



SpatialAnalyzer

Measurement Plan Command Reference



SpatialAnalyzer

Measurement Plan Command Reference

2022.11.10

SPATIALANALYZER

MP COMMAND REFERENCE

Hexagon Metrology, inc.

11/10/2022. Hexagon Metrology, inc. All rights reserved. Copyright ©1994-2022.

Your rights to the software are governed by the accompanying software license agreement. The owner or authorized user of a valid copy of SpatialAnalyzer software may reproduce this publication for the purpose of learning to use such software. No part of this publication may be reproduced or transmitted for commercial purposes, such as selling copies of this publication or for providing paid-for support services.

Every effort has been made to ensure that the information in this manual is accurate. Hexagon Metrology is not responsible for printing or clerical errors.

Because Hexagon Metrology frequently releases new versions and updates to its applications and Internet sites, images shown in this book may be slightly different from what you see on your screen.

Hexagon Metrology, inc.
436 McLaws Circle
Williamsburg, VA 23185
+1 (757) 565-1500
www.kinematics.com

SpatialAnalyzer and New River Kinematics are trademarks of Hexagon Metrology, inc. , registered in the U.S.

Windows is a registered trademark of Microsoft Corporation.

iPhone and iPod are registered trademarks of Apple Inc.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Hexagon Metrology assumes no responsibility with regard to the performance or use of these products.

Table Of Contents

| | |
|--|-----------|
| TABLE OF CONTENTS | V |
| FUNDAMENTAL TERMS | 1 |
| FILE OPERATIONS | 9 |
| New SA File | 10 |
| Open SA File | 11 |
| Save | 12 |
| Save As..... | 13 |
| Backup Now..... | 14 |
| Open Template File | 15 |
| Save As Read-Only Template | 16 |
| Exit Measurement Plan..... | 17 |
| Shut Down SA..... | 18 |
| Run Another Program | 19 |
| Run Powershell Script | 20 |
| Copy General File..... | 22 |
| Rename General File..... | 23 |
| Delete General File | 24 |
| Verify General File Exists..... | 25 |
| Verify MP File Exists..... | 26 |
| Browse For Directory | 27 |
| Browse For File..... | 28 |
| Find Files in Directory..... | 29 |
| Get Directory and Filename from Path..... | 30 |
| File Import..... | 31 |
| Import SA File..... | 32 |
| Import ASCII: Predefined Formats | 33 |
| Import ASCII: Predefined Frame Set Formats | 34 |
| Import E57 File..... | 35 |
| Import STL File..... | 36 |
| Import STEP File | 37 |
| Import IGES File..... | 38 |
| Import VDA/FS File..... | 39 |
| Import SAT File..... | 40 |
| Import File As Embedded File..... | 41 |
| Import MP File As Embedded MP | 42 |
| Import File as Picture..... | 43 |
| Direct CAD Access | 44 |
| Import SA Windows Placement..... | 46 |
| Import VSTARS .xyz File | 47 |

| | |
|--|-----------|
| Import VSTARS Cameras..... | 48 |
| Import Leica GSI File..... | 49 |
| Import Leica SDB File..... | 50 |
| Import Hidden Point Bar XML File..... | 51 |
| File Export..... | 53 |
| Export ASCII Points..... | 54 |
| Export ASCII Point Sets..... | 55 |
| Export ASCII Frame Set..... | 56 |
| Export ASCII Frames..... | 57 |
| Export ASCII Point Clouds..... | 58 |
| Export PTX Point Clouds..... | 59 |
| Export Vector Container to Excel File..... | 60 |
| Export Vector Container to ASCII File..... | 61 |
| Export STEP File - Entire Model..... | 62 |
| Export STEP File - Partial Model..... | 63 |
| Export IGES File - Entire Model..... | 64 |
| Export IGES File - Partial Model..... | 65 |
| Export VDA/FS File - Entire Model..... | 66 |
| Export VDA/FS File - Partial Model..... | 67 |
| Export Embedded File..... | 68 |
| Export DXF..... | 69 |
| Export Scan Stripe Mesh to STL File..... | 70 |
| Export Hidden Point Bar XML File..... | 71 |
| QDAS File Export... .. | 72 |
| Import QDas Catalog File..... | 73 |
| Get QDAS Catalog Entries..... | 74 |
| Set K-Field from QDAS Catalog..... | 75 |
| Get QDAS Catalog Entry Identifier..... | 76 |
| Prepare QDAS Data List..... | 77 |
| Add K-Field to QDAS Data List..... | 78 |
| Export QDAS Data List..... | 79 |
| Export QDAS Characteristics..... | 80 |
| ASCII Data File Operations..... | 82 |
| Clear All ASCII Files..... | 83 |
| Open ASCII File..... | 84 |
| Write ASCII Line..... | 85 |
| Read ASCII Line (Iterator)..... | 86 |
| Close ASCII File..... | 87 |
| Datashare Operations..... | 88 |
| Load DataShare File..... | 89 |
| Save DataShare File..... | 90 |
| Get Boolean From DataShare File..... | 91 |

| | |
|---|------------|
| Get Integer From DataShare File..... | 92 |
| Get Double From DataShare File..... | 93 |
| Get String From DataShare File..... | 94 |
| Get Vector From DataShare File..... | 95 |
| Get Transform From DataShare File..... | 96 |
| Set Boolean In DataShare File..... | 97 |
| Set Integer In DataShare File..... | 98 |
| Set Double In DataShare File..... | 99 |
| Set String In DataShare File..... | 100 |
| Set Vector In DataShare File..... | 101 |
| Set Transform In DataShare File..... | 102 |
| Load HTML Form..... | 103 |
| Database Operations..... | 106 |
| Put to ODBC Database..... | 107 |
| Get from ODBC Database..... | 108 |
| Delete from ODBC Database..... | 109 |
| XML..... | 110 |
| Open XML File..... | 111 |
| Get XML Attribute..... | 112 |
| Set XML Attribute..... | 114 |
| Close XML File..... | 115 |
| Import Nominals from XML File..... | 116 |
| Merge Measurements into XML File..... | 117 |
| PROCESS FLOW OPERATIONS..... | 119 |
| Wait for Steps to Complete..... | 120 |
| Jump To Step..... | 121 |
| Jump To Other Measurement Plan..... | 122 |
| Step Status Test..... | 123 |
| Jump Based on Ranged Status Test..... | 124 |
| Go/No Go - Range Check Results..... | 125 |
| Object Existence Test..... | 126 |
| Object Existence Test (Check Only)..... | 127 |
| Collection Existence Test..... | 128 |
| Create Counter..... | 129 |
| Increment Counter..... | 130 |
| Decrement Counter..... | 131 |
| Reset Counter..... | 132 |
| Ask for String..... | 133 |
| Ask for String (Pull-Down Version)..... | 134 |
| Ask for Point Name..... | 135 |
| Ask for Integer..... | 136 |
| Ask for Double..... | 137 |

| | |
|---|-----|
| Ask for User Decision Extended | 138 |
| Ask for User Decision(HTML) | 139 |
| Ask for User Decision (Pull-Down Version) | 140 |
| Ask for User Decision from Strings..... | 141 |
| Ask for User Decision from Image | 142 |

MP TASK OVERVIEW 145

| | |
|---------------------------------------|-----|
| Create/Clear Task Overview List | 146 |
| Set Overview Title..... | 147 |
| Set Overview Image | 148 |
| Add Task Overview Item | 149 |
| Set Task Item Status | 150 |
| Set Task Item Name..... | 151 |
| Set Task Item Comment | 152 |
| Show Progress for Task Item | 153 |
| Set Task Item Completion Values..... | 154 |
| Set Current Task..... | 155 |
| Show Task Overview List..... | 156 |

VIEW CONTROL 157

Ribbon Bar..... 158

| | |
|-------------------------------------|-----|
| Load Ribbon Bar from XML File | 159 |
| Reset Ribbon Bar to Default..... | 160 |

Point of View..... 161

| | |
|--|-----|
| Set Point of View From Frame | 162 |
| Set Point of View From Instrument Updates..... | 163 |
| Set Point of View | 164 |
| Save Point of View | 165 |
| Define Point of View | 166 |
| Get Point of View Parameters..... | 167 |

Hide/Show Operations..... 168

| | |
|---|-----|
| Show Labels..... | 170 |
| Show Objects | 171 |
| Show by Object Type..... | 172 |
| Show/Hide by Object Type | 174 |
| Show/Hide Points..... | 175 |
| Show/Hide Dimensions..... | 176 |
| Show/Hide Callout View..... | 177 |
| Hide All Callout View | 178 |
| Show/Hide Annotations for Feature Checks..... | 179 |
| Show/Hide Annotations for Datums | 180 |
| Show/Hide Instruments | 181 |

| | |
|--|------------|
| Show/Hide Instrument Probe Tip | 182 |
| Show/Hide Relationship Report..... | 183 |
| Show/Hide Relationship Watch..... | 184 |
| Show Items in Tree | 185 |
| Highlight Operations | 186 |
| Highlight Objects | 187 |
| Highlight Relationships..... | 188 |
| Highlight Point | 189 |
| Colors | 190 |
| Set Working Color..... | 191 |
| Set Working Color Auto Increment | 192 |
| Set Background Color | 193 |
| Set Object(s) Color | 194 |
| Get Object Color | 195 |
| Convert RGB Values to Integer..... | 196 |
| Convert Integer Values to RGB..... | 197 |
| Refresh Views | 198 |
| Auto-Scale | 199 |
| Set Target Labels Use Full Names | 200 |
| Set Render Mode Type..... | 201 |
| Set View Clipping Plane..... | 202 |
| Get View Clipping Plane..... | 203 |
| Set SA's Window State | 204 |
| Set SA's Window Pos | 205 |
| Set SA's Window Size | 206 |
| Set MP's Window State | 207 |
| Center Graphics About Point..... | 208 |
| Center Graphics About Object(s)..... | 209 |
| Set Object(s) Translucency | 210 |
| | |
| CLOUD VIEWER OPERATIONS | 211 |
| Cloud Display Control | 212 |
| Reset Cloud Bounding Box | 213 |
| Get Cloud Point Count..... | 214 |
| Set Cloud Default Clipping Plane | 215 |
| Clear Cloud Viewer | 216 |
| Send Cloud to SA | 217 |
| Set Filter | 218 |
| Save Point Cloud File..... | 219 |
| Load Point Cloud File | 220 |
| | |
| CONSTRUCTION OPERATIONS | 221 |
| Mirror Object(s) | 222 |

| | |
|---|-----|
| Copy Object..... | 223 |
| Copy Objects to a collection..... | 224 |
| Move Objects to a collection..... | 225 |
| Copy Objects - Point to Point Delta | 226 |
| Move Objects - Point to Point Delta..... | 227 |
| Rename Point..... | 228 |
| Rename Points with Name Pattern | 229 |
| Rename Collection..... | 230 |
| Rename Object..... | 231 |
| Rename Item | 232 |
| Delete Points | 233 |
| Delete Points WildCard Selection | 234 |
| Construct Objects From Surface Faces-Runtime Select | 235 |

Collections 236

| | |
|--|-----|
| Set (or construct) default collection..... | 237 |
| Construct Collection..... | 238 |
| Get Active Collection Name..... | 239 |
| Delete Collection | 240 |
| Delete Collections by Wildcard..... | 241 |

Points and Groups 242

| | |
|--|-----|
| Construct Point (Fit to Points)..... | 243 |
| Construct a Point in Working Coordinates | 244 |
| Construct Point From Survey Target Center | 245 |
| Set Point Position in Working Coordinates..... | 246 |
| Transform Points by Delta (About Working Frame)..... | 247 |
| Construct a Point at line MidPoint..... | 248 |
| Construct Point Group from Point Name Ref List | 249 |
| Construct Point Groups from Vector Groups..... | 250 |
| Construct Point Group from Point Cloud..... | 251 |
| Construct Point from Cloud Point - Runtime Select..... | 252 |
| Construct a Point at Circle Center..... | 253 |
| Construct Point at Intersection of Planes | 255 |
| Construct Point at Intersection of Two Lines..... | 256 |
| Construct Point at Intersection of Plane and Line | 257 |
| Construct Point at Intersection of 2 B-Splines | 258 |
| Construct Point at Intersection of B-Spline and Surfaces | 259 |
| Construct Points at Intersection of Circle and Line | 260 |
| Construct Points at Intersection of Principle Object Axes and Surfaces | 261 |
| Get Gradient At Projected Point On Surface..... | 262 |
| Get Gradient At Projected Point On Surface Edge | 263 |
| Construct Points from Cylinder | 265 |
| Construct a Point at Projection of Point onto An Object | 266 |
| Construct Points on Curves Using Max Chordal Deviation | 267 |

| | |
|--|------------|
| Construct Points at Projection on Surfaces - Parallel to WCF Axis | 268 |
| Construct Points at Projection on Surfaces - Radial from WCF Axis | 269 |
| Construct Points at Projection on Surfaces - Spherical from WCF Origin | 270 |
| Construct Points at Projection On Mesh Along Direction | 271 |
| Construct Points Spaced at a Distance on Curves | 272 |
| Construct Points N-Spaced on Curves | 273 |
| Construct Points on Object's Vertices | 274 |
| Construct Points on Surface(s) by Clicking | 275 |
| Construct Points From Surface Faces - Runtime Select | 276 |
| Construct Points From Surfaces On UV Grid | 277 |
| Construct Point at Object Origin | 278 |
| Construct Points Shifted in Working Frame | 279 |
| Construct Points Cylindrically Shifted | 280 |
| Construct Points WildCard Selection | 281 |
| Construct Points Subset with Greatest Spacing | 282 |
| Construct Points Layout on Grid | 283 |
| Construct Points Auto-Correspond 2 groups Proximity | 284 |
| Construct Points Auto-Correspond 2 groups Inter-Point Distance | 285 |
| Average a set of Groups | 286 |
| Copy Groups Excluding Obscured Points | 287 |
| Clear Hidden Point Bar Database | 288 |
| Create Hidden Point Rod | 289 |
| Create Hidden Point | 290 |
| Get Hidden Point Rod Index by Name | 291 |
| Delete Hidden point Rod | 292 |
| Point Clouds | 293 |
| Construct Point Clouds from Existing Point Group | 294 |
| Construct Point Clouds from Existing Cloud Points - Runtime Select | 295 |
| Construct Point Clouds from Existing Clouds - Uniform Spacing | 296 |
| Construct Point Cloud from Existing Clouds | 297 |
| Construct Point Clouds from Visible Cloud Points | 298 |
| Construct Boundary Points from Cloud | 299 |
| Construct Cross Section Cloud | 300 |
| Construct Cross Section Cloud - User Select | 301 |
| Extract Sphere Centers from Point Cloud | 302 |
| Scale Bars | 303 |
| Construct Scale Bar | 304 |
| Lines | 305 |
| Construct Line 2 Points | 306 |
| Construct Line 2 Points (Vector Notation) | 307 |
| Construct Line Normal to Object | 308 |

| | |
|--|------------|
| Construct Line - Project Line to Object Reference Plane | 309 |
| Construct Line - Normal to Object through Point | 310 |
| Construct Line 2 Plane Intersection | 311 |
| Construct Lines From Surface Faces-Runtime Select | 312 |
| Construct Line Center of Slot | 313 |
| Construct Line From Instrument Shot | 314 |
| Planes..... | 315 |
| Construct Plane | 316 |
| Construct Plane, Normal to Object, Through Point..... | 317 |
| Construct Planes, Bounding Point Group..... | 318 |
| Construct Plane, Bisect 2 Planes..... | 319 |
| Shift Plane | 320 |
| Construct Planes From Surface Faces-Runtime Select..... | 321 |
| Circles..... | 322 |
| Construct Circle..... | 323 |
| Construct Circles From Surface Faces-Runtime Select..... | 324 |
| Construct Circles (Lines) From Surfaces | 325 |
| Cylinders | 326 |
| Construct Cylinder | 327 |
| Construct Cylinder From End Points..... | 328 |
| Construct Cylinders From Surface Faces-Runtime Select..... | 329 |
| Spheres..... | 330 |
| Construct Sphere..... | 331 |
| Construct Spheres From Surface Faces-Runtime Select..... | 332 |
| Cones | 333 |
| Construct Cone..... | 334 |
| Construct Cones From Surface Faces-Runtime Select | 335 |
| Ellipsoids..... | 336 |
| Construct Ellipsoid | 337 |
| Construct B-Spline From Intersection of Surfaces..... | 338 |
| B-Splines | 339 |
| Construct B-Spline Fit Options | 340 |
| Construct B-Spline From Points..... | 341 |
| Construct B-Spline From Point Sets | 342 |
| Construct B-Spline From Several B-Splines..... | 343 |
| Construct B-Splines From Surfaces | 344 |
| Construct B-Splines From Lines..... | 345 |
| Construct B-Spline From Intersection of Plane and Surface..... | 346 |
| Construct B-Spline From Intersection of Plane and Mesh..... | 347 |

| | |
|---|------------|
| Surfaces..... | 348 |
| Construct Surfaces From Objects | 349 |
| Construct Surface From B-Splines | 350 |
| Construct Surface From Cylinder..... | 351 |
| Construct Surface From Plane | 352 |
| Construct Surface From Sphere | 353 |
| Construct Surface From Cone..... | 354 |
| Construct Surface From a Collection of Surfaces..... | 355 |
| Construct Surface Fit From Nominal Surfaces and Actual Data | 356 |
| Construct Surface by Dissecting Surface(s) | 357 |
| Construct Surfaces by Dissecting Surfaces from Ref List | 358 |
| Construct Surface From Point Groups | 359 |
| Construct Surfaces By Projecting Points | 360 |
| Construct Surface by offsetting a surface..... | 361 |
| Construct Surface from Annotation Links | 362 |
| Construct Geometry From Surfaces | 363 |
| Polygonized Surfaces..... | 364 |
| Construct Polygonized Surface from Point Clouds | 365 |
| Frames | 366 |
| Construct Frame with Wizard..... | 367 |
| Construct Frame..... | 368 |
| Construct Frame on Instrument Base..... | 369 |
| Construct Frame On Object | 370 |
| Construct Frame, 3 Points..... | 371 |
| Construct Frame, at Point, with working Z, and clocked axis..... | 372 |
| Construct Frame, Pick origin and point on X axis - clock Z along working Z..... | 373 |
| Construct Frame, Known Origin, Object Direction, Object Direction..... | 374 |
| Construct Frame, 3 Planes | 375 |
| Construct Frame - Copy And Make Left Handed..... | 376 |
| Construct Frame - Average of Other Object Frames..... | 377 |
| Construct Frame at Robot Link..... | 378 |
| Construct Frame from Point Measurement Probing Frames | 379 |
| Construct Mirror Cube Frame | 380 |
| Perimeters | 381 |
| Construct Perimeter From Points..... | 382 |
| Vector Groups | 383 |
| Construct a Vector Group - Group to Group Compare | 384 |
| Construct a Vector Group - Area Profile Check | 385 |
| Construct a Vector Group From Vector Name Ref List | 386 |
| Construct a Vector in Working Coordinates (Begin/Delta) | 387 |
| Construct a Vector in Working Coordinates (Begin/Direction/Mag.)..... | 388 |

| | |
|--|------------|
| Construct Vectors WildCard Selection | 389 |
| Construct a Vector Group From a Relationship | 390 |
| GD&T | 391 |
| Make Surface Face List - Runtime Select | 392 |
| Make GD&T Feature Check Annotation | 393 |
| Make GD&T Datum Annotation | 395 |
| Make Feature Checks | 396 |
| Make Annotation Ref List from a Collection | 398 |
| Make Annotation Ref List - WildCard Selection | 399 |
| Make a Feature Check Ref List from a Collection | 400 |
| Make a Feature Check Reference List - WildCard Selection | 401 |
| Delete Feature Checks | 403 |
| Make a Datum Ref List from a Collection | 404 |
| Folders | 405 |
| Construct Folder(s) | 406 |
| Delete Folders by Wildcard | 407 |
| Callouts | 408 |
| Create Vector Callout | 409 |
| Create Min/Max Vector Group Callout | 411 |
| Create Point Callout | 413 |
| Create Point Comparison Callout | 414 |
| Create Relationship Callout | 416 |
| Create Picture Callout | 417 |
| Create Text Callout | 418 |
| Make a Callout View Ref List | 419 |
| Make a Callout View Ref List - WildCard Selection | 420 |
| Add a Callout View to Callout View Ref List | 421 |
| Sort Callout View Ref List | 422 |
| Get Number of Callout Views in Callout View Ref List | 423 |
| Get i-th Callout View From Callout View Ref List | 424 |
| Set Default Callout View Properties | 425 |
| Delete Callout View | 426 |
| Rename Callout View | 427 |
| Auto Arrange Callout View | 428 |
| Other MP Types | 429 |
| Make a Boolean | 430 |
| Make a Integer | 431 |
| Make a Integer From String | 432 |
| Make a Double | 433 |
| Make a Double From an Integer | 434 |
| Make a Double from String | 435 |
| Make a Double List | 436 |

| | |
|---|-----|
| Add Double to Double List | 437 |
| Make a Boolean From an Integer..... | 438 |
| Make a String | 439 |
| Make String from Integer..... | 440 |
| Make String from Double | 441 |
| Make String from Decimal Degrees Angular Value..... | 442 |
| Make an Incremented String..... | 443 |
| Make a System String..... | 444 |
| Concatenate Strings | 446 |
| Make a String Ref List..... | 447 |
| Make a Picture Name Ref List - Runtime Select | 448 |
| Make a String From A String Ref List | 449 |
| Make Strings from a Point Name | 450 |
| Make Strings from a Collection Object Name..... | 451 |
| Make a Point Name from Strings | 452 |
| Make a Point Name - Runtime Select | 453 |
| Make a Point Name - Ensure Unique..... | 454 |
| Make a Point Name Ref List..... | 455 |
| Make a Point Name Ref List From a Group | 456 |
| Make a Point Name Ref List - Runtime Select..... | 457 |
| Make a Point Name Ref List - Wildcard Select | 458 |
| Append Two Point Name Ref Lists | 459 |
| Subtract Two Point Name Ref Lists..... | 460 |
| Make a Vector Name Ref List From a Vector Group..... | 461 |
| Make a Vector Name Ref List - Runtime Select | 462 |
| Make Vector Names Unique in Vector Group..... | 463 |
| Make a Collection Name - Runtime Select | 464 |
| Make a Collection Item Name from Strings..... | 465 |
| Make a Collection Item Name Reference List - Wildcard Selection | 466 |
| Make a Collection Object Name - Runtime Select..... | 468 |
| Make a Collection Object Name - Ensure Unique..... | 469 |
| Make a Collection Object Name Reference List - Runtime Select..... | 470 |
| Make a Collection Object Name Reference List - WildCard Selection | 471 |
| Make a Collection Object Name Ref List - By Type | 472 |
| Make a Collection Object Name Ref List - By Type and Color..... | 473 |
| Make a Collection Object Name Ref List | 474 |
| Append two Collection Object Name Ref Lists | 475 |
| Add a Collection Object Name to a Ref List..... | 476 |
| Make a Collection Object Name Ref List from all Groups in a Collection..... | 477 |
| Make a Collection Instrument Reference List..... | 478 |
| Get Collection Name and Index from Collection Instrument ID | 479 |
| Get Collection Instrument Reference List Variable..... | 480 |
| Set Collection Instrument Reference List Variable..... | 481 |
| Add a Collection Instrument to a Ref List | 482 |

| | |
|--|-----|
| Add Collection Instruments to a Ref List - WildCard Selection | 483 |
| Make a Collection Instrument Reference List - Runtime Select..... | 484 |
| Make a Relationship Reference List-WildCard Selection | 485 |
| Make a Relationship Reference List- Runtime Selection | 486 |
| Make an Event Reference List-Wildcard Selection..... | 487 |
| Append two Relationship Ref Lists | 488 |
| Make a Collection Instrument ID from a Collection and an Integer..... | 489 |
| Make a Collection Instrument ID - Runtime Select | 490 |
| Make a Collection Vector Group Name Ref List - Runtime Select | 491 |
| Make a Collection Machine ID from a Collection and an Integer..... | 492 |
| Make a Report Ref List from a Collection | 493 |
| Make a Report Ref List - Runtime Select | 494 |
| Make a Picture Name Ref List..... | 495 |
| Invert Transform..... | 496 |
| Make a Report Items Ref List | 497 |
| Make a Transform from Doubles (Fixed XYZ) | 498 |
| Make a Transform from Doubles (Matrix Elements)..... | 499 |
| Make a World Transform Operator (from Transform and Scale)..... | 500 |
| Get Working Transform of Object (Fixed XYZ) | 501 |
| Decompose Transform into Doubles (Fixed XYZ) | 502 |
| Decompose Transform into Vectors (Fixed XYZ) | 503 |
| Decompose Transform into Vectors (Origin and Axes) | 504 |
| Decompose Transform into Doubles (Matrix Elements) | 505 |
| Decompose World Transform Operator into Doubles (Fixed XYZ in World) | 506 |
| Decompose Transform into Doubles (Euler XYZ)..... | 507 |
| Decompose Transform into Doubles (Euler ZYX)..... | 508 |
| Decompose Transform into Doubles (Euler ZYZ)..... | 509 |
| Decompose Transform into Doubles (Euler ZXZ)..... | 510 |
| Decompose World Transform Operator into Vectors (Fixed XYZ in World) | 511 |
| Make a Vector from Doubles | 512 |
| Decompose Vector into Doubles..... | 513 |
| Split String into Two Strings | 514 |
| Make a Normalized Vector | 515 |
| Convert to Euler Angles from Fixed Angles | 516 |
| Make Projection Options | 517 |
| Make Axis Identifier from String..... | 518 |
| Make UDP Settings | 519 |

