Action Methods

| Name | Type |
|---------------------|-----------------|
| "ARRANGE_NOTIFY" | No return value |
| "CLOSE_NOTIFY" | Boolean |
| "GOT_FOCUS_NOTIFY" | No return value |
| "KEY_PRESSED" | Boolean |
| "LOST_FOCUS_NOTIFY" | No return value |
| "NOTIFY" | No return value |

24.1.3 Predefined Attributes

| Name | Type |
|---------------------------|-------------|
| "ABANDON_ITEM" | Dialog Item |
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "BACKGROUND_HILITE_COLOR" | Color |
| "BUSY" | Boolean |
| "CHILD_OBJECTS" | Text vector |
| "CLIPBOARD_HAS_CONTENTS" | Boolean |
| "CLIPBOARD_TEXT" | Vector |
| "CONFIRM_ITEM" | Dialog Item |
| "COPY_LOGICALS" | Boolean |

Table continues on next page

| | |
|-----------------------|------------------|
| "_DEFAULT_FOCUS_ITEM" | Dialog Item |
| "_DEFAULT_PATH" | Text |
| "_FILE_REVISION" | Text |
| "_FOCUS_ITEM" | Dialog Item |
| "_FOCUSABLE" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_ICON" | Image reference |
| "_ICONIFIED" | Boolean |
| "_ITEM_COUNT" | Integer |
| "_LEADER_DIALOG" | Dialog reference |
| "_LEADER_DIALOG_ITEM" | Dialog item |
| "_MAX_HEIGHT" | Integer |
| "_MAX_WIDTH" | Integer |
| "_MAXIMIZED" | Boolean |
| "_MIN_HEIGHT" | Integer |
| "_MIN_WIDTH" | Integer |
| "_MODAL" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OPEN" | Boolean |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_STYLE" | Vector |
| "_TITLE" | Text |
| "_VARIABLE_NAMES" | Text vector |

24.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_ITEM_INDEX" | Integer |
| "_GET_POINTER_POS" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_PLACE" | No return value |
| "_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_GEOMETRY" | No return value |
| "_PLACE_ACCORDING_TO_SCIL_GEOMETRY" | No return value |
| "_PRINT" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_RAISE" | No return value |

Table continues on next page

| | |
|--|-----------------|
| <u>"_SET_ITEM_INDEX"</u> | No return value |
| <u>"_SET_MAX_SIZE"</u> | No return value |
| <u>"_SET_MIN_SIZE"</u> | No return value |

24.2 Details

24.2.1 Action Attributes

_ARRANGE_NOTIFY

Setting this attribute specifies the action method ARRANGE_NOTIFY.

Data type: Text vector

_CLOSE_NOTIFY

Setting this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT.CLOSE_DIALOG", "")
```

_GOT_FOCUS_NOTIFY

Setting this attribute specifies the action method GOT_FOCUS_NOTIFY.

Data type: Text vector

_KEY_PRESSED

Setting this attribute specifies the action method KEY_PRESSED

Data type: Text vector

_LOST_FOCUS_NOTIFY

Setting this attribute specifies the action method LOST_FOCUS_NOTIFY.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

24.2.2 Action Methods

ARRANGE_NOTIFY

This action method is executed when the size of a dialog is changed with the mouse and the button is released.

Data type to be returned: No return value

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user closes a dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed programmatically. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

GOT_FOCUS_NOTIFY

This action method is executed immediately after the dialog has got input focus.

Data type to be returned: No return value

KEY_PRESSED

This action method is executed after a key or key combination is pressed in the dialog which has keyboard input focus. Four arguments are passed to the method: Key (text representation of the key), Alt_Enabled (boolean), Ctrl_Enabled (boolean), Shift_Enabled (boolean).

Below are text representations for Key argument:

- one letter from A to Z, case insensitive (ordinary alphabet keys)
- one number from 0 to 9 (numeric keys)
- Left, Right, Up and Down arrow keys
- HOME, END, PageUp, PageDown, INSERT, DELETE, TAB, ENTER, Backspace, ESC (corresponding keys, case insensitive)
- F1...F12 with the exception of F3 (this key is used to print the dialog which has focus).



The following key combinations are system reserved and do not activate the action method: CTRL+ALT+DEL, ALT+TAB, ALT?ENTER, ALT+ESC, CTRL+ESC.

Data type to be returned: Boolean

LOST_FOCUS_NOTIFY

This action method is executed immediately after a dialog has lost input focus.

Data type to be returned: No return value

NOTIFY

Specifies the notification procedure of an object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

24.2.3 Predefined Attributes

ABANDON_ITEM

Specifies which dialog item is the abandon item. The NOTIFY action method of the abandon item is executed when the user presses the abandon key. The meaning of the abandon key depends on the current look-and-feel. On Microsoft Windows, the abandon key is Escape (ESC).

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ABANDON_ITEM = "MY_CANCEL_BUTTON"
```

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BACKGROUND_HILITE_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLIPBOARD_HAS_CONTENTS

This attribute can be used to check if the clipboard has contents in text format that can be used from SCIL. The related attribute _CLIPBOARD_TEXT can be used to read or write the contents of the system clipboard on platforms that support the clipboard concept.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

_CLIPBOARD_TEXT

This attribute can be used to read and write the contents of the system clipboard on platforms that support the clipboard concept. The related attribute _CLIPBOARD_HAS_CONTENTS can be used to check if the clipboard has contents in text format that can be used from SCIL.

| | |
|----------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Default value: | Environment dependent |
| Access: | No limitations |

_CONFIRM_ITEM

Specifies which dialog item is the confirm item. The NOTIFY action method of the confirm item is executed when the user presses the confirm key. The meaning of the confirm key depends on the current look-and-feel. On Microsoft Windows, the confirm key is Return (ENTER). The confirm item is usually visually indicated with a thicker border than other items.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._CONFIRM_ITEM = "MY_OK_BUTTON"
```

_COPY_LOGICALS

A boolean value that defines whether temporary logical paths and representation libraries of the creating context are copied to the context of the created main dialog or not. Temporary logical paths and representation libraries are those that are defined using the "+" sign (#path my_path +/sc/apl/test/my_pict). The default value for this attribute is FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |

_DEFAULT_FOCUS_ITEM

Specifies which dialog item should receive keyboard focus when a dialog receives keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._DEFAULT_FOCUS_ITEM = "MY_TEXT"
```

_DEFAULT_PATH

Defines the path definition to be used when no logical path is used in file and library representation references within the main dialog context. If not specifies, the default path is inherited from the parent object.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Default path of creating object |

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUS_ITEM

Specifies the dialog item that currently has keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._FOCUS_ITEM = "MY_TEXT"
```

_FOCUSABLE

Designates whether a window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system, and this function only indicates that the window has use for key events. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_ICON

Specifies the icon used in the title of a window or when a window is iconified. Note that the _ICON attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether a window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ITEM_COUNT

Specifies the number of dialog items in a dialog. This attribute specifies the number of dialog items added directly to a dialog. Since certain dialog items may contain others (for example containers), the total number of dialog items in the dialog may be larger.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
@NUMBER_OF_DIALOG_ITEMS = MY_DIALOG._ITEM_COUNT
```

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like the _LEADER_DIALOG attribute, the _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one that holds the item specified. For more information about leader-follower dialogs relationship, see the description of the _LEADER_DIALOG attribute. Setting this attribute to empty text unsets the leader dialog for the given window.

This attribute can be used, for example, in a picture methods for setting picture container as the leader dialog item. This is only way to raise a new dialog from a picture context and set the picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the entire display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not the window should grab cursor and key input from other windows when it is opened. When the _MODAL attribute is TRUE, it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes the window but does not delete it.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_STYLE

The style flags determine what decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags has to be set always.
STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute, the window will contain the border along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent, and not all platforms can distinguish all combinations of style flags.

| | |
|----------------|---|
| Description: | Any combination of the following style flags given as a vector containing text strings. The flags are: STYLE_BACKGROUND - An opaque background STYLE_BORDER - A border around the content area STYLE_MOVABLE - A title bar or other dragable area STYLE_RESIZABLE - Resize handles or a grow box STYLE_ICONIFIABLE - An iconify/minimize box STYLE_MAXIMIZABLE - A maximize/zoom box STYLE_MAIN - The top-level, non-document window STYLE_DOCUMENT - Window contains a document STYLE_CONFIRM - Window is a confirmation dialog STYLE_UNCLOSABLE - No close box STYLE_PALETTE - Window is a Tool window |
| Data type: | Vector |
| Default value: | ("STYLE_BACKGROUND", "STYLE_BORDER", "STYLE_ICONIFIABLE", "STYLE_MAIN", "STYLE_MAXIMIZABLE", "STYLE_MOVABLE", "STYLE_RESIZABLE") |
| Access: | No limitations |

_TITLE

Specifies the title of a window. Note that the title may not be visible under certain circumstances, for example if the window or dialog does not have a title bar.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

24.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_ITEM_INDEX

Returns the item index of a dialog item inside the dialog. The item index specifies the tabbing and drawing order between dialog items inside a dialog.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| ITEM | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current pointer position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions a dialogs relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER, the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog reference A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),
("PLACE_TOP",
"PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialog's relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER, the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
    ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),  
    ("PLACE_CENTER"))
```

PLACE_ACCORDING_TO_GEOMETRY

Positions a dialog's relative place compared to a given rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER**, the side specification can be left out because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

RECT

List

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions a dialog's relative place compared to a given SCIL rectangle. The **RECT** parameter specifies the coordinates to be used as base for the positioning. The parameters **X_FLAGS** and **Y_FLAGS** specify the relation to **RECT** on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: **PLACE_LEFT**, **PLACE_RIGHT**, **PLACE_TOP** and **PLACE_BOTTOM**.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

```
1: Placing MY_DIALOG at TOP/LEFT of RECT:  
  
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PRINT

Prints the content of a dialog on the default printer. Note that overlapping windows which are located inside the dialog are printed, too.

Return value data type: No return value

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_ITEM_INDEX

Specifies the item index of a dialog item inside the dialog. The item index specifies the tabbing and drawing order between the dialog items inside a dialog.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ITEM | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| INDEX | Integer An integer value. |

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 25 VS_MENU

25.1 Overview

Contains the information needed to represent a single menu. Each VS_MENU contains an array of menu items, each of which represent an individual command. The parent of a VS_MENU is a VS_MENUBAR or a VS_SUB_MENU_ITEM in a hierarchical menu scheme. VS_MENUs including its parent the VS_MENUBAR and possible sub-menus can be created either dynamically or by using the dialog editor.

25.1.1 Action Attributes

| Name | Type |
|-------------------------------|-------------|
| "OPEN_NOTIFY" | Text vector |

25.1.2 Action Methods

| Name | Type |
|-------------------------------|-----------------|
| "OPEN_NOTIFY" | No return value |

25.1.3 Predefined Attributes

| Name | Type |
|------------------------------------|-------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "ITEM_COUNT" | Integer |
| "MNEMONIC" | Text |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "SOURCE_FILE_NAME" | Text |
| "TITLE" | Text |
| "VARIABLE_NAMES" | Text vector |

25.1.4 Predefined Methods

| Name | Type |
|--------------------------------------|-----------------|
| "FLAG_FOR_EXECUTION" | No return value |
| "GET_HELP" | No return value |

Table continues on next page

| | |
|--|-----------------|
| "_GET_STRING_WIDTH" | Integer |
| "_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |

25.2 Details

25.2.1 Action Attributes

_OPEN_NOTIFY

Setting this attribute specifies the action method OPEN_NOTIFY.

25.2.2 Action Methods

OPEN_NOTIFY

This action method is executed just before the menu is opened and drawn to a window. This action method can be used to change the properties of the VS_MENU.

Data type to be returned: No return value

25.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a menu is enabled or not. A change of appearance is also often combined with this attribute. On the Windows platform, the menu is dimmed when this attribute is set to TRUE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_ITEM_COUNT

Contains the number of menu items appended to a menu.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_MNEMONIC

Specifies the mnemonic character of a menu. A change of appearance is also often combined with this attribute. In Windows platform, the setting of this attribute underlines the mnemonic character in the menu.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a window. Note that the title may not be visible under certain circumstances, for example if the window or dialog does not have a title bar.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

25.2.4 Predefined Methods**_FLAG_FOR_EXECUTION**

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.spotify:user:1123248529

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialog's relative place compared to a given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

Table continues on next page

| | |
|---------|---|
| RECT | List |
| | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector |
| | Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector |
| | Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),  
("PLACE_CENTER"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 26 VS_MENU_ITEM

26.1 Overview

Contains the information needed to represent a single menu item. Each menu contains an array of MENU_ITEMS, each of which represent an individual command. A MENU_ITEM is a horizontal section of a menu. The parent of a MENU_ITEM is a MENU. MENUs including MENU_ITEMS, the MENUBAR and possible sub-menus can be created dynamically or using the Dialog Editor. The NOTIFY action method is called whenever the menu item is selected, either with the mouse or using mnemonic or accelerator keys. Accelerator keys are shortcuts to select the menu item using keyboard e.g. CTRL+S for Save menu item.

26.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| " NOTIFY" | Text vector |

26.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "NOTIFY" | No return value |

26.1.3 Predefined Attributes

| Name | Type |
|-------------------------------------|-----------------|
| " ATTRIBUTE_NAMES" | List |
| " CHILD_OBJECTS" | Text vector |
| " ENABLED" | Boolean |
| " FILE_REVISION" | Text |
| " FONT" | Font |
| " FOREGROUND_COLOR" | Color |
| " IMAGE" | Image reference |
| " MNEMONIC" | Text |
| " OBJECT_CLASS" | Text |
| " OBJECT_NAME" | Text |
| " SOURCE_FILE_NAME" | Text |
| " TITLE" | Text |
| " VARIABLE_NAMES" | Text vector |

26.1.4 Predefined Methods

| Name | Type |
|--------------------------------|-----------------|
| " <u>FLAG FOR EXECUTION</u> " | No return value |
| " <u>GET HELP</u> " | No return value |
| " <u>QUEUE FOR EXECUTION</u> " | No return value |
| " <u>SET ACCELERATOR KEY</u> " | No return value |

26.2 Details

26.2.1 Action Attributes

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

26.2.2 Action Methods

NOTIFY

Specifies the NOTIFY action method for a menu item. This is executed when a menu item is selected.

Data type to be returned: No return value

26.2.3 Predefined Attributes

_ATTRIBUTE NAMES

For more information, see [Section 3](#).

_CHILD OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether the menu item is enabled or not. Changing this attribute also affects the appearance of the menu item.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FILE REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_IMAGE

Specifies an image that is associated with an object. The way the image is displayed depends on the object. The image is given as an object reference to the image that has been loaded or created as an independent object.

| | |
|----------------|--|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. Size of the image should be of 16 % 16 preferably. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...")  
.SET ROOT\ MY_MENU_ITEM._IMAGE = "ROOT\ IMAGE1"
```

_MNEMONIC

Specifies the mnemonic character of a menu item. A change of appearance is also often combined with this attribute. In Microsoft Windows platform, the setting of this attribute underlines the mnemonic character in the menu item.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a menu item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

26.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SET_ACCELERATOR_KEY

This method can be used to set accelerator keys for a menu item.

The first argument (Key) may have one of the following text representations:

- one letter from A to Z (case insensitive, ordinary alphabet keys)
- one number from 0 to 9
- Left, Right, Up, Down arrow keys, case insensitive.
- HOME, END, PageUp, PageDown, INSERT, DELETE, TAB, ENTER, Backspace, ESC (corresponding keys, case insensitive)
- F1...F12 with the exception of F3 (this key is used to print the dialog which has focus).



The following key combinations are system reserved and should not be overridden: CTRL+ALT+DEL, ALT+TAB, ALT+ENTER, ALT+ESC, CTRL+ESC.

Return value data type: No return value

Parameter(s)

| | | |
|---------------|---------|---|
| KEY | Text | A character string containing up to 255 characters. |
| ALT_ENABLED | Boolean | A value of TRUE or FALSE. |
| CTRL_ENABLED | Boolean | A value of TRUE or FALSE. |
| SHIFT_ENABLED | Boolean | A value of TRUE or FALSE. |

Example:

```
ROOT\\MITEM._SET_ACCELERATOR_KEY("F5", FALSE, TRUE, FALSE)
```

Section 27 VS_MENUBAR

27.1 Overview

Contains the information needed to represent a single menubar. A VS_MENUBAR contains an ordered list of VS_MENUs. A menubar displays the title for each of its menus. Clicking or key traversing a title opens the associated menu. A VS_MENUBAR must be placed in a VS_DIALOG or VS_MAIN_DIALOG. VS_MENUBARs including its VS_MENUs can be created either dynamically or by using the dialog editor.

27.1.1 Action Attributes

| Name | Type |
|------------------------------------|-------------|
| "TOOLTIP_IS_SHOWN" | Text vector |

27.1.2 Action Methods

| Name | Type |
|------------------------------------|-----------------|
| "TOOLTIP_IS_SHOWN" | No return value |

27.1.3 Predefined Attributes

| Name | Type |
|------------------------------------|-------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FOCUSSED" | Boolean |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "HAS_HELP" | Boolean |
| "MENU_COUNT" | Integer |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "SCIL_GEOMETRY" | List |
| "SOURCE_FILE_NAME" | Text |
| "TITLE" | Text |
| "TOOLTIP_TEXT" | Text |
| "VARIABLE_NAMES" | Text vector |
| "VISIBLE" | Boolean |

27.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

27.2 Details

27.2.1 Action Attributes

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

27.2.2 Action Methods

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

27.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSSED

Contains the value TRUE if a dialog item has the keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_HELP

Specifies whether the last menu on the menu bar is a help menu or not. This has to be specified because the help menu may be treated differently than ordinary menus on some platforms.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MENU_COUNT

Specifies the number of top-level menus appended to a menu bar.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

27.2.4 Predefined Methods**_FLAG_FOR_EXECUTION**

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 28 VS_NOTEBOOK

28.1 Overview

A notebook item resembles a real notebook in appearance and performance. It contains one or more pages (VS_NOTEBOOK_PAGE), each of which contains dialog items that the user can view or edit. Only one page (the active page) is visible at a time. The remaining pages are stacked behind the active page. The NOTIFY action method is called every time a new page is appended or selected. If pages are appended to the notebook dynamically, the first page of the notebook is always selected as default.

28.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |

28.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

28.1.3 Predefined Attributes

| Name | Type |
|------------------------|----------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_PAGE_COUNT" | Integer |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_PAGE_INDEX" | Integer |

Table continues on next page

| | |
|----------------------------|-------------|
| <u>"_SOURCE_FILE_NAME"</u> | Text |
| <u>"_TOOLTIP_TEXT"</u> | Text |
| <u>"_VARIABLE_NAMES"</u> | Text vector |
| <u>"_VISIBLE"</u> | Boolean |

28.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| <u>"_FLAG_FOR_EXECUTION"</u> | No return value |
| <u>"_GET_HELP"</u> | No return value |
| <u>"_GET_STRING_WIDTH"</u> | Integer |
| <u>"_QUEUE_FOR_EXECUTION"</u> | No return value |

28.2 Details

28.2.1 Action Attributes

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

28.2.2 Action Methods

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

28.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_PAGE_COUNT

Specifies the number of notebook pages attached into a notebook.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_PAGE_INDEX

Specifies the page in the notebook that is selected.

| | |
|----------------|----------------------------|
| Description: | An integer greater than 0. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_NOTEBOOK._SELECTED_PAGE_INDEX=3
```

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

28.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 29 VS_NOTEBOOK_PAGE

29.1 Overview

A VS_NOTEBOOK_PAGE is a page of a VS_NOTEBOOK that can contain dialog items the same way as containers and dialogs. Only one page (the active page) is visible at a time.

29.1.1 Action Attributes

| Name | Type |
|------------------------------------|-------------|
| Section 29.2.1 | Text Vector |
| "PAGE_SELECTED" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

29.1.2 Action Methods

| Name | Type |
|------------------------------------|-----------------|
| "NOTIFY" | No return value |
| "PAGE_SELECTED" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

29.1.3 Predefined Attributes

| Name | Type |
|--|----------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DEFAULT_FOCUS_ITEM" | Dialog Item |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FLAT_KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "_FOCUS_ITEM" | Dialog Item |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_ITEM_COUNT" | Integer |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |

Table continues on next page

| | |
|---------------------|-------------|
| "_SCIL_GEOMETRY" | List |
| "_SELECTABLE" | Boolean |
| "_SELECTED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |

29.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SELECT" | No return value |

29.2 Details

29.2.1 Action Attributes

NOTIFY

Setting of this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

PAGE_SELECTED

Setting this attribute specifies the action method PAGE_SELECTED.