ANALYSIS OPERATIONS 521

| | |
|----------------------------------|-----|
| Set Object Reporting Frame | 522 |
| Get Object Reporting Frame | 523 |
| Re-Compute Calculated Items..... | 524 |
| Get Number of Collections..... | 525 |
| Get i-th Collection Name | 526 |

| | |
|--|-----|
| Get i-th Report Item From Report Items Ref List..... | 527 |
| Rename points based on proximity to reference points..... | 528 |
| Rename points based on inter-point distance to reference points..... | 529 |
| Is Object of Type..... | 530 |
| Get Number of Instruments in Collection Instrument Ref List..... | 531 |
| Get Number of Point Names in Point Name Ref List..... | 532 |
| Get Number of Objects in Collection Object Name Ref List | 533 |
| Get Number of Points in Group..... | 534 |
| Get Number of Frames In Frame Set | 535 |
| Get Number of Points In Point Set..... | 536 |
| Get Number of Strings in String Ref List..... | 537 |
| Get Number of Vectors in Vector Group..... | 538 |
| Get Number of Vectors in Vector Name Ref List | 539 |
| Get Number of characters in a string | 540 |
| Get Vector Group Properties | 541 |
| Auto-Range and Set Vector Group Colorization (Selected) | 543 |
| Auto-Range and Set Vector Group Colorization (All)..... | 544 |
| Set Vector Group Colorization Options (Selected) | 545 |
| Set Vector Group Colorization Options (All) | 546 |
| Get Vector Group Colorization Options..... | 547 |
| Create Point Uncertainty Fields..... | 548 |
| Coordinate | 549 |
| Get Point Coordinate (Cylindrical)..... | 550 |
| Get Point Coordinate (Polar)..... | 551 |
| Get Point Properties | 552 |
| Get Point Tolerance..... | 553 |
| Set Point Properties | 554 |
| Get Point To Point Distance..... | 555 |
| Get Point To Line Distance..... | 556 |
| Get i-th Point Name From Point Name Ref List..... | 557 |
| Get i-th Point Name From Point Name Ref List (Iterator) | 558 |
| Get i-th Object From Collection Object Name Ref List | 559 |
| Get i-th Object From Collection Object Name Ref List (Iterator)..... | 560 |
| Get i-th Instrument From Collection Instrument Ref List..... | 561 |
| Get i-th Point From Group..... | 562 |
| Get Timestamp for i-th Frame in Frame Set | 563 |
| Get Transform for i-th Frame in Frame Set..... | 564 |
| Get Timestamp for i-th Point in Point Set | 565 |
| Get Coordinate for i-th Point in Point Set | 566 |
| Get i-th Vector From Vector Group..... | 567 |
| Get i-th Vector From Vector Name Ref List | 568 |
| Get i-th String From String Ref List..... | 569 |
| Get i-th String From String Ref List (Iterator) | 570 |
| Get Number of Pictures in Picture Name Ref List..... | 571 |

| | |
|---|-----|
| Get i-th Picture From Picture Name Ref List..... | 572 |
| Add a Picture to Picture Name Ref List | 573 |
| Get Number of Reports in Report Ref List | 574 |
| Get i-th Report Report Ref List..... | 575 |
| Add a Report to Report Ref List..... | 576 |
| Remove i-th Object From Collection Object Name Ref List | 577 |
| Remove i-th Point Name From Point Name Ref List..... | 578 |
| Remove i-th String from String Ref List | 579 |
| Append to String Ref List..... | 580 |
| Get Vector From Vector Group By Name | 581 |
| Add a Vector to Vector Name Ref List..... | 582 |
| Delete i-th Vector From Vector Group..... | 583 |
| Delete Vector by Name..... | 584 |
| Delete Vectors..... | 585 |
| Get Sphere Properties..... | 587 |
| Get Circle Properties..... | 588 |
| Set Circle Properties..... | 589 |
| Get Cylinder Properties | 590 |
| Get Plane Properties..... | 591 |
| Get Torus Properties | 592 |
| Set Default Colorization Options | 593 |
| Set Vector Group Display Attributes..... | 594 |
| Get Vector Group Display Attributes | 596 |
| Query Groups to Objects | 598 |
| Query Points to Objects | 599 |
| Query Points to Single Point..... | 600 |
| Query Points to Circle..... | 601 |
| Query Point to Objects | 602 |
| Query Point to Point Along Curve | 603 |
| Query Frame to Frame..... | 604 |
| Transform Objects - Frame To Frame..... | 605 |
| Transform Objects by Delta (World Transform Operator)..... | 606 |
| Transform Objects by Delta (About Working Frame)..... | 607 |
| Translate Objects by Delta | 608 |
| Fit Geometry to Point Group..... | 609 |
| Fit Geometry to Point Group Projected to Plane | 610 |
| Fit Geometry to Points..... | 611 |
| Import Geometry Fit Profiles..... | 612 |
| Best Fit Transformation - Group to Group..... | 613 |
| Compute Group to Group Orientation (Rx, Ry, Rz)..... | 615 |
| Temperature Compensate a group..... | 616 |
| Query Clouds to Surface | 617 |
| Query Clouds to Objects..... | 618 |
| Get Measurement Weather Data | 619 |

| | |
|---|-----|
| Get Measurement Auxiliary Data..... | 620 |
| Set Measurement Auxiliary Data | 621 |
| Get Measurement Info Data | 623 |
| Get B-Spline Properties | 624 |
| Get Cone Properties | 625 |
| Get Slot Properties | 626 |
| Raster Scan Edge Inspection | 627 |
| New Raster Scan Edge Inspection | 628 |
| Mushroom Target Hole Inspection..... | 629 |
| Sphere Axis Check..... | 630 |
| Patch Normal Shift - Point | 631 |
| Patch Normal Shift - Hole / Pin | 632 |
| Angle Between Two Planes' normals..... | 633 |
| Angle Between Line and Plane..... | 634 |
| Angle Between Two Lines..... | 635 |
| Group To Surface Fit | 636 |
| Reverse Surface Normals | 637 |
| Reverse B-Splines..... | 638 |
| Get Surface Physical Stats..... | 639 |
| Set Inward Positive Normal..... | 640 |
| Sort Point Group in Database | 641 |
| Sort Vectors..... | 642 |
| Reverse Plane Normals | 643 |
| Get Double List Max/Min | 644 |
| Make Line Fit Profile..... | 646 |
| Make Plane Fit Profile..... | 648 |
| Make Circle Fit Profile..... | 649 |
| Make Slot Fit Profile | 650 |
| Make Ellipse Fit Profile | 651 |
| Make Sphere Fit Profile..... | 652 |
| Make Cone Fit Profile..... | 653 |
| Make Paraboloid Fit Profile | 654 |
| Make Cylinder Fit Profile | 655 |
| Set Geometry Relationship Fit Profile | 656 |
| Get Number of Feature Checks in Feature Check Ref List..... | 658 |
| Get Feature Check Datum References | 659 |
| Get i-th Feature Check From Feature Check Ref List..... | 660 |
| Get i-th Feature Check From Feature Check Ref List (Iterator) | 661 |
| Get Number of Datums in Datum Ref List..... | 662 |
| Set Feature Check Reporting Options | 663 |
| Get Feature Check Reporting Options..... | 664 |
| Get i-th Datum From Datum Ref List | 665 |
| Get i-th Datum From Datum Ref List (Iterator) | 666 |
| Get i-th Annotation From Annotation Ref List..... | 667 |

| | |
|---|-----|
| Get i-th Annotation From Annotation Ref List (Iterator) | 668 |
| Get Datum Measurements | 669 |
| Set Datum Measurements..... | 670 |
| Get Feature Check Measurements | 671 |
| Set Feature Check Measurements | 672 |
| Get Feature Check Cylinder Eval Options | 673 |
| Set Feature Check Cylinder Eval Options | 674 |
| Feature Inspection Auto Filter | 675 |
| Evaluate Feature Check | 676 |
| Evaluate Feature Checks | 678 |
| Start/Stop Feature Check Trapping | 679 |
| Enable/Disable Datum Alignment for Feature Check | 680 |
| Datum Alignment..... | 681 |
| Get GD&T Options | 682 |
| Set GD&T Options..... | 683 |
| Refresh Datums/Feature Checks from Annotations | 684 |
| Set Feature Check Reporting Frame..... | 685 |
| Get Feature Check Reporting Frame | 686 |
| Generate Geometry Relationship Summary..... | 688 |
| Get Number of Relationships in Relationship Ref List..... | 689 |
| Get i-th Relationship From Relationship Ref List..... | 690 |
| Get i-th Relationship From Relationship Ref List (Iterator) | 691 |
| Sort Relationship Ref List | 692 |
| Get Relationship Reporting Frame..... | 693 |
| Show/Hide Relationship Report..... | 694 |
| Show/Hide Relationship Watch | 695 |
| Relationship Watch Window Template | 696 |
| Make Point to Point Relationship..... | 697 |
| Make Frame to Frame Relationship | 698 |
| Make Points to Objects Relationship..... | 699 |
| Make Points to Points Relationship..... | 700 |
| Make Groups to Objects Relationship | 701 |
| Make Object to Object Direction Relationship..... | 702 |
| Get Geom Relationship Point List..... | 703 |
| Make Point Clouds to Objects Relationship..... | 704 |
| Enable All Cloud Cross Sections | 705 |
| Enable/Disable Cloud Cross Sections..... | 706 |
| Enable Single Cloud Cross Section..... | 707 |
| Get Number of Cross Sections in Cross Section Cloud | 708 |
| Make Group to Group Relationship | 709 |
| Make Group to Nominal Group Relationship..... | 710 |
| Make Average Point Relationship..... | 711 |
| Make Geometry Fit Only Relationship | 712 |
| Make Geometry Fit and Compare to Nominal Relationship..... | 713 |

| | |
|---|-----|
| Make Geometry Compare Only Relationship..... | 714 |
| Make Dynamic Point Relationship | 715 |
| Make Dynamic Line Relationship | 716 |
| Make Dynamic Plane Relationship..... | 717 |
| Make Dynamic Circle Relationship..... | 718 |
| Make Dynamic Ellipse Relationship..... | 719 |
| Make Vector Group To Vector Group Relationship..... | 720 |
| Set Vector Group To Vector Group Cylindrical Zone | 721 |
| Set Vector Group To Vector Group Fit Weights | 722 |
| Set Vector Group To Vector Group Fit Gradient Factor..... | 723 |
| Set Vector Group To Vector Group Fit Weights | 724 |
| Delete Relationship..... | 725 |
| Set Optimization Options..... | 726 |
| Set Optimization Perturbation Parameters..... | 727 |
| Do Relationship Fit..... | 728 |
| Move Collections by Minimizing Relationships..... | 729 |
| Get General Relationship Statistics | 730 |
| Get Points to Objects Relationship Statistics..... | 731 |
| Start/Stop Relationship Trapping | 732 |
| Get Point to Point Relationship Statistics..... | 733 |
| Get Relationship Associated Data | 735 |
| Set Points to Points Relationship Associated Data..... | 736 |
| Get Points to Points Relationship Associated Data | 737 |
| Auto Filter Clouds to Nominal Geometry 3D..... | 738 |
| Auto Filter Clouds to Nominal Geometry 2D..... | 739 |
| Auto Filter Points to Nominal Geometry 3D | 740 |
| Auto Filter Points/Groups/Clouds to Surface Faces | 741 |
| Create Points to Objects Map..... | 742 |
| Get Objects From Points to Objects Map..... | 743 |
| Compute Geometry Relationship Uncertainties..... | 744 |
| Make Pipe Fitting Relationship..... | 746 |
| Set Pipe Relationship Weights | 747 |
| Set Pipe Relationship Segment Properties..... | 748 |
| Get Pipe Relationship Properties | 749 |
| Get Pipe Relationship Weights..... | 750 |
| Get Pipe Relationship Cut Status | 751 |
| Make pipe Relationship Cut..... | 752 |
| Pipe Relationship Force Cut to Frame | 753 |
| Set Relationship Dormant Status..... | 755 |
| Set Relationship Weights Normalized | 756 |
| Get Relationship Type | 757 |
| Set Relationship Weighting..... | 758 |
| Get Relationship Weighting..... | 759 |
| Set Relationship Sub Sampling Options | 760 |

| | |
|--|-----|
| Get Relationship Sub Sampling Options | 761 |
| Set Relationship Reporting Frame | 762 |
| Get Relationship Reporting Frame..... | 763 |
| Set Geom Relationship Criteria | 764 |
| Get Geom Relationship Criteria..... | 765 |
| Set Geom Relationship Nominal Geometry..... | 767 |
| Get Geom Relationship Nominal Geometry..... | 768 |
| Set Geom Relationship Measured Geometry..... | 769 |
| Get Geom Relationship Measured Geometry | 770 |
| Set Geom Relationship Nominal Avg Point | 771 |
| Get Geom Relationship Nominal Avg Point..... | 772 |
| Set Geom Relationship Measured Avg Point..... | 773 |
| Get Geom Relationship Measured Avg Point..... | 774 |
| Set Geom Relationship Projection Plane | 774 |
| Get Geom Relationship Projection Plane..... | 775 |
| Set Geom Relationship Cardinal Points..... | 776 |
| Get Geom Relationship Cardinal Points | 777 |
| Set Geom Relationship Point List..... | 778 |
| Get Geom Relationship Point List..... | 778 |
| Geom Relationship Ignore Input Points..... | 779 |
| Get Geom Relationship Ignore Input Points..... | 779 |
| Geom Relationship Reuse Ignored Input Points | 780 |
| Get Geom Relationship Reuse Ignored Input Points..... | 780 |
| Geom Relationship Auto Vectors..... | 781 |
| Get Geom Relationship Auto Vectors..... | 781 |
| Set Geom Relationship Auto Vectors Nominal (AVN)..... | 782 |
| Get Geom Relationship Auto Vectors Nominal (AVN)..... | 782 |
| Set Relationship Auto Vectors Fit (AVF) | 783 |
| Get Relationship Auto Vectors Fit (AVF) | 783 |
| Set Relationship Voxel Cloud Display..... | 784 |
| Get Relationship Voxel Cloud Display..... | 784 |
| Set Relationship Desired Meas Count..... | 785 |
| Get Relationship Desired Meas Count..... | 785 |
| Make Vector Tolerance..... | 786 |
| Get Vector Tolerance..... | 786 |
| Make Vector Fit Constraint | 787 |
| Get Vector Fit Constraint | 787 |
| Set Relationship Position Fit Constraints (Vector Type)..... | 788 |
| Get Relationship Position Fit Constraints (Vector Type)..... | 788 |
| Set Relationship Orientation Fit Constraints (Vector Type)..... | 789 |
| Get Relationship Orientation Fit Constraints (Vector Type)..... | 789 |
| Set Relationship Tolerance (Vector Type)..... | 790 |
| Get Relationship Tolerance (Vector Type)..... | 790 |
| Set Relationship Projection Options..... | 791 |
| Get Relationship Projection Options..... | 792 |
| Set Relationship Outlier Rejection (Scalar Type) | 794 |
| Get Relationship Outlier Rejection (Scalar Type)..... | 795 |
| Outlier Rejection (Scalar Type)..... | 796 |
| Set Relationship Fit Constraints (Scalar Type) | 797 |
| Get Relationship Fit Constraints (Scalar Type)..... | 798 |
| Set Object to Object Direction Relationship Fit Constraints..... | 799 |
| Get Object to Object Direction Relationship Fit Constraints..... | 799 |
| Set Relationship Tolerance (Scalar Type)..... | 800 |
| Get Relationship Tolerance (Scalar Type) | 801 |
| Set Object to Object Direction Relationship Tolerance | 802 |
| Get Object to Object Direction Relationship Tolerance | 802 |
| Make Scalar Tolerance..... | 803 |
| Get Scalar Tolerance..... | 803 |
| Make Scalar Fit Constraint | 804 |
| Get Scalar Fit Constraint | 804 |
| Make Symmetric Scalar Tolerance | 805 |
| Get Symmetric Scalar Tolerance | 805 |
| Make Outlier Rejection Options..... | 806 |
| Get Outlier Rejection Options..... | 806 |

| | |
|---|-----|
| Make Symmetric Outlier Rejection Options | 807 |
| Get Number of Events in Event Ref List | 809 |
| Get i-th Event From Event Ref List | 810 |
| Get i-th Event From Event Ref List (Iterator)..... | 811 |
| Rename Event | 812 |
| Delete Event | 813 |
| Filter Clouds to Plane | 815 |
| Filter Clouds to Group..... | 816 |
| Filter Clouds to Surface | 817 |
| Filter Clouds to BSplines | 818 |
| Filter Clouds to Line Segment..... | 819 |
| Filter Clouds to Vector Groups - Resolve Points..... | 820 |
| RGB Cloud Point Filter..... | 821 |
| Delete Cloud Points by Radial Distance from Points | 822 |
| Delete Cloud Points by X Y Z Range | 823 |
| Get Scale Bar Stats..... | 825 |
| Scale Bar Check | 826 |
| Delete Scale Bar..... | 827 |
| Send Points to Geomagic | 829 |
| Send Clouds to Geomagic..... | 830 |
| Generate/Regenerate Coarse Mesh..... | 831 |
| Generate General Mesh..... | 832 |
| Mesh Volume..... | 833 |
| Create Point to Point Dimension..... | 835 |
| Create Point to Object Dimension..... | 836 |
| Create Object to Object Dimension..... | 837 |
| Create Diameter Dimension | 838 |
| Create Radius Dimension | 839 |
| Set Point to Point Dimension Properties..... | 840 |
| Set Point to Object Dimension Properties..... | 841 |
| Set Object to Object Dimension Properties..... | 842 |
| Set Diameter Dimension Properties | 843 |
| Set Radius Dimension Properties | 844 |
| Set Common Properties to Dimensions..... | 845 |
| Make a Dimension Ref List from a Collection..... | 846 |
| Make a Dimension Ref List- WildCard Selection | 847 |
| Get Number of Dimensions in Dimension Ref List..... | 848 |
| Add a Dimension to Dimension Ref List..... | 849 |
| Get i-th Dimension From Dimension Ref List..... | 850 |
| Get i-th Dimension From Dimension Ref List (Iterator) | 851 |
| Get Dimension Value..... | 852 |
| Delete Dimension..... | 853 |

REPORTING OPERATIONS

855

| | |
|---|-----|
| Set Report Tag Value From String | 856 |
| Set Report Tag Value From Integer..... | 857 |
| Set Report Tag Value From Double | 858 |
| Get Report Tag Value..... | 859 |
| Get Defined Report Tags..... | 860 |
| Remove Report Tag..... | 861 |
| Set Report Options for Object | 862 |
| Set Vector Group Report Options | 863 |
| Set Relationship Report Options | 864 |
| Set Point Delta Report Options | 865 |
| Set Point Group Report Options | 866 |
| Set Scale for Picture | 867 |
| Rename Picture | 868 |
| Delete Picture | 869 |
| Combine SA Reports | 870 |
| Quick Report | 871 |
| Close All Reports..... | 872 |
| Define Report Template | 873 |
| Make New SA Report..... | 874 |
| Append Items to SA Report..... | 875 |
| Add Item to SA Report at Location | 876 |
| Output SA Report to PDF..... | 877 |
| Output SA Report to Excel..... | 878 |
| Delete SA Report Template..... | 879 |
| Delete SA Report | 880 |
| Delete SA Doc..... | 881 |
| Generate/Update Templated Report..... | 882 |
| Make Report Graphical View Options | 883 |
| Make Report Output Options | 884 |
| Insert Page Break..... | 885 |
| Generate Custom HTML Report | 886 |
| Generate Standard HTML Report | 887 |
| Capture Screen to File (BMP/JPG/PNG/GIF/TIFF) | 888 |
| Capture Current View..... | 889 |
| Save Current View (BMP/JPG/PNG/GIF/TIFF)..... | 890 |
| Create Chart from Vector Group | 891 |
| Save Chart to JPEG File | 892 |
| Delete Chart | 893 |
| Make Utility Chart..... | 894 |
| Notify User Integer..... | 896 |
| Notify User Double..... | 897 |
| Notify User Text Array..... | 898 |
| Notify User HTML..... | 899 |
| HTML Display Board | 900 |

| | |
|--|------------|
| Close HTML Display Board | 901 |
| Report Bar | 902 |
| Add Objects to Report Bar | 903 |
| Add Callout Views to Report Bar | 904 |
| Add Charts to Report Bar | 905 |
| Add Datums to Report Bar | 906 |
| Add Dimensions to Report Bar | 907 |
| Add Events to Report Bar | 908 |
| Add Feature Checks to Report Bar | 909 |
| Add Pictures to Report Bar | 910 |
| Add Relationships to Report Bar | 911 |
| Add Scale Bars to Report Bar | 912 |
| Add Custom Tables to Report Bar | 913 |
| Generate Quick Report from Tab Order | 914 |
| Set Report Bar Visibility | 915 |
| Refresh Report Bar | 916 |
| Custom Report Tables | 917 |
| Make Custom Table | 918 |
| Clear Custom Table | 919 |
| Delete Custom Table | 920 |
| Get Custom Table Cell String | 921 |
| Get Custom Table Cell Double | 922 |
| Set Custom Table Title | 923 |
| Set Custom Table Header Row | 924 |
| Set Custom Table Header Cell | 925 |
| Set Custom Table Cell String | 926 |
| Set Custom Table Cell Double | 927 |
| Set Custom Table Cell Color | 928 |
| Set Custom Table Cell Font | 929 |
| Add Custom Table to SA Report | 930 |
| | |
| EXCEL DIRECT CONNECT | 931 |
| Open Workbook File | 932 |
| Set Workbook Address | 933 |
| Get Workbook Address | 935 |
| Set Clear After Insert | 936 |
| Run Macro | 937 |
| Save | 938 |
| Close | 939 |
| | |
| Write | 940 |
| Write Integer | 941 |
| Write Double | 942 |

| | |
|-----------------------|------------|
| Write String | 943 |
| Write Variables | 944 |
| Write Picture | 945 |
| Read..... | 946 |
| Read Integer | 947 |
| Read Double..... | 948 |
| Read String..... | 949 |
| Read Variables..... | 950 |

MS OFFICE REPORTING OPERATIONS 951

| | |
|-------------------------------------|-----|
| Initialize Office Report | 952 |
| Insert Section Break..... | 953 |
| Set Page Orientation | 954 |
| Add Section Heading to Report | 955 |
| Add Objects to Report..... | 956 |
| Insert Graphics from file..... | 957 |
| Add Graphics View to Report | 958 |
| Add User Input Notes to Report..... | 959 |
| Add Preset Notes to Report | 960 |
| Make Report Table..... | 961 |
| Close Office Report | 962 |
| Save Office Report as RTF | 963 |
| Add SADoc From File (RTF) | 964 |

INSTRUMENT OPERATIONS 965

| | |
|--|-----|
| Get Last Instrument Index..... | 966 |
| Rename Instrument..... | 967 |
| Get Instrument ID from Name | 968 |
| Get Instrument Model | 969 |
| Move Instrument to Another Collection | 970 |
| Save Instrument Configuration | 971 |
| Load Instrument Configuration..... | 972 |
| Point At Target | 973 |
| Measure Single Point Here | 974 |
| Get Current Instrument Position Update..... | 975 |
| 'Build' Target | 976 |
| Measure Existing Single Point..... | 977 |
| Measure Existing Single Point (Manual Guide)..... | 978 |
| Measure Existing Single Point and Compare..... | 979 |
| Stop Active Measurement Mode | 980 |
| Set Probe Offset Frame Online (Measure Raw Frame) | 981 |
| Set Probe Offset Frame Offline (Select Previously Measured Frame)..... | 982 |
| Enable/Disable Frame Set Scan Mode | 983 |

| | |
|---|------|
| Enable/Disable Point Set Scan Mode | 984 |
| Add New Instrument..... | 985 |
| Delete Instrument | 986 |
| Delete Measurements..... | 987 |
| Delete Measurement Observation..... | 988 |
| Move Measurement Observation..... | 989 |
| Initiate Servo-Guide..... | 990 |
| Watch Point to Point..... | 991 |
| Watch Point to Objects..... | 992 |
| Watch Point to Edge | 993 |
| Watch Closest Point | 994 |
| Watch Instrument..... | 995 |
| Watch Window Template 3D | 996 |
| Watch Point to Point With View Zooming..... | 997 |
| Start Theodolite Interface..... | 998 |
| Start Instrument Interface..... | 999 |
| Stop Instrument Interface | 1000 |
| Activate/Deactivate Instrument Toolbar..... | 1001 |
| Verify Instrument Connection..... | 1002 |
| Configure and Measure | 1003 |
| Measure..... | 1005 |
| Show/Hide Instrument Interface | 1006 |
| Dock Instrument Interface | 1007 |
| Locate Instrument (Ref. Tie-In)..... | 1008 |
| Locate Instrument (Group to Surface Quick Fit)..... | 1009 |
| Locate Instruments (USMN)..... | 1010 |
| Locate Templated Instruments (USMN)..... | 1011 |
| Create Templated Instrument (USMN)..... | 1012 |
| Make a USMN Templated Instrument List | 1013 |
| Add a USMN Templated Instrument to a USMN Templated Instrument List | 1014 |
| Locate Instrument (Best Fit - Group to Group)..... | 1015 |
| Locate Instrument (Best Fit - Nominal Geometry)..... | 1017 |
| Get Instrument Transform | 1019 |
| Set Instrument Transform..... | 1020 |
| Set Tracker/EDM Theodolite Uncertainties | 1021 |
| Get Tracker/EDM Theodolite Uncertainties..... | 1022 |
| Set PCMM Instrument XYZ Uncertainties..... | 1023 |
| Get PCMM Instrument XYZ Uncertainties..... | 1024 |
| Set Instrument XYZ Uncertainties | 1025 |
| Get Instrument XYZ Uncertainties | 1026 |
| Get Instrument Weather Setting..... | 1027 |
| Set Instrument Weather Setting..... | 1028 |
| Get Instrument Part Temperature..... | 1029 |
| Compute CTE Scale Factor | 1030 |

| | |
|---|------|
| Set (multiply) Instrument Scale Factor (CAUTION!) | 1031 |
| Set (absolute) Instrument Scale Factor (CAUTION!) | 1032 |
| Get Instrument Scale Factor | 1033 |
| Transform Instrument - Frame To Frame | 1034 |
| Transform Instrument by Delta | 1035 |
| Transform Multiple Instruments by Delta | 1036 |
| Instrument Operational Check | 1037 |
| Get Instruments with Observations on Target | 1053 |
| Get Targets Measured by Instrument | 1054 |
| Set Observation Status | 1055 |
| Get Observation Info | 1056 |
| Fabricate Observations | 1057 |
| Get Obscured Points from Instrument | 1058 |
| Set Instrument Measurement Mode/Profile | 1059 |
| Set Instrument Targeting | 1062 |
| Set Target Computation Options | 1063 |
| Set Observation Mirror Cube Shot Face | 1064 |
| Set Observation Collimation Shot Options | 1065 |
| Collimation | 1066 |
| Get Instrument Target Status | 1067 |
| Make Surface Face List from Point Proximity | 1068 |
| Scan within Perimeter | 1069 |
| Scan CAD Faces | 1071 |
| Edge Scan Measurement | 1072 |
| Track Tape Measurement | 1073 |
| Auto Measure Points | 1074 |
| Auto-Measure Vectors | 1075 |
| Auto-Measure Surface Vector Intersections | 1076 |
| Auto-Measure Specified Geometry | 1077 |
| Auto-Correspond Closest Point | 1078 |
| Auto-Correspond with Proximity Trigger | 1079 |
| Construct Mirror From Plane | 1081 |
| Drift Check | 1081 |
| Create New Dynamic Reference | 1082 |
| Calculate TCP Fixture Uncertainties | 1083 |
| Construct TCP Fixture | 1084 |
| Add Nominal Point to TCP Fixture | 1085 |
| Get Last Solved TCP Fixture Uncertainty Covariance Matrix | 1086 |
| Set Instrument Base Uncertainty Covariance Matrix WRT Base | 1087 |
| Set Instrument Base Uncertainty Covariance Matrix WRT World | 1088 |
| Get Instrument Base Uncertainty Covariance Matrix WRT World | 1089 |
| Construct Measured Point Uncertainty Ellipsoids | 1090 |
| Measure Nominal Feature | 1091 |
| Guide Objects in 6D based on Point Measurements | 1092 |

| | |
|---|-------------|
| Move Objects in 6D using Instrument Updates | 1093 |
| Align Two Targets with Axis (WCF-X) | 1095 |
| Get Instrument Interface Response Timeout | 1096 |
| Set Instrument Interface Response Timeout | 1097 |
| Get Current Trapping Status | 1098 |
| Wait For Trapping To Complete | 1099 |
| Jump Instrument to New Location | 1100 |
| Quick Align..... | 1101 |
| Align Cloud to CAD..... | 1102 |
| Start GD&T Inspection Design | 1103 |
| Start GD&T Inspection Rehearse..... | 1104 |
| Start GD&T Inspection | 1105 |
| Set Remeasure Failed Checks Only | 1106 |
| Associate Objects with Instrument..... | 1107 |
| Disassociate Objects from Instrument..... | 1108 |
| Make Collection Object Name Ref List from Objects associated with Instrument..... | 1109 |
| Combine Point Groups | 1110 |
| Dissect Point Groups | 1111 |
| Synchronized Measurement (Master/Slave) | 1112 |
| Crib Sheet Operations..... | 1113 |
| Run Crib Sheet..... | 1114 |
| Laser Projection..... | 1115 |
| Project Objects | 1116 |
| Stop Projection..... | 1117 |
| Advanced Instrument Operations..... | 1118 |
| Issue Instrument Actuator Command..... | 1119 |
| Set Instrument Axes..... | 1120 |
| Set Alignment Projector..... | 1121 |
| Nikon Metrology - Laser Radar (LR)..... | 1122 |
| LR Hardware Disconnect..... | 1123 |
| LR Verify Hardware Connection..... | 1124 |
| LR Set Red Laser Intensity..... | 1125 |
| LR Self Test..... | 1126 |
| LR Self Test - Linearization | 1127 |
| LR Self Test - Flip Test | 1128 |
| LR Self Test - LO Sep | 1129 |
| LR APDIS Perform MCM Calibration..... | 1130 |
| LR APDIS Activate MCM Calibration | 1131 |
| LR APDIS Get Active MCM Calibration | 1132 |
| ROBOT OPERATIONS | 1133 |
| Add Robot/Machine (.SAMachine)..... | 1134 |

| | |
|--|------|
| Add Robot/Machine (.ManipKin) | 1135 |
| Delete Robot/Machine | 1136 |
| Move Robot/Machine to Frame | 1137 |
| Move Robot/Machine through Path..... | 1138 |
| Move Robot/Machine to Named Destination | 1139 |
| Set Robot/Machine Parameter | 1140 |
| Get Robot/Machine Parameter..... | 1141 |
| Set Robot/Machine Model Link Parameters | 1142 |
| Get Robot/Machine Model Link Parameters | 1143 |
| Start Robot/Machine Interface..... | 1145 |
| Stop Robot/Machine Interface | 1146 |
| Compute Robot/Machine Adjusted Goal Frame..... | 1147 |
| Move Robot/Machine to Joint Pose (6DOF)..... | 1148 |
| Simulate Robot/Machine Path, Output CSV File | 1149 |
| Create Robot Calibration | 1150 |
| Delete Robot Calibration | 1151 |
| Import Poses Match to Measurements..... | 1152 |
| Import Poses Match to Frames | 1153 |
| Perform Robot Calibration | 1154 |
| Perform Robot Calibration (Alternate)..... | 1155 |
| Start/Stop Robot Calibration Trapping | 1156 |
| Set Active Robot Calibration | 1157 |
| Set Robot Calibration Tool Frame | 1158 |
| Set Robot Calibration Measurement Offset in Tool Frame | 1159 |
| Get Robot Pose for a Frame | 1160 |
| Get Calibration Appliance Integer Value..... | 1161 |
| Set Calibration Appliance Integer Value | 1162 |
| Get Calibration Appliance Real Value..... | 1163 |
| Set Calibration Appliance Real Value..... | 1164 |
| Get Calibration Appliance Data..... | 1165 |
| Set Calibration Appliance Data..... | 1166 |