Data type: Text vector

TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

29.2.2 Action Methods

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

PAGE_SELECTED

The PAGE_SELECTED program is run whenever the user selects a notebook page. The notebook page can be selected programmatically by the method _SELECT.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

29.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DEFAULT_FOCUS_ITEM

Specified which dialog item should receive keyboard focus when the container receives keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._DEFAULT_FOCUS_ITEM = "MY_LIST"
```

_ENABLED

Specifies whether the VS_NOTEBOOK_PAGE item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled VS_NOTEBOOK_PAGE is displayed as dimmed and it cannot be activated with the keyboard or the mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FLAT_KEYBOARD_TRAVERSAL_ENABLED

Specifies that the container does not interfere with the traversal of the objects in the dialog (for example when using the TAB key to move from object to object). When _FLAT_KEYBOARD_TRAVERSAL_ENABLED is TRUE, the objects appear to be on the same level although they are within a container. The appearance of the container is also different depending on this attribute.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._FLAT_KEYBOARD_TRAVERSAL_ENABLED = TRUE
```

_FOCUS_ITEM

Changes the current keyboard focus of the VS_NOTEBOOK_PAGE to a specified dialog item. The dialog item that has keyboard focus receives all normal keystrokes and certain function keys. If the object is open, the appearance may change to indicate which item has focus. For some look-and-feels, the dashed line is drawn around the objects label. Note that the focus item is cleared whenever this page is selected.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._FOCUS_ITEM = "MY_OK_BUTTON"
```

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

Description: A value of TRUE or FALSE.

Data type: Boolean

Default value: Environment dependent

Access: Read-only

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_ITEM_COUNT

Specifies the number of dialog items in an object. This attribute specifies the number of dialog items added directly into the object. Since certain dialog items may contain others (for example containers), the total number of dialog items in the object may be larger.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
@NUMBER_OF_DIALOG_ITEMS = MY_CONTAINER._ITEM_COUNT
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTABLE

Specifies whether the user can select a notebook page. The programmer must make sure that the first page that is displayed is selectable.

| | |
|--------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Access: | No limitations |

_SELECTED

Indicates whether or not a page is currently the selected page of a notebook. A page in the notebook is selected programmatically using the predefined method _SELECT.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

29.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SELECT

Selects a notebook page. This method is used when the object name of notebook page is known.

Return value data type: No return value

Section 30 VS_NOTICE_DIALOG

30.1 Overview

The VS_NOTICE_DIALOG is a dialog box intended to inform the user of an error or to ask the user for confirmation. A notice dialog usually has either a single **OK** button or **Yes** and **No** buttons. Both can have optional **Cancel** and **Help** buttons. The attribute **_IS_CAUTION_DIALOG** specifies whether the dialog is a caution or an OK dialog. An OK dialog usually has just an **OK** button and a caution dialog has **Yes** and **No** buttons. Attribute **_IS_CAUTION_DIALOG** has to be set before the action methods are defined. It is possible to change the icon of the VS_NOTICE_DIALOG by setting the attribute **_NOTICE_ICON**.

30.1.1 Action Attributes

| Name | Type |
|-----------------|-------------|
| "_APPLY" | Text vector |
| "_CANCEL" | Text vector |
| "_CLOSE_NOTIFY" | Text vector |
| "_DEFAULTS" | Text vector |
| "_HELP" | Text vector |
| "_NO" | Text vector |
| "_NOTIFY" | Text vector |
| "_OK" | Text vector |
| "_YES" | Text vector |

30.1.2 Action Methods

| Name | Type |
|----------------|-----------------|
| "APPLY" | Boolean |
| "CANCEL_HOOK" | Integer |
| "CLOSE_NOTIFY" | Boolean |
| "DEFAULTS" | Boolean |
| "HELP" | Boolean |
| "NO_HOOK" | Integer |
| "NOTIFY" | No return value |
| "OK_HOOK" | Integer |
| "YES_HOOK" | Integer |

30.1.3 Predefined Attributes

| Name | Type |
|--------------------|-------------|
| "_ABANDON_ITEM" | Dialog Item |
| "_APPLY_TITLE" | Text |
| "_ATTRIBUTE_NAMES" | List |

Table continues on next page

| | |
|-----------------------|------------------|
| "_BACKGROUND_COLOR" | Color |
| "_BUSY" | Boolean |
| "_BUTTONS_VERTICALLY" | Boolean |
| "_CANCEL_TITLE" | Text |
| "_CHILD_OBJECTS" | Text vector |
| "_CONFIRM_ITEM" | Dialog Item |
| "_DEFAULT_FOCUS_ITEM" | Dialog Item |
| "_DEFAULTS_TITLE" | Text |
| "_FILE_REVISION" | Text |
| "_FOCUS_ITEM" | Dialog Item |
| "_FOCUSABLE" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_APPLY" | Boolean |
| "_HAS_CANCEL" | Boolean |
| "_HAS_DEFAULTS" | Boolean |
| "_HAS_HELP" | Boolean |
| "_HAS_ICON" | Boolean |
| "_HAS_OK" | Boolean |
| "_HELP_TITLE" | Text |
| "_ICON" | Image reference |
| "_ICONIFIED" | Boolean |
| "_IS_CAUTION_DIALOG" | Boolean |
| "_LEADER_DIALOG" | Dialog reference |
| "_LEADER_DIALOG_ITEM" | Dialog item |
| "_MAX_HEIGHT" | Integer |
| "_MAX_WIDTH" | Integer |
| "_MAXIMIZED" | Boolean |
| "_MIN_HEIGHT" | Integer |
| "_MIN_WIDTH" | Integer |
| "_MODAL" | Boolean |
| "_NO_TITLE" | Text |
| "_NOTICE_ICON" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OK_TITLE" | Text |
| "_OPEN" | Boolean |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_STYLE" | Vector |
| "_TEXT" | Text |

Table continues on next page

| | |
|--------------------------|-------------|
| <u>"_TITLE"</u> | Text |
| <u>"_VARIABLE_NAMES"</u> | Text vector |
| <u>"_YES_TITLE"</u> | Text |

30.1.4 Predefined Methods

| Name | Type |
|---|-----------------|
| <u>"_FLAG_FOR_EXECUTION"</u> | No return value |
| <u>"_GET_HELP"</u> | No return value |
| <u>"_GET_POINTER_POS"</u> | No return value |
| <u>"_GET_STRING_WIDTH"</u> | Integer |
| <u>"_PLACE"</u> | No return value |
| <u>"_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY"</u> | No return value |
| <u>"_PLACE_ACCORDING_TO_GEOMETRY"</u> | No return value |
| <u>"_PLACE_ACCORDING_TO_SCIL_GEOMETRY"</u> | No return value |
| <u>"_QUEUE_FOR_EXECUTION"</u> | No return value |
| <u>"_RAISE"</u> | No return value |
| <u>"_SET_MAX_SIZE"</u> | No return value |
| <u>"_SET_MIN_SIZE"</u> | No return value |

30.2 Details

30.2.1 Action Attributes

_APPLY

Setting of this attribute specifies the action method APPLY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._APPLY = ("ROOT.APPLY_DIALOG", "")
```

_CANCEL

Setting of this attribute specifies the action method CANCEL.

Data type: Text vector

_CLOSE_NOTIFY

Setting of this attribute specifies the action method CLOSE_NOTIFY.

Data type: Text vector

Example:

```
.SET MY_DIALOG._CLOSE_NOTIFY = ("ROOT.CLOSE_DIALOG", "")
```

_DEFAULTS

Setting of this attribute specifies the action method DEFAULTS.

Data type: Text vector

Example:

```
.SET MY_DIALOG._DEFAULTS = ("ROOT.DEFAULTS_DIALOG", "")
```

_HELP

Setting of this attribute specifies the action method HELP.

Data type: Text vector

Example:

```
.SET MY_DIALOG._HELP = ("ROOT.HELP_DIALOG", "")
```

_NO

Setting of this attribute specifies the action method NO.

Data type: Text vector

_NOTIFY

Setting of this attribute specifies the action method NOTIFY.

Data type: Text vector

_OK

Setting of this attribute specifies the action method OK.

Data type: Text vector

_YES

Setting of this attribute specifies the action method YES.

Data type: Text vector

30.2.2 Action Methods

APPLY

Specifies the action method that is executed when the **Apply** button is clicked in the confirmation dialog. If this action method is not set, the dialog is closed by default, when the **Apply** is clicked. The dialog is automatically closed if the user returns TRUE from this action method, when the **Apply** is clicked.

Data type to be returned: Boolean

CANCEL_HOOK

Specifies the action method that is executed when the “**Cancel**” button is pressed in the notice dialog. If this action method is not set the default behaviour when the button is pressed is that the dialog is closed. The dialog is automatically closed if the user returns TRUE from this action method.

Data type to be returned: Integer

CLOSE_NOTIFY

The CLOSE_NOTIFY action method is called immediately after the user chose to close the dialog. It is called before the dialog is actually closed. If the method returns TRUE, the dialog is not closed. If it returns FALSE, the dialog is closed normally. Note that the CLOSE_NOTIFY action method is not always run. For example, it is not run when a dialog is deleted or closed programmatically. Normal cleaning up after a dialog should be done in the DELETE method of the dialog and not in the CLOSE_NOTIFY method.

Data type to be returned: Boolean

DEFAULTS

Specifies the action method that is executed when the “**Defaults**” button is pressed in the confirmation dialog.

Data type to be returned: Boolean

HELP

Specifies the action method that is executed when the “**Help**” button is pressed in the confirmation dialog.

Data type to be returned: Boolean

NO_HOOK

Specifies the action method that is executed when the “**No**” button is pressed in the notice dialog. If this action method is not set the default behaviour when the button is pressed is that the dialog is closed. The dialog is automatically closed if the user returns TRUE from this action method.

Data type to be returned: Integer

NOTIFY

Specifies the notification procedure of the object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

OK_HOOK

Specifies the action method that is executed when the “**OK**” button is pressed in the notice dialog. If this action method is not set the default behaviour when the button is pressed is that the dialog is closed. The dialog is automatically closed if the user returns TRUE from this action method.

Data type to be returned: Integer

YES_HOOK

Specifies the action method that is executed when the “**Yes**” button is pressed in the notice dialog. If this action method is not set the default behaviour when the button is pressed is that the dialog is closed. The dialog is automatically closed if the user returns TRUE from this action method.

Data type to be returned: Integer

30.2.3 Predefined Attributes

_ABANDON_ITEM

Specifies which dialog item that is the abandon item. The NOTIFY action method of the abandon item is executed when the user presses the abandon key. The meaning of the abandon key depends on current look-and-feel. On Microsoft Windows the abandon key is Escape (Esc).

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ABANDON_ITEM = "MY_CANCEL_BUTTON"
```

_APPLY_TITLE

The title of the “Apply” button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_BUTTONS_VERTICALLY

Specifies that the buttons within the confirmation dialog should be arranged vertically instead of horizontally. The default is horizontal arrangement.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._BUTTONS_VERTICALLY = TRUE
```

_CANCEL_TITLE

The title of the “Cancel” button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CONFIRM_ITEM

Specifies which dialog item is the confirm item. The NOTIFY action method of the confirm item is executed when the user presses the confirm key. The meaning of the confirm key depends on current look-and-feel. On Microsoft Windows the confirm key is Return (Enter). The confirm item is usually visually indicated with a thicker border than other items.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._CONFIRM_ITEM = "MY_OK_BUTTON"
```

_DEFAULT_FOCUS_ITEM

Specifies which dialog item should receive keyboard focus when the dialog receives keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._DEFAULT_FOCUS_ITEM = "MY_TEXT"
```

_DEFAULTS_TITLE

The title of the “Defaults“ button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUS_ITEM

Specifies the dialog item that currently has keyboard focus.

| | |
|----------------|---|
| Description: | A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| Data type: | Dialog Item |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._FOCUS_ITEM = "MY_TEXT"
```

_FOCUSABLE

Designates whether the window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system; this function only indicates that the window has use for key events. The window manager lets user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_HAS_APPLY

Specifies whether the confirmation dialog has an **Apply** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_APPLY = FALSE
```

_HAS_CANCEL

Specifies whether the confirmation dialog has a **Cancel** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_CANCEL = TRUE
```

_HAS_DEFAULTS

Specifies whether the confirmation dialog has a **Defaults** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_DEFAULTS = FALSE
```

_HAS_HELP

Specifies whether the confirmation dialog has a **Help** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_HELP = TRUE
```

_HAS_ICON

Specifies whether the icon is displayed or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

_HAS_OK

Specifies whether the confirmation dialog has an **OK** button or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._HAS_OK = TRUE
```

_HELP_TITLE

The title of the “**Help**“ button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_ICON

Specifies the icon used in the title of the window or when the window is iconified. Note that the **_ICON** attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether the window is iconified or not. The exact implication of iconification depends on the window system and window manager in use. The window manager lets user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_IS_CAUTION_DIALOG

Specifies whether the notice dialog is caution dialog or not. If _IS_CAUTION_DIALOG is TRUE the notice dialog is displayed with “Yes” and “No” buttons. Additional “Cancel” button can be added, too. If _IS_CAUTION_DIALOG is FALSE then the notice dialog is displayed with “OK” button.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side-effect. The exact implications of this designation are window system dependent; typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened. Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_LEADER_DIALOG_ITEM

Like _LEADER_DIALOG attribute _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one, which holds the item specified. For more information about leader-follower dialogs relationship see description of the _LEADER_DIALOG attribute. Setting this attribute to empty text unsets the leader dialog for the given window.

This attribute can be used, for example, in a picture methods, setting picture container as the leader dialog item. This is only way to raise a new dialog from a picture context and set the picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
; ROOT is the picture container
```

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of the window so that it encloses its maximum size or the display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not the window should grab pointer and key input from other windows when it is opened. The effect when the _MODAL attribute is TRUE is that it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_NO_TITLE

The title of the “No” button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_NOTICE_ICON

Specifies the reference to dialog item used as a icon in notice dialog.

| | |
|----------------|--|
| Description: | A text string containing one of the following keywords: ICON_INFORMATION, ICON_CAUTION or ICON_CRITICAL. |
| Data type: | Text |
| Default value: | Not applicable |
| Access: | Write-only |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OK_TITLE

The title of the “OK” button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

_OPEN

Setting the _OPEN attribute to TRUE opens the window so it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_STYLE

The style flags determine what decorations, if any, are placed around the window's border as well as how the user can directly manipulate the window. Note that the visual effect of setting style flags is platform dependent and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The STYLE_BACKGROUND and STYLE_MAIN flags has to be set always.
STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of the window. Without the STYLE_BACKGROUND attribute the window will contain the border along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent and not all platforms can distinguish all combinations of style flags.

Description:

Any combination of the following style flags given as a vector containing text strings. The flags are:
STYLE_BACKGROUND - An opaque background,
STYLE_BORDER - A border around the content area,
STYLE_MOVABLE - A title bar or other draggable area,
STYLE_RESIZABLE - Resize handles or a grow box,
STYLE_ICONIFIABLE - An iconify/minimize box,
STYLE_MAXIMIZABLE - A maximize/zoom box,
STYLE_MAIN - The top-level, non-document window,
STYLE_DOCUMENT - Window contains a document,
STYLE_CONFIRM - Window is a confirmation dialog,
STYLE_UNCLOSABLE - No close box, STYLE_PALETTE - Window is a Tool window

Data type:

Vector

Default value:

("STYLE_BACKGROUND", "STYLE_BORDER",
"STYLE_ICONIFIABLE", "STYLE_MAIN",
"STYLE_MAXIMIZABLE", "STYLE_MOVABLE",
"STYLE_RESIZABLE")

Access:

No limitations

_TEXT

Specifies the text used in notice dialog. Additional ASCII(10) characters can be added as line breaks into text string.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

Write-only

_TITLE

The title of the window. Note that the title may not be visible under certain circumstances. For example if the window or dialog does not have a title bar.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

No limitations

_VARIABLE_NAMES

For more information, see [Section 3](#).

_YES_TITLE

The title of the “Yes” button can be changed by setting this attribute.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Language dependent |
| Access: | No limitations |

30.2.4 Predefined Methods**_FLAG_FOR_EXECUTION**

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current pointer position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---------|
| Return value data type: | Integer |
| Parameter(s) | |

STRING

Text

A character string containing up to 255 characters.

_PLACE

Positions dialogs relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

BASE

Dialog reference

A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),  
("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions dialogs relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies a kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER** the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT , VS_BUTTON etc. The reference is a text reference to the name of the object. |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM , PLACE_LEFT , PLACE_RIGHT , PLACE_NEAR , PLACE_ADJACENT , PLACE_STRADDLE , PLACE_ALIGN , PLACE_INSET , PLACE_HALF , PLACE_CENTER and PLACE_TOP . |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM , PLACE_LEFT , PLACE_RIGHT , PLACE_NEAR , PLACE_ADJACENT , PLACE_STRADDLE , PLACE_ALIGN , PLACE_INSET , PLACE_HALF , PLACE_CENTER and PLACE_TOP . |

Example:

Example 1: Placing **MY_DIALOG** at TOP/LEFT of the rectangle (0, 0, 100, 100) within **YOUR_CONTAINER**:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering **MY_DIALOG** relative to **YOUR_CONTAINER** itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY
#MODIFY GEOM:V = LIST(X = 0, Y = 0)

MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),
("PLACE_CENTER"))
```

PLACE_ACCORDING_TO_GEOMETRY

Positions dialogs relative place compared to given rectangle. The **RECT** parameter specifies the coordinates to be used as base for the positioning. The parameters **X_FLAGS** and **Y_FLAGS** specify the relation to **RECT** on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: **PLACE_LEFT**, **PLACE_RIGHT**, **PLACE_TOP** and **PLACE_BOTTOM**.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin
PLACE_ADJACENT = Completely outside with no margin
PLACE_STRADDLE= Half inside, half outside
PLACE_ALIGN = Completely inside with no margin
PLACE_INSET= Completely inside with a small margin
PLACE_HALF = Opposite side touches centre of given rectangle
PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

RECT

List

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Y_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions dialogs relative place compared to given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER** the side specification can be left out because both sides are automatically used when centring.

Return value data type:

No return value

Parameter(s)

RECT

List

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system.

X_FLAGS

Vector

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Y_FLAGS

Vector
Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, **PLACE_LEFT**, **PLACE_RIGHT**,
PLACE_NEAR, **PLACE_ADJACENT**, **PLACE_STRADDLE**,
PLACE_ALIGN, **PLACE_INSET**, **PLACE_HALF**,
PLACE_CENTER and **PLACE_TOP**.

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  

("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises the dialog to the top of all other windows.

Return value data type: No return value

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 31 VS_NUMERIC_SPINNER

31.1 Overview

The VS_NUMERIC_SPINNER object is an object that can have an integral value. The appearance of the VS_NUMERIC_SPINNER is dependent on the platform. The attributes _MIN_VALUE, _MAX_VALUE and _INCREMENT_VALUE defines the behaviour of the object. The current value of an object can be read and set using the _VALUE attribute. The NOTIFY action method for this object will be called whenever the user interactively changes the value of the object.

31.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

31.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

31.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |

Table continues on next page

| | |
|---------------------|-------------|
| "_GEOMETRY" | List |
| "_INCREMENT_VALUE" | Integer |
| "_MAX_VALUE" | Integer |
| "_MIN_VALUE" | Integer |
| "_MODIFIABLE" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VALUE" | Integer |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |
| "_WRAP_ENABLED" | Boolean |

31.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

31.2 Details

31.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

31.2.2 Action Methods

GET_TABLE_EDIT_TEXT

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of the cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

31.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_INCREMENT_VALUE

Indicates the minor increment step size, for example for a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._INCREMENT_VALUE = 5
```

_MAX_VALUE

Indicates the maximum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MAX_VALUE = 10
```

_MIN_VALUE

Indicates the minimum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MIN_VALUE = 1
```

_MODIFIABLE

Specifies whether the value of a numeric spinner is modifiable or not. A change of appearance is also often combined with this attribute.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE

Indicates the current value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VALUE = 1
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_WRAP_ENABLED

Specifies whether the values displayed in a numeric spinner can increase or decrease straight from one end point of the spinner to another end point (for example, from the maximum value to the minimum value). Maximum and minimum values of a numeric spinner are specified by using the attributes _MIN_VALUE and _MAX_VALUE.

| | |
|----------------|--------------------------|
| Description: | A value of TRUE or FALSE |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

31.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 32 VS_OPTION_BUTTON

32.1 Overview

A VS_OPTION_BUTTON object is an object that can be selected but not cleared (deselected). The appearance of the option button is dependent on the graphics platform. The attribute _VALUE contains the current state of the option button. The NOTIFY action method is called when the option button is selected. When option buttons are used in a group, all other option buttons should be cleared when one is selected. Other option buttons should be cleared from SCIL in NOTIFY action method of the selected option button.

32.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

32.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

32.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |

Table continues on next page

| | |
|----------------------------|----------------|
| <u>"_BACKGROUND_COLOR"</u> | Color |
| <u>"_GEOMETRY"</u> | List |
| <u>"_MNEMONIC"</u> | Text |
| <u>"_OBJECT_CLASS"</u> | Text |
| <u>"_OBJECT_NAME"</u> | Text |
| <u>"_POP_UP_MENU"</u> | Menu reference |
| <u>"_SCIL_GEOMETRY"</u> | List |
| <u>"_SOURCE_FILE_NAME"</u> | Text |
| <u>"_TITLE"</u> | Text |
| <u>"_TOOLTIP_TEXT"</u> | Text |
| <u>"_VALUE"</u> | Integer |
| <u>"_VARIABLE_NAMES"</u> | Text vector |
| <u>"_VISIBLE"</u> | Boolean |

32.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| <u>"_FLAG_FOR_EXECUTION"</u> | No return value |
| <u>"_GET_HELP"</u> | No return value |
| <u>"_GET_STRING_WIDTH"</u> | Integer |
| <u>"_QUEUE_FOR_EXECUTION"</u> | No return value |

32.2 Details

32.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

32.2.2 Action Methods**GET_TABLE_EDIT_TEXT**

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used,

for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

32.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_MNEMONIC

Specifies the mnemonic character of a dialog item. The mnemonic may be visualized differently on different platforms.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._MNEMONIC = "A"
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a dialog item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._TITLE = "Tool Properties"
```

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE

Indicates the current value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VALUE = 1
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

32.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 33 VS_PALETTE

33.1 Overview

The VS_PALETTE object is an object that contains buttons. Buttons are added to the VS_PALETTE using the _APPEND_BUTTON method. Objects of type VS_IMAGE can be placed on buttons using the method _SET_BUTTON_IMAGE. The NOTIFY method for this object will be called either when pressing or releasing a button depending on the value of the _NOTIFY_ON_BUTTON_UP attribute.

33.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

33.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

33.1.3 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BUTTON_COUNT" | Integer |
| "_BUTTON_HEIGHT" | Integer |
| "_BUTTON_WIDTH" | Integer |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_COUNT" | Integer |
| "_CLICK_FOCUSABLE" | Boolean |
| "_COLUMN_COUNT" | Integer |
| "_ENABLED" | Boolean |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |
| "_HAS_BORDER" | Boolean |
| "_HAS_FLAT_APPEARANCE" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_KEYBOARD_SELECTION_ENABLED" | Boolean |
| "_KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_ROW_COUNT" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_BUTTON_INDEX" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_TRAVERSAL_WRAP_ENABLED" | Boolean |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

33.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| "_APPEND_BUTTON" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_REMOVE_BUTTON_AT" | No return value |
| "_SCROLL_SELECTION_INTO_VIEW" | No return value |
| "_SET_BUTTON_IMAGE" | No return value |
| "_SET_BUTTON_SIZE" | No return value |

33.2 Details

33.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

33.2.2 Action Methods

GET_TABLE_EDIT_TEXT

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method could be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

33.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_BUTTON_COUNT

Specifies the number of appended buttons in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_BUTTON_HEIGHT

Specifies the height of a button in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 20 |
| Access: | Read-only |

_BUTTON_WIDTH

Specifies the width of a button in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 20 |
| Access: | Read-only |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_COUNT

Returns the number of mouse clicks assigned into a palette. The value of attribute `_CLICK_COUNT` is purposed to read inside NOTIFY action method. This function is used to test, for example, double and triple clicks. Note that the attribute value is automatically set to default value when NOTIFY is executed.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
; Notify method of palette
@NUMBER_OF_CLICKS = ROOT\ MY_PALETTE._CLICK_COUNT
```

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_COLUMN_COUNT

Specifies the number of used columns in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

Table 1:

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Designates whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if the user grabbed it, so that the scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HAS_FLAT_APPEARANCE

Determines whether the VS_PALETTE should be drawn Flat or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_PALETTE._HAS_FLAT_APPEARANCE = TRUE
```



The mouse over feature is not available for VS_PALETTE. Flat appearance is visible only for tools opened in Modern LAF.

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_KEYBOARD_SELECTION_ENABLED

When set to TRUE, this attribute specifies whether an object allows keyboard selection through keyboard focus. This means, for example, that when a list has focus and a key is pressed, the selected cell of the list changes to the first one that starts with the character pressed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_KEYBOARD_TRAVERSAL_ENABLED

Specifies if keyboard traversal is used in list elements or not. If the value is TRUE, the object responds to keyboard commands. In this case, the user can make selections or change the focused cell with keyboard arrow keys. If value is FALSE, the object will only handle the mouse events. In this case, the user can make selections or change the focused cell only with mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking an object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_ROW_COUNT

Specifies the number of used rows in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_BUTTON_INDEX

Specifies the index of a selected button in a palette.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_TRAVERSAL_WRAP_ENABLED

Specifies whether the content of a list wraps around or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

33.2.4 Predefined Methods

_APPEND_BUTTON

Inserts a button into a palette. The new button is attached to the end of the palette. A new index is assigned into inserted button.

Return value data type: No return value

Example:

```
.CREATE ROOT\ MY_PALETTE = VS_PALETTE (-  
_GEOMETRY = LIST(...  
_OPEN = TRUE)  
  
ROOT\ MY_PALETTE._APPEND_BUTTON  
  
ROOT\ MY_PALETTE._SET_BUTTON_IMAGE(1, "ROOT\ IMAGE1")
```

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_REMOVE_BUTTON_AT

Removes a button with specified index from a palette. The indices after the removed button are decreased by one. Due to this, the indices of buttons should be rearranged after using this method.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |

Example:

```
ROOT\ MY_PALETTE._REMOVE_BUTTON_AT(1)
```

_SCROLL_SELECTION_INTO_VIEW

Scrolls a list so that the selection is displayed in the view.

Return value data type: No return value

_SET_BUTTON_IMAGE

Sets the image of a specific button in the palette.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_SET_BUTTON_SIZE

Sets the width and height of all the buttons in a palette. If images are shown on the buttons, they are placed in the centre of the button. Note that if images are to be displayed on the buttons, the images should be drawn to adjust with a specified button size. If the button size is changed later, the images will not be scaled.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| W | Integer An integer value. |
| H | Integer An integer value. |

Example:

```
ROOT\ MY_PALETTE._SET_BUTTON_SIZE(25, 25)
```


Section 34 VS_PICTURE_CONTAINER

34.1 Overview

The picture container VS_PICTURE_CONTAINER is an object that serves as a container for conventional pictures. This object works as a barrier between the SCIL context of the containing dialog and the picture. It also works as a wall in the object visibility space.

34.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

34.1.2 Action Methods

| Name | Type |
|--------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

34.1.3 Predefined Attributes

| Name | Type |
|----------------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_BOUNDS_GEOMETRY" | List |
| "_BUSY" | Boolean |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_CONTENT_GEOMETRY" | List |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |

Table continues on next page

| | |
|--------------------------------|----------------|
| "_ITEM_COUNT" | Integer |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_PICTURE_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

34.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| _GET_STRING_WIDTH | Integer |
| "_INT_PIC" | No return value |
| "_LAST_PIC" | No return value |
| "_NEW_PIC" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SHOW_BACK" | No return value |

34.2 Details

34.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

34.2.2 Action Methods**GET_TABLE_EDIT_TEXT**

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever the table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

34.2.3 Predefined Attributes**_ATTRIBUTE_NAMES**

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_BOUNDS_GEOMETRY

The BOUNDS_GEOMETRY is the entire virtual coordinate space of a container. Position (0,0) in the container coordinate system is at the lower left corner of the BOUNDS_GEOMETRY. The BOUNDS_GEOMETRY is interpreted in the same coordinate space as the GEOMETRY of the object. Each object in the container is placed relative to the lower left corner of the BOUNDS_GEOMETRY.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | No limitations |

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of an object to indicate that the object is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Write-only |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_CONTENT_GEOMETRY

The CONTENT_GEOMETRY is the portion of the coordinate space that is visible to the user at any given moment. If CONTENT_GEOMETRY is smaller than BOUNDS_GEOMETRY, the scroll

bars must be inserted for horizontal and vertical scrolling. The BOUNDS_GEOMETRY is clipped to CONTENT_GEOMETRY when the interior of a container is drawn.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Object dependent |
| Access: | Read-only |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether a container has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._HORIZONTAL_SCROLL_BAR_ENABLED = TRUE
```

_ITEM_COUNT

Specifies the number of dialog items in an object. This attribute specifies the number of dialog items added directly into the object. Since certain dialog items may contain others (for example containers), the total number of dialog items in the object may be larger.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

Example:

```
@NUMBER_OF_DIALOG_ITEMS = MY_CONTAINER._ITEM_COUNT
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_PICTURE_NAME

A read only attribute that contains the name of a picture currently shown in the picture container.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.



If user want's to use _POP_UP_MENU with this object, this object must be inserted in VS_CONTAINER and all other items should be also inserted in VS_CONTAINER or VS_NOTEBOOK.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\MENU1 = VS_MENU("...")  
.SET ROOT\MY_LIST._POP_UP_MENU = "ROOT\MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether a container has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTAINER._VERTICAL_SCROLL_BAR_ENABLED = TRUE
```

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

34.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_INT_PIC

Displays an alarm picture. Incoming alarms are put into a monitor specific picture queue. This method displays the oldest alarm picture in the queue. At the same time, the name of the picture is removed from the alarm picture queue. Note that if the alarm picture queue is empty, the method has no effect.

Return value data type: No return value

_LAST_PIC

Displays the previous picture. This method causes the picture handling unit to go one step backwards in the queue built up by _NEW_PIC method in picture container.

Return value data type: No return value

_NEW_PIC

Shows the picture with the given name in the picture container. The name can be given as a simple name, as path/name or as a full path name. The name is resolved in the callers context.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s): | |
| NAME | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SHOW_BACK

Shows the background of a specified picture in a container.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s): | |
| NAME | Text A character string containing up to 255 characters. |

Section 35 VS_PRINT_OBJECT

35.1 Overview

A VS_PRINT_OBJECT enables the user to setup a printer, print specified dialog or text to a default printer. The method _SETUP_PRINTER opens a platform dependent printer setup dialog. The method _PRINT_TEXT_VECTOR prints text vector to a default printer according to the default margins.

35.1.1 Action Attributes

| Name | Type |
|----------------------|-------------|
| "_GOT_FOCUS_NOTIFY" | Text vector |
| "_KEY_PRESSED" | Text vector |
| "_LOST_FOCUS_NOTIFY" | Text vector |

35.1.2 Action Methods

| Name | Type |
|---------------------|-----------------|
| "GOT_FOCUS_NOTIFY" | No return value |
| "KEY_PRESSED" | Boolean |
| "LOST_FOCUS_NOTIFY" | No return value |

35.1.3 Predefined Attributes

| Name | Type |
|-----------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_JOB_CONFIRMATION_ENABLED" | Boolean |
| "_LANDSCAPE_ENABLED" | Boolean |
| "_LEADER_DIALOG_ITEM" | Dialog item |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_PAGE_GEOMETRY" | List |
| "_PAPER_GEOMETRY" | List |
| "_PRINT_PAGE_RANGE_ENABLED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT_WRAP_ENABLED" | Boolean |
| "_VARIABLE_NAMES" | Text vector |

35.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| "_CLOSE_PRINT_SESSION" | Integer |
| "_CLOSE_PRINTER" | Boolean |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_DIALOG_PRINT_SCALE" | List |
| "_GET_MARGINS" | List |
| "_GET_STRING_WIDTH" | Integer |
| "_OPEN_PRINT_SESSION" | Integer |
| "_OPEN_PRINTER" | Boolean |
| "_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY" | No return value |
| "_PRINT_DIALOG" | Integer |
| "_PRINT_ROW_DATA" | No return value |
| "_PRINT_TABLE" | Integer |
| "_PRINT_TABLE_COLUMNHEADER" | No return value |
| "_PRINT_TEXT_VECTOR" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SET_DIALOG_PRINT_SCALE" | No return value |
| "_SET_MARGINS" | No return value |
| "_SET_PAGEFOOTER" | No return value |
| "_SET_PAGEHEADER" | No return value |
| "_SETUP_PRINTER" | Integer |

35.2 Details

35.2.1 Action Attributes

_GOT_FOCUS_NOTIFY

Setting this attribute specifies the action method GOT_FOCUS_NOTIFY.

Data type: Text vector

_KEY_PRESSED

Setting this attribute specifies the action method KEY_PRESSED

Data type: Text vector

_LOST_FOCUS_NOTIFY

Setting this attribute specifies the action method LOST_FOCUS_NOTIFY.

Data type: Text vector

35.2.2 Action Methods

GOT_FOCUS_NOTIFY

This action method is executed immediately after a dialog has got input focus.

Data type to be returned: No return value

KEY_PRESSED

This action method is executed after a key or a key combination is pressed in the dialog that has keyboard input focus. Four arguments are passed to the method: Key (text representation of the key), Alt_Enabled (boolean), Ctrl_Enabled (boolean), Shift_Enabled (boolean).

Data type to be returned: Boolean

LOST_FOCUS_NOTIFY

This action method is executed immediately after a dialog has lost input focus.

Data type to be returned: No return value

35.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_JOB_CONFIRMATION_ENABLED

Specifies whether the print job confirmation dialog is used or not. Some platforms may simply confirm that the printer's attributes reflect the user's preferences.



In the current version this attribute is always TRUE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_LANDSCAPE_ENABLED

Specifies whether the print page orientation is landscape or not.



_LANDSCAPE_ENABLED setting does not have any effect when printing with F3 key.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_LEADER_DIALOG_ITEM

Like the _LEADER_DIALOG attribute, the _LEADER_DIALOG_ITEM attribute defines a leader dialog for the given window. The leader dialog is the one that holds the item specified. Setting this attribute to empty text unsets the leader dialog for the given window. For more information about leader-follower dialogs relationship, see the description of the _LEADER_DIALOG attribute.

This attribute can be used, for example, in a picture methods, setting picture container as the leader dialog item. This is the only way to raise a new dialog from a picture context and set the picture holding dialog as leader because the holding dialog is not accessible directly from the picture context.

Example:

```
; in a picture container context:  
.set someDialog._LEADER_DIALOG_ITEM = "ROOT"  
;  
; ROOT is the picture container
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#)

_PAGE_GEOMETRY

Returns the page rectangle of the default printer.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Environment dependent |
| Access: | Read-only |

_PAPER_GEOMETRY

Returns the paper rectangle of the default printer.

| | |
|----------------|--|
| Description: | A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| Data type: | List |
| Default value: | Environment dependent |
| Access: | Read-only |

_PRINT_PAGE_RANGE_ENABLED

When this attribute is TRUE, the Pages selection on Print Job Confirmation dialog is enabled if supported by the operating system.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT_WRAP_ENABLED

Specifies whether text wrap is enabled or not. When this attribute is set to TRUE, the lines that are wider than the paper width are wrapped to the next line.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

35.2.4 Predefined Methods

_CLOSE_PRINT_SESSION

This method should be called after calling _PRINT_DIALOG and/or _PRINT_TEXT_VECTOR if the user has previously called _OPEN_PRINT_SESSION. _CLOSE_PRINT_SESSION starts the printing of current documents on the printer. If the user has sent documents to the printer by _PRINT_DIALOG or _PRINT_TEXT_VECTOR, those are not printed until _CLOSE_PRINT_SESSION is called.

Return value data type: Integer

_CLOSE_PRINTER

This method should be called after calling _PRINT_DIALOG and/or _PRINT_TEXT_VECTOR if the user has previously called _OPEN_PRINTER. _CLOSE_PRINTER starts the printing of current documents on the printer. If the user has sent documents to the printer by _PRINT_DIALOG or _PRINT_TEXT_VECTOR, those are not printed until calling _CLOSE_PRINTER. If the printer is not open, FALSE is returned.

Return value data type: Boolean

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_DIALOG_PRINT_SCALE

Returns the scale ratios (width/height) which are used when printing a dialog.

Return value data type: List

_GET_MARGINS

Returns the margins used in default printer.

Return value data type: List

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_OPEN_PRINT_SESSION

This method should be called prior to calling _PRINT_DIALOG or _PRINT_TEXT_VECTOR when wanting to print several dialogs or text vectors in one print session. If _JOB_CONFIRMATION_ENABLED == TRUE, _OPEN_PRINT_SESSION causes a job confirmation dialog to open if supported by the operating system.

_OPEN_PRINTER

This method should be called prior to calling _PRINT_DIALOG or _PRINT_TEXT_VECTOR when wanting to print several dialogs or text vectors in one print session. If _JOB_CONFIRMATION_ENABLED == TRUE, _OPEN_PRINTER causes a job confirmation dialog to open if supported by the operating system. If the printer is already opened, FALSE is returned and no confirmation dialog is opened.



Currently, _JOB_CONFIRMATION_ENABLED is always TRUE. If there is no local printer defined on VS Monitor, the SCIL error status 461 (SCIL_EXCEPTION_IN_METHOD_CALL) is generated.

Return value data type: Boolean

_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialog's relative place compared to a given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies the kind of container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using **PLACE_CENTER** the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
    ("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM,  
    ("PLACE_CENTER"), ("PLACE_CENTER"))
```

_PRINT_DIALOG

Prints the content of a dialog to the default printer. Note that only the specified dialog is printed, not the overlapping windows. If the user has pressed cancel on the job confirmation dialog, status code PRINT_JOB_NOT_CONFIRMED is returned. Zero (0) is returned, if the method was executed successfully.

Return value data type:

Integer

Parameter(s):

DIALOG Dialog

reference

A reference to a dialog object.



If there is no local printer defined on VS Monitor, the SCIL error status 461 (SCIL_EXCEPTION_IN_METHOD_CALL) is generated.

_PRINT_ROW_DATA

This function is used to set the row contents of a table.

Return value data type:

No return value

Parameter(s):

TEXT_VECTOR

Vector

A vector containing only text elements.

_PRINT_TABLE

Prints the table contents passed to a default printer according to the data set by the methods like _PRINT_ROW_DATA, _PRINT_TABLE_COLUMNHEADER, _SET_PAGEHEADER, and _SET_PAGEFOOTER. This function should be called only after setting all data by the previously mentioned functions.

Return value data type: Integer



If the table contents are not set or if there is any invalid or insufficient data then the SCIL error status 600 (SCIL_INVALID_VECTOR_LENGTH) is generated. If the Harmonized Print dialog library, ReportPrint.dll is not found then the SCIL error status 5007 (FILE_DOES_NOT_EXIST) is generated.

_PRINT_TABLE_COLUMNHEADER

This function is used to set the column names of the table to be printed. If the table has two rows of column name, call the function twice with the respective rows text as the vector.

Return value data type:

No return value

Parameter(s):

TEXT_VECTOR

Vector

A vector containing only text elements.

_PRINT_TEXT_VECTOR

Prints a passed text vector to a default printer. Zero (0) is returned, if the method was executed successfully.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s): | |
| TEXT_VECTOR | Vector A vector containing only text elements. |



If the harmonized print dialog library, ReportPrint.dll is not found then the SCIL error status 5007 (FILE_DOES_NOT_EXIST) is generated.

_QUEUE_FOR_EXECUTION

For more information on this item, see [Section 3](#).

_SET_DIALOG_PRINT_SCALE

Sets the scale ratios (width/height) that are used when printing a dialog.

| | |
|-------------------------|-----------------------------|
| Return value data type: | No return value |
| Parameter(s): | |
| WIDTH_RATIO | Integer An integer value |

HEIGTH_RATIO

| | |
|--------------|-----------------------------|
| HEIGTH_RATIO | Integer An integer value |
|--------------|-----------------------------|

_SET_MARGINS

Sets the margins used in the default printer. If _SET_MARGINS method is not used, the printer default margins are used.

| | |
|-------------------------|-----------------------------|
| Return value data type: | No return value |
| Parameter(s): | |
| TOP | Integer An integer value |
| BOTTOM | Integer An integer value |
| LEFT | Integer An integer value |
| RIGHT | Integer An integer value |

_SET_PAGEFOOTER

This function is used to set the footer items of a page.

| | |
|-------------------------|-----------------|
| Return value data type: | No return value |
| Parameter(s): | |
| COUNT | Text |
| MODE | Text |

_SET_PAGEHEADER

This function is used to set the header items of a page.

| | |
|-------------------------|-----------------|
| Return value data type: | No return value |
| Parameter(s): | |
| PAGE | Text |
| APPLICATION | Text |
| TOOL | Text |

_SETUP_PRINTER

Opens a setup printer dialog. Status code PRINT_SETUP_NOT_CONFIRMED is returned in case the user has pressed cancel on the setup confirmation dialog. Zero (0) is returned, if the method was executed successfully.

Return value data type: Integer



If no local printer defined on VS Monitor, the SCIL error status 461 (SCIL_EXCEPTION_IN_METHOD_CALL) is generated.

Section 36 VS_PROGRAM_EDITOR

36.1 Overview

The VS_PROGRAM_EDITOR object is a SCIL sensitive text editor that can be used to edit SCIL programs or text files. The text that is to be edited is inserted into the editor through the _PROGRAM attribute and read from the same attribute at any point of time. The _PROGRAM_IS_MODIFIED attribute indicates whether or not the text has been modified. The CLOSE_NOTIFY action method automatically runs the MENU_EXIT action method.

36.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_MENU_ABOUT" | Text vector |
| "_MENU_EXIT" | Text vector |
| "_MENU_EXPORT" | Text vector |
| "_MENU_IMPORT" | Text vector |
| "_MENU_NEW" | Text vector |
| "_MENU_OPEN" | Text vector |
| "_MENU_PRINT" | Text vector |
| "_MENU_PRINT_SETUP" | Text vector |
| "_MENU_SAVE" | Text vector |
| "_MENU_SAVE_AS" | Text vector |
| "_MENU_UPDATE" | Text vector |
| "NOTIFY" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

36.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "MENU_ABOUT" | No return value |
| "MENU_EXIT" | No return value |
| "MENU_EXPORT" | No return value |
| "MENU_IMPORT" | No return value |
| "MENU_NEW" | No return value |
| "MENU_OPEN" | No return value |
| "MENU_PRINT" | No return value |
| "MENU_PRINT_SETUP" | No return value |
| "MENU_SAVE" | No return value |
| "MENU_SAVE_AS" | No return value |
| "MENU_UPDATE" | No return value |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

36.1.3 Predefined Attributes

| Name | Type |
|------------------------------|------------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "BUSY" | Boolean |
| "CHILD_OBJECTS" | Text vector |
| "FILE_REVISION" | Text |
| "FOCUSABLE" | Boolean |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "ICON" | Image reference |
| "ICONIFIED" | Boolean |
| "INTERNAL_FILE_MENU_ENABLED" | Boolean |
| "LEADER_DIALOG" | Dialog reference |
| "MAX_HEIGHT" | Integer |
| "MAX_WIDTH" | Integer |
| "MAXIMIZED" | Boolean |
| "MIN_HEIGHT" | Integer |
| "MIN_WIDTH" | Integer |
| "MODAL" | Boolean |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "OPEN" | Boolean |
| "PROGRAM" | Vector |
| "PROGRAM_FONT" | Font |
| "PROGRAM_IS_MODIFIED" | Boolean |
| "SCIL_GEOMETRY" | List |
| "SELECTED_PROGRAM_TEXT" | Vector |
| "SOURCE_FILE_NAME" | Text |
| "STYLE" | Vector |
| "TITLE" | Text |
| "TOOLTIP_TEXT" | Text |
| "VARIABLE_NAMES" | Text vector |

36.1.4 Predefined Methods

| Name | Type |
|----------------------------|-----------------|
| "CLEAR_MODIFIED_FLAG" | No return value |
| "FLAG_FOR_EXECUTION" | No return value |
| "GET_HELP" | No return value |
| "GET_POINTER_POS" | No return value |
| "GET_SELECTED_PROGRAM_POS" | List |

Table continues on next page

| | |
|---|-----------------|
| <code>"_GET_STRING_WIDTH"</code> | Integer |
| <code>"_PLACE"</code> | No return value |
| <code>"_PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY"</code> | No return value |
| <code>"_PLACE_ACCORDING_TO_GEOMETRY"</code> | No return value |
| <code>"_PLACE_ACCORDING_TO_SCIL_GEOMETRY"</code> | No return value |
| <code>"_QUEUE_FOR_EXECUTION"</code> | No return value |
| <code>"_RAISE"</code> | No return value |
| <code>"_SET_MAX_SIZE"</code> | No return value |
| <code>"_SET_MIN_SIZE"</code> | No return value |

36.2 Details

36.2.1 Action Attributes

MENU_ABOUT

Setting this attribute specifies the action method MENU_ABOUT.

Data type: Text vector

MENU_EXIT

Setting this attribute specifies the action method MENU_EXIT.

Data type: Text vector

MENU_EXPORT

Setting this attribute specifies the action method MENU_EXPORT.

Data type: Text vector

MENU_IMPORT

Setting this attribute specifies the action method MENU_IMPORT.

Data type: Text vector

MENU_NEW

Setting this attribute specifies the action method MENU_NEW.

Data type: Text vector

MENU_OPEN

Setting this attribute specifies the action method MENU_OPEN.

Data type: Text vector

MENU_PRINT

Setting this attribute specifies the action method MENU_PRINT.

Data type: Text vector

_MENU_PRINT_SETUP

Setting this attribute specifies the action method MENU_PRINT_SETUP.

Data type: Text vector

_MENU_SAVE

Setting this attribute specifies the action method MENU_SAVE.

Data type: Text vector

_MENU_SAVE_AS

Setting this attribute specifies the action method MENU_SAVE_AS.

Data type: Text vector

_MENU_UPDATE

Setting this attribute specifies the action method MENU_UPDATE.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

36.2.2 Action Methods

MENU_ABOUT

This action method is executed when **About** has been chosen from the menu.

Data type to be returned: No return value

MENU_EXIT

This action method is executed when **Exit** has been chosen from the menu.

Data type to be returned: No return value

MENU_EXPORT

This action method is executed when **Export** has been chosen from the menu.

Data type to be returned: No return value

MENU_IMPORT

This action method is executed when **Import** has been chosen from the menu.

Data type to be returned: No return value

MENU_NEW

This action method is executed when **New** has been chosen from the menu.

Data type to be returned: No return value

MENU_OPEN

This action method is executed when **Open** has been chosen from the menu.

Data type to be returned: No return value

MENU_PRINT

This action method is executed when **Print** has been chosen from the menu.

Data type to be returned: No return value

MENU_PRINT_SETUP

This action method is executed when **Print Setup** has been chosen from the menu.

Data type to be returned: No return value

MENU_SAVE

This action method is executed when **Save** has been chosen from the menu.

Data type to be returned: No return value

MENU_SAVE_AS

This action method is executed when **Save As** has been chosen from the menu.

Data type to be returned: No return value

MENU_UPDATE

This action method is executed when **Update** has been chosen from the menu.

Data type to be returned: No return value

NOTIFY

Specifies the notification procedure of an object. The notification procedure gets called for different reasons with different objects. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

36.2.3 Predefined Attributes**_ATTRIBUTE_NAMES**

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

_BUSY

Setting the _BUSY attribute to TRUE changes the appearance of the window to indicate that the window is unavailable to the user. On Windows platform, the cursor is displayed as an hour glass. This function is normally used to indicate that a long operation is about to occur. This attribute is automatically set to FALSE when the long operation is finished.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Designates whether a window should receive keyboard focus or not. Keyboard focus policy is dependent on the window system, this function only indicates that the window has use for key events. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_GEOMETRY

For more information, see [Section 3](#).

_ICON

Specifies the icon used in the title of a window or when a window is iconified. Note that the _ICON attribute must be set from at least one object hierarchy level upper than the window itself, otherwise the delete method of the window is not allowed. The text assigned into attribute contains the image reference.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.SET MY_DIALOG._ICON = "ROOT\ IMAGE1"
```

_ICONIFIED

Designates whether a window is iconified or not. The exact implication of iconification depends on the window system and the window manager in use. The window manager lets the user manipulate rectangular, overlapping display regions known as windows. Window manager windows correspond to the top-level windows of the underlying window system (X Window System and Microsoft Windows, for example).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_INTERNAL_FILE_MENU_ENABLED

Specifies whether or not to use the internal file menu that does not contain certain menu items such as **New**, **Open**, **Save** and **Save As**.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Write-only |

_LEADER_DIALOG

Defines a leader-follower relationship between dialogs. When a dialog's leader is opened, closed, iconified, uniconified, blocked, or unblocked, any follower dialog receive the same message as a side effect. The exact implications of this designation are window system dependent. Typically, dialogs with the same leader are placed in a group by the window system. If either dialog is destroyed, the other dialog's reference to it is deleted automatically. The _LEADER_DIALOG attribute of a dialog must be set before the dialog is opened.

Dialogs also inherit attributes from their leaders if the attributes are not specified in the dialog itself or its class. For example, _ICON, _FOREGROUND_COLOR, _BACKGROUND_COLOR and _FONT are inherited.

| | |
|----------------|---------------------------------|
| Description: | A reference to a dialog object. |
| Data type: | Dialog reference |
| Default value: | Empty text |
| Access: | Write-only |

_MAX_HEIGHT

For more information, see [Section 3](#).

_MAX_WIDTH

For more information, see [Section 3](#).

_MAXIMIZED

Setting the _MAXIMIZED attribute to TRUE changes the size of a window so that it encloses its maximum size or the entire display, whichever is smaller. The window's current position and size are saved for later restoration by setting the attribute to FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_MIN_HEIGHT

For more information, see [Section 3](#).

_MIN_WIDTH

For more information, see [Section 3](#).

_MODAL

Designates whether or not a window should grab pointer and key input from other windows when it is opened. When the _MODAL attribute is TRUE, it is impossible to activate another window as long as the MODAL window is open.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN

Setting the _OPEN attribute to TRUE opens the window so that it is visible at its current position and size on the display. Setting the attribute to FALSE closes but does not delete the window.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_PROGRAM

A text vector containing the program or text that is going to be edited or the already edited program. This attribute can be read or written at any time during the editing of the text. Note that the maximum number of elements in text vector is 10 000.

| | |
|----------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Default value: | Empty vector |
| Access: | No limitations |

_PROGRAM_FONT

Specifies the font that is being used to display the program that is being edited.

| | |
|----------------|-----------------------------------|
| Description: | A font as referenced within SCIL. |
| Data type: | Font |
| Default value: | Environment dependent |
| Access: | No limitations |

_PROGRAM_IS_MODIFIED

A boolean flag that indicates if the program that is being edited has been changed. This flag should be checked, for example, before saving a program. The flag is cleared from SCIL using the method _CLEAR_MODIFIED_FLAG. The flag is not used internally by program editor. The program editor only sets the flag to TRUE when changes have been made to the text that is being edited.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Read-only |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_PROGRAM_TEXT

A text vector containing the program or text that is selected in the program editor. This attribute can be read or written at any time during the editing of the text. Note that the maximum number of elements in text vector is 10 000.

| | |
|--------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Access: | Read-only |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_STYLE

The style flags determine what decorations, if any, are placed around the window's border, as well as how the user can directly manipulate the window. Note that the visual effect of setting

style flags is platform dependent and not all platforms can distinguish all combinations of style flags. The best way to use style flags is to first obtain the original set of flags from the dialog and then append or delete the desired flags from this set.