Robot Calibration Appliance Node Operations..... 1167

UTILITY OPERATIONS **1169**

| | |
|------------------------------------|------|
| OPC DA Server | 1170 |
| Set OPC DA Tag Value Double..... | 1171 |
| Get OPC DA Tag Value Double | 1172 |
| Set OPC DA Tag Value Integer..... | 1173 |
| Get OPC DA Tag Value Integer | 1174 |
| Set OPC DA Tag Value String..... | 1175 |
| Get OPC DA Tag Value String..... | 1176 |
| Language | 1177 |

| | |
|---|-------------|
| Set Active Integrated Language..... | 1178 |
| Set Active Custom Language..... | 1179 |
| Get Active Language..... | 1180 |
| Folders | 1181 |
| Delete Folder..... | 1182 |
| Move Collection to Folder..... | 1183 |
| Move Folder to Folder..... | 1184 |
| Get Folders by Wildcard..... | 1185 |
| Get Folder Notes..... | 1186 |
| Set Folder Notes..... | 1187 |
| Get Folder Collections..... | 1188 |
| Notes..... | 1189 |
| Get Collection Notes..... | 1190 |
| Set Collection Notes..... | 1191 |
| Get Object Notes..... | 1192 |
| Set Object Notes..... | 1193 |
| Get Point Notes..... | 1194 |
| Set Point Notes..... | 1195 |
| Units..... | 1196 |
| Get Active Units..... | 1197 |
| Set Active Units..... | 1198 |
| Set Decimal Digits for Display..... | 1199 |
| Set Angular Representation..... | 1200 |
| Get Angular Representation..... | 1201 |
| Step Comment..... | 1202 |
| Set User Interface Profile..... | 1203 |
| Set MP Step Mode..... | 1204 |
| Delay for Specified Time..... | 1205 |
| Get Tick Count..... | 1206 |
| Speak To User..... | 1207 |
| Set Working Frame..... | 1208 |
| Get Working Frame Properties..... | 1209 |
| Set Working Color..... | 1210 |
| Set Working Color Auto Increment..... | 1211 |
| Set Object(s) Color..... | 1212 |
| Get Object Color..... | 1213 |
| Set Object(s) Translucency..... | 1214 |
| Send MP Result to External Device..... | 1215 |
| Send MP Step's Status to External Device..... | 1216 |
| Delete Objects..... | 1217 |
| Delete Items..... | 1218 |
| Lock Imported Items..... | 1219 |

| | |
|--|------|
| Lock/Unlock Selected Items | 1220 |
| Highlight Objects | 1221 |
| Highlight Objects | 1222 |
| Highlight Point | 1223 |
| Move Objects Drag Graphically..... | 1224 |
| Scale Objects..... | 1225 |
| Move Instruments Drag Graphically..... | 1226 |
| Set Special MP Mode..... | 1227 |
| Increment Point Name..... | 1228 |
| Refresh Views | 1229 |
| Set Logging State | 1230 |
| Trim Log File | 1231 |
| Set Auto Event Creation | 1232 |
| Write to Log | 1233 |
| Remove Specified Characters From String..... | 1234 |
| Close All Watch Windows..... | 1235 |
| Status Dialog | 1236 |
| Set Working Directory..... | 1237 |
| Make Directory | 1238 |
| Directory Existence | 1239 |
| Delete Directory..... | 1240 |
| Copy Directory..... | 1241 |
| Generate Random Number..... | 1242 |
| Set View Idle Update Frequency | 1243 |
| Set Automatic Backup State | 1244 |
| Set Notification Cancel Override..... | 1245 |
| Set Interaction Mode..... | 1246 |
| UDP Send String..... | 1247 |
| UDP Receive String | 1248 |
| Get Screen Resolution | 1249 |
| Set Wild Card Asterisk Mode..... | 1250 |
| Lock/Unlock Trapping Control..... | 1251 |

ACCUMULATOR MATH OPERATIONS 1253

| | |
|-------------------------------|------|
| Accumulator Clear..... | 1254 |
| Accumulator Add..... | 1255 |
| Accumulator Subtract | 1256 |
| Accumulator Multiply | 1257 |
| Accumulator Divide | 1258 |
| Accumulator Power | 1259 |
| Accumulator Invert..... | 1260 |
| Accumulator Change Sign | 1261 |

SCALAR MATH OPERATIONS 1263

| | |
|--------------------------------------|------|
| Integer Math Operation | 1264 |
| Double Math Operation | 1265 |
| Double Comparison | 1266 |
| Double Comparison (result) | 1267 |
| Integer Comparison..... | 1268 |
| Integer Comparison (result)..... | 1269 |
| String Comparison | 1270 |
| String Comparison (result) | 1271 |
| Change String Case..... | 1272 |
| Does String Contain Sub-String | 1273 |
| Boolean Comparison..... | 1274 |
| Boolean Comparison (result)..... | 1275 |
| Color Comparison..... | 1276 |
| Color Comparison (result) | 1277 |
| Integer Absolute Value | 1278 |
| Double Absolute Value..... | 1279 |
| Double Square Root | 1280 |
| Round Double..... | 1281 |
| Trig Function | 1282 |
| Logarithmic Function..... | 1283 |
| Double Angle Conversion | 1284 |

VECTOR MATH OPERATIONS 1285

| | |
|--------------------------------|------|
| Vector Addition | 1286 |
| Vector Subtraction | 1287 |
| Vector Dot Product | 1288 |
| Vector Cross Product..... | 1289 |
| Vector Scaling | 1290 |
| Vector Normalize | 1291 |
| Vector Magnitude (Length)..... | 1292 |

MP SUBROUTINES 1293

| | |
|--------------------------------------|------|
| Run Subroutine | 1294 |
| Define Subroutine Input Values | 1295 |
| Return from Subroutine Now..... | 1296 |
| Define Subroutine Return Values..... | 1297 |

VARIABLES 1299

| | |
|--------------------------------------|------|
| Set Integer Variable..... | 1300 |
| Get Integer Variable..... | 1301 |
| Set Double Variable | 1302 |
| Get Double Variable..... | 1303 |
| Set Named Double List Variable..... | 1304 |
| Get Named Double List Variable | 1305 |

| | |
|--|------|
| Add Double to Named Double List Variable..... | 1306 |
| Get Named Double List Variable Min/Max | 1307 |
| Clear Named Double List Variable | 1308 |
| Get i-th Double From List | 1309 |
| Get number of Doubles in List..... | 1310 |
| Clear Double List..... | 1311 |
| Set String Variable | 1312 |
| Get String Variable | 1313 |
| Set Boolean Variable..... | 1314 |
| Get Boolean Variable..... | 1315 |
| Set Vector Variable | 1316 |
| Get Vector Variable..... | 1317 |
| Set Font Variable | 1318 |
| Get Font Variable | 1319 |
| Set Transform Variable | 1320 |
| Get Transform Variable | 1321 |
| Set Point Name Variable..... | 1322 |
| Get Point Name Variable..... | 1323 |
| Set Collection Object Ref List Variable..... | 1324 |
| Get Collection Object Ref List Variable | 1325 |
| Set Relationship Ref List Variable..... | 1326 |
| Get Relationship Ref List Variable | 1327 |
| Set Report Items Reference List Variable | 1328 |
| Get Report Items Reference List Variable..... | 1329 |
| Set Point Name Ref List Variable | 1330 |
| Get Point Name Ref List Variable..... | 1331 |
| Set String Ref List Variable..... | 1332 |
| Get String Ref List Variable | 1333 |
| Set Collection Object Name Variable | 1334 |
| Get Collection Object Name Variable..... | 1335 |
| Delete Variable | 1336 |
| Delete Variables - Wildcard Match..... | 1337 |

1

FUNDAMENTAL TERMS

About the Command Reference Manual

The MP Command Reference Manual is intended to provide detailed and specific information on each available command offered through Measurement Plan (MP) scripts. If you need to figure out what is expected as an input or return from a command, this is where you will find the answer. The Measurement Plan chapter of the SA Users Manual is also available and includes an overview of the MP Editor and the scripting process.

MP Step Fundamentals:

Step Arguments

Each MP Step operates either as a direct switch to control an aspect of SA, as a computational unit, or as a combination of the two. It can be given information which is then used when the step is run (executed) to

produce a return with the results of the operation.

- **Input Arguments.** These are values that can be supplied to a command as input. They help the command determine what it is supposed to do. Most input arguments are required, but some are optional.
- **Return Arguments.** Return arguments are used to return information or results back from a command after it executes.

Success/Partial Success/Failure Step Status

Steps can Fail for a number of reasons. As a rule of thumb Steps can return the following conditions:

- **Failed.** A step fails when it is missing critical information in order to complete. This typically causes the script to stop in order for the problem to be addressed.
- **Partial Success.** Partial failures can be returned. These typically indicate an unexpected result rather than a problem with the MP. For example a best-fit can have a tolerance set. If the points cannot be found the step will fail, but if it fits but the fit is out of tolerance you get a partial success.
- **Success.** Everything worked as expected.

The response of the MP script to these events depend on your settings. Use the command *Set Interaction Mode* to change this behavior such that a failed step is ignored (when possible) and error handling can be built in to the script. This is call "Silent mode".

Additional Job Configuration

Additional Steps that should be considered when running MP:

- **Set Logging State.** With logging active each MP step will be recorded into the log file. This can be very

helpful in tracking changes to the job file, but can also grow the log into a huge and cumbersome file if repeated MP executions are used.

- **Set Automatic Backup State.** Is another one to consider carefully. Automatic backups can interrupt operations but not backing up can also be dangerous.
- **Clear All Ascii Files.** When opening and closing external files such as text or excel files these files can be left open as background processes. This command clears these open files.

Naming and Reference Structure for Items Within SA.

Referencing specific items or objects in the tree requires a naming sequence. We use the `Collection::Object::Name` convention (which can also include a `::type` specification at the end) in order to provide a path to a specific instance of an item.

Items within an SA job file can be categorized as follows:

- **Items.** Items are the largest most generic category and includes anything within SA. Commands such as *Show Items in Tree* provide a means to access any item of any *Type*.
- **Instruments.** Instruments are handled separately from all other types of items within SA and are referenced using a *Collection::Instrument ID*. This includes the collection name and the index of the instrument within that collection.
- **Objects.** Objects are a subcategory that includes only 3D features within SA's graphics such as CAD entities, Geometry, point groups or point clouds. They do not include pictures, callouts or reports. They are measurable features displayed within the graphics.
- **Points.** Points are not objects in themselves but are included as identifiers within a specific Point Group. A *Point Name Ref List* therefore include both the Col-

lection Object Name for the Point Group and a point name. As an example the point group *A::Nominals* would include the point *A::Nominals::P1*.

Data Types

Every argument has a data type. A data type indicates the type of information that an argument is expecting. You must feed an argument data of a compatible data type otherwise, SA will prevent the operation or the script will fail at runtime.

Basic Data Types Within MP Include:

- **Boolean.** Booleans are the simplest data type. They are either True or False, much like a switch is either on or off. You can use the shorthand "1" for True and "0" for False in entering a Boolean value.
- **String.** Strings are sequence of one or more alphanumeric characters, such as a sentence or prompt.
- **Integer.** Integers are whole numeric values without any decimals. Integers are represented exactly and can be used to reference a particular item in a list.
- **Double.** Doubles are numeric values with decimal precision, either positive or negative. The precision of the decimal representation (such as such as 3.1415926535) beyond 8-10 decimals will depend on they system and any rounding applied which can influence trying to compare one double value to another. You may expect them to be the same but they may actually be very slightly different.
- **Reference Lists.** Reference lists are also a data type. A point list for example is composed of a list of points in a specific order.

Working with Variables

Most data types can also be designated as a variable. Variables allow you to define a name that is associated with a value. This allows you to refer to that value by name. The advantage of this is that while a script is running, you may change or overwrite the value of that variable, but anywhere you refer to it by name, you'll always retrieve its current value.

Defined variables appear in the variables view panel. Some lists, such as Point Name Reference Lists and Collection Object Name Lists also support a special **Ctrl+Shift** double click option to display a new panel with the names of the items defined in the variable while in debug mode.

A variable name can also be used to define a variable. This allows a placeholder variable to be established in a script that references a specific name to retrieve the variable values.

* Note that the variables panel cannot track changes in variable type.

Search and Selection Tools

- **Runtime Select.** Runtime select provides a means to bypasses the need to identify items as part of the script. It allows a user to select the necessary items manually while the script is running (runtime).
- **Wildcard Selection.** Wildcard search terms provide a means to build a list based upon name matching. An asterisks (*) represent a set of contiguous characters, and a question mark (?) represent a single character. You can also use brackets ([]) as an escape sequence if needed to search for special characters, for example [*] will search for names containing an asterisk (*). Also [!] provides a means to exclude items from the search. Wildcard selection criteria, like all names in SpatialAnalyzer, are not case sensitive.

Supported wild-characters include: '*', '?'; sets: [a-z],

'!' negation

Examples include:

'[a-g]l*i?n' matches 'florian'

'[!abc]*e' matches 'smile'

'[-z] matches 'a'

Wildcard selection automatically adds an asterisk (*) at the beginning and end of your search text. Each search string is automatically wrapped as *search string*. This allows you to search for "P1" and find "AP123". However, this may result in selection of more items than desired. To control this operation use the command [Set Wild Card Asterisk Mode](#).

Loop Step Structures

Loops within SA are performed using the following step structure:

- **Counter.** A counter is required for a loop in order to keep track of the index or status as an element is retrieved from a list and an action is performed.
- **Get i-th.** In order to sequentially perform operations with items from within a list a script needs to be able to access those items in an ordered way. Each item in a list contains an index (i), item 1, 2, 3, etc. "Get i-th" commands provide a means to access a specific item at a specific index in a list by returning the i-th object. Note that a list (array) starts at 0 as the first entry and 1 is the second.
- **Increment Counter.** In order for an MP to work through a list the index of a counter must be adjusted to retrieve the next element in the list.
- **Integer Comparison.** In order to exit a loop a comparison needs to be made. Either to verify that a target has been found or that the end of the list has been reached. An integer comparison that references the counter is an effective way to verify that the end

of the list has been reached. A comparison step also provides a means to jump back to the Get-ith step in order to retrieve the next element from the list.

Iterator Step Structures

Many command such as *Get i-th Point Name From Point Name Ref List* also have an (*iterator*) option. These steps have a built in counter and integer comparison. Each time the step is run the internal counter indexes by one and checks to see if it has reached the end of the list. If it has, it uses the “*Step to Jump at End of List*” argument to jump out of the loop.

A loop using an iterator is as simple as:

1. Build a list
2. Add a get i-th step with (iterator)
3. Perform any operations you wish
4. Jump to Get i-th to get the next element.

This built in Iterator concept works great when an MP is executed a single time. It will not work correctly when a loop is set within a loop (nested loop) as an iterator will not reset once it reaches the end of the list.

Relative Paths

Command arguments that designate a directory support both absolute paths (“C:\Analyzer Data\MP Work\test.xit”) and relative paths (“.\test.xit”). Relative paths are both faster to type and allow a directory to be moved without breaking the MP path designation. Note that relative paths may also relate to the working directory, depending on the command, which may be the SA install directory unless specified using *Set Working Directory*.

“.” designates the current directory.

If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location. But, if executing an external MP the file that will be opened will be

opened with respect to the MPs directory.

Relative paths can also be expanded relative to the current directory. For example, to access a text file in the directory (C:\Analyzer Data\MP Work\Fit\Nominals.txt) from an SA file running from ".\MP Work" folder you can enter:

```
“.\Fit\Nominals.txt”
```

This can be expanded further to browse up:

“..” designates one folder above the current directory

Following the same example, a template file in “C:\Analyzer Data\Templates” could be accessed from C:\Analyzer Data\MP Work using “..\Templates\MyTemplate.xit64”. To browse up two levels use “..\..\”.

2 FILE OPERATIONS

New SA File

Closes the current SA file and opens a new file using the current default template (if defined). The user is not prompted to save the current file.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------|
| SUCCESS | When a new SA file is created. |
|---------|--------------------------------|

Remarks

Creating a new file will cause problems if you attempt to call additional embedded MP files from the original SA file, since that file is no longer open. However, the current MP will continue to execute.

The default template file is loaded from `Analyzer Data\Templates\Default.xit`.

Open SA File

Discards the currently open file and opens the specified SA file. The currently opened file is not saved.

Input Arguments

| | | | |
|---|---------------------------------|--------------|------------------------|
| 0 | File Path or Embedded File Name | SA File Name | The .XIT file to open. |
|---|---------------------------------|--------------|------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was opened successfully. |
| FAILURE | The file could not be found or could not be opened. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Examples

The following absolute path opens the SA file at the given specific drive location:

| | | | |
|---|---------------------------------|--------------|-----------------------------|
| 0 | File Path or Embedded File Name | SA File Name | c:\Analyzer Data\sample.xit |
|---|---------------------------------|--------------|-----------------------------|

The following relative path opens the SA file in the “sub” subdirectory of the currently executing MP’s location:

| | | | |
|---|---------------------------------|--------------|------------------|
| 0 | File Path or Embedded File Name | SA File Name | .\sub\sample.xit |
|---|---------------------------------|--------------|------------------|

This means that if the executing external MP is located at `c:\myMPs`, the file that will be opened will be `c:\myMPs\sub\sample.xit`.

Save

Saves the current SA file using the current filename and path.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The file was saved successfully. |
| FAILURE | The file could not be saved. |

Remarks

The *File* ► *Save As* dialog is opened if the file has not been previously named.

Save As

Saves the current SA file with the provided filename and path. An optional serial number will be appended to the filename.

Input Arguments

| | | | |
|---|----------------------------|--------------------|--|
| 0 | File Path or Embedded File | File Name | The .XIT file to use when saving the current file. |
| 1 | Boolean | Add Serial Number? | Specify whether or not to append a serial number to the filename. |
| 2 | Integer | Optional Number | The serial number to append to the filename. Only applies when Argument 1 is TRUE. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was saved successfully. |
| FAILURE | The file could not be saved. (Perhaps the path was not accessible). |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Examples

The following absolute path saves the SA file at the given specific drive location:

| | | | |
|---|---------------------------------|--------------|-----------------------------|
| 0 | File Path or Embedded File Name | SA File Name | c:\Analyzer Data\sample.xit |
|---|---------------------------------|--------------|-----------------------------|

The following relative path saves the SA file in the “sub” subdirectory of the currently executing MP’s location:

| | | | |
|---|---------------------------------|--------------|------------------|
| 0 | File Path or Embedded File Name | SA File Name | .\sub\sample.xit |
|---|---------------------------------|--------------|------------------|

This means that if the executing external MP is located at `c:\myMPs`, the file that will be saved will be `c:\myMPs\sub\sample.xit`.

Backup Now

Saves a backup copy of the current SA file to the SA backup location (`C:\Analyzer Data\Backup` by default).

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was saved successfully. |
| FAILURE | The file could not be saved. Check permissions on the save path. |

Remarks

None.

Open Template File

Opens a read-only SA template file from the specified path and file name.

Input Arguments

| | | | |
|---|----------------------------|--------------------|--------------------------------|
| 0 | File Path or Embedded File | Template File Name | The default .XIT file to open. |
|---|----------------------------|--------------------|--------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The template file was opened successfully. |
| FAILURE | The template file could not be found or could not be opened. |

Remarks

This command supports both absolute paths (ex. `C:\Analyzer Data\Templates\test.xit`) and relative paths (ex. `.\Templates\test.xit`).

If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Examples

The following absolute path opens the SA template file at the given specific drive location:

| | | | |
|---|---------------------------------|--------------|---------------------------------------|
| 0 | File Path or Embedded File Name | SA File Name | c:\Analyzer Data\Templates\sample.xit |
|---|---------------------------------|--------------|---------------------------------------|

The following relative path opens the SA template file in the “sub” subdirectory of the currently executing MP’s location:

| | | | |
|---|---------------------------------|--------------|----------------------------|
| 0 | File Path or Embedded File Name | SA File Name | .\Templates\sub\sample.xit |
|---|---------------------------------|--------------|----------------------------|

This means that if the executing external MP is located at `c:\myMPs`, the file that will be opened will be `c:\myMPs\Templates\sub\sample.xit`.

Save As Read-Only Template

Saves the current SA file as a read-only template file with the provided filename and path. An optional serial number will be appended to the filename.

Input Arguments

| | | | |
|---|----------------------------|--------------------|--|
| 0 | File Path or Embedded File | Template File Name | The .XIT file to use when saving the current file as a template. |
|---|----------------------------|--------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The template file was saved successfully. |
| FAILURE | The template file could not be saved. (Perhaps the path was not accessible). |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Examples

The following absolute path saves the SA file at the given specific drive location:

| | | | |
|---|---------------------------------|--------------|-----------------------------|
| 0 | File Path or Embedded File Name | SA File Name | c:\Analyzer Data\sample.xit |
|---|---------------------------------|--------------|-----------------------------|

The following relative path saves the SA file in the “sub” subdirectory of the currently executing MP’s location:

| | | | |
|---|---------------------------------|--------------|------------------|
| 0 | File Path or Embedded File Name | SA File Name | .\sub\sample.xit |
|---|---------------------------------|--------------|------------------|

This means that if the executing external MP is located at `c:\myMPs`, the file that will be saved will be `c:\myMPs\sub\sample.xit`.

Exit Measurement Plan

Exits the Measurement Plan.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | When the MP is exited. (Returned status is never used). |
|---------|---|

Remarks

If omitted, when a Measurement Plan proceeds past the last line in a script, a dialog stating “Inspection Done!” will be displayed. Using this command at the end of the script will suppress this dialog.

This command is often used as the destination for a Jump to Step command when jumping to immediately exit a script.

Shut Down SA

Exits the SpatialAnalyzer application.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | When SA is successfully exited. (Return status is never used). |
|---------|--|

Remarks

Displays a “Save As” dialog if the SA file has not been saved.

Run Another Program

Initiates an external application with optional command-line arguments.

Input Arguments

| 0 | File Path or Embedded File | Program Path | The path to the program to execute. |
|---|----------------------------|-----------------------------------|--|
| 1 | String | Command Line Arguments (optional) | Optional command-line arguments to pass to the executing program. |
| 2 | Boolean | Wait for Program Completion | Indicates whether the MP should pause until the program finishes. |
| 3 | Integer | Process Exit Code | Windows Process Exit Code return as part of trying to run the application. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The program was executed successfully. |
| FAILURE | The program could not be found or executed. |

Remarks

This command supports both absolute paths (ex. `C:\test.exe`) and relative paths (ex. `.\test.exe`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

The file to execute does not necessarily need to be an executable (.exe) file. Batch files (.BAT), VBScript files (.VBS), and other file types that execute when double-clicked will work with this command.

Command line arguments should not be enclosed in quotes. If more than one argument is provided, separate them by a space--not a comma. For example, to run a program with two command-line arguments, argument #1 might be `-x file.txt`.

If the *Wait for Program Completion* argument is set to TRUE, the MP will pause while the other program is running and SA will not receive events from Windows. As a result, SA will appear to have hung or crashed. This is normal, and once the application completes and returns control back to SA, the interface will be updated and become responsive again. For this reason, it is recommended to minimize the SA interface (using the Set SA's Window State command) prior to running an external program with this method.

If the *Wait for Program Completion* argument is set to FALSE, the MP will start to execute the specified program and immediately continue to the next step in the MP.

Process Exit Code is a windows process level integer error code returned as part of any attempt to run a separate program. It returns values such as:

0. Operation completed successfully
 1. Incorrect function.
 2. System cannot find the file specified
- ... (a full listing can be found from Microsoft).

Run Powershell Script

PowerShell is a cross-platform task automation framework. It is built on top of the .NET Common Language Runtime (CLR), and accepts and returns .NET objects.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | File Path or Embedded File | Powershell Script Path | The path to the powershell script to execute. |
| 1 | String | Script Arguments (optional) | Optional script arguments to pass to the executing program. |
| 2 | Boolean | Wait for Program Completion | Indicates whether the MP should pause until the program finishes. |

Return Arguments

| | | | |
|---|---------|-------------------|----------------------------------|
| 3 | Integer | Process Exit Code | The powershell script exit code. |
|---|---------|-------------------|----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The program was executed successfully. |
| FAILURE | The program could not be found or executed. |

Remarks

One example of how to use a powershell script is as follows: “*Run Powershell Script*” which can be used to execute the following script (as an example).

ExportTo-ExcelPDF.ps1

```
if ($args.Count -gt 0)
{
    $path = $args[0]
    $xlFixedFormat = "Microsoft.Office.Interop.Excel.xlFixedFormatType" -as [type]
    $excelFiles = Get-ChildItem -Path $path -include *.xls, *.xlsx -recurse
    $objExcel = New-Object -ComObject excel.application
    $objExcel.visible = $false
    foreach($wb in $excelFiles)
    {
        $filepath = Join-Path -Path $path -ChildPath ($wb.BaseName + ".pdf")
        $workbook = $objExcel.workbooks.open($wb.fullname, 3)
        $workbook.Saved = $true
        "saving $filepath"
        $workbook.ExportAsFixedFormat($xlFixedFormat::xlTypePDF, $filepath)
        $objExcel.Workbooks.close()
    }
    $objExcel.Quit()
}
```

The first argument of the MP command must be the full path of the powershell cmdlet – in this case *ExportTo-ExcelPDF.ps1*. This file is included in the installation directory of SA by default.

The second argument of the MP command must be the full path to the directory in which EXCEL files have been put

awaiting print to PDF.

This powershell cmdlet will print out ALL EXCEL files of *.xls or .xlsx types to PDF files in the specified directory (or any subordinate directory).

Many other tasks could be accomplished by executing a powershell script in this way.

Copy General File

Copies a source file to a destination file, optionally overwriting the destination file if it already exists.

Input Arguments

| | | | |
|---|----------------------------|-----------------------|--|
| 0 | File Path or Embedded File | Source File Name | The path for the source file. |
| 1 | File Path or Embedded File | Destination File Name | The path for the destination file. |
| 2 | Boolean | Overwrite? | Indicates whether the destination file should be overwritten if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully copied. |
| FAILURE | The source file could not be found, or the source/destination path could not be accessed. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

If the *Overwrite?* argument (Argument 2) is set to TRUE, and the destination file already exists, then the source file will be copied over the destination file. If this argument is set to FALSE and the destination file already exists, no change will occur.

Rename General File

Renames a source file to a destination file, optionally overwriting the destination file if it already exists. Since this is a rename operation, after successful completion the source file will no longer exist.

Input Arguments

| | | | |
|---|----------------------------|-----------------------|--|
| 0 | File Path or Embedded File | Source File Name | The path for the source file. |
| 1 | File Path or Embedded File | Destination File Name | The path for the destination file. |
| 2 | Boolean | Overwrite? | Indicates whether the destination file should be overwritten if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully renamed. |
| FAILURE | The source file could not be found, or the source/destination path could not be accessed. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

If the *Overwrite?* argument (Argument 2) is set to TRUE, and the destination file already exists, then the source file will be renamed over the destination file. If this argument is set to FALSE and the destination file already exists, no change will occur.

Delete General File

Deletes a file from the file system.

Input Arguments

| | | | |
|---|---------------------------------|-----------|---------------------------------|
| 0 | File Path or Embedded File Name | File Name | The path of the file to delete. |
|---|---------------------------------|-----------|---------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was deleted successfully. |
| FAILURE | The file could not be deleted or could not be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Verify General File Exists

Determines whether a file currently exists at the specified path.

Input Arguments

| 0 | File Path or Embedded File | File Name | The path of the file to check. |
|---|----------------------------|----------------------------|---|
| 1 | Step ID | Step if File Does Exist | Step to jump to if file exists. |
| 2 | Step ID | Step if File Doesn't Exist | Step to jump to if file does not exist. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was found at the specified path. |
| FAILURE | The provided path was not valid or the file was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Verify MP File Exists

Determines whether an MP file currently exists at the specified path.

Input Arguments

| 0 | File Path or Embedded File | MP File Name | The path of the mp file to check. |
|---|----------------------------|----------------------------|---|
| 1 | Step ID | Step if File Does Exist | Step to jump to if file exists. |
| 2 | Step ID | Step if File Doesn't Exist | Step to jump to if file does not exist. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mp file was found at the specified path. |
| FAILURE | The provided path was not valid or the file was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Browse For Directory

Displays a standard Windows *File > Browse for Folder* dialog prompting the user to select a file.

Input Arguments

| | | | |
|---|--------|-------------------------|--|
| 0 | String | Dialog Title (Optional) | The title to appear in the dialog's title bar. |
|---|--------|-------------------------|--|

Return Arguments

| | | | |
|---|--------|------|--------------------|
| 1 | String | Path | The selected path. |
|---|--------|------|--------------------|

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The command completed successfully. |
|---------|-------------------------------------|

Remarks

None.

Browse For File

Displays a standard Windows *File>Open* or *File>Save As* dialog prompting the user to select a file.

Input Arguments

| | | | |
|---|----------------|--|---|
| 0 | Boolean | File Open Dialog? (FALSE=Save As Dialog) | TRUE displays a File Open dialog. FALSE specifies a File Save dialog. |
| 1 | Directory Path | Working Directory (Optional) | The default location in the dialog when it is opened. |
| 2 | String | File Extension (Optional) | Specify a file extension filter to apply to the dialog. |
| 3 | String | Dialog Title (Optional) | The title to appear in the dialog's title bar. |

Return Arguments

| | | | |
|---|---------|---------------------------------|--|
| 4 | Boolean | File Selected (FALSE=Cancelled) | Contains TRUE if a file was selected, or FALSE if the user closed the dialog box or clicked the CANCEL button. |
| 5 | String | File Name | The selected file name only (without its path). |
| 6 | String | Path | The selected path only (without the file name). |
| 7 | String | Path with File Name | Both the path and file name. |

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The command completed successfully. |
|---------|-------------------------------------|

Remarks

Argument 1 supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location. If an argument is not provided at all, the default path will be the most recent save path used in SA.

Argument 2 should be a single file extension, without any other characters. For example, `txt` is a valid filter for `.txt` files.

Find Files in Directory

Retrieves a list of files on the file system matching a wildcard pattern.

Input Arguments

| | | | |
|---|---------|-------------------|--|
| 0 | String | Directory | The directory in which to search. |
| 1 | String | File Name Pattern | The Windows-standard wildcard pattern to search. |
| 2 | Boolean | Recursive? | Indicates whether subdirectories of the supplied directory should be recursively searched. |

Return Arguments

| | | | |
|---|-----------------|-------|--|
| 3 | String Ref List | Files | The list of file paths matching the provided search details. |
|---|-----------------|-------|--|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The list was retrieved successfully. |
| FAILURE | The provided directory was not found. |

Remarks

This command supports absolute paths (ex. `C:\test.xit`), or relative paths (ex. `.\test.xit`) from the *working directory* which may be SA install directory unless specified using *Set Working Directory*. To make this command behave like others that accept relative paths, first use “Set Working Directory” with `.\` as the working directory path in argument 0.

Get Directory and Filename from Path

Splits a path into a separate directory and filename.

Input Arguments

| | | | |
|---|--------|------|-----------------------|
| 0 | String | Path | The source file path. |
|---|--------|------|-----------------------|

Return Arguments

| | | | |
|---|--------|-----------|-------------------------------|
| 1 | String | Directory | The path's directory portion. |
| 2 | String | Filename | The path's filename portion. |

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The path was split successfully. |
| FAILURE | An invalid path was provided. |

Remarks

None.

File Import

Import SA File

Imports an SA file into the current job.

Input Arguments

| 0 | File Path or Embedded File | SA File Name | The path for the .XIT file to import. |
|---|----------------------------|---------------------------------|---|
| 1 | Boolean | Allow Operator Selections | TRUE to allow the user to select specific collections to import. FALSE to import all collections. |
| 2 | String Ref List | Selected Collections (Optional) | List of Collection Names to import from the specified job file |

Return Arguments

None

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found, or could not be imported. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location. Imported collections will be renamed as necessary to avoid duplicate collection names.

Import ASCII: Predefined Formats

Imports an ASCII file containing points, clouds, vectors, frames, or planes into the current job based on a predefined format.

Input Arguments

| | | | |
|---|----------------------------|-----------------|--|
| 0 | File Path or Embedded File | ASCII File Path | The path to the ASCII file to import. |
| 1 | File Format | File Format | The format of the source ASCII file. |
| 2 | Units | Units | The units of the source ASCII file. |
| 3 | Angular Units | Angular Units | The angular units of the source ASCII file. |
| 4 | Collection Object Name | Group Name | The collection and group in which to place imported points. |
| 5 | Boolean | Import As Cloud | TRUE to import points as a point cloud. FALSE to import as points. |

Return Arguments

None

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

The Collection Object Name specified (Arg.4) is used differently depending on the file format being imported (Arg. 1). A format that specifies collection and group will ignore this name entirely, while a format that includes a group name but not collection name will use the specified group name from the file and place it in the collection specified in Arg 4. This is true unless a format including uncertainty is imported which would also add an instrument, in which case, both the instrument and data are imported into the working collection.

This command supports both absolute paths (ex. C:\test.xit) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Note that this command will not fail if the specified file did not match the prescribed format. In that case, data will not be imported. Upon import, values will be converted from the source units of the file (Argument 2) to the units of the current SA file. When a format contains a group name, it will override the group name provided in the arguments.

Import ASCII: Predefined Frame Set Formats

Imports an ASCII file containing frames into the current job based on a predefined format as a Frame Set.

Input Arguments

| | | | |
|---|----------------------------|--------------------------|---|
| 0 | File Path or Embedded File | ASCII File Path | The path to the ASCII file to import. |
| 1 | File Format | File Format | The format of the source ASCII file. |
| 2 | Units | Units | The units of the source ASCII file. |
| 3 | Angular Units | Angular Units | The angular units of the source ASCII file. |
| 4 | Collection Object Name | Frame Set Container Name | The name of the resulting Frame Set |
| 5 | Boolean | Ensure Unique Name | TRUE checks to ensure the name is unique. |

Return Arguments

None

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

This command supports both absolute paths (ex. C:\test.xit) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Note that this command will not fail if the specified file did not match the prescribed format. In that case, data will not be imported. Upon import, values will be converted from the source units of the file (Argument 2) to the units of the current SA file. When a format contains a group name, it will override the group name provided in the arguments.

Import E57 File

Imports an E57 file into the current job file.

Input Arguments

| | | | |
|---|----------------------------|---------------------------------------|--|
| 0 | File Path or Embedded File | E57 File Path | The path to the STEP file to import. |
| 1 | Boolean | Saved Converted File | True saves the file after conversion |
| 2 | Boolean | Use Square Root of Intensity | True applies the square root |
| 3 | Boolean | Automatically Close Converter | True closes the converter dialog, while false allows user interaction. |
| 4 | Boolean | Prioritize Color Over Intensity | True uses color when available |
| 5 | Boolean | Import Scan Blocks As Separate Clouds | True imports the blocks separately |
| 6 | Units | Units | Units of the imported file |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import STL File

Imports a STL file into the current job file.

Input Arguments

| | | | |
|---|----------------------------|--------------------|---|
| 0 | File Path or Embedded File | STEP File Path | The path to the STEP file to import. |
| 1 | Units | Units | Units of the imported STL file |
| 2 | Boolean | Import Mesh | When TRUE, the STL file will be imported as a mesh |
| 3 | Boolean | Import Point Cloud | When TRUE, the STL file will be imported as a Point Cloud |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import STEP File

Imports a STEP file into the current job file.

Input Arguments

| 0 | File Path or Embedded File | STEP File Path | The path to the STEP file to import. |
|---|----------------------------|------------------------|---|
| 1 | Boolean | Display Entity Filters | When TRUE, display a list of entities to the user to select which to import. When FALSE, all entities are imported. |
| 2 | Boolean | Display Residuals | When TRUE, displays a detailed summary about the imported file. When FALSE, no summary is displayed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import IGES File

Imports an IGES file into the current job file.

Input Arguments

| | | | |
|---|----------------------------|----------------|--------------------------------------|
| 0 | File Path or Embedded File | IGES File Path | The path to the IGES file to import. |
|---|----------------------------|----------------|--------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import VDA/FS File

Imports a VDA/FS file into the current job file.

Input Arguments

| | | | |
|---|----------------------------|------------------|--|
| 0 | File Path or Embedded File | VDA/FS File Path | The path to the VDA/FS file to import. |
|---|----------------------------|------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import SAT File

Imports a SAT file into the current job file.

Input Arguments

| | | | |
|---|----------------------------|---------------|-------------------------------------|
| 0 | File Path or Embedded File | SAT File Path | The path to the SAT file to import. |
|---|----------------------------|---------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import File As Embedded File

Imports any file and embeds it into the active collection.

Input Arguments

| 0 | File Path or Embedded File | External File Name | The name of the file to embed. |
|---|----------------------------|--------------------|--|
| 1 | Boolean | Replace Existing? | Specifies whether the imported file will replace an existing embedded file with the same name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import MP File As Embedded MP

Imports an external MP file and embeds it into the active collection.

Input Arguments

| 0 | File Path or Embedded File | External MP File Name | The name of the MP file to embed. |
|---|----------------------------|-----------------------|--|
| 1 | Boolean | Replace Existing? | Specifies whether the imported file will replace an existing embedded MP with the same name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If a relative path is provided and the MP is embedded, the path will be relative to the current SA file location.

Import File as Picture

Imports an external file as a picture file in the active collection.

Input Arguments

| 0 | File Path or Embedded File | External MP File Name | The name of the picture to import |
|---|----------------------------|-----------------------|--|
| 1 | Boolean | Replace Existing? | Specifies whether the imported file will replace an existing picture with the same name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

This command supports standard picture formats such as *.jpg, *.png.

Direct CAD Access

Imports a file directly from a CAD file format (SA Ultimate + Native CAD only).

Input Arguments

| 0 | File Path or Embedded File | CAD File Name | The name of the file to import. |
|----|----------------------------|---|---|
| 1 | Boolean | Import Solids | Indicates whether solids should be imported from the file. |
| 2 | Boolean | Import Surfaces | Indicates whether surfaces should be imported from the file. |
| 3 | Boolean | Import Polygonized Surfaces | Indicates whether polygonized surfaces should be imported from the file. |
| 4 | Boolean | Import Annotations | Indicates whether annotations should be imported from the file. |
| 5 | Boolean | Import Vectors | Indicates whether vectors should be imported from the file. |
| 6 | Boolean | Import Points | Indicates whether points should be imported from the file. |
| 7 | String | Point Group Name | Point Group name used for the imported points. |
| 8 | Boolean | Import Attributes/Metadata | Indicates whether metadata should be imported from the file. |
| 9 | Boolean | Import Coordinate Frames | Indicates whether coordinate frames should be imported from the file. |
| 10 | Boolean | Import Planes | Indicates whether planes should be imported from the file. |
| 11 | Boolean | Import 3D Curves - Lines | Indicates whether lines should be imported from the file. |
| 12 | Boolean | Import 3D Curves - Circles | Indicates whether circles should be imported from the file. |
| 13 | Boolean | Import 3D Curves - General Curves | Indicates whether general curves should be imported from the file. |
| 14 | Boolean | Import Construction Geometry | Indicates whether construction geometry should be imported from the file. |
| 15 | Boolean | Import Hidden Entities | Indicates whether hidden entities should be imported from the file. |
| 16 | Boolean | Import all Surfaces as Mesh Graphical Entities | Indicates whether surfaces should be imported as mesh graphical entities (see Remarks below). |
| 17 | Boolean | Do Not Import Fillets | Surfaces marked as fillets in the source format will not be imported (to save memory/improve responsiveness). |
| 18 | Boolean | Do Not Import Dittos | Objects marked as dittos in the source format will not be imported. |
| 19 | Integer | Ditto Threshold | Indicates the maximum number of dittos that should be imported. |
| 20 | Boolean | Center View on Imported Objects | Indicates whether the graphical view should be centered on the imported objects. |
| 21 | Boolean | Import into Folders matching CAD file hierarchy | Indicates whether objects will be imported into folders in the tree matching the hierarchy in the CAD file. |
| 22 | Boolean | Remove Empty Folders | Indicates whether empty folders imported from the CAD hierarchy should be deleted. |

| | | | |
|----|---------|---|--|
| 23 | Integer | Surface Normals Mode (1 or 2) | Indicates the surface normals mode to use when importing. If mode 1 causes issues, try mode 2. |
| 24 | Boolean | Prompt on Missing Components | Notify user when components are missing from the selected CAD model. |
| 25 | Boolean | Selective Import | True to display the Selective Import dialog when opening the selecting file |
| 26 | Boolean | Surface Compatibility Mode | True enables Surface Compatibility mode. |
| 27 | Boolean | Explode Surfaces | True enables exploded surfaces |
| 28 | String | CAD File Units (leave blank to use the units specified in the file) | The units used by the selected CAD file. |
| 29 | Boolean | Build Callout Views | True will build callout views included the import GD&T annotations |

Return Arguments

| | | | |
|----|---------|-------------------------|---|
| 30 | Boolean | Import Warnings | Indicates if import warnings were generated. |
| 31 | String | Import Warning Messages | Contains a list of any import warning messages. |
| 32 | Vector | Extents Min | The minimum X/Y/Z extents of the imported CAD (in the active frame). |
| 33 | Vector | Extents Max | The maximum X/Y/Z extents of the imported CAD (in the active coordinate frame). |

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

Surfaces imported as Mesh Graphical Entities save a considerable amount of memory and processing power--but the entities cannot be analyzed. They are purely a “picture” and cannot be used for point-to-surface analyses.

Argument 26 provides a “Surface Compatibility Mode” option to CAD import. If you turn it on, we run the surfaces through a conditioning tool which should create a more compatible representation of the model. The idea is that it may import a model which is otherwise causing either import or graphical problems.

Argument 27, “Explode Surfaces”, provides a means to dissect a model on import such that each face is imported as a separate object.

Import SA Windows Placement

Imports a windows placement file and updates stored dialog/window positions and sizes to match what is stored in the file.

Input Arguments

| | | | |
|---|----------------------------|-----------|---|
| 0 | File Path or Embedded File | File Path | The path to the windows placement file. |
|---|----------------------------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

A windows placement file can be created by choosing *File* ► *Export* ► *Window Placements*.

Import VSTARS .xyz File

Imports an .xyz VSTARS file into the current SA job file.

Input Arguments

| | | | |
|---|----------------------------|-----------|---|
| 0 | File Path or Embedded File | File Path | The path to the VSTARS .xyz file to import. |
|---|----------------------------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

None.

Import VSTARS Cameras

Imports a camera file (outstar.txt) from VSTARS into the current SA job file.

Input Arguments

| | | | |
|---|----------------------------|-----------|---|
| 0 | File Path or Embedded File | File Path | The path to the VSTARS camera file to import. |
|---|----------------------------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found or the format was not recognized. |

Remarks

None.

Import Leica GSI File

Imports a Leica GSI file.

Input Arguments

| | | | |
|---|----------------------------|---------------|---|
| 0 | Collection Instrument ID | Instrument ID | The ID of the instrument with which to associate the file's data. |
| 1 | Collection Object Name | Group Name | The group into which to import the data. |
| 2 | File Path or Embedded File | File Path | The path to the file to import. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file or instrument could not be found. |

Remarks

None.

Import Leica SDB File

Imports a Leica SDB scan file.

Input Arguments

| | | | |
|---|----------------------------|-----------------|---|
| 0 | Collection Instrument ID | Instrument ID | The ID of the instrument with which to associate the file's data. |
| 1 | Collection Object Name | Scan Cloud Name | The point cloud into which to import the data. |
| 2 | File Path or Embedded File | File Path | The path to the file to import. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file or instrument could not be found. |

Remarks

None.

Import Hidden Point Bar XML File

Imports an XML file containing a list of hidden point bar definitions. This is specific to the hidden-point bar database definitions in the Users Options. Refer to the Points chapter of the Users Manual for more information on hidden point bar definitions.

Input Arguments

| 0 | File Path or Embedded File | XML File Path | The path to the file to import. |
|---|----------------------------|---------------------------|---|
| 1 | Boolean | Replace Existing Entries? | True replaces any existing hidden point definitions in the User Options with those defined in the specified file. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully imported. |
| FAILURE | The source file could not be found. |

Remarks

Here is a basic example of an XML file with one of each type of bar definition included as an example of the expected formatting:

```
<?xml version="1.0" encoding="UTF-8"?>
-<HiddenPointAdapters>
<!-- Units: M = 0, CM = 1, MM = 2, FT = 3, IN = 4 -->
<Units Units="4"/>
-<HiddenPointAdapter>
    <Type Type="Two Point Bar"/>
    <Name Name="PtBar0"/>
    <AToB AToB="10.000000"/>
    <ToC ToC="15.000000"/>
    <FromA FromA="TRUE"/>
    <InterPointTolerance InterPointTolerance="0.000000"/>
    <PlanarOffset PlanarOffset="0.000000"/>
    <RadialOffset RadialOffset="0.000000"/>
</HiddenPointAdapter>
```

```

-<HiddenPointAdapter>
  <Type Type="Gravity Bar"/>
  <Name Name="PtBar1"/>
  -<FrameOfReference Name="A::WORLD">
    <VectorNamed Name="X-Axis" Z="0.000000" Y="0.000000" X="1.000000"/>
    <VectorNamed Name="Y-Axis" Z="0.000000" Y="1.000000" X="0.000000"/>
    <VectorNamed Name="Z-Axis" Z="1.000000" Y="0.000000" X="0.000000"/>
  </FrameOfReference>
  <VerticalOffset VerticalOffset="8.000000"/>
</HiddenPointAdapter>

```

```

-<HiddenPointAdapter>
  <Type Type="Multiple Point Fixture"/>
  <Name Name="PtBar2"/>
  <InterPointTolerance InterPointTolerance="0.000000"/>
  <PlanarOffset PlanarOffset="0.000000"/>
  <RadialOffset RadialOffset="0.000000"/>
  -<ReferencePoints>
    <VectorNamed Name="r0" Z="0.800592" Y="53.596606" X="39.176611"/>
    <VectorNamed Name="r1" Z="-0.563036" Y="95.184179" X="101.150548"/>
    <VectorNamed Name="r2" Z="0.318155" Y="35.621204" X="139.086581"/>
  </ReferencePoints>
  -<ProbePoint>
    <VectorNamed Name="r3" Z="0.261696" Y="95.217750" X="107.248146"/>
  </ProbePoint>
</HiddenPointAdapter>
</HiddenPointAdapters>

```

File Export

Export ASCII Points

Exports one or more groups of points to an ASCII-formatted file.

Input Arguments

| | | | |
|----|--------------------------------|--|---|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Group Name Ref List | Group Names to export | A list of point groups to export. |
| 2 | Export Delimiter Spec | Data Delimiter | A character separating data values. (Space or Comma). |
| 3 | Target Name Format | Target Name Format | Specifies how point/target names are specified in the ASCII file. |
| 4 | Coordinate System Type | Desired Coordinate System | Specify the type of coordinate system in which to export the points. |
| 5 | Boolean | Include Target Offsets? | Specify whether to include the stored offsets for each point. |
| 6 | Boolean | Include Target Comments? | Specify whether to include the comments stored with each point. |
| 7 | Boolean | Include Timestamps? | Specify whether to include measurement timestamps with each point. |
| 8 | Boolean | Include Tolerances? | Specify whether to include stored tolerances with each point. |
| 9 | Boolean | Include Coordinate Uncertainties? | Specify whether to include calculated uncertainties with each point. |
| 10 | Boolean | Include SA version and frame comments? | Specify whether to include the version of SA used and the frame comments. |
| 11 | Boolean | Include Axis Comments? | Specify whether to include axis comments with each point. |
| 12 | Boolean | Include Export Format Info? | Specify whether to include export format information. |
| 13 | Boolean | Include Weights? | Specify whether to include weights of each point. |
| 14 | Boolean | Include Measurement Details? | Specify whether to include details with each measurement. |
| 15 | Boolean | Maximum Precision (Scientific Notation?) | Specify whether to export numbers with full internal precision (15 digits). |
| 16 | Integer | Decimal Precision | The number of digits following the decimal. |
| 17 | Boolean | Append? | Specify whether to add the ASCII data to an existing file, if it exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file could not be saved, or none of the provided point groups could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 13 is set to FALSE, the specified file will be overwritten if it already exists.

Export ASCII Point Sets

Exports a select Point Set to an ASCII-formatted file.

Input Arguments

| | | | |
|----|----------------------------|--|---|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name | Point Set Container | Point Set containing the points to export. |
| 2 | Export Delimiter Spec | Data Delimiter | A character separating data values. (Space or Comma). |
| 3 | Target Name Format | Target Name Format | Specifies how point/target names are specified in the ASCII file. |
| 4 | Coordinate System Type | Desired Coordinate System | Specify the type of coordinate system in which to export the points. |
| 5 | Boolean | Include Target Offsets? | Specify whether to include the stored offsets for each point. |
| 6 | Boolean | Include Timestamps? | Specify whether to include measurement time-stamps with each point. |
| 7 | Boolean | Include SA version and frame comments? | Specify whether to include the version of SA used and the frame comments. |
| 8 | Boolean | Include Axis Comments? | Specify whether to include axis comments with each point. |
| 9 | Boolean | Include Export Format Info? | Specify whether to include export format information. |
| 10 | Boolean | Maximum Precision (Scientific Notation?) | Specify whether to export numbers with full internal precision (15 digits). |
| 11 | Integer | Decimal Precision | The number of digits following the decimal. |
| 12 | Boolean | Append? | Specify whether to add the ASCII data to an existing file, if it exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file could not be saved, or none of the provided point groups could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 13 is set to FALSE, the specified file will be overwritten if it already exists.

Export ASCII Frame Set

Exports a Frame Set to an ASCII-formatted file.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|--|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name | Frame Set Container | The Frame Set to be exported |
| 2 | Export Delimiter Spec | Data Delimiter | A character separating data values. (Space or Comma). |
| 3 | File Format | File Format | Format to be used in the Ascii file |
| 4 | Boolean | Include Export Format Info? | Create a header with the format information? |
| 5 | Integer | Decimal Precision | Decimal precision to use in the resulting file |
| 6 | Boolean | Append? | Specify whether to add the ASCII data to an existing file, if it exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully exported. |
| FAILURE | The export file could not be written, or frame select could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 6 is set to FALSE, the specified file will not be written if it already exists.

Export ASCII Frames

Exports a selection of Frames to an ASCII-formatted file.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|--|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name Ref List | Object List | The list of frames to be exported as part of the ascii file. |
| 2 | Export Frame Mode | Export Frame Mode | Selection list of frame export formats. |
| 3 | Boolean | Overwrite existing file? | Specify whether to add the ASCII data to an existing file, if it exists. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully exported. |
| FAILURE | The export file could not be written, or none of the provided frames could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 6 is set to FALSE, the specified file will not be written if it already exists.

Export ASCII Point Clouds

Exports one or more point clouds to an ASCII-formatted file.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|---|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name Ref List | Point Cloud List | A list of point clouds to export. |
| 2 | Export Delimiter Spec | Data Delimiter | A character separating data values. (Space or Comma). |
| 3 | Boolean | Overwrite existing file? | Specify whether to add the ASCII data to an existing file, if it exists. |
| 4 | Boolean | Show Progress Dialog? | Specify whether a progress dialog should be shown as clouds are exported. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully exported. |
| FAILURE | The export file could not be written, or none of the provided point clouds could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 3 is set to FALSE, the specified file will not be written if it already exists.

Export PTX Point Clouds

Exports one or more point clouds to an PTX formatted file.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|--|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name Ref List | Point Cloud List | A list of point clouds to export. |
| 2 | Boolean | Overwrite existing file? | Specify whether to replace the a file with the same name if it already exists. |
| 3 | Boolean | Show Progress Dialog? | Specify whether a progress dialog should be shown as clouds are exported. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully exported. |
| FAILURE | The export file could not be written, or none of the provided point clouds could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 3 is set to FALSE, the specified file will not be written if it already exists.

Export Vector Container to Excel File

Exports data associated with a vector group to an Excel file. The command exports begin, end, delta, and magnitude information for each vector.

Input Arguments

| | | | |
|---|----------------------------|--------------------------|---|
| 0 | File Path or Embedded File | Excel File Path | The path to the file to save. |
| 1 | Vector Group Name | Vector group to export | The Vector group to export to the file. |
| 2 | Boolean | Overwrite existing file? | Specify whether to overwrite the file if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file could not be saved, or none of the provided point groups could be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). To avoid an error dialog when opening the file, the path should match the version of Excel installed on the machine. If Office 2000/2003, use a `.xls` extension. If Office 2007 and later is installed, use a `.xlsx` extension. If Argument 2 is set to `FALSE`, the specified file will not be written if it already exists, and the step will return `FAILURE`. If Argument 2 is `TRUE`, and the file already exists, the user will be prompted to confirm the overwriting process.

Export Vector Container to ASCII File

Exports data associated with a vector group to a comma-delimited ASCII file. The command exports begin, delta, and magnitude information for each vector.