The **STYLE_BACKGROUND** and **STYLE_MAIN** flags has to always be set.
STYLE_BACKGROUND causes the dialog to update its background.

STYLE_BACKGROUND

This attribute sets the opacity of a window. Without the **STYLE_BACKGROUND** attribute the window will contain a border along with header information, and the Iconifiable, Maximize & Unclosable icons.

The attribute can be set as follows:

```
set._style = vector("STYLE_BACKGROUND")
```

STYLE_DOCUMENT

This attribute has no effect on the windows platform.

The visual effect of setting style flags is platform dependent. Not all platforms can distinguish all combinations of style flags.

Description:

Any combination of the following style flags given as a vector containing text strings. The flags are:
STYLE_BACKGROUND - An opaque background
STYLE_BORDER - A border around the content area
STYLE_MOVABLE - A title bar or other draggable area
STYLE_RESIZABLE - Resize handles or a grow box
STYLE_ICONIFIABLE - An iconify/minimize box
STYLE_MAXIMIZABLE - A maximize/zoom box
STYLE_MAIN - The top-level, non-document window
STYLE_DOCUMENT - Window contains a document
STYLE_CONFIRM - Window is a confirmation dialog
STYLE_UNCLOSABLE - No close box
STYLE_PALETTE - Window is a Tool window

Data type:

Vector

Default value:

("**STYLE_BACKGROUND**", "**STYLE_BORDER**",
"**STYLE_ICONIFIABLE**", "**STYLE_MAIN**",
"**STYLE_MAXIMIZABLE**", "**STYLE_MOVABLE**",
"**STYLE_RESIZABLE**")

Access:

No limitations

_TITLE

Specifies the title of a window. Note that the title may not be visible under certain circumstances, for example if the window or dialog does not have a title bar.

Description:

A character string containing up to 255 characters.

Data type:

Text

Default value:

Empty text

Access:

No limitations

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

36.2.4 Predefined Methods

_CLEAR_MODIFIED_FLAG

Clears the _PROGRAM_IS_MODIFIED flag. This flag should be cleared, for example, when a new program is being edited and when the program has been saved. The flag is not used internally by SCIL Program Editor. The SCIL Program Editor only sets the flag to TRUE when changes have been made to the text that is being edited. Note that when attaching text into SCIL Program Editor by assigning text vector into _PROGRAM attribute, the flag is automatically set to TRUE. Due to this, the _CLEAR_MODIFIED_FLAG method should be used after the text has been placed into SCIL Program Editor.

Return value data type: No return value

Example:

```
.CREATE ROOT\ MY_PROGRAM_EDITOR = VS_PROGRAM_EDITOR(-  
_PROGRAM = %My_Program,-  
_OPEN = TRUE)  
ROOT\ MY_PROGRAM_EDITOR._CLEAR_MODIFIED_FLAG
```

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_POINTER_POS

Returns a list containing the attribute names X and Y. X and Y are integers referring to the current cursor position. The coordinates are given relative to the coordinate system of the object that is executing the method call.

Return value data type: No return value

_GET_SELECTED_PROGRAM_POS

Returns a list containing the positions of the selected part of a program. The list contains the items START_LINE, START_COLUMN, END_LINE and END_COLUMN.

Return value data type: List

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_PLACE

Positions a dialog's relative place compared to another dialog. The BASE parameter specifies the dialog to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

- PLACE_NEAR = Completely outside with a small margin
- PLACE_ADJACENT = Completely outside with no margin
- PLACE_STRADDLE = Half inside, half outside
- PLACE_ALIGN = Completely inside with no margin
- PLACE_INSET = Completely inside with a small margin
- PLACE_HALF = Opposite side touches centre of base dialog
- PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER, the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| BASE | Dialog reference A reference to a dialog type object. A dialog can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_LEFT", "PLACE_ALIGN"),  
("PLACE_TOP",
```

```
"PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_DIALOG:

```
MY_DIALOG._PLACE("YOUR_DIALOG", ("PLACE_CENTER"), ("PLACE_CENTER"))
```

PLACE_ACCORDING_TO_DIALOG_ITEM_GEOMETRY

Positions a dialog's relative place compared to the given rectangle in the coordinate system of a dialog item. The BASE and RECT parameters specify the dialog item and the rectangle to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to BASE on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

Usually, BASE specifies a container (VS_CONTAINER, VS_NOTEBOOK_PAGE, etc.), and RECT is the geometry of a direct child item of that container. However, it is possible to specify any dialog item as BASE, and any rectangle as RECT.

The following flags indicate how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE= Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET= Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of base dialog

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centering.

| | |
|-------------------------|-----------------|
| Return value data type: | No return value |
|-------------------------|-----------------|

| | |
|--------------|--|
| Parameter(s) | |
|--------------|--|

| | |
|------|--|
| BASE | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
|------|--|

Table continues on next page

RECT**List**

A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system.

X_FLAGS**Vector**

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Y_FLAGS**Vector**

Any combination of the following style flags given as a vector containing text strings. The flags are:
PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT,
PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE,
PLACE_ALIGN, PLACE_INSET, PLACE_HALF,
PLACE_CENTER and PLACE_TOP.

Example:

Example 1: Placing MY_DIALOG at TOP/LEFT of the rectangle (0, 0, 100, 100) within YOUR_CONTAINER:

```
MY_DIALOG._PLACE("YOUR_CONTAINER", LIST(X=0, Y=0, W=100, H=100),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

Example 2: Centering MY_DIALOG relative to YOUR_CONTAINER itself:

```
@GEOM = YOUR_CONTAINER._GEOMETRY  
#MODIFY GEOM:V = LIST(X = 0, Y = 0)  
  
MY_DIALOG._PLACE("YOUR_CONTAINER", %GEOM, ("PLACE_CENTER"),  
("PLACE_CENTER"))
```

_PLACE_ACCORDING_TO_GEOMETRY

Positions a dialog's relative place compared to a given rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the Visual SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

```
1: Placing MY_DIALOG at TOP/LEFT of RECT:
MY_DIALOG._PLACE_ACCORDING_TO_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_PLACE_ACCORDING_TO_SCIL_GEOMETRY

Positions a dialog's relative place compared to a given SCIL rectangle. The RECT parameter specifies the coordinates to be used as base for the positioning. The parameters X_FLAGS and Y_FLAGS specify the relation to RECT on the different coordinate axes. The following flags specify which side is used when determining the new coordinates: PLACE_LEFT, PLACE_RIGHT, PLACE_TOP and PLACE_BOTTOM.

The following flags indicates how to locate the side of the dialog specified above:

PLACE_NEAR = Completely outside with a small margin

PLACE_ADJACENT = Completely outside with no margin

PLACE_STRADDLE = Half inside, half outside

PLACE_ALIGN = Completely inside with no margin

PLACE_INSET = Completely inside with a small margin

PLACE_HALF = Opposite side touches centre of given rectangle

PLACE_CENTER = Centered inside

The flags above should be used together in a vector so that one value specifies which side to align and another value how to align the side. When using PLACE_CENTER the side specification can be left out because both sides are automatically used when centring.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| RECT | List A list containing the attributes X, Y, W, H. The attributes describe coordinates within the SCIL coordinate system. |
| X_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |
| Y_FLAGS | Vector Any combination of the following style flags given as a vector containing text strings. The flags are: PLACE_BOTTOM, PLACE_LEFT, PLACE_RIGHT, PLACE_NEAR, PLACE_ADJACENT, PLACE_STRADDLE, PLACE_ALIGN, PLACE_INSET, PLACE_HALF, PLACE_CENTER and PLACE_TOP. |

Example:

1: Placing MY_DIALOG at TOP/LEFT of RECT:

```
MY_DIALOG._PLACE_ACCORDING_TO_SCIL_GEOMETRY(LIST(X=200,Y=200,W=0,H=0),  
("PLACE_LEFT", "PLACE_ALIGN"), ("PLACE_TOP", "PLACE_ALIGN"))
```

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_RAISE

This method raises a dialog to the top of all other windows.

Return value data type: No return value

_SET_MAX_SIZE

For more information, see [Section 3](#).

_SET_MIN_SIZE

For more information, see [Section 3](#).

Section 37 VS_SCROLL_BAR

37.1 Overview

A VS_SCROLL_BAR is a control object that allows the user to drag the scroll box over a linear range of values in the scroll bar. The NOTIFY action method is called whenever the scroll box is moved either by dragging or pressing the scroll arrows.

37.1.1 Action Attributes

| Name | Type |
|---------------------|-------------|
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |

37.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

37.1.3 Predefined Attributes

| Name | Type |
|-------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_INCREMENT_VALUE" | Integer |
| "_MAX_VALUE" | Integer |
| "_MIN_VALUE" | Integer |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_PAGE_INCREMENT_VALUE" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VALUE" | Integer |

Table continues on next page

| | |
|---------------------------|-------------|
| " <u>VARIABLE_NAMES</u> " | Text vector |
| " <u>VERTICAL</u> " | Boolean |
| " <u>VIEW_VALUE</u> " | Integer |
| " <u>VISIBLE</u> " | Boolean |

37.1.4 Predefined Methods

| Name | Type |
|--------------------------------|-----------------|
| " <u>FLAG_FOR_EXECUTION</u> " | No return value |
| " <u>GET_HELP</u> " | No return value |
| " <u>GET_STRING_WIDTH</u> " | Integer |
| " <u>QUEUE_FOR_EXECUTION</u> " | No return value |

37.2 Details

37.2.1 Action Attributes

NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

37.2.2 Action Methods

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

37.2.3 Predefined Attributes

ATTRIBUTE NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3s](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_INCREMENT_VALUE

Indicates the minor increment step size, for example, for a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._INCREMENT_VALUE = 5
```

_MAX_VALUE

Indicates the maximum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MAX_VALUE = 10
```

_MIN_VALUE

Indicates the minimum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MIN_VALUE = 1
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_PAGE_INCREMENT_VALUE

Indicates the major increment step size, for example, for a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._PAGE_INCREMENT_VALUE = 5
```

_SCIL_GEOmetry

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE

Indicates the current value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VALUE = 1
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL

Indicates that the control object lies along the Y axis instead of along the X axis. Note that this attribute is not relevant to all kinds of control objects.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VERTICAL = TRUE
```

_VIEW_VALUE

Indicates the visible area of the external object that a control object is controlling. This attribute is relevant, for example, when using a scroll bar to scroll text within a text field. For some look-and-feels, the size of the thumb of the scroll bar indicates the relative size of the document that is visible.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

37.2.4

Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 38 VS_SEPARATOR_MENU_ITEM

38.1 Overview

A VS_SEPARATOR_MENU_ITEM is a special kind of menu item that has no command attached to it. It acts as a separator within the menu, displayed as a horizontal line. The VS_SEPARATOR_MENU_ITEM is used to group related menu items. When using VS_TOGGLE_MENU_ITEMS with the _EXCLUSIVE attribute set to TRUE, the VS_SEPARATOR_MENU_ITEM also defines the scope of the VS_TOGGLE_MENU_ITEMS that are mutually exclusive. This means that only one menu item can be selected at a time.

38.1.1 Predefined Attributes

| Name | Type |
|---------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_FILE_REVISION" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SOURCE_FILE_NAME" | Text |
| "_VARIABLE_NAMES" | Text vector |

38.1.2 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |

38.2 Details

38.2.1 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_FILE_REVISION

For more information, see [Section 3](#).

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_VARIABLE_NAMES

For more information, see [Section 3](#).

38.2.2 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 39 VS_SLIDER

39.1 Overview

A VS_SLIDER is a control object that allows the user to drag over a linear range of values.

39.1.1 Action Attributes

| Name | Type |
|--------------------|-------------|
| "_NOTIFY" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

39.1.2 Action Methods

| Name | Type |
|--------------------|-----------------|
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

39.1.3 Predefined Attributes

| Name | Type |
|------------------------|-------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "CLICK_FOCUSABLE" | Boolean |
| "ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FOCUSABLE" | Boolean |
| "FOCUSED" | Boolean |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "INCREMENT_VALUE" | Integer |
| "MAX_VALUE" | Integer |
| "MIN_VALUE" | Integer |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "PAGE_INCREMENT_VALUE" | Integer |
| "SCIL_GEOMETRY" | List |
| "SOURCE_FILE_NAME" | Text |
| "TOOLTIP_TEXT" | Text |
| "VALUE" | Integer |
| "VARIABLE_NAMES" | Text vector |

Table continues on next page

| | |
|---------------|---------|
| "_VERTICAL" | Boolean |
| "_VIEW_VALUE" | Integer |
| "_VISIBLE" | Boolean |

39.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |

39.2 Details

39.2.1 Action Attributes

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

39.2.2 Action Methods

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

39.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_INCREMENT_VALUE

Indicates the minor increment step size, for example, for a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._INCREMENT_VALUE = 5
```

_MAX_VALUE

Indicates the maximum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MAX_VALUE = 10
```

_MIN_VALUE

Indicates the minimum value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._MIN_VALUE = 1
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_PAGE_INCREMENT_VALUE

Indicates the major increment step size, for example, for a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._PAGE_INCREMENT_VALUE = 5
```

_SCIL_GEOmetry

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VALUE

Indicates the current value of a control object.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VALUE = 1
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL

Indicates that a control object lies along the Y axis instead of along the Xaxis. Note that this attribute is not relevant to all control objects.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_CONTROL._VERTICAL = TRUE
```

_VIEW_VALUE

Indicates the visible area of the external object that a control object is controlling. This attribute is relevant, for example, when using a scroll bar to scroll text within a text field. For some look-and-feels, the size of the thumb of the scroll bar indicates the relative size of the document that is visible.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

Table 2:

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

39.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 40 VS_SPLIT_BAR

40.1 Overview

The split bar is an object meant to be used when the user has to be able to dynamically split an area into two. The split bar is placed between, for example, two containers. It contains the functionality needed to make it possible for the user to grab the split bar with the mouse cursor and resize the two objects that are connected to the sides of the split bar. Note that only one of the objects connected to the split bar may have an internal spring connection at most. The NOTIFY action method is executed every time the split bar is interactively moved by the user.

40.1.1 Action Attributes

| Name | Type |
|------------------------------------|-------------|
| "TOOLTIP_IS_SHOWN" | Text vector |

40.1.2 Action Methods

| Name | Type |
|------------------------------------|-----------------|
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

40.1.3 Predefined Attributes

| Name | Type |
|-------------------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_GEOMETRY" | List |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POSITION" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL" | Boolean |

40.1.4 Predefined Methods

| Name | Type |
|--------------------------------|-----------------|
| " <u>FLAG FOR EXECUTION</u> " | No return value |
| " <u>GET HELP</u> " | No return value |
| " <u>GET STRING WIDTH</u> " | Integer |
| " <u>QUEUE FOR EXECUTION</u> " | No return value |

40.2 Details

40.2.1 Action Attributes

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

40.2.2 Action Methods

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

40.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3s](#).

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POSITION

By using this attribute, it is possible to programmatically set a default value for the split bar. The value is the X or Y coordinate of the centre of the VS_SPLIT_BAR in the coordinate space of the object that contains the split bar.

| | |
|--------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |

Access: No limitations

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over it.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL

Specifies whether the split bar splits the objects vertically or horizontally.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

40.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 41 VS_SUB_MENU_ITEM

41.1 Overview

The VS_SUB_MENU_ITEM behaves like the VS_MENU_ITEM except for the fact that it can contain a sub-menu that is an ordinary VS_MENU.

41.1.1 Predefined Attributes

| Name | Type |
|---------------------|-----------------|
| "_ATTRIBUTE_NAMES" | List |
| "_CHILD_OBJECTS" | Text vector |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_IMAGE" | Image reference |
| "_MNEMONIC" | Text |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SOURCE_FILE_NAME" | Text |
| "_TITLE" | Text |
| "_VARIABLE_NAMES" | Text vector |

41.1.2 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_QUEUE_FOR_EXECUTION" | No return value |

41.2 Details

41.2.1 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a menu item is enabled or not. Changing this attribute also affects the appearance of the menu item.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_IMAGE