Input Arguments

| | | | |
|---|----------------------------|---|--|
| 0 | File Path or Embedded File | ASCII File Path | The path to the file to save. |
| 1 | Collection Object Name | Vector group to export | The collection object name for the vector group to export. |
| 2 | Boolean | Overwrite existing file? | Specify whether to overwrite the file if it already exists. |
| 3 | Boolean | Use Full Precision (Scientific Notation)? | If TRUE, the data will be exported with full decimal precision |
| 4 | Vector Name Format | Vector Name Format | The format for the vector name. |
| 5 | Boolean | Include Length? | If TRUE, the length of the vector will be exported as well. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was successfully exported. |
| FAILURE | The export file could not be written, the vector group could not be found, or Argument 2 was FALSE and the file already exists. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). If Argument 2 is set to FALSE, the specified file will not be written if it already exists, and the step will return FAILURE. If Argument 2 is TRUE, and the file already exists, the user will be prompted to confirm the overwriting process.

Export STEP File - Entire Model

Exports a STEP file containing all STEP-compatible objects from the current SA job file.

Input Arguments

| | | | |
|---|----------------------------|----------------|-------------------------------|
| 0 | File Path or Embedded File | STEP File Path | The path of the file to save. |
|---|----------------------------|----------------|-------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file path is valid and the STEP file was successfully exported. |
|---------|---|

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export STEP File - Partial Model

Exports a STEP file containing a list of selected objects.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | File Path or Embedded File | STEP File Path | The path of the file to save. |
| 1 | Collection Object Name Ref List | Object Name List | A list of objects to export to the file. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The file path is valid and the STEP file was successfully exported. |
| PARTIAL SUCCESS | One or more objects in the list was not valid. |
| FAILURE | The object list is empty, or no objects were valid. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export IGES File - Entire Model

Exports an IGES file containing all IGES-compatible objects from the current SA job file.

Input Arguments

| | | | |
|---|----------------------------|----------------|-------------------------------|
| 0 | File Path or Embedded File | IGES File Path | The path of the file to save. |
|---|----------------------------|----------------|-------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file path is valid and the IGES file was successfully exported. |
|---------|---|

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export IGES File - Partial Model

Exports an IGES file containing a list of selected objects.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | File Path or Embedded File | IGES File Path | The path of the file to save. |
| 1 | Collection Object Name Ref List | Object Name List | A list of objects to export to the file. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The file path is valid and the IGES file was successfully exported. |
| PARTIAL SUCCESS | One or more objects in the list was not valid. |
| FAILURE | The object list is empty, or no objects were valid. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export VDA/FS File - Entire Model

Exports a VDA/FS file containing all compatible objects from the current SA job file.

Input Arguments

| | | | |
|---|----------------------------|------------------|-------------------------------|
| 0 | File Path or Embedded File | VDA/FS File Path | The path of the file to save. |
|---|----------------------------|------------------|-------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file path is valid and the VDA/FS file was successfully exported. |
|---------|---|

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export VDA/FS File - Partial Model

Exports a VDA/FS file containing a list of selected objects.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | File Path or Embedded File | VDA/FS File Path | The path of the file to save. |
| 1 | Collection Object Name Ref List | Object Name List | A list of objects to export to the file. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The file path is valid and the VDA/FS file was successfully exported. |
| PARTIAL SUCCESS | One or more objects in the list was not valid. |
| FAILURE | The object list is empty, or no objects were valid. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export Embedded File

Exports an embedded file to an external file.

Input Arguments

| | | | |
|---|----------------------------|-------------------------------|---|
| 0 | Collection Name | Embedded File Collection Name | The collection containing the embedded file to export. |
| 1 | String | Embedded File Name | The name of the embedded file to export. |
| 2 | File Path or Embedded File | External File Name | The filename of the external file to save. |
| 3 | Boolean | Replace Existing? | Specify whether to overwrite the provided external file if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file could not be exported. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Export DXF

Exports a set of points to a DXF file.

Input Arguments

| | | | |
|---|----------------------------|---------------|---|
| 0 | File Path or Embedded File | DXF File Path | The path of the file to save. |
| 1 | Point Name Ref List | Point Names | A list of points to export to the DXF format. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file path is valid and the DXF file was successfully exported. |
| FAILURE | The file path is invalid or the points could not be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.dxf`) and relative paths (ex. `.\test.dxf`).

Export Scan Stripe Mesh to STL File

Exports a Scan Strip Mesh in STL format.

Input Arguments

| | | | |
|---|----------------------------|---------------|--|
| 0 | File Path or Embedded File | STL File Path | The path of the file to save. |
| 1 | Collection Object Name | Mesh | The name of the Scan Strip Mesh to export. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file path is valid and the STL file was successfully exported. |
| FAILURE | The file path is invalid or the mesh could not be found. |

Remarks

None.

Export Hidden Point Bar XML File

Exports an XML file containing a list of hidden point bar definitions. This is specific to the hidden-point bar database definitions in the Users Options. Refer to the Points chapter of the Users Manual for more information on hidden point bar definitions.

Input Arguments

| | | | |
|---|----------------------------|---------------------------|---|
| 0 | File Path or Embedded File | XML File Path | The path to the file to import. |
| 1 | Boolean | Replace Existing Entries? | True replaces any existing hidden point definitions in the User Options with those defined in the specified file. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file path is invalid. |

Remarks

None.

QDAS File Export...

Import QDas Catalog File

The command imports a DFD Catalog file for use while the MP is running.

Input Arguments

| | | | |
|---|----------------------------|--------------------|---|
| 0 | File Path or Embedded File | QDAS DFD File Path | The path of the (*.dfd) file to import. |
|---|----------------------------|--------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file path is valid and the file was successfully imported. |
| FAILURE | The file could not be found. |

Remarks

None.

Get QDAS Catalog Entries

The command creates a string list of QDAS catalog entries applicable to a target Kxxxx field. For example, for the target K0008 field the result is a list of operators' names recorded in the Q-DAS catalog.

Input Arguments

| | | | |
|---|-----------------|-----------------|-------------------------|
| 0 | String | K-Field Target | K-Field Target |
| 1 | String Ref List | Catalog Entries | List of Catalog Entries |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | List of entries were returned successfully |
| FAILURE | The catalog could not be found or the K-field could not be accessed. |

Remarks

None.

Set K-Field from QDAS Catalog

The command provides the ability to re-load a Q-DAS catalog if needed and select a record identifier in the QDAS catalog per each Kxxxx target field.

The record identifier selection is optional. If nothing is selected (default value -1), the target field will not be added to QDAS export file.

Input Arguments

None.

Return Arguments

| | | | |
|---|---------|---------------------------------|--------------------------|
| 0 | Integer | K0005: Event Identifier | Event Identifier |
| 1 | Integer | K0007: Cavity Identifier | Cavity Identifier |
| 2 | Integer | K0008: Operator Identifier | Operator Identifier |
| 3 | Integer | K0010: Machine Identifier | Machine Identifier |
| 4 | Integer | K0011: Process Param Identifier | Process Param Identifier |
| 5 | Integer | K0012: Gage Identifier | Gage Identifier |

Returned Status

| | |
|---------|--|
| SUCCESS | User selected K-Fields were returned successfully. |
| FAILURE | The catalog could not be found. |

Remarks

This command returns the index of the user selected string, not the string its self. But this integer is what is needed in the *.dfq export step. String 0 is always “-1: N/A” for each K-Field if no user selection is made or no options were available in the catalog.

Get QDAS Catalog Entry Identifier

Much like *Set K-Field from QDAS Catalog* this command offers the user an option to select a record identifier for the target Kxxxx field from the imported QDAS catalog. The advantage here is that you can select any K-Field and build a question to ask using a string reference list.

Input Arguments

| | | | |
|---|-----------------|----------------|----------------|
| 0 | String | K-Field Target | K-Field Target |
| 1 | String Ref List | User Prompt | User Prompt |
| 2 | Font Type | Font | Font to use |

Return Arguments

| | | | |
|---|---------|------------------|------------------|
| 3 | Integer | Entry Identifier | Returned Integer |
|---|---------|------------------|------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The user selection was returned successfully |
| FAILURE | The catalog could not be found. |

Remarks

This command returns the index of the user selected string, not the string its self. But this integer is what is needed in the *.dfq export step. String 0 is always “-1: N/A” for each K-Field if no user selection is made or no options were available in the catalog.

Prepare QDAS Data List

This command creates a QDAS data list of the most common Kxxxx fields needed for part's description and its characteristics. These are defined in advance and can be exported using *Export QDAS Data List*.

A comprehensive export option is to use *Prepare QDAS Data List* along with *Add Ke-Field to QDAS Data List* and finally *Export QDAS Data List* to build a complete *.dfq file for your part.

Input Arguments

| | | | |
|----|---------------------------------|-----------------------------|---|
| 0 | String | K1001: Part Number | Part Number |
| 1 | String | K1002: Part Description | Part Description |
| 2 | String | K1071: Supplier Number | Supplier Number |
| 3 | String | K1072: Supplier Description | Supplier Description |
| 4 | String | K1203: Reason for Test | Reason for Test |
| 5 | String | K1303: Plant | Plant |
| 6 | String | K1900: Part Remark | Part Remark |
| 7 | String | K0006: Batch Number | Batch Number |
| 8 | String | K0014: Part ID | Part ID |
| 9 | String | K0053: Order Number | Order Number |
| 10 | String | K0004: Date Time Stamp | Date Time Stamp |
| 11 | Integer | K0008: Operator Identifier | Operator Identifier |
| 12 | integer | K0010: Machine Identifier | Machine Identifier |
| 13 | Integer | K0012: Gage Identifier | Gage Identifier |
| 14 | Relationship Ref List | Relationship List | Relationships from which criteria will be recorded |
| 15 | Feature Check Ref List | Feature Check List | List of GD&T Checks to record |
| 16 | Dimension Ref List | Dimension List | List of dimension to record |
| 17 | Collection Object Name Ref List | Vector Group List | List of Vector Groups from which criteria will be recorded. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The QDAS Data List has been defined successfully. |
| FAILURE | The catalog could not be found or reference was invalid. |

Remarks

A fair bit of argument verification has been added to ensure data completeness and validity.

The following arguments must have values:

- A0 - K1001 Part Number
- A1 - K1002 Part Description
- A14, A15, A16, or A17 – at least one of the arguments must have a value

Add K-Field to QDAS Data List

Once “Prepare QDAS Data List” command is in the script (prerequisite), then based on customer needs, this command provides the ability to add to the QDAS data list.

A comprehensive export option is to use *Prepare QDAS Data List* along with *Add Ke-Field to QDAS Data List* and finally *Export QDAS Data List* to build a complete *.dfq file for your part.

Input Arguments

| | | | |
|---|-----------------|--------------------------|--|
| 0 | String | Field Name | Optional info to simplify MP script Reading |
| 1 | String | K-Field | Mandatory field in the following format: Kxxxx, where xxxx is a numeric value between 1-9999 |
| 2 | Integer | Characteristic Number | Number to indicate to which characteristic it is applicable. |
| 3 | Integer | Value Max Length | Mandatory field defined by QDAS Specifications |
| 4 | QDAS Value Type | Value Type | Mandatory field defined by QDAS Specifications - Pick from List. |
| 5 | String | Value | A Kxxxx field value in string format that should satisfy requirements defined by A3 and A4 |
| 6 | Integer | Catalog Entry Identifier | A Kxxxx field value as a reference to a catalog entry. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The K-Field was defined successfully. |
| FAILURE | The K-Field is invalid or one of its entries. |

Remarks

Additional Kxxxx field can be added except the “structural info” K5xxx block. The structural information of logical groups in the output DFQ file is generated automatically based on the SA Tree structure. If it is needed user may re-organize SA tree in the desired groups before export.

The structural information of logical groups in the output DFQ file is generated automatically based on SA Tree structure. If needed a user may re-organize the SA tree in the desired groups before initiating Q-DAS export.

Important Notes:

- It is a responsibility of user to use Q-DAS specifications to properly define the required K-Field type, max length, and field type. According to entered information, SA will validate user input.
- Either A5 or A6 argument must be defined, but not simultaneously. If both arguments have values by mistake, the A6 will define the field's value.

Export QDAS Data List

This command exports the existing QDAS Data List into the specified *.DFQ file.

A comprehensive export option is to use *Prepare QDAS Data List* along with *Add Ke-Field to QDAS Data List* and finally *Export QDAS Data List* to build a complete *.dfq file for your part.

Input Arguments

| | | | |
|---|----------------------------|-----------------------|---|
| 0 | File Path or Embedded File | QDAS Export File Path | Path and file name for the exported file. |
|---|----------------------------|-----------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The QDAS Data List was exported successfully. |
| FAILURE | The catalog could not be found or the path is invalid. |

Remarks

None.

Export QDAS Characteristics

Exports a *.dfq file for use in Hexagon's Q-DAS packages. This command provides a baseline export option.

A comprehensive export option is to use *Prepare QDAS Data List* along with *Add Ke-Field to QDAS Data List* and finally *Export QDAS Data List* to build a complete *.dfq file for your part.

Input Arguments

| 0 | File Path or Embedded File | QDAS Export File Path | The path of the file to save. |
|----|---------------------------------|-----------------------------|---|
| 1 | String | K1001: Part Number | Part Number to save with the file |
| 2 | String | K1002: Part Description | Part Description |
| 3 | String | K1071:Supplier Number | Supplier Number |
| 4 | String | K1072: Supplier Description | Supplier Description |
| 5 | String | K1203: Reason for Test | Reason for Test |
| 6 | String | K1303: Plant | Plant |
| 7 | String | K1900: Part Remark | Part Remark |
| 8 | String | K0006: Batch Number | Batch Number |
| 9 | String | K0014: Part ID | Part ID |
| 10 | String | K0053: Order Number | Order Number |
| 11 | String | K0004: Date Time Stamp | Date Time Stamp |
| 12 | Integer | K0008: Operator Identifier | Operator Identifier |
| 13 | Integer | K0010: Machine Identifier | Machine Identifier |
| 14 | Integer | K0012: Gage Identifier | Gage Identifier |
| 15 | Relationship Ref List | Relationship List | Relationships from which criteria will be recorded |
| 16 | Feature Check Ref List | Feature Check List | List of GD&T Checks to record |
| 17 | Dimensions Ref List | Dimension list | List of dimension to record |
| 18 | Collection Object Name Ref List | Vector Group List | List of Vector Groups from which criteria will be recorded. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully exported. |
| FAILURE | The file path is invalid or the referenced items could not be found. |

Remarks

9 new arguments for supplier and part description were added in 2022.3. The export was also limited to only relationship criteria with reporting enabled and tolerances set. Vector Characteristics - User can control what vector components to export through Vector Group report options.

This command duplicates the manual export capabilities available under File>Export>Other>Q-DAS Part Data. For more information refer to the *Q-DAS ASCII Transferformat* document from Q-DAS.

Example:

K2001/1 Diameter-E78177CA

Characteristic Number

| | |
|---------------------------------------|--|
| K2002/1 A::BoltHole Pattern1-Diameter | Characteristic Description (Collection Item Name - Criteria) |
| K2009/1 202 | Measurement type (202 = diameter) |
| K0001/1 6.533426e-01 | Measured Value |
| K2101/1 6.456693e-01 | Nominal Value |
| K2112/1 -5.000000e-03 | Lower Allowance |
| K2113/1 5.000000e-03 | Upper Allowance |
| K2120/1 1 | Lower Limit Type |
| K2121/1 1 | Upper Limit Type |
| K2142/1 in | Unit Description |
| K2022/1 4 | Decimal Places |

ASCII Data File Operations

Clear All ASCII Files

Clears all Ascii files.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Open ASCII File

Opens an ASCII file and returns a handle to that file in preparation for reading from or writing to it.

Input Arguments

| | | | |
|---|----------------------------|-----------------|-------------------------------------|
| 0 | File Path or Embedded File | ASCII File Path | The path to the ASCII file to open. |
|---|----------------------------|-----------------|-------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------------------|--|
| 1 | Integer | ASCII File Handle | The handle to the opened ASCII file (for later referencing). |
| 2 | Integer | ASCII File Size (Lines) | The number of lines in the ASCII file. |


Returned Status

| | |
|---------|--|
| SUCCESS | The file was opened successfully. |
| FAILURE | The file path is invalid or the referenced items could not be found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

The file need not necessarily exist, since you may be creating a new file for writing.

 **Warning:** Be careful to close every ASCII file that you open. Failing to close an open ASCII file could lead to unusual behavior with the ASCII commands until SA is restarted.

Write ASCII Line

Writes one or more boolean, integer, double, string, vector, transform, or double lists to a line in an ASCII file, separated by commas (comma-delimited).

Input Arguments

| | | | |
|---|---------|-------------------|--|
| 0 | Integer | ASCII File Handle | The handle of the open ASCII file to write to. |
| 1 | Boolean | MakeCSVRow | Indicates whether the arguments should be combined in CSV-compliant fashion. If set to FALSE, the arguments will be added directly with no separators or other processing. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The line was read successfully. |
| FAILURE | The file reference is invalid or write failed. |

Remarks

A valid file handle that has previously been opened must be provided. This command will always write to the end of the file.

If you need to write quotation marks to the file, set the *MakeCSVRow* argument to FALSE.

Read ASCII Line (Iterator)

Reads one or more boolean, integer, double, string, vector, transform, or double lists values from a line in an ASCII file. This command allows the script writer to define the number and type return arguments

Input Arguments

| | | | |
|---|---------|-----------------------------|---|
| 0 | Integer | ASCII File Handle | The handle of the open ASCII file to read from. |
| 1 | Integer | Line Index | The line to start reading from. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to once the iterator has finished reading all lines from the file. |

Return Arguments

Matching the items added to the list to be read from the file. To add arguments select the step in your script. When you do, additional buttons will appear at the top of the MP editor providing Add, Remove, Edit and Read controls (*Figure 2-1*).

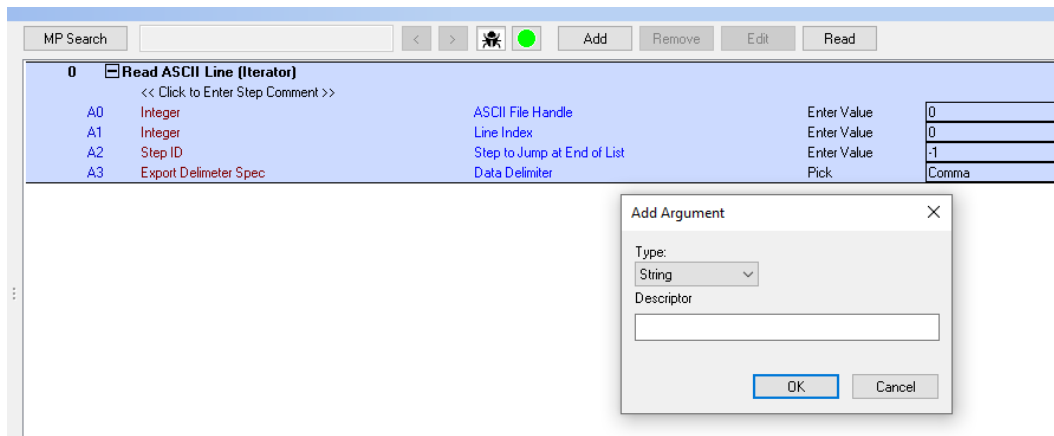


Figure 2-1. Defining a return argument in the Read Ascii Line command.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was read successfully. |
| FAILURE | The file path is invalid or the referenced items could not be found. |

Remarks

A valid file handle that has previously been opened must be provided. The line index is zero-based, so the first line in the file is line zero.

You can jump to a specific comma-delimited entry in the line by using the [Add Column to Read](#) button, in which you specify the zero-based column to read from. This will place the “cursor” at this column in preparation for reading an item, which is defined by clicking the [Add](#) button.

You can jump around to different nonsequential columns in this way.

Close ASCII File

Closes an ASCII file that has previously been opened via Open ASCII File, optionally saving it.

Input Arguments

| | | | |
|---|---------|-------------------|---|
| 0 | Integer | ASCII File Handle | The handle of the open ASCII file to close. |
| 1 | Boolean | Save? | Indicates whether the file should be saved when closed (whether changes should be discarded). |

Return Arguments

None.

Returned Status

| | |
|---------|-----------------------------------|
| SUCCESS | The file was closed successfully. |
| FAILURE | The file reference is invalid. |

Remarks

A valid file handle that has previously been opened must be provided.

Datashare Operations

Load DataShare File

Loads a set of arguments from a DataShare file. DataShare files are text files that define a set of named values for easy data storage. They are read and parsed adding additional arguments to the MP command, providing a convenient way to have access to data.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | Boolean | Make Variables? | True builds variables from the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully loaded. |
| FAILURE | The file could not be loaded or had an invalid format. |

Remarks

In order to populate this command with the proper list of arguments (so that they can be referenced later in the script), you must first specify a path to an actual DataShare file of the same format, then click the [Refresh Arguments](#) button. This will populate the command with the appropriate arguments, and the file path can subsequently be set to the desired value.

The expected dataShare format is as follows:

<ASCII>

<S: String Name>

"String"

<B: Boolean Name>

0

<I: Integer Name>

32

<D: Double Name>

3.20000000000000002e+00

<V: Vector Name>

1.000000 0.020000 0.030000

<T: Transform name>

0.010000 0.020000 0.030000 0.040000 0.050000 0.060000

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`). This command is not supported via the SA SDK--data can be loaded from files using language-specific commands.

Make Variables? If enabled, the dataShare values are saved as variables, defined using the argument names, and can be used while a script is running. This option was added in 2022.3 and will build variables as long as the argument names are unique.

Save DataShare File

Saves a set of arguments to a DataShare file.

Input Arguments

| | | | |
|------|----------------------------|--------------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | Boolean | Save in Binary Format? | Specify whether to save a file in binary or ASCII format. |
| 2 | Boolean | Append to existing file? | Specify whether to append the arguments to an existing file, if that file already exists. |
| 3->n | USER | USER | ADDITIONAL CUSTOM ARGUMENTS |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The file was successfully saved. |
| FAILURE | The file could not be saved. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

In order to populate this command with a list of arguments, click the [Add/Remove/Edit](#) buttons. Specify the argument type and a description, and any number of additional arguments will be added to the list.

ASCII DataShare files are human-readable. Saving DataShare files to binary format removes human readability, but processing binary files is significantly faster and requires smaller file sizes.

Get Boolean From DataShare File

Retrieves a boolean value from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Boolean Name | The Boolean name to retrieve from datashare file. |

Return Arguments

| | | | |
|---|---------|---------------|---|
| 2 | Boolean | Boolean Value | The boolean value retrieved from the datashare file, if that file exists. |
|---|---------|---------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The boolean value was successfully retrieved. |
| FAILURE | The datashare file or boolean name could not be found. |

Remarks

None.

Get Integer From DataShare File

Loads an integer argument (by name) from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Integer Name | The named argument for an integer in the selected DataShare file. |

Return Arguments

| | | | |
|---|---------|---------------|---|
| 2 | Integer | Integer Value | The value of the integer loaded from the command. |
|---|---------|---------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully loaded. |
| FAILURE | The file could not be loaded or the named integer argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Get Double From DataShare File

Loads a double argument (by name) from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Double Name | The named argument for a double in the selected DataShare file. |

Return Arguments

| | | | |
|---|--------|--------------|--|
| 2 | Double | Double Value | The value of the double loaded from the command. |
|---|--------|--------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The file and argument were successfully loaded. |
| FAILURE | The file could not be loaded or the named double argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Get String From DataShare File

Loads a string argument (by name) from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | String Name | The named argument for a string in the selected DataShare file. |

Return Arguments

| | | | |
|---|--------|--------------|--|
| 2 | String | String Value | The value of the string loaded from the command. |
|---|--------|--------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The file and argument were successfully loaded. |
| FAILURE | The file could not be loaded or the named string argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Get Vector From DataShare File

Loads a vector argument (by name) from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Vector Name | The named argument for a vector in the selected DataShare file. |

Return Arguments

| | | | |
|---|--------|--------------|--|
| 2 | Vector | Vector Value | The value of the vector loaded from the command. |
|---|--------|--------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The file and argument were successfully loaded. |
| FAILURE | The file could not be loaded or the named vector argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Get Transform From DataShare File

Loads a transform argument (by name) from a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Transform Name | The named argument for a transform in the selected DataShare file. |

Return Arguments

| | | | |
|---|-----------|-----------------|---|
| 2 | Transform | Transform Value | The value of the transform loaded from the command. |
|---|-----------|-----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully loaded. |
| FAILURE | The file could not be loaded or the named transform argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Set Boolean In DataShare File

Sets a boolean value in a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to load. |
| 1 | String | Boolean Name | The boolean name to set in datashare file. |
| 2 | Boolean | Boolean Value | Specify boolean condition. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The boolean value was successfully set. |
| FAILURE | The datashare file could not be found. |

Remarks

None.

Set Integer In DataShare File

Saves an integer value (by name) to an existing DataShare file.

Input Arguments

| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to update. |
|---|----------------------------|---------------------|--|
| 1 | String | Integer Name | The named argument for an integer in the specified DataShare file. |
| 2 | Integer | Integer Value | The integer value to save to the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully updated. |
| FAILURE | The file could not be loaded or the named argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Set Double In DataShare File

Saves a double value (by name) to an existing DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to update. |
| 1 | String | Double Name | The named argument for a double in the specified DataShare file. |
| 2 | Double | Double Value | The double value to save to the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully updated. |
| FAILURE | The file could not be loaded or the named argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Set String In DataShare File

Saves a string value (by name) to an existing DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to update. |
| 1 | String | String Name | The named argument for a string in the specified DataShare file. |
| 2 | String | String Value | The string value to save to the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully updated. |
| FAILURE | The file could not be loaded or the named argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Set Vector In DataShare File

Saves a vector value (by name) to an existing DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|--|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to update. |
| 1 | String | Vector Name | The named argument for a vector in the specified DataShare file. |
| 2 | Vector | Vector Value | The vector value to save to the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully updated. |
| FAILURE | The file could not be loaded or the named argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Set Transform In DataShare File

Saves a transform value (by name) to an existing DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---------------------|---|
| 0 | File Path or Embedded File | DataShare File Path | The path to the DataShare file to update. |
| 1 | String | Transform Name | The named argument for a transform in the specified DataShare file. |
| 2 | Transform | Transform Value | The transform value to save to the DataShare file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file and argument were successfully updated. |
| FAILURE | The file could not be loaded or the named argument was not found. |

Remarks

This command supports both absolute paths (ex. `C:\test.xit`) and relative paths (ex. `.\test.xit`).

Load HTML Form

Loads a set of arguments from a DataShare file, Displays those values in an HTML prompt, and then saves user entered values back into a DataShare file.

Input Arguments

| | | | |
|---|----------------------------|---|---|
| 0 | File Path or Embedded File | Input HTML Form Path | Location of the displayed HTML page |
| 1 | File Path or Embedded File | Input DataShare File Path | Location of the input values for the HTML |
| 2 | File Path or Embedded File | Output DataShare File Path | Location to write the HTML inputs |
| 3 | Boolean | Save in Binary Format? | Binary or Text output |
| 4 | Integer | Step to jump to if Canceled (-1 will fail Step if Cancel) | Step to jump to if canceled. |
| 5 | String | Save Button Text | Custom text used for the Save Button |
| 6 | String | Cancel Button Text | Custom text used for the Cancel Button |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was successfully loaded. |
| FAILURE | The file could not be loaded or had an invalid format. |

Remarks

Currently, the application supports the following data types extracted from HTML form:

- String (S), Integer (I), Double (D), Boolean (B)

In order to identify each value the HTML id attribute must be specified. The id attribute can be unique as needed. In order to identify type of entered value, the application uses first character of HTML id attribute.

For example, id="S123" or id="S87654" will be identified as Strings.

Number fields: number fields can be either integer or double and would be specified as follows:

```
<input type="number" id = "D1" name="Circle Diameter" value=0>
```

```
<input type="number" id = "I" name="Desired Measurement Count" value=0>
```

Radio buttons: It is important for radio buttons of the same group to have identical name attributes and data type. As in example below, id indicates String and name "Gender" for all <input> tags:

```
<fieldset> <legend>Gender</legend>
```

```
<input align="left" type="radio" id="S3" name="Gender" value="male" checked> Male
```

```
<input align="left" type="radio" id="S3" name="Gender" value="female"> Female
```

```
<input align="left" type="radio" id="S3" name="Gender" value="other"> Other
```

```
</fieldset>
```

File Browser: The selected filename path should have String identification as in example below:

```
<input type="file" size = "50" id="S4" name="CAD File Path" value="">
```

Note on File Browse: Before presenting the HTML form to user, the application restores previously saved values from a DataShare file. An input field of "file" type has read only "value" attribute for security purpose. The application can't populate (write) pre-stored filename to reopened HTML form. The re-opened form filename path initial value is always empty. User doesn't have to re-enter filename again on the re-opened form; unless he wants to change the path.

CheckBoxes: The HTML checkboxes represent Boolean values. See example below for proper type/name identification:

```
<input type="checkbox" id="B1" name="Prompt for Tooling Selection" value="">Prompt for Tooling Selection<br>
```

Dropdown List: The type/name identification should be added to <select> tag. See below:

```
<select id="S5" name="Tracker">
    <option value="Faro">Faro Vantage Laser Tracker</option>
    <option value="Leica">Lieca AT960 Laser Tracker</option>
    <option value="API">API Radian Laser Tracker</option>
</select>
```

jQuery and JavaScript: Adding: <meta http-equiv="x-ua-compatible" content="IE=edge"> into the header of an html file sent to the MP command "Load HTML Form" allows the use of jQuery commands.

jQuery and other libraries need to be linked to the form as well, for example in the header:

```
<script type="text/javascript" src="javascripts/jquery-3.3.1.min.js"></script>
```

if the jQuery JavaScript is saved locally.

Note on Buttons: Buttons created in your HTML can serve the purpose of submitting or canceling the form. This is especially helpful if the button is tied to a JavaScript function that writes values related to the selected button to a datashare file accessible to the MP following an "onclick" event. It prevents the operator from having to select the button and submitting the form, and allows the Save and Cancel buttons to be hidden in the HTML window (argument 7).

In the style section of your HTML header, create the two classes that are recognized by the MP:

```
<head>
<style>
.classSaveButton {}
.classCancelButton {}
</style>
</head>
```

Then add the classes to the button(s) that will act as submit or cancel. The buttons can support multiple classes, so

simply add the new class to the button element with a space in between, i.e.

```
<input type="button" class="classSaveButton" value="Import Control Points" onclick="nextFunction('this.value)' />
```

An example MP with this Load HTML form can be found on the website here:

<http://www.kinematics.com/ftp/SA/Install/Examples/Instructional/>

Database Operations

Put to ODBC Database

Sends data to any ODBC-capable database such as MySQL, Microsoft Access, and Oracle databases.

Input Arguments

| | | | |
|------|-------------------|-------------------|--|
| 0 | Connection String | Connection String | The connection string to the database. |
| 1 | Table Name | Table Name | The name of the table to update. |
| 2->n | USER | USER | ADDITIONAL CUSTOM ARGUMENTS |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The data was successfully saved to the database. |
| FAILURE | The database or table could not be accessed. |

Remarks

Unless you know how to format the connection string, the easiest thing to do is to change the entry method to *Browse* and then click the down-arrow button in the *Value* field. This opens the Windows ODBC data source picker, which guides you through selecting the proper database.

Unless you know how to format the table name string, the easiest thing to do is to change the entry method to *Browse* (once the connection string has been specified) so that you can enter the value using a graphical user interface.

To add, remove, or edit any number of arguments, use the buttons listed under the comment area of the MP editor. When adding arguments, the Description should match the column name, and the data type should be compatible with the data type in the database (values will be converted to the proper type, if possible). To enter the values using a graphical interface, use the [Add Using Column Picker](#) button, but be aware that a valid connection string must be specified first before using this option.

Get from ODBC Database

Retrieves data from any ODBC-capable database such as MySQL, Microsoft Access, and Oracle databases.

Input Arguments

| | | | |
|------|-------------------|-------------------|--|
| 0 | Connection String | Connection String | The connection string to the database. |
| 1 | Table Name | Table Name | The name of the table to access. |
| 2 | String | WHERE | A string containing the SQL selection criteria for the data. |
| 3->n | USER | USER | ADDITIONAL USER-DEFINED ARGUMENTS |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The data was successfully retrieved from the database. |
| FAILURE | The database or table could not be accessed. |

Remarks

Unless you know how to format the connection string, the easiest thing to do is to change the entry method to *Browse* and then click the down-arrow button in the *Value* field. This opens the Windows ODBC data source picker, which guides you through selecting the proper database.

Unless you know how to format the table name string, the easiest thing to do is to change the entry method to *Browse* (once the connection string has been specified) so that you can enter the value using a graphical user interface.

If the WHERE string results in the selection of more than one record from the database, only the first selected record is considered. For more details on valid SQL strings, search the internet or look into one of many texts on SQL databases.

To add, remove, or edit the arguments that will be populated from the table, use the buttons listed under the comment area of the MP editor. When adding arguments, the Description should match the column name, and the data type should be compatible with the data type in the database (values will be converted to the proper type, if possible). To enter the values using a graphical interface, use the [Add Using Column Picker](#) button, but be aware that a valid connection string must be specified first before using this option.

Delete from ODBC Database

Deletes data from any ODBC-capable database such as MySQL, Microsoft Access, and Oracle databases.

Input Arguments

| | | | |
|---|-------------------|-------------------|--|
| 0 | Connection String | Connection String | The connection string to the database. |
| 1 | Table Name | Table Name | The name of the table to access. |
| 2 | String | WHERE | A string containing the SQL selection criteria for the data to delete. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The data was successfully deleted from the database. |
| FAILURE | The database or table could not be accessed. |

Remarks

Unless you know how to format the connection string, the easiest thing to do is to change the entry method to *Browse* and then click the down-arrow button in the *Value* field. This opens the Windows ODBC data source picker, which guides you through selecting the proper database.

Unless you know how to format the table name string, the easiest thing to do is to change the entry method to *Browse* (once the connection string has been specified) so that you can enter the value using a graphical user interface.

For more details on valid SQL strings, search the internet or look into one of many texts on SQL databases.

XML

Open XML File

Opens an XML file for reading or writing.

Input Arguments

| | | | |
|---|----------------------------|---------------|---------------------------|
| 0 | File Path or Embedded File | XML File Path | The path to the XML file. |
|---|----------------------------|---------------|---------------------------|

Return Arguments

| | | | |
|---|---------|-----------------|------------------------------------|
| 1 | Integer | XML File Handle | The handle to the opened XML file. |
|---|---------|-----------------|------------------------------------|

Returned Status

| | |
|---------|-----------------------------------|
| SUCCESS | The file was opened successfully. |
| FAILURE | The file could not be opened. |

Remarks

None.

Get XML Attribute

Retrieves one or more attribute values from the specified XML element.

Input Arguments

| | | | |
|---|---------|-----------------|--|
| 0 | Integer | XML File Handle | The handle to the XML file as returned from the Open XML File command. |
| 1 | String | XPath | The XPath to the node to read from. |

Return Arguments

Return arguments are user-defined.

Returned Status

| | |
|---------|---|
| SUCCESS | The attributes were retrieved successfully. |
| FAILURE | One or more attributes or a valid node was not found. |

Remarks

You can retrieve one or more attribute values from a specific node by specifying the XPath to that node. (*XPath is widely-adopted standardized query language for selecting nodes from an XML document*).

Once the node of interest is specified by the XPath, the actual attribute values are retrieved by adding return arguments in the command. The *Type* of the argument specifies the data type the value will be returned as, whereas the *Descriptor* for the argument must match the XML attribute name perfectly. For example, for an XML file excerpt that looks like this,

```
<ROOT>
  <BOOK TITLE="Mastering Metrology" AUTHOR="S. M. Arr" PAGES="1023">
    <BOOKMARK PAGE="144" />
  </BOOK>
  <BOOK TITLE="A Christmas Carol" AUTHOR="Charles Dickens" PAGES="95">
    <BOOKMARK PAGE="3" />
    <BOOKMARK PAGE="15" />
  </BOOK>
</ROOT>
```

you can retrieve the author and length of *A Christmas Carol* by specifying the following XPath string:

```
/ROOT/BOOK[@TITLE="A Christmas Carol"]
```

and adding two return arguments: one of type *String* with the descriptor *AUTHOR*, and the other of type *Integer* with the descriptor *PAGES*.

In layman's terms the command will search through the xml file as specified in the XPath. It will start with *<Root>* then look for *<Book>* and then when it finds a matching entry it will look for where *TITLE="A Christmas Carol"* and

then all the descriptors under /Root/Book become available (*TITLE*, *AUTHOR*, *PAGES*). To go one level deeper you could specify:

```
/ROOT/BOOK[@TITLE="A Christmas Carol"]/BOOKMARK[2]
```

and then return a descriptor *PAGE*. The command will find the first bookmark in the section and return "15". Therefore, by using a counter you can index through a list even if a specific descriptor is not known.

**A nice trick is to install Notepad ++ with the XML plug-in or similar resource. This allows you to open an XML page, select an attribute and have it generate the path for you which can then be pasted into the MP command.

Set XML Attribute

Writes one or more attribute values to the specified XML element.

Input Arguments

| | | | |
|-----|--------------|-----------------|--|
| 0 | Integer | XML File Handle | The handle to the XML file as returned from the Open XML File command. |
| 1 | String | XPath | The XPath to the node to write to. |
| 2-n | User-defined | User-defined | Attribute values to write to the specified node. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The attributes were written successfully. |
| FAILURE | The specified node was not found. |

Remarks

See [Get XML Attribute](#) for indexing information.

Close XML File

Closes an open XML file, optionally saving it in the process.

Input Arguments

| | | | |
|---|---------|-----------------|--|
| 0 | Integer | XML File Handle | The handle to the XML file as returned from the Open XML File command. |
| 1 | Boolean | Save? | TRUE if the file should be saved; FALSE otherwise. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The file was closed successfully. |
| FAILURE | The file handle was invalid, or could not be saved. |

Remarks

None.

Import Nominals from XML File

Imports nominal points from an XML file. Equivalent to *File>Import>XML*.

Input Arguments

| | | | |
|---|----------------------------|-----------|---------------------------|
| 0 | File Path or Embedded File | File Path | The path to the XML file. |
|---|----------------------------|-----------|---------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The import was successful. |
| FAILURE | The file was not found, or had an invalid format. |

Remarks

Refer to the SA documentation for the proper format of the XML file and the behavior of this command.

Merge Measurements into XML File

Merges measured points into an existing XML file. Equivalent to the [Merge Measurements and Save](#) button in the *File>Import>XML* command.

Input Arguments

| | | | |
|---|----------------------------|------------|--|
| 0 | File Path or Embedded File | File Path | The path to the XML file. |
| 1 | Collection Object Name | Group Name | The name of the point group to push to the XML file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The merge was successful. |
| FAILURE | The file was not found, the group was not found, or the file had an invalid format. |

Remarks

Refer to the SA documentation for the proper format of the XML file and the behavior of this command.

This Page Intentionally Left Blank.

3

PROCESS FLOW OPERATIONS

Wait for Steps to Complete

Waits for a set of one or more steps to complete before continuing execution.

Input Arguments

| | | | |
|---|------------------|---------------------------|---|
| 0 | Step ID Ref List | Steps to Await Completion | A set of one or more steps that must be marked as complete before this command will continue. |
|---|------------------|---------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Several MP commands allow concurrent operation while the MP continues executing. The “Wait for Completion” argument of the Configure and Measure command is one example.

Jump To Step

Jumps to a specified step in the current script.

Input Arguments

| | | | |
|---|---------|-----------------|----------------------|
| 0 | Step ID | Step to Jump To | The step to jump to. |
|---|---------|-----------------|----------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Jump To Other Measurement Plan

Jumps to a specified step in a different Measurement Plan script.

Input Arguments

| | | | |
|---|----------------------------|------------------------|---|
| 0 | File Path or Embedded File | MP filename to Jump To | The path to the other MP file to jump to. |
| 1 | Step ID | Step index to begin at | The step to jump to in the other MP file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The jump was successful. |
| FAILURE | The specified MP or step was not found. |

Remarks

None.

Step Status Test

Checks the status of a specific step (success, partial success, or failure) and jumps to a step depending on the result.

Input Arguments

| | | | |
|---|---------|-------------------------|--|
| 0 | Step ID | Step in question | The step whose status should be checked. |
| 1 | Step ID | Step if success | The step to jump to if the step in question has succeeded. |
| 2 | Step ID | Step if partial success | The step to jump to if the step in question has partially succeeded. |
| 3 | Step ID | Step if failure | The step to jump to if the step in question has failed. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This is the primary command that enables error checking in MPs.

Jump Based on Ranged Status Test

Jumps to a step based on the most erroneous status from a range of steps.

Input Arguments

| | | | |
|---|---------|---------------------|---|
| 0 | Step ID | First Step in Range | The first step in the range of steps to check. |
| 1 | Step ID | Last Step in Range | The last step in the range of steps to check. |
| 2 | Step ID | Step if Go | The step to jump to if all steps have succeeded. |
| 3 | Step ID | Step if Partial Go | The step to jump to if the range of steps had one or more statuses of partial success, but no failures. |
| 4 | Step ID | Step if No Go | The step to jump to if at least one step in the specified range has failed. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Go/No Go - Range Check Results

Displays a custom message based on the status of a range of steps.

Input Arguments

| | | | |
|---|---------|---------------------|--|
| 0 | Step ID | First Step in Range | The first step in the range of steps to check. |
| 1 | Step ID | Last Step in Range | The last step in the range of steps to check. |
| 2 | Boolean | Minor Error is OK? | Indicates whether minor errors (partial successes) are considered "GO" or not. |
| 3 | String | Message for Go | The message to display if all commands are considered successful. |
| 4 | String | Message for No Go | The message to display if one or more commands are considered failed. |

Return Arguments

| | | | |
|---|---------|--------|---|
| 5 | Boolean | Result | Indicates a "Go" or "No Go" based on the result of the check. |
|---|---------|--------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Object Existence Test

Tests whether an object exists in the current file.

Input Arguments

| | | | |
|---|------------------------|------------------------------|---|
| 0 | Collection Object Name | Object Name | The name of the object to check for. |
| 1 | Step ID | Step if Object does exist | The step to jump to if the specified object exists. |
| 2 | Step ID | Step if Object doesn't exist | The step to jump to if the specified object does not exist. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The specified object exists. |
| FAILURE | The specified object does not exist. |

Remarks

None.

Object Existence Test (Check Only)

Tests whether an object exists in the current file.

Input Arguments

| | | | |
|---|------------------------|-------------|--------------------------------------|
| 0 | Collection Object Name | Object Name | The name of the object to check for. |
|---|------------------------|-------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|---------|---------------------------------|
| 1 | Boolean | Exists? | The object entered by the user. |
|---|---------|---------|---------------------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The specified object exists. |
| FAILURE | The specified object does not exist. |

Remarks

None.

Collection Existence Test

Tests whether a collection exists in the current file.

Input Arguments

| 0 | Collection Name | Collection Name to check | The name of the collection to check for. |
|---|-----------------|----------------------------------|---|
| 1 | Step ID | Step if Collection does exist | The step to jump to if the specified collection exists. |
| 2 | Step ID | Step if Collection doesn't exist | The step to jump to if the specified collection does not exist. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The specified collection exists. |
| Partial Success | The step executed correctly but the collection was not found |
| FAILURE | The command did not complete correctly. |

Remarks

None.

Create Counter

Creates an integer counter that keeps track of a number and can be incremented/decremented.

Input Arguments

| | | | |
|---|---------|---------------|------------------------------------|
| 0 | Counter | Counter Value | The initial value for the counter. |
|---|---------|---------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Counters are used most often in loops to keep track of the current iteration through the loop.

If a “jump to step” command causes this command to be re-executed, the counter will be automatically reset to the initial value.

Increment Counter

Increases a counter's value by 1.

Input Arguments

| | | | |
|---|-------------------|-------------------|-------------------------------------|
| 0 | Counter Reference | Counter Reference | A reference to an existing counter. |
|---|-------------------|-------------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decrement Counter

Decreases a counter's value by 1.

Input Arguments

| | | | |
|---|-------------------|-------------------|-------------------------------------|
| 0 | Counter Reference | Counter Reference | A reference to an existing counter. |
|---|-------------------|-------------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Reset Counter

Resets a counter to the specified value.

Input Arguments

| | | | |
|---|-------------------|-------------------|-------------------------------------|
| 0 | Counter Reference | Counter Reference | A reference to an existing counter. |
| 1 | Integer | Counter Value | The value to reset the counter to. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Ask for String

Prompts the user to enter a string value.

Input Arguments

| | | | |
|---|-----------|---|---|
| 0 | String | Question to ask | A question or prompt to display to the user. |
| 1 | String | Initial Answer | An optional default value to enter into the answer field. |
| 2 | Font Type | Font | The font to use. |
| 3 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

| | | | |
|---|--------|--------|---------------------------------|
| 4 | String | Answer | The string entered by the user. |
|---|--------|--------|---------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The answer was accepted successfully. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 3 was -1. |

Remarks

None.

Ask for String (Pull-Down Version)

Prompts the user to select from a dropdown list of possible answers, returning a string in the process.

Input Arguments

| | | | |
|---|-----------------|---|--|
| 0 | String Ref List | Question or Statement | A list of one or more strings to concatenate together as a prompt to the user. |
| 1 | String Ref List | Possible Answers | A list of possible answers for the dropdown list. |
| 2 | Font Type | Font | The font to use. |
| 3 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

| | | | |
|---|--------|--------|----------------------------------|
| 4 | String | Answer | The string selected by the user. |
|---|--------|--------|----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The answer was accepted successfully. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 3 was -1. |

Remarks

None.

Ask for Point Name

Prompts the user to enter a point name, returning a point name.

Input Arguments

| | | | |
|---|------------|---|---|
| 0 | String | Question to ask | A prompt to display to the user. |
| 1 | Point Name | Initial Value | A default value to optionally display in the answer field. |
| 2 | Font Type | Font | The font to use. |
| 3 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog box. |

Return Arguments

| | | | |
|---|------------|--------|-------------------------------------|
| 4 | Point Name | Answer | The point name entered by the user. |
|---|------------|--------|-------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The point name was accepted successfully. |
| FAILURE | The user cancelled the dialog and argument 3 was -1, or entered the point name in an incorrect format. |

Remarks

None.

Ask for Integer

Prompts the user to enter an integer.

Input Arguments

| | | | |
|---|-----------|---|---|
| 0 | String | Question to ask | A prompt to display to the user. |
| 1 | Integer | Initial Value | A default value to optionally display in the answer field. |
| 2 | Boolean | Enforce Min/Max Values? | Indicates whether the supplied min/max range should be enforced on the input. |
| 3 | Integer | Min Value | The minimum allowable value (if argument 2 is TRUE). |
| 4 | Integer | Max Value | The maximum allowable value (if argument 2 is TRUE). |
| 5 | Font Type | Font | The font to use. |
| 6 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

| | | | |
|---|---------|--------|----------------------------------|
| 7 | Integer | Answer | The integer entered by the user. |
|---|---------|--------|----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The integer was accepted successfully. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 6 was -1. |

Remarks

Data is validated automatically. If the user enters a non-integer value into the field, a warning will be displayed and the user will be prompted to re-enter a value.

If argument 2 is TRUE and the user enters a value outside the supplied min/max range, a warning will be displayed and the user will be prompted to re-enter a value.

Ask for Double

Prompts the user to enter a floating point double value.

Input Arguments

| | | | |
|---|-----------|---|--|
| 0 | String | Question to ask | A prompt to display to the user. |
| 1 | Double | Initial Value | A default value to optionally display in the answer field. |
| 2 | Boolean | Enforce Min/Max Values? | Indicates whether the provided minimum/maximum values should be enforced on the input. |
| 3 | Double | Min Value | The minimum allowable value (if argument 2 is TRUE). |
| 4 | Double | Max Value | The maximum allowable value (if argument 2 is TRUE). |
| 5 | Font Type | Font | The font to use. |
| 6 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

| | | | |
|---|--------|--------|---------------------------------|
| 7 | Double | Answer | The double entered by the user. |
|---|--------|--------|---------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The double was accepted successfully. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 6 was -1, or entered a non-double value into the field. |

Remarks

Data is validated automatically. If the user enters a non-numeric value into the field, a warning will be displayed and the user will be prompted to re-enter a value.

If argument 2 is TRUE and the user enters a value outside the supplied min/max range, a warning will be displayed and the user will be prompted to re-enter a value.

Ask for User Decision Extended

Prompts the user to select one of three custom buttons.

Input Arguments

| | | | |
|---|---------------|---|--|
| 0 | Edit Text | Question or Statement | A prompt to display to the user. |
| 1 | Font Type | Font | The font to use for the prompt and button text. |
| 2 | Decision List | Button Answers | A list of button titles and the associated step each should jump to. |
| 3 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The user clicked a button. |
| FAILURE | The user clicked the Cancel or Close button and argument 3 was -1. |

Remarks

None.

Ask for User Decision(HTML)

Prompts the user to select one of three custom buttons, displaying an HTML file as a prompt.

Input Arguments

| | | | |
|---|----------------------------|---|---|
| 0 | File Path or Embedded File | Path to HTML File | The path to the HTML file to use as the prompt. |
| 1 | Font Type | Font | The font to use for the prompt and button text. |
| 2 | String | Button1 Text | The text to display on the first button. |
| 3 | String | Button2 Text | The text to display on the second button. |
| 4 | String | Button3 Text | The text to display on the third button. |
| 5 | Step ID | Step to jump to for Button1 (-1 to hide button) | The step to jump to when the first button is clicked. |
| 6 | Step ID | Step to jump to for Button2 (-1 to hide button) | The step to jump to when the second button is clicked. |
| 7 | Step ID | Step to jump to for Button3 (-1 to hide button) | The step to jump to when the third button is clicked. |
| 8 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The user clicked a button. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 8 was -1. |

Remarks

If a step ID (arguments 5-7) is set to -1, that associated button will not appear in the dialog.

Ask for User Decision (Pull-Down Version)

Prompts the user to make a selection from a dropdown list. Each selection in the dropdown list has an associated step file to which the MP jumps after this command is successful.

Input Arguments

| | | | |
|---|---------------|---|--|
| 0 | Edit Text | Question or Statement | The prompt to display to the user. |
| 1 | Font Type | Font | The font to use for the prompt. |
| 2 | Decision List | Possible Answers | A list of possible answers and their associated "jump to" steps. |
| 3 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The user made a selection. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 3 was -1. |

Remarks

None.

Ask for User Decision from Strings

Prompts the user to make a selection from up to three buttons. The text on the button becomes the string return argument for the command.

Input Arguments

| | | | |
|---|-----------|---|---|
| 0 | Edit Text | Question or Statement | The prompt to display to the user. |
| 1 | Font Type | Font | The font to use for the prompt and button text. |
| 2 | String | Button1 Text (Empty to hide button) | The text for the first button. |
| 3 | String | Button2 Text (Empty to hide button) | The text for the second button. |
| 4 | String | Button3 Text (Empty to hide button) | The text for the third button. |
| 5 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user cancels or closes the dialog. |

Return Arguments

| | | | |
|---|--------|--------|------------------------------------|
| 6 | String | Answer | The text from the selected button. |
|---|--------|--------|------------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The user made a selection. |
| FAILURE | The user clicked the Cancel or Close buttons and argument 5 was -1. |

Remarks

None.

Ask for User Decision from Image

Displays a clickable image to the user which can have user-defined click regions for making a decision.

Input Arguments

| | | | |
|---|----------------------------|---|---|
| 0 | File Path or Embedded File | Image File | The file containing the image to display. |
| 1 | File Path or Embedded File | Image Map XML File | The path to the XML file containing the click regions for the image (see below). |
| 2 | String | Window Caption | The caption for the window displaying the image. |
| 3 | Integer | Window Width (0 = default) | The width for the window (in pixels). Use 0 to use the default arbitrary width. |
| 4 | Integer | Window Height (0 = default) | The height for the window (in pixels). Use 0 to use the default arbitrary height. |
| 5 | Step ID | Step to jump to if Canceled (-1 will fail step on Cancel) | The step to jump to if the user closes or cancels the dialog. |

Return Arguments

| | | | |
|---|--------|-------------|--|
| 6 | String | User Choice | The resulting decision made by the user. |
|---|--------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The user made a selection. |
| FAILURE | The image or XML file could not be found, or the user cancelled the dialog (and argument 5 was -1). |

Remarks

The XML map file defines regions in the image—either 2-point rectangles (top left/bottom right) or polygons—with names. The image is displayed for the user and if they click in a region which is defined in the map, then the window closes and the MP result is the name of the clicked region. If the user cancels, the returned string will be empty.