Specifies an image that is associated with the object. The way the image is displayed depends on the object. The image is given as an object reference to the image that has been loaded or created as an independent object.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. The size of the image should preferably be 16 % 16. |
| Data type: | Image reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ IMAGE1 = VS_IMAGE("...  
.SET ROOT\ MY_SUB_MENU_ITEM._IMAGE = "ROOT\ IMAGE1"
```

_MNEMONIC

Specifies the mnemonic character of a menu item. A change of appearance is also often combined with this attribute. In Microsoft Windows platform, the setting of this attribute underlines the mnemonic character in menu item.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a menu item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

41.2.2 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 42 VS_TABLE

42.1 Overview

The VS_TABLE object implements what most developers would refer to as a “spreadsheet widget” or a “grid”. Essentially, it’s a two dimensional list of data that can have header information for each row and/or each column of that list. The VS_TABLE objects can be used to display output information, but it can also be used to allow the user to enter new data or modify the displayed information. For more information about the VS_TABLE, see SYS600 Visual SCIL Application Design.



The VS_TABLE does not handle mouse motion events, if it is not inserted inside a dialog.

42.1.1 Action Attributes

| Name | Type |
|---|-------------|
| "CELL_EDITING_ACCEPTED" | Text vector |
| "CELL_EDITING_CANCELLED" | Text vector |
| "CELL_EDITING_STARTED" | Text vector |
| "COLUMN_TITLE_CLICKED" | Text vector |
| "COLUMN_TITLE_DOUBLE_CLICKED" | Text vector |
| "DRAGGED_AND_DROPPED" | Text vector |
| "FOCUSSED_CELL_CHANGED" | Text vector |
| "NOTIFY" | Text vector |
| "SELECTION_HAS_CHANGED" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

42.1.2 Action Methods

| Name | Type |
|---|-----------------|
| "CELL_EDITING_ACCEPTED" | No return value |
| "CELL_EDITING_CANCELLED" | No return value |
| "CELL_EDITING_STARTED" | No return value |
| "COLUMN_TITLE_CLICKED" | No return value |
| "COLUMN_TITLE_DOUBLE_CLICKED" | No return value |
| "DRAGGED_AND_DROPPED" | No return value |
| "FOCUSSED_CELL_CHANGED" | No return value |
| "SELECTION_HAS_CHANGED" | No return value |
| "TOOLTIP_IS_SHOWN TOOLTIP_IS_SHOWN" | No return value |

42.1.3 Predefined Attributes

| Name | Type |
|-----------------------------------|----------------|
| "ATTRIBUTE_NAMES" | List |
| "AUTO_EDIT_ENABLED" | Boolean |
| "AUTO_SCROLL_ENABLED" | Boolean |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "CLICK_FOCUSABLE" | Boolean |
| "COLUMN_COUNT" | Integer |
| "COLUMN_DRAG_AND_DROP_ENABLED" | Boolean |
| "COLUMN_HEADER_RESIZABLE" | Boolean |
| "COLUMN_HEADER_SELECTION_ENABLED" | Boolean |
| "DOUBLE_CLICK_EDIT_ENABLED" | Boolean |
| "DRAWING_SELECTION_ENABLED" | Boolean |
| "ENABLED" | Boolean |
| "EXTEND_COLUMN_SELECTION_ENABLED" | Boolean |
| "EXTEND_ROW_SELECTION_ENABLED" | Boolean |
| "FILE_REVISION" | Text |
| "FIRST_VISIBLE_COLUMN" | Integer |
| "FIRST_VISIBLE_ROW" | Integer |
| "FOCUSABLE" | Boolean |
| "FOCUSED" | Boolean |
| "FOCUSED_CELL" | Integer |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "GEOMETRY" | List |
| "GRABBING_HAND_ENABLED" | Boolean |
| "HAS_BORDER" | Boolean |
| "HAS_COLUMN_HEADER" | Boolean |
| "HAS_ROW_HEADER" | Boolean |
| "HORIZONTAL_GRID_ENABLED" | Boolean |
| "HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "KEYBOARD_SELECTION_ENABLED" | Boolean |
| "KEYBOARD_SELECTION_ORDER" | Text |
| "KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "POP_UP_MENU" | Menu reference |
| "ROW_COUNT" | Integer |
| "ROW_DRAG_AND_DROP_ENABLED" | Boolean |
| "ROW_HEADER_RESIZABLE" | Boolean |
| "ROW_HEADER_SELECTION_ENABLED" | Boolean |
| "ROW_HEADER_WIDTH" | Integer |

Table continues on next page

| | |
|--------------------------------|----------------------|
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_CELLS" | Integer |
| "_SELECTED_ROW" | Integer |
| "_SELECTION_POLICY" | Text |
| "_SELECTION_SENSE_ENABLED" | Boolean |
| "_SORT_ARROW_VISIBLE" | Boolean |
| "_SORT_CRITERIA" | Any valid SCIL value |
| "_SORTING_ENABLED" | Boolean |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_GRID_ENABLED" | Boolean |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

42.1.4 Predefined Methods

| Name | Type |
|--|-----------------|
| "_ACCEPT_EDIT" | No return value |
| "_ADD_COLUMN_GROUP" | No return value |
| "_ADD_COLUMNS" | No return value |
| "_ADD_ROW_GROUP" | No return value |
| "_ADD_ROWS" | No return value |
| "_CANCEL_EDIT" | No return value |
| "_DELETE_COLUMNS" | No return value |
| "_DELETE_ROWS" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_FREEZE_COLUMNS" | No return value |
| "_GET_CELL_BACKGROUND_COLOR" | Color |
| "_GET_CELL_FONT" | Font |
| "_GET_CELL_FOREGROUND_COLOR" | Color |
| "_GET_CELL_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_CELL_OBJECT_ID" | Text |
| "_GET_CELL_TEXT" | Text |
| "_GET_CELL_VERTICAL_JUSTIFICATION" | Text |
| "_GET_COLUMN_BACKGROUND_COLOR" | Color |
| "_GET_COLUMN_FONT" | Font |
| "_GET_COLUMN_FOREGROUND_COLOR" | Color |
| "_GET_COLUMN_GROUP_BACKGROUND_COLOR" | Color |
| "_GET_COLUMN_GROUP_COUNT" | Integer |
| "_GET_COLUMN_GROUP_FONT" | Font |
| "_GET_COLUMN_GROUP_FOREGROUND_COLOR" | Color |
| "_GET_COLUMN_GROUP_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_COLUMN_GROUP_OBJECT_ID" | Text |

Table continues on next page

| | |
|--|-----------------|
| "_GET_COLUMN_GROUP_TITLE" | Text |
| "_GET_COLUMN_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_COLUMN_OBJECT_ID" | Text |
| "_GET_COLUMN_TITLE" | Text |
| "_GET_COLUMN_TITLE_BACKGROUND_COLOR" | Color |
| "_GET_COLUMN_TITLE_FONT" | Font |
| "_GET_COLUMN_TITLE_FOREGROUND_COLOR" | Color |
| "_GET_COLUMN_TITLE_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_COLUMN_VERTICAL_JUSTIFICATION" | Text |
| "_GET_COLUMN_WIDTH" | Integer |
| "_GET_EDITED_CELL" | Integer |
| "_GET_HELP" | No return value |
| "_GET_ROW_BACKGROUND_COLOR" | Color |
| "_GET_ROW_FONT" | Font |
| "_GET_ROW_FOREGROUND_COLOR" | Color |
| "_GET_ROW_GROUP_BACKGROUND_COLOR" | Color |
| "_GET_ROW_GROUP_COUNT" | Integer |
| "_GET_ROW_GROUP_FONT" | Font |
| "_GET_ROW_GROUP_FOREGROUND_COLOR" | Color |
| "_GET_ROW_GROUP_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_ROW_GROUP_OBJECT_ID" | Text |
| "_GET_ROW_GROUP_TITLE" | Text |
| "_GET_ROW_HEIGHT" | Integer |
| "_GET_ROW_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_ROW_OBJECT_ID" | Text |
| "_GET_ROW_TITLE" | Text |
| "_GET_ROW_TITLE_BACKGROUND_COLOR" | Color |
| "_GET_ROW_TITLE_FONT" | Font |
| "_GET_ROW_TITLE_FOREGROUND_COLOR" | Color |
| "_GET_ROW_TITLE_HORIZONTAL_JUSTIFICATION" | Text |
| "_GET_ROW_VERTICAL_JUSTIFICATION" | Text |
| "_GET_STRING_WIDTH" | Integer |
| "_IS_CELL_READ_ONLY" | Boolean |
| "_IS_CELL_TEXT_WWRAPPED" | Boolean |
| "_IS_COLUMN_READ_ONLY" | Boolean |
| "_IS_COLUMN_RESIZABLE" | Boolean |
| "_IS_COLUMN_SELECTABLE" | Boolean |
| "_IS_COLUMN_SORTING_ENABLED" | Boolean |
| "_IS_COLUMN_TEXT_WWRAPPED" | Boolean |
| "_IS_ROW_READ_ONLY" | Boolean |
| "_IS_ROW_RESIZABLE" | Boolean |
| "_IS_ROW_SELECTABLE" | Boolean |
| "_IS_ROW_TEXT_WWRAPPED" | Boolean |
| "_QUEUE_FOR_EXECUTION" | No return value |

Table continues on next page

| | |
|--|-----------------|
| "_SCROLL_POSITION_INTO_VIEW" | No return value |
| "_SCROLL_SELECTION_INTO_VIEW" | No return value |
| "_SET_CELL_BACKGROUND_COLOR" | No return value |
| "_SET_CELL_EDIT_TYPE_TEXT" | No return value |
| "_SET_CELL_FONT" | No return value |
| "_SET_CELL_FOREGROUND_COLOR" | No return value |
| "_SET_CELL_HORIZONTAL_JUSTIFICATION" | No return value |
| "_SET_CELL_OBJECT_ID" | No return value |
| "_SET_CELL_READ_ONLY" | No return value |
| "_SET_CELL_TEXT" | No return value |
| "_SET_CELL_TEXT_RANGE" | No return value |
| "_SET_CELL_TEXT_WWRAPPED" | No return value |
| "_SET_CELL_VALUE_HANDLER" | No return value |
| "_SET_CELL_VERTICAL_JUSTIFICATION" | No return value |
| "_SET_COLUMN_BACKGROUND_COLOR" | No return value |
| "_SET_COLUMN_EDIT_TYPE_TEXT" | No return value |
| "_SET_COLUMN_FONT" | No return value |
| "_SET_COLUMN_FOREGROUND_COLOR" | No return value |
| "_SET_COLUMN_GROUP_BACKGROUND_COLOR" | No return value |
| "_SET_COLUMN_GROUP_FONT" | No return value |
| "_SET_COLUMN_GROUP_FOREGROUND_COLOR" | No return value |
| "_SET_COLUMN_GROUP_HORIZONTAL_JUSTIFICATION" | No return value |
| "_SET_COLUMN_GROUP_OBJECT_ID" | No return value |
| "_SET_COLUMN_GROUP_TITLE" | No return value |
| "_SET_COLUMN_HORIZONTAL_JUSTIFICATION" | No return value |
| "_SET_COLUMN_OBJECT_ID" | No return value |
| "_SET_COLUMN_READ_ONLY" | No return value |
| "_SET_COLUMN_SORTING" | No return value |
| "_SET_COLUMN_TEXT_WWRAPPED" | No return value |
| "_SET_COLUMN_TITLE" | No return value |
| "_SET_COLUMN_TITLE_BACKGROUND_COLOR" | No return value |
| "_SET_COLUMN_TITLE_FONT" | No return value |
| "_SET_COLUMN_TITLE_FOREGROUND_COLOR" | No return value |
| "_SET_COLUMN_TITLE_HORIZONTAL_JUSTIFICATION" | No return value |
| "_SET_COLUMN_TITLE_RESIZABLE" | No return value |
| "_SET_COLUMN_TITLE_SELECTABLE" | No return value |
| "_SET_COLUMN_VALUE_HANDLER" | No return value |
| "_SET_COLUMN_VERTICAL_JUSTIFICATION" | No return value |
| "_SET_COLUMN_WIDTH" | No return value |
| "_SET_ROW_BACKGROUND_COLOR" | No return value |
| "_SET_ROW_EDIT_TYPE_TEXT" | No return value |
| "_SET_ROW_FONT" | No return value |
| "_SET_ROW_FOREGROUND_COLOR" | No return value |
| "_SET_ROW_GROUP_BACKGROUND_COLOR" | No return value |

Table continues on next page

| | |
|---|-----------------|
| " <u>SET_ROW_GROUP_FONT</u> " | No return value |
| " <u>SET_ROW_GROUP_FOREGROUND_COLOR</u> " | No return value |
| " <u>SET_ROW_GROUP_HORIZONTAL_JUSTIFICATION</u> " | No return value |
| " <u>SET_ROW_GROUP_OBJECT_ID</u> " | No return value |
| " <u>SET_ROW_GROUP_TITLE</u> " | No return value |
| " <u>SET_ROW_HEIGHT</u> " | No return value |
| " <u>SET_ROW_HORIZONTAL_JUSTIFICATION</u> " | No return value |
| " <u>SET_ROW_OBJECT_ID</u> " | No return value |
| " <u>SET_ROW_READ_ONLY</u> " | No return value |
| " <u>SET_ROW_TEXT_WWRAPPED</u> " | No return value |
| " <u>SET_ROW_TITLE</u> " | No return value |
| " <u>SET_ROW_TITLE_BACKGROUND_COLOR</u> " | No return value |
| " <u>SET_ROW_TITLE_FONT</u> " | No return value |
| " <u>SET_ROW_TITLE_FOREGROUND_COLOR</u> " | No return value |
| " <u>SET_ROW_TITLE_HORIZONTAL_JUSTIFICATION</u> " | No return value |
| " <u>SET_ROW_TITLE_RESIZABLE</u> " | No return value |
| " <u>SET_ROW_TITLE_SELECTABLE</u> " | No return value |
| " <u>SET_ROW_VALUE_HANDLER</u> " | No return value |
| " <u>SET_ROW_VERTICAL_JUSTIFICATION</u> " | No return value |
| " <u>SORT</u> " | No return value |
| " <u>UNFREEZE_COLUMNS</u> " | No return value |

42.2 Details

42.2.1 Action Attributes

_CELL_EDITING_ACCEPTED

Setting this attribute specifies the action method CELL_EDITING_ACCEPTED.

Data type: Text vector

_CELL_EDITING_CANCELLED

Setting this attribute specifies the action method CELL_EDITING_CANCELLED.

Data type: Text vector

_CELL_EDITING_STARTED

Setting this attribute specifies the action method CELL_EDITING_STARTED

Data type: Text vector

_COLUMN_TITLE_CLICKED

Setting this attribute specifies the action method COLUMN_TITLE_CLICKED.

Data type: Text vector

_COLUMN_TITLE_DOUBLE_CLICKED

Setting this attribute specifies the action method COLUMN_TITLE_DOUBLE_CLICKED.

Data type: Text vector

_DRAGGED_AND_DROPPED

Setting this attribute specifies the action method DRAGGED_AND_DROPPED.

Data type: Text vector

_FOCUSSED_CELL_CHANGED

Setting this attribute specifies the action method FOCUSSED_CELL_CHANGED.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_SELECTION_HAS_CHANGED

Setting this attribute specifies the action method SELECTION_HAS_CHANGED.

Data type: Text vector

42.2.2 Action Methods

CELL_EDITING_ACCEPTED

This action method is executed whenever the user has accepted a new value of an edited cell. Two arguments are passed to this action method specifying the indexes of the cell. The arguments are integer values specifying ROW and COLUMN.

Data type to be returned: No return value

CELL_EDITING_CANCELLED

This action method is executed whenever the user cancels the editing of a cell. Two arguments are passed to this action method specifying the indexes of the cell. The arguments are integer values specifying ROW and COLUMN.

Data type to be returned: No return value

CELL_EDITING_STARTED

This action method is executed whenever the user starts editing a cell. Two arguments are passed to this action method specifying the indexes of the cell. The arguments are integer values specifying ROW and COLUMN.

Data type to be returned: No return value

COLUMN_TITLE_CLICKED

This method is executed whenever the user clicks a column title. One argument is passed to this action method specifying the number of the column title, which was clicked. The argument is an integer value specifying COLUMN NUMBER.

Data type to be returned: No return value

COLUMN_TITLE_DOUBLE_CLICKED

This method is executed whenever the user double-clicks a column title. One argument is passed to this action method specifying the number of the column title that was double-clicked. The argument is an integer value specifying COLUMN NUMBER.

Data type to be returned: No return value

DRAGGED_AND_DROPPED

This method is executed whenever the user has dragged and dropped a column/row or multiple columns/rows. Five arguments are passed to these action methods: START_NUM = previous column/row order number, COUNT = number of columns/rows been dragged, LEVEL = 0 for a single column/row, otherwise is a group level, END_NUM = new order number of the column/row, TYPE = text ROW or COLUMN specifies the moved items.

Data type to be returned: No return value

FOCUSSED_CELL_CHANGED

This method is executed whenever the user has changed the focused cell with the mouse or arrow keys. Two arguments are passed to this action method specifying the indexes of the new focused cell. The arguments are integer values specifying ROW and COLUMN.

Data type to be returned: No return value

SELECTION_HAS_CHANGED

This method is executed whenever the selection is going to be changed, is changing and has changed. One argument is passed to this action method. The argument can be used to detect how the selection is changing.

Data type to be returned: No return value

Example:

```
@state=argument(1)

#case %arg

#when 1 @stateText="Selection is going to change"
#when 2 @stateText="Selection is changing"
#when 3 @stateText="Selection has changed"
#otherwise @stateText="Selection error"

#case_end
```

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

42.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_EDIT_ENABLED

This attribute will specify how to enter edit mode. If this attribute is TRUE, the editing will start automatically when entering an editable cell. If this attribute is FALSE, the user must press ENTER to start editing. This attribute only affects keyboard handling.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_AUTO_SCROLL_ENABLED

Specifies whether the table scrolls automatically or not when the user drags the mouse outside of the area currently visible in the table.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#)

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

Description: A value of TRUE or FALSE.

Data type: Boolean

Default value: Object dependent

Access: No limitations

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_COLUMN_COUNT

Specifies the number of used columns in a table.



Setting this attribute to zero will clear all columns and their properties. In this case, also the following properties and attributes are initialized to default values: default column width, default column title, _COLUMN_DRAG_AND_DROP_ENABLED, _COLUMN_HEADER_RESIZABLE, _COLUMN_HEADER_SELECTABLE, _EXTEND_COLUMN_SELECTION_ENABLED, _HAS_COLUMN_HEADER, _SORTING_ENABLED.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_COLUMN_DRAG_AND_DROP_ENABLED

This attribute specifies whether dragging and dropping columns or column groups is enabled. If this attribute is TRUE, the user can drag and drop a column or a column group. If this attribute is FALSE, the user cannot drag and drop columns or column groups.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_COLUMN_HEADER_RESIZABLE

Specifies whether the user can resize column titles and groups or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_COLUMN_HEADER_SELECTION_ENABLED

Specifies the column header behavior. This attribute affects all column titles and column groups. If this attribute is TRUE, clicking a column title will select the entire column.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_DOUBLE_CLICK_EDIT_ENABLED

This attribute will specify how to enter edit mode. If this attribute is TRUE, the user must double-click a cell to start editing. If this attribute is FALSE, the user needs to click a cell only once to start editing. This attribute only affects mouse handling.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_DRAWING_SELECTION_ENABLED

Specifies whether a drawing selection is used or not. If _DRAWING_SELECTION_ENABLED is set to TRUE, all the table cells that were under the cursor when dragging the mouse became selected. If _DRAWING_SELECTION_ENABLED is set to FALSE, the selected area is rectangular.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_EXTEND_COLUMN_SELECTION_ENABLED

When this attribute is set to TRUE, the entire column is selected when any of the cells in the column is selected. The _SELECTION_POLICY attribute works as before, except that it operates on columns instead of single cells. Only one column can be selected when _SELECTION_POLICY is ONE_ONLY_SELECTION.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_EXTEND_ROW_SELECTION_ENABLED

When this attribute is set to TRUE, the entire row is selected when any of the cells in the row is selected. The SELECTION_POLICY attribute works as before, except that it operates on rows instead of single cells. Only one row can be selected when SELECTION_POLICY is ONE_ONLY_SELECTION.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_FILE_REVISION

For more information, see [Section 3](#).

_FIRST_VISIBLE_COLUMN

Contains the index of the leftmost visible column.

| | |
|----------------|---|
| Description: | An integer from 1 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | Read-only |

_FIRST_VISIBLE_ROW

Contains the index of the topmost visible row.

| | |
|----------------|---|
| Description: | An integer from 1 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | Read-only |

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FOCUSSED_CELL

Specifies the current focused cell. When the user is using keyboard to travel around the table, the _FOCUSSED_CELL is changing. The _FOCUSSED_CELL can sometimes be one of _SELECTED_CELLS.

| | |
|----------------|---|
| Description: | A vector containing zero or more items. The items are vectors containing two integer items. |
| Data type: | Integer |
| Default value: | Empty vector |
| Access: | No limitations |

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Designates whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if you grabbed it, so that the scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Specifies the border of an entire VS_TABLE object. If this attribute is set to TRUE, the table has a border. If this attribute is set to FALSE, the table does not have a border.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_HAS_COLUMN_HEADER

Specifies the column header visibility. If this attribute is TRUE, column header is visible. Setting this attribute to FALSE will hide the column header (all column titles and column groups).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_HAS_ROW_HEADER

Specifies the row header visibility. If this attribute is TRUE, row header is visible. Setting this attribute to FALSE will hide the row header (all row titles and row groups).

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_HORIZONTAL_GRID_ENABLED

Specifies whether a horizontal grid is enabled or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_KEYBOARD_SELECTION_ENABLED

Specifies whether a table allows keyboard selection through keyboard focus. When a table has focus and a key is pressed, the selected cell of the table changes to the first one that starts with the character pressed. If the characters are pressed with delay smaller than one second, the characters are collected to a string and a matching cell text is searched.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_KEYBOARD_SELECTION_ORDER

Specifies the searching order of matching cell texts after pressing of character keys. It has meaning only if _KEYBOARD_SELECTION_ENABLED attribute value is TRUE. Possible values are ROW_MAJOR and COLUMN_MAJOR. In the first case, when a character key is pressed, all rows of the first column are scanned from top to bottom then all rows of the second column and so forth. In the second case the scanning occurs by all columns of the first row then by all columns of the second row and so forth.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: ROW_MAJOR or COLUMN_MAJOR |
| Data type: | Text |
| Default value: | "ROW_MAJOR" |
| Access: | No limitations |

_KEYBOARD_TRAVERSAL_ENABLED

Specifies if the keyboard traversal is used in table cells or not. If the value is TRUE, the VS_TABLE responds to keyboard commands. In this case, the user can make selections or change the focused cell with keyboard arrow keys. If the value is FALSE, the VS_TABLE will only handle mouse events. In this case, the user can make selections or change the focused cell only with mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_ROW_COUNT

Specifies the number of used rows in a table.



Setting this attribute to zero will clear all rows and their properties. In this case, the following properties and attributes are also initialized to default values:
`_default_row_title, _default_row_height, _EXTEND_ROW_SELECTION_ENABLED, _HAS_ROW_HEADER, _ROW_DRAG_AND_DROP_ENABLED, _ROW_HEADER_RESIZABLE, _ROW_HEADER_SELECTABLE, _SORTING_ENABLED, _ROW_HEADER_WIDTH.`

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_ROW_DRAG_AND_DROP_ENABLED

This attribute specifies whether dragging and dropping of rows or row groups is enabled. If this attribute is TRUE, the user can drag and drop a row or a row group. If this attribute is FALSE, the user cannot drag and drop rows or row groups.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ROW_HEADER_RESIZABLE

Specifies whether the user can resize row titles and groups or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_ROW_HEADER_SELECTION_ENABLED

Specifies the row header behavior. This attribute affects to all row titles and row groups. If this attribute is TRUE, clicking a row title will select the entire row.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ROW_HEADER_WIDTH

Specifies the width of the row header as a percentage value of the VS_TABLE width. The calculation of the actual VS_TABLE row header width is done so that the VS_TABLE keeps at

least a sliver of a cell visible. For example, the width of the VS_TABLE is 200. After .SET MY_TABLE._ROW_HEADER_WIDTH=50, the header width is 100 (in pixels).

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 25 |
| Access: | No limitations |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_CELLS

Specifies the selected cells in a table. Note that if the attribute _SELECTION_POLICY is set to ONE_ONLY_SELECTION, only one cell can be selected at a time.

| | |
|----------------|---|
| Description: | A vector containing zero or more items. The items are vectors containing two integer items. |
| Data type: | Integer |
| Default value: | Empty vector |
| Access: | No limitations |

_SELECTED_ROW

Specifies the selected row number.

| | |
|----------------|----------------|
| Description: | An integer |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

Example:

```
@rownumber = my_table._selected_row
.set my_table._selected_row = 5
```

_SELECTION_POLICY

Specifies the selection policy in a table.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: FREE_SELECTION, ONE_ONLY_SELECTION, RECTANGULAR_SELECTION and NO_SELECTION. |
| Data type: | Text |
| Default value: | "ONE_ONLY_SELECTION" |
| Access: | No limitations |

_SELECTION_SENSE_ENABLED

If _SELECTION_SENSE_ENABLED is TRUE and the user adjust-clicks on selected cells, the cells are removed from the selection. When _SELECTION_SENSE_ENABLED is FALSE, clicking cells with the mouse only adds cells to the selection. Note that the implementation of adjust-click

may vary between platforms. On the Windows platform, adjust-click is implemented as CTRL +left mouse button.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_SORT_ARROW_VISIBLE

Specifies whether the arrow showing the sort order will be visible or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SORT_CRITERIA

This attribute specifies the sorting criteria of a table. Sorting can be based on one or multiple columns. The sorting criterion is an ordered vector, which contains a sorting keys. The sorting key is a vector of two elements. This element specifies the column and the sort order of that column. Sort order is either ASCENDING, DESCENDING or NO_ORDER.

| | |
|----------------|-----------------------|
| Description: | Any valid SCIL value. |
| Data type: | Any valid SCIL value |
| Default value: | Empty vector |
| Access: | No limitations |

Example:

```
.SET MY_TABLE._SORT_CRITERIA=vector((3,"ASCENDING"), (1,"DESCENDING"))

MY_TABLE._SORT
```

This means that the column 3 is the first sorting key and it is sorted in ascending order and the second sorting key is the column 1 and it is sorted in descending order. Also this means that for the columns 1 and 3 the sorting is enabled.

_SORTING_ENABLED

Specifies the default sorting. By default, sorting is disabled for a table.



Setting this attribute to TRUE requires that there are some rows but no groups in the table.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_GRID_ENABLED

Specifies whether a vertical grid is enabled or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

42.2.4 Predefined Methods

_ACCEPT_EDIT

Terminates any active cell editing. After this call, a new value has been stored to the cell.

Return value data type: No return value

_ADD_COLUMN_GROUP

This method adds a column group to the given level LEVEL on a table. All columns in range [START_COLUMN, END_COLUMN] belong to this column group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| END_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| LEVEL | Integer An integer value. |

_ADD_COLUMNS

Inserts COUNT columns before (AFTER=FALSE) or after (AFTER=TRUE) column with number START_COLUMN. If the column START_COLUMN is in a column group, new columns are added to the same group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| AFTER | Boolean A value of TRUE or FALSE. |
| COUNT | Integer An integer value. |

_ADD_ROW_GROUP

This method adds a row group to the given level LEVEL to the table. Rows in range [START_ROW, END_ROW] belongs to this row group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_ROW | Integer An integer from 1 to the length of a SCIL vector. |
| END_ROW | Integer An integer from 1 to the length of a SCIL vector. |
| LEVEL | Integer An integer value. |

_ADD_ROWS

Inserts COUNT rows before (AFTER=FALSE) or after (AFTER=TRUE) row with number START_ROW. If the row START_ROW is in a row group, new rows are added to the same group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_ROW | Integer An integer from 1 to the length of a SCIL vector. |
| AFTER | Boolean A value of TRUE or FALSE. |
| COUNT | Integer An integer value. |

_CANCEL_EDIT

Terminates any active cell editing. The cell's value is returned to the value before the editing started.

Return value data type: No return value

_DELETE_COLUMNS

Deletes columns as a range from START_COLUMN to END_COLUMN



This attribute also deletes the corresponding column groups.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| END_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_DELETE_ROWS

Deletes rows as a range from START_ROW to END_ROW.



This attribute also deletes the corresponding row group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_ROW | Integer An integer from 1 to the length of a SCIL vector. |
| END_ROW | Integer An integer from 1 to the length of a SCIL vector. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_FREEZE_COLUMNS

This method freezes all columns from 1 to END_COLUMN. These columns always become visible independently of the scrolling process. These columns must not belong to any group, or the group these columns belong to must not contain other columns. Otherwise, this function generates an exception.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| END_COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_CELL_BACKGROUND_COLOR

Returns the background color being used for the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|---|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_CELL_FONT

Returns the font being used for the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|---|
| Return value data type: | Font |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

COLUMN **Integer**
 An integer from 1 to the length of a SCIL vector.

_GET_CELL_FOREGROUND_COLOR

Returns the foreground color being used for the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|---|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_CELL_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification being used for the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|---|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

COLUMN **Integer**
 An integer from 1 to the length of a SCIL vector.

_GET_CELL_OBJECT_ID

Returns the object id saved in the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_CELL_TEXT

Returns the current value of the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_CELL_TEXT_RANGE

Returns the text vector ranging from the given cell in a table. The text area can be fetched horizontally or vertically.

| | |
|--------------|--|
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| DIRECTION | Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL. |
| COUNT | Integer An integer value. |

Example:

```
@VECTOR = MY_TABLE._GET_CELL_TEXT_RANGE(1, 1, "HORIZONTAL", 10)
```

_GET_CELL_VERTICAL_JUSTIFICATION

Returns the vertical justification being used for the cell specified by the ROW and COLUMN arguments.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_BACKGROUND_COLOR

Returns the background color being used for the column specified by argument COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_FONT

Returns the font being used for the column specified by argument COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_FOREGROUND_COLOR

Returns the foreground color being used for the column specified by argument COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_BACKGROUND_COLOR

Returns the background color being used for the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_COUNT

Returns the count of column groups at given level LEVEL. Note that this attribute returns the value -1 if there is no group at all, or zero (0) if there are groups in other levels than LEVEL.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_FONT

Returns the font being used for the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Font |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_FOREGROUND_COLOR

Returns the foreground color being used for the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification being used for the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_OBJECT_ID

Returns the object id saved in the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_GROUP_TITLE

Returns the title of the column group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification being used for drawing the text of the COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_OBJECT_ID

Returns the object id saved in the column COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_TITLE

Returns the title of the column COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_TITLE_BACKGROUND_COLOR

Returns the column title being used for the column title specified by the column argument, as well as the color of the top left cell.

In the example below, the top left cell is set to the color of the first column title's background color.

| | VS_LOCAL | VS_REMOTE | X_REMOTE |
|----|---------------------------|-------------------------------|-------------------------------|
| U1 | Times New Roman, Bold, 28 | times-bold-r--28--x--iso8859- | times-bold-r--28--x--iso8859- |
| U2 | | | |
| U3 | | | |
| U4 | | | |

Figure 1: Column title background color

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_TITLE_FONT

Returns the font being used for the column title specified by the COLUMN argument.

| | |
|-------------------------|--|
| Return value data type: | Font |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_TITLE_FOREGROUND_COLOR

Returns the foreground color being used for the column title specified by the COLUMN argument.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_TITLE_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification being used for the column title specified by the COLUMN argument.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_VERTICAL_JUSTIFICATION

Returns the vertical justification being used for the column specified by COLUMN argument.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_COLUMN_WIDTH

Returns the width of the column COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_GET_EDITED_CELL

Returns a vector containing a vector with the indexes (row and column) of the currently edited cell in a table (eg. vector((%i_Row,%i_Col))). When no cell is being edited, this attribute returns an empty vector.

Return value data type: Integer

Example:

```
@VECTOR = MY_TABLE._GET_EDITED_CELL
```

_GET_HELP

For more information, see [Section 3](#).

_GET_ROW_BACKGROUND_COLOR

| | |
|-------------------------|---|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

Returns the background color being used for the row ROW.

_GET_ROW_FONT

Returns the font being used for the row ROW.

| | |
|-------------------------|---|
| Return value data type: | Font |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_FOREGROUND_COLOR

Returns the foreground color being used for the row ROW.

| | |
|-------------------------|---|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_BACKGROUND_COLOR

Returns the background color being used for the row group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|---|
| Return value data type: | Color |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_COUNT

Returns the count of row groups at given level LEVEL. Note that this attribute returns the value -1 if there is no group at all, or zero (0) if there are groups in other levels than LEVEL.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_FONT

Returns the font being used for the row group title specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Font |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_FOREGROUND_COLOR

Returns the foreground color used for the row group title specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification used for the row group title specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_OBJECT_ID

Returns the object id saved in the row group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_GROUP_TITLE

Returns the title of the row group specified by arguments LEVEL and ORDER_NUMBER.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_HEIGHT

Returns the height of the row ROW.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification of the row ROW.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_OBJECT_ID

Returns the object id saved in the row ROW.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_TITLE

Returns the title of the row ROW.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_TITLE_BACKGROUND_COLOR

Returns the background color being used for the row title specified by the ROW argument.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_TITLE_FONT

Returns the font being used for the row title specified by the ROW argument.

| | |
|-------------------------|--|
| Return value data type: | Font |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_TITLE_FOREGROUND_COLOR

Returns the foreground color being used for the row title specified by the ROW argument.

| | |
|-------------------------|--|
| Return value data type: | Color |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_TITLE_HORIZONTAL_JUSTIFICATION

Returns the horizontal justification being used for the row title specified by the ROW argument.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_ROW_VERTICAL_JUSTIFICATION

Returns the vertical justification being used for the row specified by the ROW argument.

| | |
|-------------------------|--|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_IS_CELL_READ_ONLY

Returns the value being used to determine whether the cell specified by the ROW and COLUMN arguments is considered read-only or not.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_CELL_TEXT_WRAPPED

Returns the value being used to determine whether the cell text specified by the ROW and COLUMN arguments is considered wrapped or not.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_COLUMN_READ_ONLY

Returns the value being used to determine whether the specified COLUMN is considered read-only or not.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_COLUMN_RESIZABLE

Returns TRUE if the column title of the column COLUMN is resizable, otherwise returns FALSE.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_COLUMN_SELECTABLE

Returns TRUE if the column title of the column COLUMN is selectable, otherwise returns FALSE.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_COLUMN_SORTING_ENABLED

Is sorting enabled or disabled for column COLUMN.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_COLUMN_TEXT_WWRAPPED

Returns the value being used to determine whether or not the column text specified by the COLUMN argument is considered wrapped.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_IS_ROW_READ_ONLY

Returns the value being used to determine whether the specified ROW is considered read-only or not.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_IS_ROW_RESIZABLE

Returns the value being used to determine whether the specified ROW title is considered resizable or not.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_IS_ROW_SELECTABLE

Returns the value being used to determine whether the specified ROW title is considered selectable or not.

| | |
|-------------------------|---|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_IS_ROW_TEXT_WWRAPPED

Returns the value being used to determine whether the specified ROW texts are considered wrapped or not.

| | |
|-------------------------|---|
| Return value data type: | Boolean |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SCROLL_POSITION_INTO_VIEW

Scrolls the table so that the specified cell is at least partly visible.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |

_SCROLL_SELECTION_INTO_VIEW

Scrolls the table so that the text that is currently selected is at least partly visible.

Return value data type: No return value

_SET_CELL_BACKGROUND_COLOR

Sets the background color of the cell specified by the ROW and COLUMN to the COLOR.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_CELL_EDIT_TYPE_TEXT

This method adds a default text edit type to a cell described by parameters ROW and COLUMN.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| IS_ONE_LINE | Boolean A value of TRUE or FALSE. |
| WRAP_ENABLED | Boolean A value of TRUE or FALSE. |
| TEXT | Text A character string containing up to 255 characters. |

SET_CELL_FONT

Sets the font of the cell specified by the ROW and COLUMN to the FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

SET_CELL_FOREGROUND_COLOR

Sets the foreground color of the cell specified by the ROW and COLUMN to the COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

SET_CELL_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification of the cell specified by the ROW and COLUMN to the JUSTIFICATION.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_CELL_OBJECT_ID

Sets the identification string of the specified cell. The identification can be any valid text string. The OBJECT_ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| OBJECT_ID | Text A character string containing up to 255 characters. |

_SET_CELL_READ_ONLY

Sets the cell specified by the ROW and COLUMN to be READ_ONLY. If READ_ONLY is TRUE, the cell is read only. If READ_ONLY is FALSE, the cell is editable.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| READ_ONLY | Boolean A value of TRUE or FALSE. |

_SET_CELL_TEXT

Sets the text of the given cell in the table. The table is automatically expanded if text is set into a row or column that does not exist.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| TEXT | Text A character string containing up to 255 characters. |

Example:

```
MY_TABLE._SET_CELL_TEXT(1, 1, "Inserted Text")
```

_SET_CELL_TEXT_RANGE

Sets the text area ranging from a given cell in the table. The text area can be set horizontally or vertically. The table is automatically expanded if text area is set into a row or column that does not exist. The table can be cleared by setting the attribute _ROW_COUNT to zero.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| DIRECTION | Text A text string containing one of the following keywords: VERTICAL and HORIZONTAL. |
| TEXT_VECTOR | Vector A vector containing only text elements. |

_SET_CELL_TEXT_WWRAPPED

Sets the wrapping of the cell text specified by ROW and COLUMN.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| WRAPPED | Boolean A value of TRUE or FALSE. |

_SET_CELL_VALUE_HANDLER

Defines a user defined value handler for the cell. See programming guidelines from SYS600 Visual SCIL Application Design.



Edit and view objects must be created or loaded as childs of an ancestor object of the table in the objects tree. If a sufficient time is passed between creating/loading the object and setting it as value handler, the object should be made invisible and disabled if it is not expected to appear on the dialog and take mouse and keyboard events. One object can be set as edit or view object only for one cell, row or column.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

Table continues on next page

| | |
|-------------|--|
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| VIEW_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| EDIT_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

Example:

```
.LOAD ROOT\ A=VS_COMBO_POPDOWN(-
"MYTYPES.VSO", "EDIT1_OBJECT")

.LOAD ROOT\ B=VS_TEXT(-
"MYTYPES.VSO", "VIEW1_OBJECT")

ROOT\ MY_TABLE._SET_CELL_VALUE_HANDLER(1,1, "ROOT\ B", "ROOT\ A")
```

_SET_CELL_VERTICAL_JUSTIFICATION

Sets the vertical justification of the cell specified by the ROW and COLUMN to the JUSTIFICATION.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: BOTTOM_JUSTIFY, TOP_JUSTIFY or CENTER_JUSTIFY |

_SET_COLUMN_BACKGROUND_COLOR

Sets the background color being used for the whole column specified by argument COLUMN to color COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_EDIT_TYPE_TEXT

This method adds a default text edit type to a column described by parameter COLUMN.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| IS_ONE_LINE | Boolean A value of TRUE or FALSE. |
| WRAP_ENABLED | Boolean A value of TRUE or FALSE. |
| TEXT | Text A character string containing up to 255 characters. |

_SET_COLUMN_FONT

Sets the font being used for the whole column specified by argument COLUMN to FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_COLUMN_FOREGROUND_COLOR

Sets the foreground color being used for the whole column specified by argument COLUMN to color COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_GROUP_BACKGROUND_COLOR

Sets the background color being used for the column group specified by arguments LEVEL and ORDER_NUMBER to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_GROUP_FONT

Sets the font being used for the column group specified by arguments LEVEL and ORDER_NUMBER to FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_COLUMN_GROUP_FOREGROUND_COLOR

Sets the foreground color being used for the column group specified by arguments LEVEL and ORDER_NUMBER to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_GROUP_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification being used for the column group specified by arguments LEVEL and ORDER_NUMBER to JUSTIFICATION.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_COLUMN_GROUP_OBJECT_ID

Sets the identification string of the specified column group. The identification can be any valid text string. The OBJECT_ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| OBJECT_ID | Text A character string containing up to 255 characters. |

_SET_COLUMN_GROUP_TITLE

Sets the title of the column group specified by arguments LEVEL and ORDER_NUMBER to TITLE. It is possible to use automatic numbering in the column group. The title can have “%d” to show the index of the group and “%l” to show the level of the group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| TITLE | Text A character string containing up to 255 characters. |

Example:

This example shows how to use automatic numbering of a column group:

```
._add_column_group(1,2,1)
._set_column_group_title(1,1,"Level %l, Index %d")
```

_SET_COLUMN_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification being used for drawing the text of the COLUMN to JUSTIFICATION. The JUSTIFICATION is a text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_COLUMN_OBJECT_ID

Sets the identification string of the specified column. The identification can be any valid text string. The OBJECT_ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| OBJECT_ID | Text A character string containing up to 255 characters. |

_SET_COLUMN_READ_ONLY

Sets the specified column COLUMN to be read only if the READ_ONLY argument is TRUE, otherwise the column is editable. By default, all columns are read only.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| READ_ONLY | Boolean A value of TRUE or FALSE. |

_SET_COLUMN_SORTING

Sets the specified column COLUMN to be sortable, if the SORTING_ENABLED argument is TRUE. By default, sorting is disabled for all columns.



Sorting is disabled, if there is column or row groups.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| SORTING_ENABLED | Boolean A value of TRUE or FALSE. |

_SET_COLUMN_TEXT_WWRAPPED

Sets the specified column to be wrapped if the WRAPPED argument is TRUE, otherwise the column is not wrapped. By default, columns are not wrapped.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| WRAPPED | Boolean A value of TRUE or FALSE. |

_SET_COLUMN_TITLE

Sets the title of the specified column COLUMN to TITLE. The title is a string to be displayed in the column header if the header is visible.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| TITLE | Text A character string containing up to 255 characters. |

_SET_COLUMN_TITLE_BACKGROUND_COLOR

Sets the background color of the specified column title COLUMN to the specified color COLOR. Setting this color also sets the color of the Top Left Cell in VS_TABLE.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_TITLE_FONT

Sets the font of the specified column COLUMN to the specified font FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_COLUMN_TITLE_FOREGROUND_COLOR

Sets the foreground color of the specified column title COLUMN to the specified color COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_COLUMN_TITLE_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification being used for the column title specified by the COLUMN argument. The JUSTIFICATION is a text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_COLUMN_TITLE_RESIZABLE

Sets the behavior of the column title specified by COLUMN. If RESIZABLE is TRUE, the user can resize the column width with the cursor.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| RESIZABLE | Boolean A value of TRUE or FALSE. |

_SET_COLUMN_TITLE_SELECTABLE

Sets the behavior of the column title specified by COLUMN. If SELECTABLE is TRUE, the user can select the entire column by clicking the title of the column.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| SELECTABLE | Boolean A value of TRUE or FALSE. |

_SET_COLUMN_VALUE_HANDLER

Defines a user defined value handler for a column. See programming guidelines from SYS600 Visual SCIL Application Design.



Edit and view objects must be created or loaded as childs of an ancestor object of the table in the objects tree. If a sufficient time is passed between creating/loading the object and setting it as value handler, the object should be made invisible and disabled if it is not expected to appear on the dialog and take mouse and keyboard events. One object can be set as edit or view object only for one cell, row or column.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| VIEW_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| EDIT_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

Example:

```
.LOAD ROOT\ A=VS_COMBO_POPDOWN (-  
"MYTYPES.VSO", "EDIT1_OBJECT")  
  
.LOAD ROOT\ B=VS_TEXT (-  
"MYTYPES.VSO", "VIEW1_OBJECT")  
  
ROOT\ MY_TABLE._SET_COLUMN_VALUE_HANDLER(1, "ROOT\ B", "ROOT\ A")
```

_SET_COLUMN_VERTICAL_JUSTIFICATION

Sets the vertical justification being used for drawing the text of the COLUMN to JUSTIFICATION. The JUSTIFICATION is a text string containing one of the following keywords: TOP_JUSTIFY, BOTTOM_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: BOTTOM_JUSTIFY, TOP_JUSTIFY or CENTER_JUSTIFY |

_SET_COLUMN_WIDTH

Sets the width of the column COLUMN to WIDTH.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| COLUMN | Integer An integer from 1 to the length of a SCIL vector. |
| WIDTH | Integer An integer value. |

_SET_ROW_BACKGROUND_COLOR

Sets the background color of the row ROW to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_EDIT_TYPE_TEXT

This method adds a default text edit type to a row described by parameter ROW.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| IS_ONE_LINE | Boolean A value of TRUE or FALSE. |
| WRAP_ENABLED | Boolean A value of TRUE or FALSE. |
| TEXT | Text A character string containing up to 255 characters. |

_SET_ROW_FONT

Sets the font of the row ROW to FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_ROW_FOREGROUND_COLOR

Sets the foreground color of the row ROW to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_GROUP_BACKGROUND_COLOR

Sets the background color being used for the row group specified by arguments LEVEL and ORDER_NUMBER to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_GROUP_FONT

Sets the font being used for the row group specified by arguments LEVEL and ORDER_NUMBER to FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_ROW_GROUP_FOREGROUND_COLOR

Sets the foreground color being used for the row group specified by arguments LEVEL and ORDER_NUMBER to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_GROUP_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification being used for the row group specified by arguments LEVEL and ORDER_NUMBER to JUSTIFICATION. JUSTIFICATION is a text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_ROW_GROUP_OBJECT_ID

Sets the identification string of the specified row group. The identification can be any valid text string. The OBJECT_ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| OBJECT_ID | Text A character string containing up to 255 characters. |

_SET_ROW_GROUP_TITLE

Sets the title of the row group specified by arguments LEVEL and ORDER_NUMBER to TITLE. It is possible to use automatic numbering in the row group. The title can have "%d" to show the index of the group and "%l" to show the level of the group.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| LEVEL | Integer An integer from 1 to the length of a SCIL vector. |
| ORDER_NUMBER | Integer An integer from 1 to the length of a SCIL vector. |
| TITLE | Text A character string containing up to 255 characters. |

Example:

This example shows how to use automatic numbering of a row group:

```
._add_row_group(1,2,1)  
.set_row_group_title(1,1,"%l / %d")
```

_SET_ROW_HEIGHT

Sets the height of the row ROW to HEIGHT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| HEIGHT | Integer An integer value. |

_SET_ROW_HORIZONTAL_JUSTIFICATION

Sets the horizontal justification of the row ROW to JUSTIFICATION. JUSTIFICATION is a text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_ROW_OBJECT_ID

Sets the identification string of the specified row. The identification can be any valid text string. The OBJECT_ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| OBJECT_ID | Text A character string containing up to 255 characters. |

_SET_ROW_READ_ONLY

Sets the specified row ROW to be READ ONLY if the READ_ONLY argument is TRUE, otherwise the row is editable. By default, all rows are READ ONLY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| READ_ONLY | Boolean A value of TRUE or FALSE. |

_SET_ROW_TEXT_WWRAPPED

Sets the specified row to be WRAPPED if the WRAPPED argument is TRUE, otherwise the row is not wrapped. By default, rows are not wrapped.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| WRAPPED | Boolean A value of TRUE or FALSE. |

_SET_ROW_TITLE

Sets the title of the specified row ROW to TITLE. TITLE is a string to be displayed in the row header if the header is visible.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| TITLE | Text A character string containing up to 255 characters. |

_SET_ROW_TITLE_BACKGROUND_COLOR

Sets background color of the title of the specified row ROW to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_TITLE_FONT

Sets font of the title of the specified row ROW to FONT.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| FONT | Font A font as referenced within SCIL. |

_SET_ROW_TITLE_FOREGROUND_COLOR

Sets foreground color the title of the specified row ROW to COLOR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| COLOR | Color A color as referenced within SCIL. |

_SET_ROW_TITLE_HORIZONTAL_JUSTIFICATION

Sets horizontal justification of the title of the specified row ROW to JUSTIFICATION. JUSTIFICATION is a text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: DEFAULT, LEFT_JUSTIFY, RIGHT_JUSTIFY or CENTER_JUSTIFY |

_SET_ROW_TITLE_RESIZABLE

Sets the behavior of the row title specified by ROW. If RESIZABLE is TRUE, the user can resize the row height with the cursor.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| RESIZABLE | Boolean A value of TRUE or FALSE. |

_SET_ROW_TITLE_SELECTABLE

Sets the behavior of the row title specified by ROW. If SELECTABLE is TRUE, the user can select the entire row by clicking the title of the row.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| SELECTABLE | Boolean A value of TRUE or FALSE. |

_SET_ROW_VALUE_HANDLER

Defines a user defined value handler for the row ROW. See programming guidelines from SYS600 Visual SCIL Application Design.



Edit and view objects must be created or loaded as childs of an ancestor object of the table in the objects tree. If a sufficient time is passed between creating/ loading the object and setting it as value handler, the object should be made invisible and disabled if it is not expected to appear on the dialog and take mouse and keyboard events. One object can be set as edit or view object only for one cell, row or column.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| VIEW_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |
| EDIT_OBJECT | Dialog Item A reference to a dialog item such as VS_TEXT, VS_BUTTON etc. The reference is a text reference to the name of the object. |

Example:

```
.LOAD ROOT\ A=VS_COMBO_POPDOWN (-  
"MYTYPES.VSO", "EDIT1_OBJECT")  
.LOAD ROOT\ B=VS_TEXT (-  
"MYTYPES.VSO", "VIEW1_OBJECT")  
ROOT\ MY_TABLE._SET_ROW_VALUE_HANDLER(1, "ROOT\ B", "ROOT\ A")
```

_SET_ROW_VERTICAL_JUSTIFICATION

Sets vertical justification of the row of the specified row ROW to JUSTIFICATION. JUSTIFICATION is a text string containing one of the following keywords: TOP_JUSTIFY, BOTTOM_JUSTIFY or CENTER_JUSTIFY.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |
| JUSTIFICATION | Text A text string containing one of the following keywords: BOTTOM_JUSTIFY, TOP_JUSTIFY or CENTER_JUSTIFY |

_SORT

This method forces the table to be sorted based on the _SORT_CRITERIA.

Return value data type: No return value

Example:

```
.SET MY_TABLE._SORT_CRITERIA = VECTOR( (1,"ASCENDING") , (3,"DESCENDING") )  
MY_TABLE._SORT
```

_UNFREEZE_COLUMNS

This method unfreezes columns previously frozen. If there are no frozen columns, the method does nothing.

Return value data type: No return value

Section 43 VS_TEXT

43.1 Overview

A VS_TEXT object is a text field for editing single or multilined texts. An object can be set to read only by setting the _MODIFIABLE attribute.

43.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_CHANGE_FILTER" | Text vector |
| "_DELETE_FILTER" | Text vector |
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_GOT_FOCUS" | Text vector |
| "_INPUT_FILTER" | Text vector |
| "_LOST_FOCUS" | Text vector |
| "_TEXT_DIMENSION_ERROR" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

43.1.2 Action Methods

| Name | Type |
|-------------------------------------|-----------------|
| "CHANGE_FILTER" | Boolean |
| "DELETE_FILTER" | Boolean |
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "GOT_FOCUS" | No return value |
| "INPUT_FILTER" | Boolean |
| "LOST_FOCUS" | No return value |
| "TEXT_DIMENSION_ERROR" | No return value |
| "TOOLTIP_IS_SHOWN_TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

43.1.3 Predefined Attributes

| Name | Type |
|--------------------|---------|
| "_ACTIVE" | Boolean |
| "_ATTRIBUTE_NAMES" | List |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_AUTO_INSERT_ENABLED" | Boolean |
| "_AUTO_SCROLL_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DRAG_DROP_EDIT_ENABLED" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FIRST_VISIBLE_LINE" | Integer |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_HORIZONTAL_SCROLL_LOCKED" | Boolean |
| "_IS_ONE_LINE" | Boolean |
| "_LEFT_MARGIN" | Integer |
| "_LINE_COUNT" | Integer |
| "_MODIFIABLE" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SELECTABLE" | Boolean |
| "_SELECTED_TEXT_POS_BEHAVIOUR" | Text |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT" | Text |
| "_TEXT_VECTOR" | Vector |
| "_TOP_MARGIN" | Integer |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VERTICAL_SCROLL_LOCKED" | Boolean |
| "_VISIBLE" | Boolean |
| "_VISIBLE_LINE_COUNT" | Integer |

43.1.4 Predefined Methods

| Name | Type |
|-----------------------|-----------------|
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |

Table continues on next page

| | |
|--|-----------------|
| <code>"_GET_SELECTED_TEXT"</code> | Vector |
| <code>"_GET_SELECTED_TEXT_POS"</code> | List |
| <code>"_GET_STRING_WIDTH"</code> | Integer |
| <code>"_GET_VISIBLE_TEXT_POS"</code> | List |
| <code>"_QUEUE_FOR_EXECUTION"</code> | No return value |
| <code>"_SCROLL_POSITION_INTO_VIEW"</code> | No return value |
| <code>"_SCROLL_SELECTION_INTO_VIEW"</code> | No return value |
| <code>"_SELECT_TEXT_RANGE"</code> | No return value |
| <code>"_SET_SELECTED_TEXT"</code> | Boolean |
| <code>"_SET_SELECTED_TEXT_POS"</code> | No return value |

43.2 Details

43.2.1 Action Attributes

_CHANGE_FILTER

Setting this attribute specifies the action method CHANGE_FILTER.

Data type: Text vector

_DELETE_FILTER

Setting this attribute specifies the action method DELETE_FILTER.

Data type: Text vector

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_GOT_FOCUS

Setting this attribute specifies the action method GOT_FOCUS.

Data type: Text vector

_INPUT_FILTER

Setting this attribute specifies the action method INPUT_FILTER.

Data type: Text vector

_LOST_FOCUS

Setting this attribute specifies the action method LOST_FOCUS.

Data type: Text vector

_TEXT_DIMENSION_ERROR

Setting this attribute specifies the action method TEXT_DIMENSION_ERROR.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

43.2.2 Action Methods

CHANGE_FILTER

This action method is executed whenever a change is about to take place in a text field. The result of the change, if fulfilled, is passed to the method as argument 1. This way the user can check the result of the change before it has been executed, and either acknowledge it by returning TRUE or stopping it by returning FALSE. Note that this action method is executed only for text objects, in which the _IS_ONE_LINE attribute is set to TRUE. Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: Boolean

DELETE_FILTER

The action method DELETE_FILTER is executed when the object is set programmatically.

Data type to be returned: Boolean

GET_TABLE_EDIT_TEXT

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of a cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell.

This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

GOT_FOCUS

This program is run whenever an object has received keyboard focus. Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: No return value

INPUT_FILTER

This function provides a possibility to filter and/or reject a text that is inserted into a text field. The INPUT_FILTER action method gets the text that is going to be inserted into the field as the first argument to this function. The user can examine the text and return TRUE if the text should be accepted and FALSE if the text should be rejected. The action method INPUT_FILTER is executed when setting the object programmatically. Note that the text that is going to be inserted into the field can contain more than one character, for example if the text is assigned into the field by setting the attribute _TEXT.

Data type to be returned: Boolean

LOST_FOCUS

This program is run whenever an object has lost keyboard focus. This function can be used, for example, to validate the contents of a text field when the user has left the field. Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: No return value

TEXT_DIMENSION_ERROR

Executed if the user tries to type or paste text, resulting the exceeding of allowed text dimensions. This action method is also called if the user deletes LINEFEED characters causing the text to exceed the allowed text dimensions.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of a view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for a view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell, to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

43.2.3 Predefined Attributes

_ACTIVE

Specifies whether a text is active or not. This attribute is TRUE if the text view is active or has focus, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_INSERT_ENABLED

Specifies whether the auto insert function is used or not. This attribute is TRUE if auto inserting is enabled, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | Read-only |

_AUTO_SCROLL_ENABLED

Specifies whether the auto scroll function is used or not. This attribute is TRUE if auto scrolling is enabled, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DRAG_DROP_EDIT_ENABLED

Specifies whether the drag and drop edit is used or not. This attribute is TRUE if drag and drop editing is enabled, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FIRST_VISIBLE_LINE

The line number of the first visible line. The text can be scrolled using this attribute.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Specifies whether the Grabbing hand is used or not. This attribute returns TRUE if the Grabbing hand is active when mouse clicks occur in the text, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HORIZONTAL_SCROLL_LOCKED

Specifies whether horizontal scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_IS_ONE_LINE

Specifies if a text is a one line text or not. A one line text does not allow RETURN to be entered from the keyboard. This attribute should be set to TRUE for an input field that contains only one line to prevent the text from scrolling out of view if RETURN is pressed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

Example:

```
.SET MY_TEXT._IS_ONE_LINE = TRUE
```

_LEFT_MARGIN

Specifies the left margin for a text. The margin is the amount of space in pixels between the left edge of the text dialog item and the left edge of the text rectangle (where text is displayed).

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_LINE_COUNT

The number of text lines in the object.

Description: An integer value.

Data type: Integer

Default value: Environment dependent

Access: Read-only

_MODIFIABLE

Specifies whether the text of a text item is modifiable or not. A non-modifiable text item will not allow text to be entered via the keyboard. However, it will allow the text to be changed programmatically.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")
```

```
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTABLE

Specifies whether the text string of a text item is selectable or not. An unselectable text object will not allow selection via the mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_SELECTED_TEXT_POS_BEHAVIOUR

The enum with possible values NCLUDE_LAST_CARRIAGE_RETURN and EXCLUDE_LAST_CARRIAGE_RETURN which defines the behavior of the _SET_SELECTED_TEXT_POS method.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: INCLUDE_LAST_CARRIAGE_RETURN, EXCLUDE_LAST_CARRIAGE_RETURN. |
| Data type: | Text |
| Default value: | "INCLUDE_LAST_CARRIAGE_RETURN" |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT

Specifies the text string of a text item. The maximum length of a text line is 255 characters.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_TEXT_VECTOR

Specifies the multiline text string of the a item.

| | |
|----------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Default value: | Empty vector |
| Access: | No limitations |

_TOP_MARGIN

Specifies the top margin for a text. The margin is the amount of space in pixels between the top edge of a text dialog item and the top edge of the text rectangle (where text is displayed).

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VERTICAL_SCROLL_LOCKED

Specifies whether vertical scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_VISIBLE_LINE_COUNT

Specifies the number of lines currently visible in the object.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | Read-only |

43.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_SELECTED_TEXT

Returns the text that is currently selected.

Return value data type: Vector

_GET_SELECTED_TEXT_POS

Returns a list containing the positions of a selected text. The function also returns the position of a LINEFEED character if it is selected.

Return value data type: List

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_GET_VISIBLE_TEXT_POS

Returns a list containing the first and the last positions of the visible text.

Return value data type: List

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SCROLL_POSITION_INTO_VIEW

Scrolls the text so that the specified character is at least partly visible. The character is displayed as close to the left upper corner as possible.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| START_CHAR_IN_LINE | Integer An integer value. |

_SCROLL_SELECTION_INTO_VIEW

Scrolls the text so that the text that is currently selected is also at least partly visible.

Return value data type: No return value

_SELECT_TEXT_RANGE

Selects the range of text in a text object or places the cursor in the text. When the start parameter is 1, the starting position for the selection is before the first character. When the start parameter is -1, the starting position for the selection is after the last character. Length parameter 0 indicates placing the cursor in the text without selecting any characters. A positive length parameter defines the number of selected characters to the right from the starting point. A negative length parameter defines the number of selected characters to the left from the starting point. The length parameter is automatically truncated to the length of the text. Note that this method is only valid for text objects where the _IS_ONE_LINE attribute is set to TRUE.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| START | Integer An integer value. |
| LENGTH | Integer An integer value. |

_SET_SELECTED_TEXT

Modifies the current text selection. If no text is selected, the given text is inserted where the cursor is positioned. If some text is selected, it will be replaced. If a text vector that exceeds the allowed text dimension is given as a parameter, the function returns FALSE and the text is not modified.

| | |
|-------------------------|---|
| Return value data type: | Boolean |
| Parameter(s) | |
| TEXT | Vector A vector containing only text elements. |

_SET_SELECTED_TEXT_POS

Selects the text within the specified bounds. Bounds specify positions between symbols starting from 1. Behavior of this method can be controlled by the attribute _SELECTED_TEXT_POS_BEHAVIOUR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |

Table continues on next page

| | |
|--------------------|--|
| START_CHAR_IN_LINE | Integer An integer value. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_CHAR_IN_LINE | Integer An integer value. |

Section 44 VS_TEXT_EDITOR

44.1 Overview

The VS_TEXT_EDITOR object is a base component for implementing different kinds of text editors. The object itself contains editing functionalities, such as find/replace, indenting, etc.

44.1.1 Action Attributes

| Name | Type |
|-------------------------|-------------|
| "_CHANGE_FILTER" | Text vector |
| "_DELETE_FILTER" | Text vector |
| "_GOT_FOCUS" | Text vector |
| "_INPUT_FILTER" | Text vector |
| "_LOST_FOCUS" | Text vector |
| "_TEXT_DIMENSION_ERROR" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |

44.1.2 Action Methods

| Name | Type |
|------------------------|-----------------|
| "CHANGE_FILTER" | Boolean |
| "DELETE_FILTER" | Boolean |
| "GOT_FOCUS" | No return value |
| "INPUT_FILTER" | Boolean |
| "LOST_FOCUS" | No return value |
| "TEXT_DIMENSION_ERROR" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

44.1.3 Predefined Attributes

| Name | Type |
|---------------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_INDENT_ENABLED" | Boolean |
| "_AUTO_SCROLL_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_DRAG_DROP_EDIT_ENABLED" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FIRST_VISIBLE_LINE" | Integer |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_HORIZONTAL_SCROLL_LOCKED" | Boolean |
| "_INDENT_CHARACTER_COUNT" | Integer |
| "_LEFT_MARGIN" | Integer |
| "_LINE_COUNT" | Integer |
| "_MODIFIABLE" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_POP_UP_MENU" | Menu reference |
| "_SCIL_GEOMETRY" | List |
| "_SELECTABLE" | Boolean |
| "_SELECTED_TEXT_POS_BEHAVIOUR" | Text |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT_IS_MODIFIED" | Boolean |
| "_TEXT_VECTOR" | Vector |
| "_TOP_MARGIN" | Integer |
| "_TOOLTIP_TEXT" | Text |
| "_UNDO_ENABLED" | Boolean |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VERTICAL_SCROLL_LOCKED" | Boolean |
| "_VISIBLE" | Boolean |
| "_VISIBLE_LINE_COUNT" | Integer |

44.1.4 Predefined Methods

| Name | Type |
|--------------------------|-----------------|
| "_COMMENT_LINES" | Boolean |
| "_FIND" | List |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_CURRENT_WORD" | Text |
| "_GET_CURRENT_WORD_POS" | List |
| "_GET_HELP" | No return value |
| "_GET_SELECTED_TEXT" | Vector |
| "_GET_SELECTED_TEXT_POS" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_GET_VISIBLE_TEXT_POS" | No return value |

Table continues on next page

| | |
|--------------------------------------|-----------------|
| <u>"_INDENT_LINES"</u> | Boolean |
| <u>"_QUEUE_FOR_EXECUTION"</u> | No return value |
| <u>"_REDO"</u> | Boolean |
| <u>"_SCROLL_POSITION_INTO_VIEW"</u> | No return value |
| <u>"_SCROLL_SELECTION_INTO_VIEW"</u> | No return value |
| <u>"_SELECT_BLOCK"</u> | List |
| <u>"_SET_SELECTED_TEXT"</u> | Boolean |
| <u>"_SET_SELECTED_TEXT_POS"</u> | No return value |
| <u>"_UNCOMMENT_LINES"</u> | Boolean |
| <u>"_UNDO"</u> | Boolean |
| <u>"_UNINDENT_LINES"</u> | Boolean |

44.2 Details

44.2.1 Action Attributes

_CHANGE_FILTER

Setting this attribute specifies the action method CHANGE_FILTER.

Data type: Text vector

_DELETE_FILTER

Setting this attribute specifies the action method DELETE_FILTER.

Data type: Text vector

_GOT_FOCUS

Setting this attribute specifies the action method GOT_FOCUS.

Data type: Text vector

_INPUT_FILTER

Setting this attribute specifies the action method INPUT_FILTER.

Data type: Text vector

_LOST_FOCUS

Setting this attribute specifies the action method LOST_FOCUS.

Data type: Text vector

_TEXT_DIMENSION_ERROR

Setting this attribute specifies the action method TEXT_DIMENSION_ERROR.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

44.2.2 Action Methods

CHANGE_FILTER

This action method is executed whenever a change is about to take place in a text field. If the result of the change is fulfilled, it is passed to the method as argument 1. This way, the user can check the result of the change before it has been executed, and either acknowledge it by returning TRUE or stopping it by returning FALSE. Note that this action method is executed only for text objects which can only have one line (set in dialog editor). Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: Boolean

DELETE_FILTER

The action method DELETE_FILTER is executed when setting the object programmatically.

Data type to be returned: Boolean

GOT_FOCUS

This program is run whenever an object has received keyboard focus. Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: No return value

INPUT_FILTER

This function provides a possibility to filter and/or reject the text that is inserted into a text field. The INPUT_FILTER action method gets the text that is going to be inserted into the field as the first argument to this function. The user can examine the text and return TRUE if the text should be accepted, and FALSE if the text should be rejected. The action method INPUT_FILTER is executed when the object is set programmatically. Note that the text that is going to be inserted into the field can contain more than one character, for example if the text is assigned into the field by setting the attribute _TEXT_VECTOR.

Data type to be returned: Boolean

LOST_FOCUS

This program is run whenever an object has lost keyboard focus. This function can be used, for example, to validate the contents of a text field when the user has left the field. Within this action method, no commands that may cause the input focus to change are allowed. If such commands are needed, they should be executed outside of this action method using the method QUEUE_FOR_EXECUTION.

Data type to be returned: No return value

TEXT_DIMENSION_ERROR

Executed if the user tries to type or paste text, resulting in exceeding the allowed text dimensions. This action method is also called if the user deletes LINEFEED characters causing the text to exceed the allowed text dimensions.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

44.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_INDENT_ENABLED

Specifies whether automatic indentation is enabled or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_AUTO_SCROLL_ENABLED

Specifies whether auto scroll is used or not. This attribute is TRUE if autoscrolling is enabled, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DRAG_DROP_EDIT_ENABLED

Specifies whether the drag and drop edit feature is used or not. This attribute is TRUE if drag and drop editing is enabled, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FIRST_VISIBLE_LINE

Specifies the line number of the first visible line. The text can be scrolled using this attribute.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | Environment dependent |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Specifies whether the Grabbing hand is used or not. This attribute returns TRUE if the Grabbing hand is active when mouse clicks occur in the text, otherwise FALSE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Object dependent |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HORIZONTAL_SCROLL_LOCKED

Specifies whether horizontal scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_INDENT_CHARACTER_COUNT

Specifies the number of characters used when indenting text either by using the TAB key or the _INDENT_LINES method.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 3 |
| Access: | No limitations |

_LEFT_MARGIN

Specifies the left margin for the text. The margin is the amount of space in pixels between the left edge of the text dialog item and the left edge of the text rectangle (where text is displayed).

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_LINE_COUNT

Specifies the number of text lines in the object.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | Read-only |

_MODIFIABLE

Specifies whether the text of a text item is modifiable or not. A non-modifiable text item will not allow text to be entered via the keyboard. However, it will allow the text to be changed programmatically.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")  
.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTABLE

Specifies whether the text string of a text item is selectable or not. An unselectable text object will not allow selection via the mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_SELECTED_TEXT_POS_BEHAVIOUR

The enum with possible values INCLUDE_LAST_CARRIAGE_RETURN and EXCLUDE_LAST_CARRIAGE_RETURN which defines the behavior of the _SET_SELECTED_TEXT_POS method.

| | |
|----------------|---|
| Description: | A text string containing one of the following keywords: INCLUDE_LAST_CARRIAGE_RETURN, EXCLUDE_LAST_CARRIAGE_RETURN. |
| Data type: | Text |
| Default value: | "INCLUDE_LAST_CARRIAGE_RETURN" |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT_IS_MODIFIED

Indicates whether a text has been modified since the last time this flag was cleared (the attribute set to FALSE). This flag should be cleared, for example, when saving the text. If _UNDO_ENABLED is TRUE and the user clears the flag, the undo action list is cleared.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | No limitations |

_TEXT_VECTOR

Specifies the multiline text string of a text item.

| | |
|----------------|---|
| Description: | A vector containing only text elements. |
| Data type: | Vector |
| Default value: | Empty vector |
| Access: | No limitations |

_TOP_MARGIN

Specifies the top margin for the text. The margin is the amount of space in pixels between the top edge of a text dialog item and the top edge of the text rectangle (where text is displayed).

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | No limitations |

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_UNDO_ENABLED

Specifies whether the undo functionality is enabled or not. If this attribute is TRUE, editing actions can be undone and redone by running the _UNDO and _REDO methods.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VERTICAL_SCROLL_LOCKED

Specifies whether vertical scrolling is locked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_VISIBLE_LINE_COUNT

Specifies the number of lines currently visible in the object.

| | |
|----------------|-----------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Environment dependent |
| Access: | Read-only |

44.2.4 Predefined Methods

_COMMENT_LINES

Comments the specified range of lines. The comment sign is places at the beginning of the lines.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |

_FIND

The parameter CASE_SENSITIVE should be TRUE or FALSE depending on whether the user wants the find function to be case sensitive or not. The parameter DIRECTION should be UP or DOWN depending on in which direction the user wants to execute the find function. If the user wants to search to the end of the text without having to know the position of text end, the user can give the maximum amount (length of a SCIL vector) of lines as the parameter END_LIN, and the maximum amount (1023) of text in one line as the parameter END_CHAR_IN_LINE .

| | |
|-------------------------|--|
| Return value data type: | List |
| Parameter(s) | |
| WHAT | Text A character string containing up to 1023 characters. |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| START_CHAR_IN_LINE | Integer An integer value. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_CHAR_IN_LINE | Integer An integer value. |
| CASE_SENSITIVE | Boolean A value of TRUE or FALSE. |
| DIRECTION | Text. UP or DOWN |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_CURRENT_WORD

This method returns the word on which the cursor currently is positioned. If the user has selected some text, the method returns currently selected text.

Return value data type: Text

_GET_CURRENT_WORD_POS

This method returns the position of the word on which the cursor currently is positioned. If the user has selected some text, the method returns currently selected text position.

Return value data type: List

_GET_HELP

For more information, see [Section 3](#).

_GET_SELECTED_TEXT

Returns the text that is currently selected.

Return value data type: Vector

_GET_SELECTED_TEXT_POS

Returns a list containing the positions of the selected text. The function also returns the position of LINEFEED character if it is selected.

Return value data type: No return value

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_GET_VISIBLE_TEXT_POS

Returns a list containing the first and the last positions of the visible text.

Return value data type: No return value

_INDENT_LINES

Indents the specified range of lines.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_REDO

Redoes a previously undone editing action.

Return value data type: Boolean

_SCROLL_POSITION_INTO_VIEW

Scrolls the text so that the specified character is at least partly visible. The character is displayed as close to the upper left corner as possible.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| START_CHAR_IN_LINE | Integer An integer value. |

_SCROLL_SELECTION_INTO_VIEW

Scrolls the text so that the text that is currently selected is at least partly visible.

Return value data type: No return value

_SELECT_BLOCK

If the user wants to search to the end of a text without having to know the position of the text end, the user can give the maximum amount (10000) of lines as the "endline" parameter, and the maximum amount (255) of text in one line as the "char" parameter for endline.

| | |
|-------------------------|--|
| Return value data type: | List |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| START_CHAR_IN_LINE | Integer An integer value. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_CHAR_IN_LINE | Integer An integer value. |

_SET_SELECTED_TEXT

Modifies the current text selection. If no text is selected, the text given is inserted where the cursor is positioned. If some text is selected, it is replaced. If a text vector that exceeds the allowed text dimension is given as a parameter, the function returns FALSE and the text is not modified.

| | |
|-------------------------|---|
| Return value data type: | Boolean |
| Parameter(s) | |
| TEXT | Vector A vector containing only text elements. |

_SET_SELECTED_TEXT_POS

Selects the text within the specified bounds. Bounds specify the positions between symbols starting from 1. The behavior of this method can be controlled by the attribute _SELECTED_TEXT_POS_BEHAVIOUR.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| START_CHAR_IN_LINE | Integer An integer value. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_CHAR_IN_LINE | Integer An integer value. |

_UNCOMMENT_LINES

Uncomments the specified range of lines. One comment sign at the beginning of each row is removed.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |

_UNDO

Undoes the previous editing action. The undo operation revokes the last editing operation. The maximum number of actions that can be undone is 50. The _UNDO method returns FALSE if called after max count is reached.

Return value data type: Boolean

_UNINDENT_LINES

Unindents the specified range of lines.

| | |
|-------------------------|--|
| Return value data type: | Boolean |
| Parameter(s) | |
| START_LINE | Integer An integer from 1 to the length of a SCIL vector. |
| END_LINE | Integer An integer from 1 to the length of a SCIL vector. |

Section 45 VS_TEXT_SPINNER

45.1 Overview

The VS_TEXT_SPINNER object is an object that allows the selection of a text value. The appearance of the object is dependent on the platform. Text can be added to the object by using the _APPEND_TEXT method. The _VALUE attribute contains the ordinal number of the currently selected text. The NOTIFY method for this object will be called whenever the user interactively changes the value of the object.

45.1.1 Action Attributes

| Name | Type |
|---------------------------|-------------|
| "_GET_TABLE_EDIT_TEXT" | Text vector |
| "_GET_TABLE_VIEW_TEXT" | Text vector |
| "_NOTIFY" | Text vector |
| "_TOOLTIP_IS_SHOWN" | Text vector |
| "_UPDATE_TABLE_EDIT_TEXT" | Text vector |
| "_UPDATE_TABLE_VIEW_TEXT" | Text vector |

45.1.2 Action Methods

| Name | Type |
|-------------------------------------|-----------------|
| "GET_TABLE_EDIT_TEXT" | Text |
| "GET_TABLE_VIEW_TEXT" | Text |
| "NOTIFY" | No return value |
| "TOOLTIP_IS_SHOWN TOOLTIP_IS_SHOWN" | No return value |
| "UPDATE_TABLE_EDIT_TEXT" | No return value |
| "UPDATE_TABLE_VIEW_TEXT" | No return value |

45.1.3 Predefined Attributes

| Name | Type |
|----------------------|-------------|
| "_ATTRIBUTE_NAMES" | List |
| "_AUTO_SORT_ENABLED" | Boolean |
| "_BACKGROUND_COLOR" | Color |
| "_CHILD_OBJECTS" | Text vector |
| "_CLICK_FOCUSABLE" | Boolean |
| "_ENABLED" | Boolean |
| "_FILE_REVISION" | Text |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |

Table continues on next page

| | |
|---------------------|-------------|
| "_BACKGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_MODIFIABLE" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_INDEX" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TEXT" | Text |
| "_TEXT_COUNT" | Integer |
| "_TOOLTIP_TEXT" | Text |
| "_VARIABLE_NAMES" | Text vector |
| "_VISIBLE" | Boolean |
| "_WRAP_ENABLED" | Boolean |

45.1.4 Predefined Methods

| Name | Type |
|------------------------|-----------------|
| "_APPEND_TEXT" | No return value |
| "_DELETE_TEXT" | No return value |
| "_DELETE_TEXT_AT" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_STRING_WIDTH" | Integer |
| "_GET_TEXT_AT" | Text |
| "_GET_TEXT_INDEX" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_SELECT_TEXT" | No return value |
| "_SELECT_TEXT_AT" | No return value |
| "_SET_TEXT_AT" | No return value |
| "_SORT_LIST" | No return value |

45.2 Details

45.2.1 Action Attributes

_GET_TABLE_EDIT_TEXT

Setting this attribute specifies the action method GET_TABLE_EDIT_TEXT.

Data type: Text vector

_GET_TABLE_VIEW_TEXT

Setting this attribute specifies the action method GET_TABLE_VIEW_TEXT.

Data type: Text vector

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

Example:

```
.SET MY_OBJECT._NOTIFY = ("ROOT.OPEN_EDIT_DIALOG", "")
```

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

_UPDATE_TABLE_EDIT_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_EDIT_TEXT.

Data type: Text vector

_UPDATE_TABLE_VIEW_TEXT

Setting this attribute specifies the action method UPDATE_TABLE_VIEW_TEXT.

Data type: Text vector

45.2.2 Action Methods**GET_TABLE_EDIT_TEXT**

This action method is executed whenever a table needs a value from an edit object. This method is executed just before the edit object of the cell is closed (just before the CELL_EDITING_ACCEPTED action method). This function can be used, for example, to convert internal edit object value to a text value, which is stored in the cell. This action method is called with two parameters: ROW and COLUMN of the current cell. This action method should return a text string.

Data type to be returned: Text

GET_TABLE_VIEW_TEXT

This action method is executed whenever a table needs the value for the view object of the current cell. For example, this method is executed when the view object is drawn into the cell. This function can be used, for example, to convert edit object value of the cell to the text value of the cell.

Data type to be returned: Text

NOTIFY

Specifies the notification procedure of a dialog item. The notification procedure gets called for different reasons with different dialog items. See the general object description for a description of NOTIFY for a specific object.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

UPDATE_TABLE_EDIT_TEXT

This action method is executed whenever the edit object of a cell needs a value to be shown. This method is executed just before the editing of the cell is started (just after the CELL_EDITING_STARTED action method). This action method has three parameters: ROW and COLUMN of the cell and the text of the cell. This action method can be used to convert the text value of the view object to the edit object value.

Data type to be returned: No return value

UPDATE_TABLE_VIEW_TEXT

This action method is executed whenever a table needs a value for the view object. This method is executed just before the view object of the cell is drawn. This function can be used, for example, to convert the text value of the cell to the current value of the view object, which is shown in the view object. This action method has three parameters: ROW and COLUMN of the cell and the TEXT of the cell.

Data type to be returned: No return value

45.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_AUTO_SORT_ENABLED

Specifies whether the auto sort function is used or not. The texts in text spinner are sorted automatically when set to TRUE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether the dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_MODIFIABLE

Specifies whether the value of a text spinner is modifiable or not. A change of appearance is also often combined with this attribute.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_INDEX

Specifies the index of a selected text string in the text spinner.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TEXT

Specifies the current text in text spinner.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | Read-only |

_TEXT_COUNT

Specifies the number of texts in text spinner.

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | 0 |
| Access: | Read-only |

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

_WRAP_ENABLED

Specifies whether the values displayed in the numeric spinner can increase or decrease straight from one end point of the spinner to another end point, for example from the maximum value to the minimum value. Maximum and minimum values of the numeric spinner are specified by using attributes _MIN_VALUE and _MAX_VALUE.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

45.2.4 Predefined Methods

_APPEND_TEXT

Appends a text string to the text spinner. If the attribute _AUTO_SORT_ENABLED is set to FALSE then the text is appended to the end of the list. Otherwise, the text spinner is sorted automatically.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| TEXT | Text A character string containing up to 255 characters. |

_DELETE_TEXT

Deletes the specified text string if it exists in the text spinner.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| TEXT | Text A character string containing up to 255 characters. |

_DELETE_TEXT_AT

Deletes the text string at a specific index.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_GET_TEXT_AT

Returns the text string at a specified index.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_GET_TEXT_INDEX

Returns the index at a specified text string.

| | |
|-------------------------|---|
| Return value data type: | Integer |
| Parameter(s) | |
| TEXT | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_SELECT_TEXT

Selects the specified text string in the text spinner.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| TEXT | Text A character string containing up to 255 characters. |

_SELECT_TEXT_AT

Selects the text string at the specified index in the text spinner.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| INDEX | Integer An integer value. |

_SET_TEXT_AT

Sets the text string at the specified index in the text spinner.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| TEXT | Text A character string containing up to 255 characters. |
| INDEX | Integer An integer value. |

_SORT_LIST

Sorts the texts in the text spinner. Note that when appending or deleting text into list, the list is not automatically sorted. Use attribute `_AUTO_SORT_ENABLED` instead.

Return value data type: No return value

Section 46 VS_TOGGLE_MENU_ITEM

46.1 Overview

A VS_TOGGLE_MENU_ITEM is a special kind of menu item that has a state. It has an _ON attribute that can be TRUE or FALSE. The indication of the state can vary according to the look-and-feel on different platforms. The VS_TOGGLE_MENU_ITEM also has an attribute _EXCLUSIVE, which indicates that only one toggle menu item in a group can be in the _ON state at a time. The group of VS_TOGGLE_MENU_ITEMS must be separated from other menu items using the VS_SEPARATOR_MENU_ITEM.

46.1.1 Action Attributes

| Name | Type |
|----------|-------------|
| "NOTIFY" | Text vector |

46.1.2 Action Methods

| Name | Type |
|----------|-----------------|
| "NOTIFY" | No return value |

46.1.3 Predefined Attributes

| Name | Type |
|--------------------|-------------|
| "ATTRIBUTE_NAMES" | List |
| "CHILD_OBJECTS" | Text vector |
| "ENABLED" | Boolean |
| "EXCLUSIVE" | Boolean |
| "FILE_REVISION" | Text |
| "FONT" | Font |
| "FOREGROUND_COLOR" | Color |
| "MNEMONIC" | Text |
| "OBJECT_CLASS" | Text |
| "OBJECT_NAME" | Text |
| "ON" | Boolean |
| "SOURCE_FILE_NAME" | Text |
| "TITLE" | Text |
| "VARIABLE_NAMES" | Text vector |

46.1.4 Predefined Methods

| Name | Type |
|--------------------------------|-----------------|
| " <u>FLAG_FOR_EXECUTION</u> " | No return value |
| " <u>GET_HELP</u> " | No return value |
| " <u>QUEUE_FOR_EXECUTION</u> " | No return value |

46.2 Details

46.2.1 Action Attributes

_NOTIFY

Setting this attribute specifies the action method NOTIFY.

Data type: Text vector

46.2.2 Action Methods

NOTIFY

Specifies the NOTIFY action method for the menu item. This is executed when the menu item is selected.

Data type to be returned: No return value

46.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_CHILD_OBJECTS

For more information, see [Section 3](#).

_ENABLED

Specifies whether a menu item is enabled or not. Changing this attribute also affects the appearance of the menu item.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_EXCLUSIVE

If this attribute is TRUE, it indicates that the toggle menu item is part of a group of menu items, of which only one should be checked at a time. The _ON attribute, which is automatically set by the system, specifies whether the menu item is checked or not. The group of menu items must be separated within the menu with separator menu items.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_FILE_REVISION

For more information, see [Section 3](#).

_FONT

For more information, see [Section 3](#).

_FOREGROUND_COLOR

For more information, see [Section 3](#).

_MNEMONIC

Specifies the mnemonic character of a menu item. A change of appearance is also often combined with this attribute. In Microsoft Windows platform, the setting of this attribute underlines the mnemonic character in menu item.

| | |
|----------------|--|
| Description: | A character string containing 1 character. |
| Data type: | Text |
| Default value: | ASCII(0) |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_ON

The _ON attribute specifies whether a toggle menu item is checked or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TITLE

Specifies the title of a menu item.

| | |
|----------------|---|
| Description: | A character string containing up to 255 characters. |
| Data type: | Text |
| Default value: | Empty text |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

46.2.4 Predefined Methods

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

Section 47 VS_TREE

47.1 Overview

A VS_TREE is an object used to display hierarchical data. The behavior of the tree is defined through its different action methods. The most important action method is HANDLE_EXPAND_NODE that is run whenever the user tries to open a specific node. All nodes or leafs (except the root node) that is to be displayed must be added within this method. When the user tries to open/close a node or selects a leaf, the row number is passed as argument. The row number indexing begins from 1.

47.1.1 Action Attributes

| Name | Type |
|------------------------|-------------|
| "HANDLE_COLLAPSE_NODE" | Text vector |
| "HANDLE_EXPAND_NODE" | Text vector |
| "LEAF_SELECTED" | Text vector |
| "NODE_SELECTED" | Text vector |
| "NONE_SELECTED" | Text vector |
| "TOOLTIP_IS_SHOWN" | Text vector |

47.1.2 Action Methods

| Name | Type |
|------------------------|-----------------|
| "HANDLE_COLLAPSE_NODE" | Boolean |
| "HANDLE_EXPAND_NODE" | Boolean |
| "LEAF_SELECTED" | No return value |
| "NODE_SELECTED" | No return value |
| "NONE_SELECTED" | No return value |
| "TOOLTIP_IS_SHOWN" | No return value |

47.1.3 Predefined Attributes

| Name | Type |
|------------------------|-----------------|
| "ATTRIBUTE_NAMES" | List |
| "BACKGROUND_COLOR" | Color |
| "CHILD_OBJECTS" | Text vector |
| "CLICK_FOCUSABLE" | Boolean |
| "DEFAULT_CLOSED_IMAGE" | Image reference |
| "DEFAULT_LEAF_IMAGE" | Image reference |
| "DEFAULT_OPEN_IMAGE" | Image reference |
| "ENABLED" | Boolean |
| "FILE_REVISION" | Text |

Table continues on next page

| | |
|----------------------------------|----------------|
| "_FIRST_VISIBLE_ROW" | Integer |
| "_FOCUSABLE" | Boolean |
| "_FOCUSED" | Boolean |
| "_FONT" | Font |
| "_FOREGROUND_COLOR" | Color |
| "_GEOMETRY" | List |
| "_GRABBING_HAND_ENABLED" | Boolean |
| "_HAS_BORDER" | Boolean |
| "_HORIZONTAL_SCROLL_BAR_ENABLED" | Boolean |
| "_KEYBOARD_SELECTION_ENABLED" | Boolean |
| "_KEYBOARD_TRAVERSAL_ENABLED" | Boolean |
| "_OBJECT_CLASS" | Text |
| "_OBJECT_NAME" | Text |
| "_OPEN_ICON_ON_SELECT" | Boolean |
| "_POP_UP_MENU" | Menu reference |
| "_ROW_COUNT" | Integer |
| "_SCIL_GEOMETRY" | List |
| "_SELECTED_ROW" | Integer |
| "_SOURCE_FILE_NAME" | Text |
| "_TOOLTIP_TEXT" | Text |
| "_TRAVERSAL_WRAP_ENABLED" | Boolean |
| "_VARIABLE_NAMES" | Text vector |
| "_VERTICAL_SCROLL_BAR_ENABLED" | Boolean |
| "_VISIBLE" | Boolean |

47.1.4 Predefined Methods

| Name | Type |
|-------------------------------|-----------------|
| "_ADD_LEAF" | No return value |
| "_ADD_NODE" | No return value |
| "_ADD_ROOT_NODE" | No return value |
| "_COLLAPSE_NODE" | No return value |
| "_EXPAND_NODE" | No return value |
| "_FLAG_FOR_EXECUTION" | No return value |
| "_GET_HELP" | No return value |
| "_GET_ROW_ID" | Text |
| "_GET_ROW_TEXT" | Text |
| "_GET_STRING_WIDTH" | Integer |
| "_QUEUE_FOR_EXECUTION" | No return value |
| "_REMOVE_ROW" | No return value |
| "_SCROLL_ROW_INTO_VIEW" | No return value |
| "_SCROLL_SELECTION_INTO_VIEW" | No return value |

Table continues on next page

| | |
|-------------------------|-----------------|
| <u>"_SET_ROW_ID"</u> | No return value |
| <u>"_SET_ROW_IMAGE"</u> | No return value |
| <u>"_SET_ROW_TEXT"</u> | No return value |

47.2 Details

47.2.1 Action Attributes

_HANDLE_COLLAPSE_NODE

Setting this attribute specifies the action method HANDLE_COLLAPSE_NODE.

Data type: Text vector

Example:

```
.SET ROOT\ MY_TREE._HANDLE_COLLAPSE_NODE = ("ROOT.CLOSENODE (ARGUMENT(1))",
"...
("ROOT.CLOSENODE (ARGUMENT(1))", "...
```

_HANDLE_EXPAND_NODE

Setting this attribute specifies the action method HANDLE_EXPAND_NODE.

Data type: Text vector

Example:

```
.SET ROOT\ MY_TREE._HANDLE_EXPAND_NODE = ("ROOT.OPENNODER (ARGUMENT(1))",
"..."
```

_LEAF_SELECTED

Setting this attribute specifies the action method LEAF_SELECTED.

Data type: Text vector

_NODE_SELECTED

Setting this attribute specifies the action method NODE_SELECTED.

Data type: Text vector

_NONE_SELECTED

Setting this attribute specifies the action method NONE_SELECTED.

Data type: Text vector

_TOOLTIP_IS_SHOWN

Setting this attribute specifies the action method TOOLTIP_IS_SHOWN.

Data type: Text vector

47.2.2 Action Methods

HANDLE_COLLAPSE_NODE

This action method is run when the user collapses an opened node by double-clicking it. This method gets one input parameter, which is the row number of the node that is being collapsed. If the node can be collapsed, this method should return TRUE, otherwise FALSE.

Data type to be returned: Boolean

HANDLE_EXPAND_NODE

This action method is run when the user expands a closed node by double-clicking it. This method gets one input parameter, which is the row number of the node that is being expanded. Within this method, the user should add any nodes or leafs that need to be displayed when the node is expanded. If the node can be expanded, this method should return TRUE otherwise FALSE.

Data type to be returned: Boolean

LEAF_SELECTED

This action method is run when a leaf is selected. The method gets one input parameter, which is the row number of the leaf that has been selected.

Data type to be returned: No return value

NODE_SELECTED

This action method is run when a node is selected. The method gets one input parameter, which is the row number of the node that has been selected.

Data type to be returned: No return value

NONE_SELECTED

This action method is run when all nodes and leafs in the tree are unselected.

Data type to be returned: No return value

TOOLTIP_IS_SHOWN

This action method is executed whenever a tooltip is shown over a dialog item.

Data type to be returned: No return value

47.2.3 Predefined Attributes

_ATTRIBUTE_NAMES

For more information, see [Section 3](#).

_BACKGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 65535, 65535)
```

_CHILD_OBJECTS

For more information, see [Section 3](#).

_CLICK_FOCUSABLE

Specifies whether a dialog item receives focus when it is clicked.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._CLICK_FOCUSABLE = FALSE
```

_DEFAULT_CLOSED_IMAGE

Specifies the default image to be used to visualise a closed node. If a default image is not set, the system provides a standard image. The most frequently used image for closed node situation is preferred to use as a default image.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Not applicable |
| Access: | Write-only |

Example:

```
.LOAD ROOT\ CLOSEDIMAGE = VS_IMAGE("...")  
.SET ROOT\ MY_TREE._DEFAULT_CLOSED_IMAGE = "ROOT\ CLOSEDIMAGE"
```

_DEFAULT_LEAF_IMAGE

Specifies the default image to be used to visualise a leaf. If a default image is not set, the system provides a standard image. The most frequently used image for leaf is preferred to use as a default image.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Not applicable |
| Access: | Write-only |

_DEFAULT_OPEN_IMAGE

Specifies the default image to be used to visualise an open node. If a default image is not set, the system provides a standard image. The most frequently used image for open node situation is preferred to use as a default image.

| | |
|----------------|---|
| Description: | A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |
| Data type: | Image reference |
| Default value: | Not applicable |
| Access: | Write-only |

_ENABLED

Specifies whether a dialog item responds to various events. A change of appearance is also often combined with this attribute. When setting this attribute to FALSE, the disabled dialog item is displayed as dimmed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._ENABLED = FALSE
```

_FILE_REVISION

For more information, see [Section 3](#).

_FIRST_VISIBLE_ROW

Contains the index of the uppermost visible row.

| | |
|----------------|---|
| Description: | An integer from 1 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | 1 |
| Access: | Read-only |

_FOCUSABLE

Specifies whether a dialog item receives focus during normal keyboard traversal.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._FOCUSABLE = FALSE
```

_FOCUSED

Contains the value TRUE if a dialog item has keyboard focus.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | Environment dependent |
| Access: | Read-only |

Example:

```
@HAS_FOCUS = MY_OBJECT._FOCUSSED
```

_FONT

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._FONT = ("M", 1)
```

_FOREGROUND_COLOR

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._BACKGROUND_COLOR = (65535, 0, 0)
```

_GEOMETRY

For more information, see [Section 3](#).

Example:

```
.SET MY_OBJECT._GEOMETRY = LIST(X = 10, Y = 10, W = 100, H = 100)
```

_GRABBING_HAND_ENABLED

Specifies whether clicking and dragging the mouse in a view invokes the Grabbing hand or simply selects objects. The Grabbing hand tracks the mouse while the mouse button is down and scrolls the view as if the user has grabbed it, so that the scrolling of the view and the movement of the mouse are synchronous.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_HAS_BORDER

Determines whether a dialog item should be drawn with a border or not. The exact interpretation of this attribute is dependent on the object type.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._HAS_BORDER = FALSE
```

_HORIZONTAL_SCROLL_BAR_ENABLED

Specifies whether an object has a horizontal scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_KEYBOARD_SELECTION_ENABLED

Specifies whether an object allows keyboard selection through keyboard focus, FALSE otherwise. This means, for example, that when a list has focus and a key is pressed, the selected cell of the list changes to the first one that starts with the character pressed.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_KEYBOARD_TRAVERSAL_ENABLED

Specifies if a keyboard traversal is used in list elements or not. If the value is TRUE, the object responds to keyboard commands. In this case, the user can make selections or change the focused cell with keyboard arrow keys. If the value is FALSE, the object will only handle the mouse events. In this case, the user can make selections or change the focused cell only with mouse.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_OBJECT_CLASS

For more information, see [Section 3](#).

_OBJECT_NAME

For more information, see [Section 3](#).

_OPEN_ICON_ON_SELECT

Specifies whether a tree should change icon when a node is opened.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | FALSE |
| Access: | No limitations |

_POP_UP_MENU

Specifies the popup menu that is associated with an object. This menu is activated by right-clicking the object.

| | |
|----------------|---|
| Description: | A reference to a VS_MENU object. A VS_MENU can be created using the SCIL function .CREATE or loaded from a file using the function .LOAD. |
| Data type: | Menu reference |
| Default value: | Empty text |
| Access: | No limitations |

Example:

```
.LOAD ROOT\ MENU1 = VS_MENU("...")

.SET ROOT\ MY_LIST._POP_UP_MENU = "ROOT\ MENU1"
```

_ROW_COUNT

Read only attribute that contains the number of currently visible tree items in the tree structure (the number of rows in the tree).

| | |
|----------------|-------------------|
| Description: | An integer value. |
| Data type: | Integer |
| Default value: | Not applicable |
| Access: | Read-only |

_SCIL_GEOMETRY

For more information, see [Section 3](#).

_SELECTED_ROW

Specifies the row number of the row in the tree that is currently selected.

| | |
|----------------|---|
| Description: | An integer from 0 to the length of a SCIL vector. |
| Data type: | Integer |
| Default value: | Not applicable |
| Access: | No limitations |

_SOURCE_FILE_NAME

For more information, see [Section 3](#).

_TOOLTIP_TEXT

Specifies the tooltip text that appears over an object when the cursor is held over the object.

Data type: Text

Example:

```
.SET MY_OBJECT._TOOLTIP_TEXT = "Tool Properties"
```

_TRAVERSAL_WRAP_ENABLED

Specifies whether the content of a list wraps around or not.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VARIABLE_NAMES

For more information, see [Section 3](#).

_VERTICAL_SCROLL_BAR_ENABLED

Specifies whether an object has a vertical scroll bar.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

_VISIBLE

Specifies whether a dialog item is visible or not. Setting this attribute to FALSE may be useful if the dialog item should be invisible when opening the dialog.

| | |
|----------------|---------------------------|
| Description: | A value of TRUE or FALSE. |
| Data type: | Boolean |
| Default value: | TRUE |
| Access: | No limitations |

Example:

```
.SET MY_OBJECT._VISIBLE = FALSE
```

47.2.4 Predefined Methods

_ADD_LEAF

Adds a leaf to the tree. A leaf cannot contain any sub-levels of nodes or leafs. The first parameter specifies the row number of the parent node, and the second parameter specifies the index where to insert the new leaf within the parent node.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| PARENT_ROW | Integer An integer value. |
| POSITION | Integer An integer value. |

_ADD_NODE

Adds a node to the tree. A node can contain more nodes or leafs. The first parameter specifies the row number of the parent node, and the second parameter specifies the index where to insert the new node within the parent node.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| PARENT_ROW | Integer An integer value. |
| POSITION | Integer An integer value. |

_ADD_ROOT_NODE

Adds a root node to the tree. Note that every tree object must contain a root node.

Return value data type: No return value

_COLLAPSE_NODE

Collapses the specified node.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |

_EXPAND_NODE

Expands the specified node.

| | |
|-------------------------|------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |

_FLAG_FOR_EXECUTION

For more information, see [Section 3](#).

_GET_HELP

For more information, see [Section 3](#).

_GET_ROW_ID

Returns the identification string associated with the specified row in the tree.

| | |
|-------------------------|------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer value. |

_GET_ROW_TEXT

Returns the text associated with the specified row in the tree.

| | |
|-------------------------|-------------------------------------|
| Return value data type: | Text |
| Parameter(s) | |
| ROW | Integer An integer value. |

_GET_STRING_WIDTH

Returns the width of the given string in the current font of the object.

| | |
|-------------------------|--|
| Return value data type: | Integer |
| Parameter(s) | |
| STRING | Text A character string containing up to 255 characters. |

_QUEUE_FOR_EXECUTION

For more information, see [Section 3](#).

_REMOVE_ROW

Removes the specified row.

| | |
|-------------------------|-------------------------------------|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |

_SCROLL_ROW_INTO_VIEW

Scrolls the specified row so that it becomes at least partly visible.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer from 1 to the length of a SCIL vector. |

_SCROLL_SELECTION_INTO_VIEW

Scrolls the list so that the selection is displayed in the view.

Return value data type: No return value

_SET_ROW_ID

Sets the identification string of a specified row. Identification can be any valid text string. The ID is not used internally in any way, so the SCIL programmer may use it freely.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |
| ID | Text A character string containing up to 255 characters. |

_SET_ROW_IMAGE

Sets the image associated with a specified row. If no image is specified, a default image is used.

| | |
|-------------------------|--|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |
| IMAGE | Image reference A reference to a VS_IMAGE object. A VS_IMAGE can be created using the SCIL function .CREATE or loaded using the function .LOAD. |

_SET_ROW_TEXT

Sets the text associated with the specified row.

| | |
|-------------------------|---|
| Return value data type: | No return value |
| Parameter(s) | |
| ROW | Integer An integer value. |
| TEXT | Text A character string containing up to 255 characters. |

Index

| | | |
|-------------------------------|-----------------------------|---|
| P | Predefined attributes..... | 17 |
| | Predefined methods..... | 17, 21 |
| R | | |
| | ROW..... | 36 |
| T | | |
| | TEXT..... | 36 |
| | TOOLTIP_IS_SHOWN..... | 27, 36, 45, 53, 66, 121, 134, 142, 156, 175, 220, 226, 232, 280, 287, 315, 343, 365, 410, 459, 475, 489, 504 |
| U | | |
| | UPDATE_TABLE_EDIT_TEXT..... | 27, 36, 45 |
| | UPDATE_TABLE_VIEW_TEXT..... | 27, 36, 46 |
| V | | |
| | VS_BOX..... | 25 |
| | VS_BUTTON..... | 33 |
| | VS_CHECK_BOX..... | 43 |
| | VS_IMAGE..... | 33 |
| ACCEPT | | 52 |
| ALIGNMENT | | 36 |
| ATTRIBUTE_NAMES | | 17, 28, 36 |
| BACKGROUND_COLOR | | 18, 28, 36 |
| BOTTOM_SHADOW_COLOR | | 226 |
| CHILD_OBJECTS | | 18, 28, 37 |
| CLICK_FOCUSABLE | | 37 |
| ENABLED | | 28, 37 |
| FILE_REVISION | | 18, 28, 37 |
| FLAG_FOR_EXECUTION | | 21, 31, 42 |
| FOCUSABLE | | 37 |
| FOCUSED | | 38 |
| FONT | | 19, 28, 38 |
| FOREGROUND_COLOR | | 19, 28, 38, 226 |
| GEOMETRY | | 19, 29, 38 |
| GET_HELP | | 22, 31, 42 |
| GET_STRING_WIDTH | | 31, 42 |
| GET_TABLE_EDIT_TEXT | | 25, 26, 34, 44 |
| GET_TABLE_VIEW_TEXT | | 25, 26, 34, 44 |
| HAS_BORDER | | 29, 38 |
| HAS_IMAGE_MASK | | 29, 39 |
| IMAGE | | 29, 39 |
| LEADER_DIALOG_ITEM | 93, 108, 194, 210, 253, | 303, 354 |
| LINE_STYLE | | 227 |
| MAX_HEIGHT | | 19 |
| MAX_WIDTH | | 20 |
| MIN_HEIGHT | | 20 |
| MIN_WIDTH | | 20 |
| MNEMONIC | | 39 |
| NOTIFY | | 34, 44 |
| OBJECT_CLASS | | 20, 30, 40 |
| OBJECT_NAME | | 20, 30, 40 |
| POP_UP_MENU | | 40 |
| QUEUE_FOR_EXECUTION | | 22, 31, 42 |
| SCIL_GEOMETRY | | 21, 30, 40 |
| SELECT | | 52 |
| SET_MAX_SIZE | | 23 |
| SET_MIN_SIZE | | 23 |
| SOURCE_FILE_NAME | | 21, 30, 40 |
| TITLE | | 30, 40 |
| TITLE_VECTOR | | 41 |
| TOOLTIP_IS_SHOWN | | 25, 26, 35, 44, 52, 65, 77, 120, 134, 141, 155, 175, 220, 226, 231, 280, 286, 314, 342, 364, 409, 458, 473, 489, 503 |
| TOOLTIP_TEXT | | 30, 41, 48, 58, 71, 83, 124, 136, 146, 162, 182, 222, 227, 239, 283, 291, 319, 347, 370, 381, 397, 421, 466, 480, 493 |
| TOP_SHADOW_COLOR | | 227 |
| UPDATE_TABLE_EDIT_TEXT | | 25, 26, 35, 45 |
| UPDATE_TABLE_VIEW_TEXT | | 25, 26, 35, 45 |
| VARIABLE_NAMES | | 21, 30, 41 |
| VISIBLE | | 30, 41 |
| C | | |
| COLUMN | | 36 |
| G | | |
| GET_TABLE_EDIT_TEXT | | 27, 35, 45 |
| GET_TABLE_VIEW_TEXT | | 27, 35, 45 |
| N | | |
| NOTIFY | | 33, 35, 45 |

Hitachi ABB Power Grids
Grid Automation Products
PL 688
65101 Vaasa, Finland



Scan this QR code to visit our website

<https://hitachiabb-powergrids.com/microscadax>