The format of the XML file is outlined below. Inside the `<ImageMap>` element are `<Region>` blocks. The region has one attribute: `name`, which describes the string returned when that region is clicked. The `<Region>` block must contain at least two `<Point>` elements. Each `<Point>` element has two attributes: `x` and `y`, indicating the position in pixels (from the top left of the image) for that vertex of the image map. Any region that has exactly two `<Point>` elements is assumed to be a rectangular region. Any region with more than two `<Point>` elements is assumed to be a closed polygon. The opening/closing vertex need not be repeated.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ImageMap>
  <Region name="1 Point Perpendicular">
    <Point x="332" y="140" />
    <Point x="674" y="425" />
  </Region>
  <Region name="Circle Parallel">
```

```
<Point x="685" y="436" />  
<Point x="822" y="400" />  
<Point x="971" y="693" />  
</Region>  
</ImageMap>
```

This Page Intentionally Left Blank.

4 MP TASK OVERVIEW

Create/Clear Task Overview List

Creates or clears a task overview list for an MP. A task overview list must be created and configured before being displayed onscreen.

Input Arguments

| | | | |
|---|-----------|-------------------|---------------------------------------|
| 0 | Font Type | Task Name Font | The font to use for the task name. |
| 1 | Font Type | Task Comment Font | The font to use for the task comment. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Overview Title

Sets the title for the task overview.

Input Arguments

| | | | |
|---|--------|----------------|----------------------------------|
| 0 | String | Overview Title | The title for the task overview. |
|---|--------|----------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Overview Image

Sets the main image for the task overview.

Input Arguments

| | | | |
|---|----------------------------|------------|-------------------------------|
| 0 | File Path or Embedded File | Image Path | The path to the image to use. |
|---|----------------------------|------------|-------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The image was loaded successfully. |
| FAILURE | The file was not found or was not in a compatible format. |

Remarks

If no image is used, the SA logo will be displayed instead.

Add Task Overview Item

Adds a new item to a task overview list.

Input Arguments

| | | | |
|---|--------|--------------|---|
| 0 | String | Task Name | The name for the item. |
| 1 | String | Comment Text | The subtext for the item. |
| 2 | Double | Effort Index | A number indicating the relative difficulty or duration of the item. Used to determine overall progress for the task overview progress bar. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Task Item Status

Sets the completion status for a specific task item.

Input Arguments

| | | | |
|---|-------------------------|------------|---|
| 0 | Integer | Task Index | The zero-based index of the item of interest in the task overview list. |
| 1 | Measurement Plan Result | Status | The status of the index in question. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero. When a task item status is changed, the following task item becomes the “in work” task item.

Set Task Item Name

Sets the name for a specific task item.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Task Item Index | The zero-based index of the item of interest in the task overview list. |
| 1 | String | Task Name | The new name for the task item of interest. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero.

Set Task Item Comment

Sets the comment for a specific task item.

Input Arguments

| | | | |
|---|---------|--------------|---|
| 0 | Integer | Task Index | The zero-based index of the item of interest in the task overview list. |
| 1 | String | Task Comment | The new comment for the task item of interest. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero.

Show Progress for Task Item

Shows or hides the progress bar for a specific task item.

Input Arguments

| | | | |
|---|---------|----------------|---|
| 0 | Integer | Task Index | The zero-based index of the item of interest in the task overview list. |
| 1 | Boolean | Show Progress? | Indicates whether a progress bar should be displayed for the specified task item. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero.

Set Task Item Completion Values

Updates the completion state for a specific task item, which influences the percent complete for its individual progress bar.

Input Arguments

| | | | |
|---|---------|----------------------|---|
| 0 | Integer | Task Index | The zero-based index of the item of interest in the task overview list. |
| 1 | Integer | Increments Completed | The number of increments completed. |
| 2 | Integer | Total Increments | The total number of increments for the task item. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero.

“Increments” are abstract integer values--they can represent anything. The percent complete for the individual task item is computed by the ratio of “Increments Completed” to “Total Increments”.

Set Current Task

Sets the specified task item as the current task. This causes an animated gear icon to be displayed next to the item.

Input Arguments

| | | | |
|---|---------|------------|---|
| 0 | Integer | Task Index | The zero-based index of the item of interest in the task overview list. |
|---|---------|------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task indices are zero-based. The first item added to a task list is index zero.

Show Task Overview List

Shows or hides a task overview list.

Input Arguments

| | | | |
|---|---------|-------|---|
| 0 | Boolean | Show? | Indicates whether the task overview list should be displayed or hidden. |
|---|---------|-------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Task overview lists should be constructed and configured before being displayed onscreen.

5 VIEW CONTROL

Ribbon Bar

Load Ribbon Bar from XML File

Provides a means to load a ribbon bar configuration from an existing file.

Input Arguments

| | | | |
|---|----------------------------|-----------|---|
| 0 | File Path or Embedded File | File Path | File Path and Name of the xml file to load. |
|---|----------------------------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | This XLM file was loaded successfully |
| FAILURE | The xml file could not be found or is the wrong format. |

Remarks

None.

Reset Ribbon Bar to Default

Provides a means to restore the default ribbon bar configuration.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Point of View

Set Point of View From Frame

Sets the point of view to match the orientation of a given coordinate frame.

Input Arguments

| | | | |
|---|------------------------|-------|--|
| 0 | Collection Object Name | Frame | The frame to use for setting the view. |
|---|------------------------|-------|--|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The view was set successfully. |
| FAILURE | The specified frame was not found. |

Remarks

None.

Set Point of View From Instrument Updates

Sets the point of view dynamically based upon an instruments probing direction.

Input Arguments

| | | | |
|---|--------------------------|---|---|
| 0 | Collection Instrument ID | Instrument's ID | Instrument used for view updates |
| 1 | Boolean | Display View Control | True displays the Control dialog |
| 2 | Boolean | Enable Set View from Instrument Updates | True enables view control from the instrument's updates |
| 3 | Double | Update View Percent | The percent of view within which view updates are ignored. This defines a working zone. |
| 4 | Boolean | Clip Behind Probe | True enables clipping behind the probe |
| 5 | Boolean | Automatic Zoom When Trapping | True enables automatic zoom. The window will size to fit the probe tip and the nominal feature. |
| 6 | Boolean | Enable Directional Cloud Points | True enables directional cloud display, clipping data measured from the far side of a part. |
| 7 | Double | Angle Reset Threshold | The number of degrees the probing vector must rotate before a view update is triggered |
| 8 | Integer | Animation Steps | The number of animation steps between view positions. |
| 9 | Collection Object Name | Reference Frame Object | The reference used to orient the graphics. Typically the Z axis of world is up. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The view was set successfully. |
| FAILURE | The specified frame was not found. |

Remarks

None.

Set Point of View

Sets the view orientation to match a saved (named) view.

Input Arguments

| | | | |
|---|-----------|-----------|-------------------------------|
| 0 | View Name | View Name | The name of the view to load. |
|---|-----------|-----------|-------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The view was set successfully. |
| FAILURE | The named view was not found in the current file. |

Remarks

Note that viewpoints are saved with the SA file. Therefore, if the view is not available in the file that is open when this command is executed, it will fail.

Save Point of View

Saves the current view orientation as a named view.

Input Arguments

| 0 | View Name | View Name | The name of the view to save. |
|---|-----------|------------------------|---|
| 1 | Boolean | Restore Zoom Settings? | Indicate whether to save the zoom settings so that they will later be restored with the view. |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The view was saved successfully. |
|---------|----------------------------------|

Remarks

Note that viewpoints are saved with the SA file.

Define Point of View

Defines a view orientation with a specified view angle, zoom setting, view position, and render mode.

Input Arguments

| | | | |
|---|------------------|------------------------|--|
| 0 | View Name | View Name | The name of the view to define. |
| 1 | Double | Rotation (x) | The rotation of the viewpoint about the active frame's X axis. |
| 2 | Double | Rotation (y) | The rotation of the viewpoint about the active frame's Y axis. |
| 3 | Double | Rotation (z) | The rotation of the viewpoint about the active frame's Z axis. |
| 4 | Boolean | Restore Zoom Settings? | Set to TRUE to save zoom settings with the view. |
| 5 | Double | Scale Factor | The zoom setting. Higher values are zoomed in more. |
| 6 | Integer | Origin (x) | The horizontal point at which the view is centered, in screen coordinates, and defined from the center of the view. For example, higher X values shift the viewpoint to the right. |
| 7 | Integer | Origin (y) | The vertical point at which the view is centered, in screen coordinates, and defined from the center of the view. For example, higher Y value shift the viewpoint up. |
| 8 | Boolean | Restore Render Mode? | Set to TRUE to recall the rendering mode with the view. |
| 9 | Render Mode Type | Rendering Mode | The render mode to use when a rendering mode is saved with the view. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Note that viewpoints are saved with the SA file.

Get Point of View Parameters

Retrieves the parameters (viewpoint, rotation, zoom settings, etc.) for a named view.

Input Arguments

| | | | |
|---|-----------|-----------|----------------------------------|
| 0 | View Name | View Name | The name of the view to examine. |
|---|-----------|-----------|----------------------------------|

Return Arguments

| | | | |
|---|------------------|------------------------|--|
| 1 | Double | Rotation (x) | The rotation of the viewpoint about the active frame's X axis. |
| 2 | Double | Rotation (y) | The rotation of the viewpoint about the active frame's Y axis. |
| 3 | Double | Rotation (z) | The rotation of the viewpoint about the active frame's Z axis. |
| 4 | Boolean | Restore Zoom Settings? | TRUE if zoom settings are saved with the view. |
| 5 | Double | Scale Factor | The zoom setting. Higher values are zoomed in more. |
| 6 | Double | Origin (x) | The horizontal point at which the view is centered, in screen coordinates, and defined from the center of the view. For example, higher X values shift the viewpoint to the right. |
| 7 | Double | Origin (y) | The vertical point at which the view is centered, in screen coordinates, and defined from the center of the view. For example, higher Y value shift the viewpoint up. |
| 8 | Boolean | Restore Render Mode? | TRUE if the named view is set to recall the rendering mode with the view. |
| 9 | Render Mode Type | Rendering Mode | The render mode used when a rendering mode is saved with the view. |

Returned Status

| | |
|---------|--|
| SUCCESS | The view parameters were retrieved successfully. |
| FAILURE | The named view was not found. |

Remarks

Note that viewpoints are saved with the SA file.

The Origin parameters (A6 & A7) were converted to doubled in after 2017.08.11 to increase placement accuracy.

Hide/Show Operations

Set Toolkit Visibility

Hides or shows SA Toolkit.

Input Arguments

| | | | |
|---|--------------|-----------------|---|
| 0 | Boolean | Show Toolkit? | Whether the toolkit will be visible or not. |
| 1 | Toolkit Page | Page to Display | The tab of the toolkit to be shown. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show Labels

Sets the visibility of point labels. Equivalent to turning on or off Point Labels in the [View](#) menu.

Input Arguments

| | | | |
|---|---------|-------------------|--|
| 0 | Boolean | Point Labels On? | Specify whether labels should be visible. |
| 1 | Boolean | Object Labels On? | Specifies whether object labels should be visible. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show Objects

Shows specified objects so that they are no longer hidden in the graphical view.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | Collection Object Name Ref List | Objects to Show | The list of objects to show. |
| 1 | Collection Object Name Ref List | Object Name List | A list of objects to export to the file. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | All objects in the list were shown successfully. |
| PARTIAL SUCCESS | At least one object was shown successfully, but at least one object was not found. |
| FAILURE | None of the specified objects were found. |

Remarks

None.

Show by Object Type

Shows a set of objects so that they are no longer hidden in the graphical view. The objects that are shown are those that match the type of the specified object.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Object Type to Show | Specify the object defining the type of object to show. |
| 1 | Boolean | All Collections? | Show objects in all collections (TRUE) or just the active collection (FALSE)? |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The specified object was found. Any objects matching the type of that object were shown. |
| FAILURE | The specified object was not found. |

Remarks

None.

Hide Objects

Hides specified objects so that they no longer appear in the graphical view.

Input Arguments

| | | | |
|---|---------------------------------|-----------------|------------------------------|
| 0 | Collection Object Name Ref List | Objects to Hide | The list of objects to hide. |
|---|---------------------------------|-----------------|------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | All objects in the list were hidden successfully. |
| PARTIAL SUCCESS | At least one object was hidden successfully, but at least one object was not found. |
| FAILURE | None of the specified objects were found. |

Remarks

Point Cloud selection can be refined as follows:

Cloud = Cloud, Scan Stripe Cloud, or Cross Section Cloud

Scan Stripe Cloud = Scan Stripe Cloud or Cross Section Cloud (but not basic clouds)

Cross Section Cloud = allows Cross Section Cloud selection only

Show/Hide by Object Type

Shows or hides objects based on their type (Point Groups, Planes, etc.). Can be applied to object types in a specific collection or across all collections.

Input Arguments

| | | | |
|---|-----------------|--------------------------|--|
| 0 | Boolean | All Collections? | Specify whether all objects in all collections matching the specified type should be shown/hidden. |
| 1 | Collection Name | Specific Collection | Collection whose objects will be shown/hidden. Only applies if Argument 0 is FALSE. |
| 2 | Object Type | Object Type To Show/Hide | Specify an object type to show or hide. |
| 3 | Boolean | Hide? (Show = FALSE) | Specify whether to show (FALSE) or hide (TRUE) the specified objects. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All objects matching the requested type were shown/hidden successfully. |
| FAILURE | The specified collection (if applicable) was not found. |

Remarks

None.

Show/Hide Points

Shows or hides specified points.

Input Arguments

| 0 | Point Name Ref List | Point Names | The list of points to show or hide. |
|---|---------------------|----------------------|--|
| 1 | Boolean | Show? (Hide = FALSE) | Indicates whether the points should be shown (TRUE) or hidden (FALSE). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If a point group is hidden, it will be shown if any point inside it is shown.

Show/Hide Dimensions

Shows or hides specified dimensions.

Input Arguments

| | | | |
|---|-----------------------------|------------------|---|
| 0 | Collection Object Name List | Dimension Names | The list of dimensions to show or hide. |
| 1 | Boolean | Show Dimensions? | Choosing True will Show the selected dimensions |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show/Hide Callout View

Shows or hides an existing callout view.

Input Arguments

| | | | |
|---|------------------------------|----------------------|--|
| 0 | Collection Callout View Name | Callout View to Show | The name of the callout view to show or hide. |
| 1 | Boolean | Show Callout View? | Specify whether to show (TRUE) or hide (FALSE) the callout view. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The callout view was found. |
| FAILURE | The specified callout view was not found. |

Remarks

None.

Hide All Callout View

Hides all existing callout views.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show/Hide Annotations for Feature Checks

Shows or hides the annotations associated with a list of feature checks.

Input Arguments

| | | | |
|---|------------------------|-------------------------|--|
| 0 | Feature Check Ref List | Feature Check Name List | The list of feature checks for which annotations should be shown or hidden. |
| 1 | Boolean | Show? | If TRUE, shows the annotations. If FALSE, hides them. |
| 2 | Boolean | Highlight? | Indicates whether the list of associated objects or faces should be highlighted in the graphical view. |
| 3 | Boolean | Set Inspection View? | If set to TRUE, the view will snap to the inspection view for the last feature check in the list. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The annotations were shown or hidden successfully. |
| FAILURE | The specified feature checks were not found. |

Remarks

None.

Show/Hide Annotations for Datums

Shows or hides the annotations associated with a list of datums.

Input Arguments

| | | | |
|---|----------------|----------------------|--|
| 0 | Datum Ref List | Datum Name List | The list of datums to modify. |
| 1 | Boolean | Show? | If TRUE, shows the annotations. If FALSE, hides them. |
| 2 | Boolean | Highlight? | Indicates whether the list of associated objects or faces should be highlighted in the graphical view. |
| 3 | Boolean | Set Inspection View? | If set to TRUE, the view will snap to the inspection view for the last datum in the list. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The annotations were shown or hidden successfully. |
| FAILURE | The specified datums were not found. |

Remarks

None.

Show/Hide Instruments

Shows or hides a set of instruments in the graphical view.

Input Arguments

| | | | |
|---|-----------------------------------|-------------------|---|
| 0 | Collection Instrument ID Ref List | Instrument IDs | A list of the instrument IDs to show or hide. |
| 1 | Boolean | Show Instruments? | Specify whether to show (TRUE) or hide (FALSE) the instruments. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The specified instruments were shown/hidden. |
| PARTIAL SUCCESS | At least one instrument was shown/hidden, and at least one was not found. |
| FAILURE | The specified instruments were not found. |

Remarks

Instrument IDs are zero-based, meaning they counted starting from zero.

Show/Hide Instrument Probe Tip

Shows or hides an instrument's probe tip in the graphical view. Equivalent to the "Draw Probe Tip" setting in the User Options>Display tab.

Input Arguments

| | | | |
|---|---------|----------------------------|---|
| 0 | Boolean | Show Instrument Probe Tip? | Indicates whether the tip should be shown (TRUE) or hidden (FALSE). |
|---|---------|----------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show/Hide Relationship Report

Shows or hides the relationship report for the selected collection.

Input Arguments

| 0 | Collection Name | Collection Name | Collection to consider |
|---|-----------------|--------------------------|------------------------------|
| 1 | Boolean | Show Relationship Report | True will display the report |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Show/Hide Relationship Watch

Shows or hides a Watch Window displaying the specified parameters using the referenced relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | Name of the relationship to watch |
|---|------------------------|--------------------------------------|-----------------------------------|
| 1 | Boolean | Show Relationship Watch | True displays the watch window |
| 2 | Collection Object Name | Relationship Watch Window Properties | Properties to use when displayed |
| 3 | Integer | Window Top Left X Position | |
| 4 | Integer | Window Top Left Y Position | |
| 5 | Integer | Window Width | |
| 6 | Integer | Window Height | |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | This Watch Window display status was updated. |
| FAILURE | The specified relationship were not found. |

Remarks

None.

Show Items in Tree

Shows specified items in the tree, optionally collapsing the categories for other items.

Input Arguments

| | | | |
|---|-----------------------------------|---------------------------|---|
| 0 | Boolean | Collapse all other Items? | Indicates whether SA should collapse everything in the tree not specifically specified. |
| 1 | Point Name Ref List | Points | The list of point to display in the tree. |
| 2 | Collection Object Name Ref List | Objects | The list of objects to display in the tree. |
| 3 | Collection Instrument ID Ref List | Instruments | The list of instruments to display in the tree. |
| 4 | Feature Check Ref List | Feature Checks | The list of feature checks to display in the tree. |
| 5 | Datum Ref List | Datums | The list of datums to display in the tree. |
| 6 | String Ref List | Collections | The list of collections to display in the tree. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Highlight Operations

Highlight Objects

Highlights the desired object(s).

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------------|---|
| 0 | Collection Object Name Ref List | Object Names (Empty to clear all) | The list of objects that will be highlighted. |
| 1 | Boolean | HighLight Objects? | Indicates highlight condition of object(s). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Highlight Relationships

Highlights the desired relationship(s).

Input Arguments

| | | | |
|---|-----------------------|------------------------------------|---|
| 0 | Relationship Ref List | Relationships (Empty to clear all) | The list of relationships that will be highlighted. |
| 1 | Boolean | HighLight Objects? | Indicates highlight condition of relationship(s). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Will highlight all relationship's entities shown in graphical view and its tree node.

Highlight Point

Highlights the desired point.

Input Arguments

| | | | |
|---|------------|---------------------------------|---|
| 0 | Point Name | Point Name (Empty to clear all) | The name of the point that will be highlighted. |
| 1 | Boolean | Show Point? | Indicates highlight condition of point. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Colors

Set Working Color

Sets the current working color which will be applied to the next object created.

Input Arguments

| | | | |
|---|-------|------------------------|------------------------|
| 0 | Color | New Working Color Name | Pick the color desired |
|---|-------|------------------------|------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Working Color Auto Increment

Sets the current working color auto increment status.

Input Arguments

| | | | |
|---|---------|----------------|--|
| 0 | Boolean | Auto increment | True enables the auto increment colors process |
|---|---------|----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Background Color

Sets the current background and highlight colors.

Input Arguments

| | | | |
|---|-------------------------------|---------------------------|--|
| 0 | Background Color Type | Background Color Type | Specifies whether the background color is solid or a gradient. |
| 1 | Color | Solid Color Name | The color (when solid is the background color type). |
| 2 | Color | Gradient Start Color Name | The start color for the gradient (when gradient is the background color type). |
| 3 | Color | Gradient End Color Name | The end color for the gradient (when gradient is the background color type). |
| 4 | Color Gradient Direction Type | Gradient Color Direction | Specifies whether the gradient is horizontal or vertical. |
| 5 | Color | Highlight Color | The highlight color to use. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Object(s) Color

Sets the current color for the selected objects.

Input Arguments

| 0 | Collection Object Name Ref List | Objects to Change | List of objects to recolor |
|---|---------------------------------|------------------------|---|
| 1 | Color | New Working Color Name | Color |
| 2 | Boolean | Auto Increment | True automatically increments the colors. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Object Color

Returns the color of the selected object.

Input Arguments

| | | | |
|---|------------------------|-------------|------------------------------|
| 0 | Collection Object Name | Object Name | Name of the reference object |
|---|------------------------|-------------|------------------------------|

Return Arguments

| | | | |
|---|-------|--------------|-----------------------|
| 1 | Color | Object Color | Returned Object Color |
|---|-------|--------------|-----------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The color was returned successfully. |
| FAILURE | The object was not found. |

Remarks

None.

Convert RGB Values to Integer

Takes a reference RGB color and splits into individual Red, Green and Blue integer components returning the values from 0 to 255.

Input Arguments

| | | | |
|---|-------|-----|----------------------|
| 0 | Color | RGB | The color to convert |
|---|-------|-----|----------------------|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Red Color | Red component of the reference color |
| 2 | Integer | Green Color | Green component of the reference color |
| 3 | Integer | Blue Color | Blue component of the reference color |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Convert Integer Values to RGB

This command takes a set of reference integer values and builds a color from the individual Red, Green and Blue integer components.

Input Arguments

| | | | |
|---|---------|---------------------|--|
| 0 | Integer | Red Color (0-255) | Red component of the reference color |
| 1 | Integer | Green Color (0-255) | Green component of the reference color |
| 2 | Integer | Blue Color (0-255) | Blue component of the reference color |

Return Arguments

| | | | |
|---|-------|-----|----------------------|
| 3 | Color | RGB | The color to convert |
|---|-------|-----|----------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Refresh Views

Triggers a refresh for the graphical view and menus.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Auto-Scale

Scales the view so that all visible objects appear in the graphical view.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Target Labels Use Full Names

Specifies whether or not target labels should use full names.

Input Arguments

| | | | |
|---|---------|-----------------|--|
| 0 | Boolean | Use Full Names? | Indicates whether full names should be used. |
|---|---------|-----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Render Mode Type

Sets the current graphical view rendering mode.

Input Arguments

| | | | |
|---|------------------|----------------|--|
| 0 | Render Mode Type | Rendering Mode | Specify whether the view should be rendered in wireframe, hidden line removed, solid+edges, or solid shading mode. |
|---|------------------|----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set View Clipping Plane

Activates a clipping plane on the XY plane of the specified object's base frame. This clipping plane is added to the global view clipping plane list.

Input Arguments

| | | | |
|---|------------------------|------------------------|---|
| 0 | Collection Object Name | Object | The object to which the clipping plane should be attached. |
| 1 | Boolean | Remove Clipping Plane? | Indicates whether the specified object should have a clipping plane added or removed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The clipping plane was set successfully. |
| FAILURE | The object was not found. |

Remarks

None.

Get View Clipping Plane

Returns a list of the defined clipping plane reference objects in the active clipping plane view list. This includes global clipping planes set under view control and/or clipping planes embedded in a callout and enabled.

Input Arguments

None

Return Arguments

| | | | |
|---|---------------------------------|---------|--|
| 0 | Collection Object Name Ref List | Objects | A list of the objects used to define the clipping entities |
|---|---------------------------------|---------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command does not reflect Cloud Clipping. It always succeeds and will return an empty list if no clipping is enabled.

Set SA's Window State

Sets the window state for the SpatialAnalyzer application. Allows you to show, hide, maximize, minimize, or restore the SA application window.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | Window State | SA Window State | The state for the SA window. |
| 1 | Collection Object Name Ref List | Object Name List | A list of objects to export to the file. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set SA's Window Pos

Sets the position of the SA window.

Input Arguments

| | | | |
|---|---------|-------|--|
| 0 | Integer | Pos X | The x position of the window, where zero represents the left side of the screen. |
| 1 | Integer | Pos Y | The y position of the window, where zero represents the top side of the screen. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set SA's Window Size

Sets the size of the SA window.

Input Arguments

| | | | |
|---|---------|--------|--------------------------------------|
| 0 | Integer | Width | The width of the window, in pixels. |
| 1 | Integer | Height | The height of the window, in pixels. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set MP's Window State

Sets the window state of the executing MP. “Minimize” shrinks the executing MP to a corner of the MP bar. “Maximize” reverses this effect and has the same behavior as “Restore”.

Input Arguments

| | | | |
|---|--------------|-----------------|--|
| 0 | Window State | MP Window State | The window state for the executing MP. |
| 1 | Integer | Height | The height of the window, in pixels. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The “Show” and “Hide” options have no effect.

Center Graphics About Point

Centers the graphical view on the specified point.

Input Arguments

| | | | |
|---|------------|------------|----------------------------------|
| 0 | Point Name | Point Name | The name of the point to center. |
|---|------------|------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The point was found. |
| FAILURE | The specified point was not found. |

Remarks

None.

Center Graphics About Object(s)

Centers the graphical view on one or more specified objects of a specified type.

Input Arguments

| 0 | Object Type | Object Type | The type of the objects to center on. |
|---|-------------|------------------------------|---|
| 1 | String | Collection Wildcard Criteria | A wildcard specifying the name of the collection containing the objects to center on. |
| 2 | String | Object Wildcard Criteria | A wildcard specifying the name of the objects to center on. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The specified objects were found. |
| FAILURE | The specified objects were not found. |

Remarks

Wildcards can be entered using asterisks(*) or question marks(?) to denote a multi-character wildcard or a single-character wildcard.

Examples (enclosing quotes would not be entered):

- “Line?” would find all names beginning with “Line” and followed by a single character (ex. Line1, Line 2, Line 3).
- “Line*” would find all names beginning with “Line” and followed by one or more characters (ex. Line1, Line10, Line4023).
- “??a?e” would find all names with 2 characters, followed by an a, followed by another character, and ending in an e. (ex. Frame, Slate, Grate).

Use an asterisk (*) to specify “all collections” or “all objects” of the specified type.

Set Object(s) Translucency

Input Arguments

| | | | |
|---|----------------------|-------------------|---------------------------------|
| 0 | Object Name Ref List | Objects to change | List of objects to edit |
| 1 | Translucency Type | Render Type | Display rendering to set |
| 2 | Double | Opacity Value | Degree of translucency to apply |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The specified objects were edited. |
| FAILURE | The specified objects were not found. |

Remarks

6 CLOUD VIEWER OPERATIONS

Cloud Display Control

Sets parameters for display of point clouds.

Input Arguments

| | | | |
|---|---------|-----------------------|--|
| 0 | Integer | Thin (Draw Increment) | Indicates what n-th number of points should be drawn (1 indicates every cloud point, 2 is every other, etc.) |
| 1 | Integer | Point Size | The drawn size for each cloud point (in pixels). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Reset Cloud Bounding Box

Provides the ability to adjust the bounding box display settings for the selected point cloud.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Cloud Name | Name of the Cloud to Edit. |
| 1 | Coordinate System Type | Cloud Box Type | Pick from available box display computation methods |
| 2 | Boolean | Show Bounding Box? | True - displays the bounding box |
| 3 | Boolean | User All Points? | True enables the use of all points |
| 4 | integer | Desired Point Count | Number of cloud points to use if "Use All Points" is not enabled. |

Return Arguments

| | | | |
|----|-----------|----------------------------------|---|
| 5 | Double | X Dimension | Box Length |
| 6 | Double | Y Dimension | Box Width |
| 7 | Double | Z Dimension | Box Height |
| 8 | Vector | X-Axis (in WORLD) | X-axis direction of the box |
| 9 | Vector | Y-Axis (in WORLD) | Y-axis direction of the box |
| 10 | Vector | Z-Axis (in WORLD) | Z-axis direction of the box |
| 11 | Vector | Centroid (in WORLD) | Center point for the box |
| 12 | Transform | Reference Transform (in WORLD) | Base reference frame for the box in WORLD |
| 13 | Transform | Reference Transform (in Working) | Base reference frame for the box in Working |
| 14 | Integer | Points Used for Bounding Box | Number of points used for the box |

Returned Status

| | |
|---------|--|
| SUCCESS | The cloud bounding box display was adjusted successfully |
| FAILED | The specified cloud could not be found |

Remarks

The box reference frame is based on the centroid of the box as the origin and the X, Y, Z vectors define the box component directions.

Get Cloud Point Count

Returns basic information on the selected point cloud

Input Arguments

| | | | |
|---|------------------------|------------|-------------------|
| 0 | Collection Object Name | Cloud Name | Cloud to Consider |
|---|------------------------|------------|-------------------|

Return Arguments

| | | | |
|---|---------|------------------------|--|
| 1 | Integer | Point Count | Number of cloud points in the cloud |
| 2 | Double | Planar Offset | Planar Offset saved with the cloud |
| 3 | Double | Radial Offset | Radial Offset Saved with the cloud |
| 4 | Integer | Active Clipping Planes | Number of active clipping planes enabled |

Returned Status

| | |
|---------|---|
| SUCCESS | Cloud settings were returned successfully |
| FAILURE | Reference Object could not be found |

Remarks

None.

Set Cloud Default Clipping Plane

Establishes a cloud clipping plane definition that is then passed to newly measured point clouds.

Input Arguments

| | | | |
|---|-------------------------|------------------------|---|
| 0 | Boolean | Enable Cloud Clipping? | Turns on/off the default cloud clipping plane |
| 1 | Collection Object Name | Reference Object | Select object to define clipping plane definition using its base frame. |
| 2 | Clipping Entity Options | Clipping Options | Dialog selection of the X,Y,Z clipping components. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Default Clipping Plane status has been updated |
| FAILURE | Reference Object could not be found |

Remarks

None.

Clear Cloud Viewer

Clears an instrument interface's cloud viewer.

Input Arguments

| | | | |
|---|----------|---------------|--|
| 0 | Inst. ID | Instrument ID | The instrument ID of the instrument of interest. |
|---|----------|---------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The viewer was cleared successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Send Cloud to SA

Sends all of the visible cloud points in a cloud viewer to the SA job file.

Input Arguments

| | | | |
|---|------------|---------------|--|
| 0 | Inst. ID | Instrument ID | The instrument ID of the instrument of interest. |
| 1 | Cloud Name | Cloud Name | The name to use for the new point cloud. |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The cloud was sent successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

The point cloud will be placed into the active collection.

Set Filter

Sets the quality threshold for an instrument's cloud viewer. All cloud points with quality values below the specified value are not displayed in the viewer.

Input Arguments

| | | | |
|---|----------|---------------|--|
| 0 | Inst. ID | Instrument ID | The instrument ID of the instrument of interest. |
| 1 | Integer | Filter Value | The quality threshold (0-100). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The threshold was set successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Save Point Cloud File

Saves the cloud points in a cloud viewer to a file.

Input Arguments

| | | | |
|---|----------------------------|---------------|---|
| 0 | Inst. ID | Instrument ID | The instrument ID of the instrument of interest. |
| 1 | File Path or Embedded File | File Path | The path of the file to save. |
| 2 | Boolean | Save as Ascii | Indicates whether a file should be saved in ASCII format (TRUE) or binary format (FALSE). |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The file was saved successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Binary files will save and load much faster, and use less disk space--but they are not human-readable and cannot be easily imported back into SA (without using the "Load Point Cloud File" command).

Load Point Cloud File

Loads the cloud points for a file into an instrument's cloud viewer.

Input Arguments

| | | | |
|---|----------------------------|---------------|--|
| 0 | Inst. ID | Instrument ID | The instrument ID of the instrument of interest. |
| 1 | File Path or Embedded File | File Path | The path of the file to load. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file was loaded successfully. |
| FAILURE | The instrument or file was not found, or the file was not of the correct format. |

Remarks

The instrument must be in the active collection, and its interface must be active.

The format of the cloud file (ASCII vs. binary) will be detected automatically.

7 CONSTRUCTION OPERATIONS

Mirror Object(s)

Mirrors one or more objects across one of a specified frame's orthogonal planes.

Input Arguments

| 0 | Collection Object Name Ref List | Object(s) | A list of the objects to mirror. |
|---|---------------------------------|------------------------------|--|
| 1 | Collection Object Name | Frame Name | The name of a frame whose axes define a plane to mirror across. |
| 2 | MP Plane Type | Frame Plane to Mirror Around | Choose to mirror across the selected frame's XY, XZ, or YZ plane. |
| 3 | Boolean | Copy? [FALSE=Move] | Specify whether to make a copy of the existing objects (TRUE) first, or to just move the existing objects. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | One or more specified objects were found and mirrored. |
| FAILURE | No specified objects were found, or the specified frame was not found. |

Remarks

The mirrored objects inherit the name of the originals, but with “-mirror” appended to the end.

Copy Object

Copies a specified object.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Source Object | The source object to copy. |
| 1 | Collection Object Name | New Object Name | The name of the newly copied object. |
| 2 | Boolean | Overwrite If Exists? | Specify whether to overwrite an object if it already exists (TRUE) or to not (FALSE). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The object was copied successfully. |
| FAILURE | The specified source object was not found, or "Overwrite If Exists?" was set to FALSE and the object already exists. |

Remarks

None.

Copy Objects to a collection

Copies one or more objects to a specific collection.

Input Arguments

| 0 | Collection Object Name Ref List | Source Objects | A list of the objects to copy. |
|---|---------------------------------|-----------------------------|--|
| 1 | Collection Name | Destination Collection Name | The name of the collection into which to copy the specified objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The object(s) were copied successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the specified objects to copy were found. |
| FAILURE | No specified objects to copy were found. |

Remarks

If the destination collection does not exist, it will be created for you.

Move Objects to a collection

Moves one or more objects to a specific collection.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name Ref List | Source Objects | A list of the objects to move. |
| 1 | Collection Name | Destination Collection Name | The name of the collection into which to move the specified objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The object(s) were moved successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the specified objects to move were found. |
| FAILURE | No specified objects to move were found. |

Remarks

If the destination collection does not exist, it will be created for you.

Copy Objects - Point to Point Delta

Copies one or more objects, translates them based on a delta between two points, and places the copies in a specific collection.

Input Arguments

| | | | |
|---|---------------------------------|--|--|
| 0 | Collection Object Name Ref List | Objects to Copy | A list of the objects to copy. |
| 1 | Point Name | First Delta Point | A point defining the start of the translation vector. |
| 2 | Point Name | Second Delta Point | A point defining the end of the translation vector. |
| 3 | Collection Name | Destination Collection Name (optional) | The name of the collection into which to place the copied objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The object(s) were copied successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the specified objects to copy were found. |
| FAILURE | No specified objects to copy were found, or the first/second delta point was not found. |

Remarks

If the destination collection is left blank, copies will be placed in the active collection. If the destination collection is specified and does not exist, it will be created for you.

Move Objects - Point to Point Delta

Translates one or more objects based on a delta between two points.

Input Arguments

| | | | |
|---|---------------------------------|--------------------|---|
| 0 | Collection Object Name Ref List | Objects to Move | A list of the objects to translate. |
| 1 | Point Name | First Delta Point | A point defining the start of the translation vector. |
| 2 | Point Name | Second Delta Point | A point defining the end of the translation vector. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The object(s) were moved successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the specified objects to move were found. |
| FAILURE | No specified objects to copy were found, or the first/second delta point was not found. |

Remarks

None.

Rename Point

Renames a point.

Input Arguments

| | | | |
|---|------------|----------------------|---|
| 0 | Point Name | Original Point Name | The name of the point to rename. |
| 1 | Point Name | New Point Name | A new name for the point. |
| 2 | Boolean | Overwrite if exists? | Specify whether a point should be overwritten if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was renamed successfully. |
| FAILURE | The original point was not found, or a point already exists with the new point name and "Overwrite if exists?" is set to FALSE. |

Remarks

Renaming a point will preserve its status as a measured or constructed point. (In other words, measured points will maintain their association with the instrument that measured them).

If a specified point group does not already exist, it will be created for you.

* Note with Overwrite if exists? (A1) the original point will be renamed to the new point name and the new point will be deleted if present in the job rather than built from strings.

Rename Points with Name Pattern

Renames a list of points incrementally based on a supplied name pattern and a starting value.

Input Arguments

| | | | |
|---|---------------------|--------------|---|
| 0 | Point Name Ref List | Point Names | The list of points to rename. |
| 1 | String | Name Pattern | A prefix or pattern to use for the new point names. |
| 2 | Integer | Start Value | The starting value to apply to the first point. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The points were renamed successfully. |
| FAILURE | The points were not found. |

Remarks

If the Name Pattern argument uses the `%d` wildcard, it will be replaced by the current number value. For example, if the start value is 5 and the name pattern is `Test%dPoint`, then the first point will be named `Test5Point`, the second will be named `Test6Point`, and so on. If the `%d` wildcard is omitted, the current number will be appended to the name (with a hyphen). For example, a name pattern of `NewPoint` results in `NewPoint-5`, `NewPoint-6`, etc. The `%d` wildcard may be used multiple times in a name pattern.

Rename Collection

Renames a collection.

Input Arguments

| | | | |
|---|-----------------|--------------------------|---------------------------------------|
| 0 | Collection Name | Original Collection Name | The name of the collection to rename. |
| 1 | Collection Name | New Collection Name | A new name for the collection. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The collection was renamed successfully. |
| FAILURE | A collection with the new collection name already exists, or the original collection was not found. |

Remarks

None.

Rename Object

Renames an object.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Original Object Name | The collection object name of the object to rename. |
| 1 | Collection Object Name | New Object Name | A new collection object name for the object. |
| 2 | Boolean | Overwrite if exists? | Specify whether or not to overwrite an object if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The object was renamed successfully. |
| FAILURE | The original object was not found, or an object with the new name already exists and "Overwrite if exists?" is set to FALSE. |

Remarks

None.

Rename Item

Renames an item.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Original Item Name | The collection item name of the object to rename. |
| 1 | Collection Object Name | New Item Name | A new collection item name for the object. |
| 2 | Boolean | Overwrite if exists? | Specify whether or not to overwrite an item if it already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The item was renamed successfully. |
| FAILURE | The original item was not found, or an item with the new name already exists and "Overwrite if exists?" is set to FALSE. |

Remarks

None.

Delete Points

Deletes one or more points.

Input Arguments

| | | | |
|---|---------------------|-------------|---------------------------------|
| 0 | Point Name Ref List | Point Names | A list of the points to delete. |
|---|---------------------|-------------|---------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If a measured target is deleted, all observations comprising that target are also deleted.

Delete Points WildCard Selection

Deletes points from a specified set of groups that match a specified wildcard point name.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|--|
| 0 | Collection Object Name Ref List | Groups to Delete From | A list of the source point groups from which the points may be selected. |
| 1 | Point Name | WildCard Selection Names | A point name containing the point selection criteria. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | Any matching points were deleted successfully, and the specified groups were all found. |
| PARTIAL SUCCESS | At least one (but not all) of the source groups was not found. |
| FAILURE | No source groups could be found. |

Remarks

Enter wildcard values for the collection, point group, and point name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all points that start with s1 followed by two characters, the point name defining the selection criteria would be ***::*::s1??**.

Construct Objects From Surface Faces-Runtime Select

Creates a set of primitive geometric shapes (planes, cylinders, spheres, cones, lines, points, and circles) from CAD surfaces selected by the user at runtime.

Input Arguments

| | | | |
|---|---------|----------------------|--|
| 0 | Boolean | Construct Planes? | Indicates whether planes should be constructed. |
| 1 | Boolean | Construct Cylinders? | Indicates whether cylinders should be constructed. |
| 2 | Boolean | Construct Spheres? | Indicates whether spheres should be constructed. |
| 3 | Boolean | Construct Cones? | Indicates whether cones should be constructed. |
| 4 | Boolean | Construct Lines? | Indicates whether lines should be constructed. |
| 5 | Boolean | Construct Points? | Indicates whether points should be constructed. |
| 6 | Boolean | Construct Circles? | Indicates whether circles should be constructed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | All possible surfaces were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Collections

Set (or construct) default collection

Sets a collection as the active collection. If the specified collection does not exist, the collection is first created, then activated.

Input Arguments

| | | | |
|---|-----------------|-----------------|---|
| 0 | Collection Name | Collection Name | The name of the collection to activate (or create). |
|---|-----------------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Construct Collection

Creates a collection, optionally placing it into a specific tree folder and activating it.

Input Arguments

| 0 | Collection Name | Collection Name | The name of the collection to create. |
|---|-----------------|--------------------------|--|
| 1 | String | Folder Path | The folder path into which to create the collection. |
| 2 | Boolean | Make Default Collection? | Specify whether to activate the collection after its creation. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Folder paths should be entered with each folder separated by a backslash (\). For example, a folder path of **Measured\Day 1** would create a root folder named **Measured**, a subfolder named **Day 1**, and construct the collection in that folder. If the specified collection already exists, a new collection will be created that increments the name of the collection.

Get Active Collection Name

Returns the name of the active default collection.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|----------------------------------|--|
| 0 | String | Currently Active Collection Name | The name of the currently active default collection. |
|---|--------|----------------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Delete Collection

Deletes a collection.

Input Arguments

| | | | |
|---|-----------------|------------------------------|---------------------------------------|
| 0 | Collection Name | Name of Collection to Delete | The name of the collection to delete. |
|---|-----------------|------------------------------|---------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The collection was successfully deleted. |
| FAILURE | The collection to delete was not found, or an attempt was made to delete the last collection in the job. |

Remarks

You are not permitted to delete the last collection in the job. Attempting to do so will cause this command to fail.

If you attempt to delete the active collection, the adjacent collection above the active collection will be activated first.

Delete Collections by Wildcard

Deletes a set of collections that match the wildcard search criteria.

Input Arguments

| | | | |
|---|---------|--------------------------------|---|
| 0 | String | Search String | The wildcard selection criteria |
| 1 | Boolean | Case Sensitive Search | Indicates whether or not the search should be case-sensitive. |
| 2 | Boolean | Allow Deleting all Collections | If set to <i>FALSE</i> , the command will fail if all collections meet the search criteria. |

Return Arguments

| | | | |
|---|---------|-------------|---|
| 3 | Integer | Num Deleted | The number of collections that were successfully deleted. |
| 4 | Integer | Num Failed | The number of collections which matched the search criteria but could not be deleted. |

Returned Status

| | |
|---------|---|
| SUCCESS | The collections were deleted successfully. |
| FAILURE | Argument 2 was set to <i>FALSE</i> and all collections met the search criteria. |

Remarks

Enter wildcard search criteria using the same convention as elsewhere in SA: asterisks (*) are wildcard placeholders for one or more characters, and a question mark (?) is a wildcard character for a single character. Specific characters can also be found using brackets[].

Points and Groups

Construct Point (Fit to Points)

Constructs a point at a point representing the mathematical average of the source points.

Input Arguments

| | | | |
|---|---------------------|----------------------|--|
| 0 | Point Name Ref List | Point Names | A list of the source point names to fit. |
| 1 | Point Name | Resulting Point Name | The name for the resulting point. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | All source points were successfully averaged. |
| PARTIAL SUCCESS | One or more source points (but not all) were not found. |
| FAILURE | No source points were found. |

Remarks

If a point with the resulting point name already exists, the point will be given a new name by appending an asterisk to the end.

Construct a Point in Working Coordinates

Constructs a point in the specified coordinates of the working frame.

Input Arguments

| 0 | Point Name | Point Name | The name of the point to create. |
|---|------------|---------------------|---|
| 1 | Vector | Working Coordinates | The coordinates of the point to create, expressed in the working frame. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If a point with the resulting point name already exists, the point will be given a new name by appending a number to the end.

Construct Point From Survey Target Center

Constructs a point detected from intensity information saved in a point cloud. Used for rectangular survey targets.

Input Arguments

| | | | |
|---|------------------------|--------------------------|---|
| 0 | Collection Object Name | Cloud Containing Target | Name of the cloud to consider |
| 1 | Point Name | Reference Seed Point | Name of a reference point to use as a seed for the search |
| 2 | GD&T Zone Type | Survey Target Type | Pick the zone type to use |
| 3 | Double | Search Diameter | Proximity zone definition for used to define which cloud points to include in the search. |
| 4 | Point Name | Result Center Point Name | The name of the point to create. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The survey point was constructed successfully |
| PARTIAL SUCCESS | No target point was found. |
| FAILURE | The cloud or seed point were not found |

Remarks

This command uses a seed point as a reference and searches the point cloud within the designated zone with respect to it.

Set Point Position in Working Coordinates

Allows direct editing of point location in the specified coordinates of the working frame.

Input Arguments

| 0 | Point Name | Point Name | The name of the point to edit. |
|---|------------|---------------------|---|
| 1 | Vector | Working Coordinates | The coordinates of the point to create, expressed in the working frame. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | Point position updated successfully. |
| FAILURE | Point could not be found. |

Remarks

This command operates like directly opening the properties of a point and pressing the edit button. It will not update or adjust the observations on this point.

Transform Points by Delta (About Working Frame)

Allows direct editing of point locations in the coordinates of the current working frame.

Input Arguments

| | | | |
|---|---------------------|------------------------------|--|
| 0 | Point Name Ref List | Point Name List | The name of the points to edit. |
| 1 | Vector | Delta In Working Coordinates | The relative position change to apply. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | Point positions updated successfully. |
| FAILURE | Points could not be found. |

Remarks

This command operates like directly opening the properties of a point and pressing the edit button. It will not update or adjust the observations on the selected points.

Construct a Point at line MidPoint

Constructs a point at the midpoint of a line.

Input Arguments

| | | | |
|---|------------------------|------------|-----------------------------------|
| 0 | Collection Object Name | Line Name | The name of the line to consider. |
| 1 | Point Name | Point Name | The name of the point to create. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The point was created successfully. |
| FAILURE | The specified line was not found. |

Remarks

If a point with the resulting point name already exists, the point will be given a new name by appending an asterisk to the end.

Construct Point Group from Point Name Ref List

Constructs a point group containing copies of a list of points.

Input Arguments

| | | | |
|---|------------------------|-----------------|--|
| 0 | Point Name Ref List | Point Name List | The list of points to copy. |
| 1 | Collection Object Name | Group Name | The name of the group into which to copy the points. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | All specified points were found, and the point group was created successfully. |
| PARTIAL SUCCESS | One or more specified points (but not all) were found, and the point group was created successfully with the points. |
| FAILURE | None of the specified points was found. |

Remarks

If the point group already exists, the specified points are added to the group. If any of the specified points already exists in the specified group, then they are sequenced to have a unique name.

The copied points will have the same offsets as the source points (offsets will be preserved).

Construct Point Groups from Vector Groups

Constructs one or more point groups from a list of one or more vector groups.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------|--|
| 0 | Collection Object Name Ref List | Vector Groups | The list of vector groups to use as the source for the point groups. |
| 1 | String | Optional Group Name Suffix | An optional suffix to attach to each constructed point group name. |
| 2 | Boolean | Make Vector Begin Points | Specify whether to construct points at the beginning of vectors. |
| 3 | Boolean | Make Vector End Points | Specify whether to construct points at the end of vectors. |

Return Arguments

| | | | |
|---|---------------------------------|--------------|---|
| 0 | Collection Object Name Ref List | Point Groups | A list containing the point groups that were created. |
|---|---------------------------------|--------------|---|

Returned Status

| | |
|-----------------|--|
| SUCCESS | All specified vector groups were found, and the point groups were created successfully. |
| PARTIAL SUCCESS | One or more specified vector groups (but not all) were found, and the point groups were created successfully from those vector groups. |
| FAILURE | None of the specified vector groups were found. |

Remarks

If a resulting point group already exists, the specified points are added to the group. If any of the points already exists in the specified group, then they are sequenced to have a unique name.

If only the beginning points or only the end points are created, the resulting points will match the vector names. If both the begin and end points are created, the begin points will have a “_Begin” suffix, and the end points will have an “_End” suffix.

The point group names match the names of their source vector group, but with the optional suffix appended.

Construct Point Group from Point Cloud

Constructs a point group from a point cloud.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | Cloud Name | The name of the point cloud to use as the source. |
| 1 | Collection Object Name | Point Group Name | The name of the point group into which to place the resulting points. |
| 2 | String | Point Prefix | A prefix applied to each point name. |
| 3 | Integer | Starting Point Number | The index at which to start numbering the resulting points. |
| 4 | Double | Point Offset | The planar/radial offset to assign to each resulting point. |
| 5 | Boolean | Sub-Sampling? | Specify whether to sub-sample the point cloud when creating the points. |
| 6 | Double | Sub-Sampling Distance | If sub-sampling is enabled, indicates the minimum distance between the resulting points. |
| 7 | Boolean | Show Progress? | Specify whether a progress bar should be displayed while calculating results. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point group was created successfully. |
| FAILURE | The point cloud was not found. |

Remarks

If a resulting point group already exists, the specified points are added to the group. If any of the resulting point names already exists in the specified group, then they are sequenced to have a unique name.

Sub-sampling will ensure that each point in the resulting group is approximately, but no less than, the specified sub-sampling distance apart.

Construct Point from Cloud Point - Runtime Select

Constructs a point from a selected cloud point.

Input Arguments

| | | | |
|---|------------|------------------------|---|
| 0 | String | Selection Prompt | "Select a cloud point" or whatever prompt desired |
| 1 | Boolean | Construct Point? | True - builds a new point |
| 2 | Point Name | Constructed Point Name | Name to use for the extracted point |

Return Arguments

| | | | |
|---|--------|---------------------------------|------------------------------------|
| 3 | Vector | Selected Cloud Point Coordinate | XYZ position of the selected point |
|---|--------|---------------------------------|------------------------------------|

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The point was created successfully. |
| FAILURE | The circle was not found. |

Remarks

The returned XYZ point location is reported in WORLD coordinates.

Construct a Point at Circle Center

Constructs a point at the center of a circle.

Input Arguments

| | | | |
|---|------------------------|-------------|----------------------------------|
| 0 | Collection Object Name | Circle Name | The name of the circle to use. |
| 1 | Point Name | Point Name | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The point was created successfully. |
| FAILURE | The circle was not found. |

Remarks

If the resulting point name already exists, then it will be sequenced to have a unique name.

Construct Point at Intersection of Planes

Constructs a point at the intersection of three non-parallel planes.

Input Arguments

| | | | |
|---|------------------------|--------------|--|
| 0 | Collection Object Name | Plane 1 Name | The name of the first plane to intersect. |
| 1 | Collection Object Name | Plane 2 Name | The name of the second plane to intersect. |
| 2 | Collection Object Name | Plane 3 Name | The name of the third plane to intersect. |
| 3 | Point Name | Point Name | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The point was created successfully. |
| FAILURE | One of the planes was not found, or an intersection could not be determined. |

Remarks

No two planes may be parallel, and all three must not have normals that lie in the same plane. If the resulting point name already exists, the new point name will be sequenced to avoid duplicate plane names.

Construct Point at Intersection of Two Lines

Constructs a point at the intersection or mutual perpendicular midpoint between two lines.

Input Arguments

| | | | |
|---|------------------------|------------------|---|
| 0 | Collection Object Name | First Line Name | The name of the first line to intersect. |
| 1 | Collection Object Name | Second Line Name | The name of the second line to intersect. |
| 2 | Point Name | Point Name | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was created successfully. |
| FAILURE | One or both of the lines was not found. |

Remarks

If the point already exists with the same name as specified, it will be renamed to avoid duplicate names.

Construct Point at Intersection of Plane and Line

Constructs a point at the intersection of a line and a plane.

Input Arguments

| | | | |
|---|------------------------|----------------------|-------------------------------------|
| 0 | Collection Object Name | Plane Name | The name of the plane to intersect. |
| 1 | Collection Object Name | Line Name | The name of the line to intersect. |
| 2 | Point Name | Resulting Point Name | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was created successfully. |
| FAILURE | The line or plane was not found, or do not intersect. |

Remarks

If the point already exists with the same name as specified, it will be renamed to avoid duplicate names.

Construct Point at Intersection of 2 B-Splines

Constructs a point at the intersection of 2 B-Splines, or at the closest point between the two if they do not intersect.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | First B-Spline Name | The name of the first B-spline to intersect. |
| 1 | Collection Object Name | Second B-Spline Name | The name of the second B-spline to intersect. |
| 2 | Point Name | Point Name | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The point was created successfully. |
| FAILURE | One or both of the B-Splines was not found, or the resulting point already exists. |

Remarks

None.

Construct Point at Intersection of B-Spline and Surfaces

Constructs a point at the intersection of a B-Spline and each of a list of surfaces.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------|--|
| 0 | Collection Object Name | B-Spline Name | The B-spline to intersect. |
| 1 | Collection Object Name Ref List | Surface List | A list of one or more surfaces to intersect with. |
| 2 | Double | Approximation Tolerance | A value indicating the greatest deviation from the B-Spline approximation and its actual position. |
| 3 | Point Name | Point Name | The name of the resulting intersected point. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The point was intersected successfully. |
| PARTIAL SUCCESS | One or more (but not all) of the surfaces was not found. |
| FAILURE | The B-spline or surfaces were not found, or no intersection exists. |

Remarks

The intersection point will be created at the first intersection of the B-Spline with the first surface it encounters, in the direction of the B-Spline.

Construct Points at Intersection of Circle and Line

Constructs points at the intersection of a circle and a line, after the line has been projected to the plane of the circle.

Input Arguments

| | | | |
|---|------------------------|-----------------------------|---------------------------------------|
| 0 | Collection Object Name | Circle Name | The name of the circle to intersect. |
| 1 | Collection Object Name | Line Name | The name of the line to intersect. |
| 2 | Point Name | Base Point Name for results | The base name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point or points were created successfully. |
| FAILURE | The circle or line could not be found, an intersection could not be determined. |

Remarks

If two points are created from the intersection, they will be suffixed with a “1” or “2” appropriately.

Construct Points at Intersection of Principle Object Axes and Surfaces

Constructs a point at the intersection of an object's principle axis and a surface.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------|---|
| 0 | Collection Object Name Ref List | Axis Object List | One or more objects defining the axes along which to project. |
| 1 | Collection Object Name Ref List | Surface List | One or more surfaces with which to intersect. |
| 2 | String | Point Suffix (optional) | An optional suffix to add to each generated point. |
| 3 | Collection Object Name | Resultant Group Name | The group into which to place the generated points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The point or points were created successfully, or no intersections were found. |
| FAILURE | No axis or surface objects could be found. |

Remarks

The “principle object direction” is along the Z axis for frames, along the line direction for lines, along the normal for planar geometry, and through axes for other geometry.

Each axis object produces just one intersection point at the first intersection with one of the supplied surfaces. Additional intersections are not performed for a given axis object once it has already intersected with one surface.

Get Gradient At Projected Point On Surface

This command projects a point onto the selected surface and returns the normal vector of the surfaces as well as the U and V direction vectors of the surface at the projected point.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|--|
| 0 | Point Name | Point to Project | Reference point used for analysis. |
| 1 | Collection Object Name | Surface Name | Reference Surface used for projection. |
| 2 | Boolean | Generate output vector lines? | Option to construct lines defining the surface vector at the point projection. |

Return Arguments

| | | | |
|---|--------|-----------------|--|
| 3 | Vector | Projected Point | XYZ location of the projected Point |
| 4 | Vector | Normal Vector | Normal vector for the surface direction at the projected point |
| 5 | Vector | U Direction | The U direction vector at the projected point |
| 6 | Vector | V Direction | The V direction vector at the projected point |

Returned Status

| | |
|---------|---|
| SUCCESS | The point was projected successfully and direction vectors obtained |
| FAILURE | No point or surfaces could be found. |

Remarks

If Argument 2 is True, three lines will be created. The names of the lines will be based on the point name from Argument 0; [Collection]_[Group]_[Target] Projected [Direction Description], where [Direction Description] is Normal, U-Direction, or V-Direction.

All vectors returned are in the working frame.

This command returns pure U and V surface directions when the point is projected onto the a surface, but can return undesirable vector directions if the input point is projected to the edge of the surface. In the situation where the input points are generated from a B-Spline that represents the edge of the surface, a similar command is more appropriate – Get Gradient At Projected Point On Surface Edge.

Get Gradient At Projected Point On Surface Edge

This command offsets an edge point in the specified direction, then projects a point onto the selected surface. The surface U direction is determined here, then the point is projected back to the reference B-Spline. The command returns the normal vector of the surface as well as the U and V direction vectors of the surface at the projected point on the spline. The edge offset avoids ambiguous surface UV directions directly on the surface edge.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|---|
| 0 | Point Name | Point to Project | Reference point (on b-spline) used for analysis. |
| 1 | Collection Object Name | B-Spline Name | Reference B-Spline from Surface edge used for projection. |
| 2 | Collection Object Name | Surface Name | Reference Surface used for projection. |
| 3 | Vector | Edge Offset Direction | Working direction to offset Point to Project such that it will intersect the surface. |
| 4 | Double | Edge Offset Distance | Distance to offset the Point to Project such that it will intersect the surface. |
| 3 | Boolean | Generate output vector lines? | Option to construct lines defining the surface vector at the point projection. |

Return Arguments

| | | | |
|---|--------|-----------------|--|
| 3 | Vector | Projected Point | XYZ location of the projected Point |
| 4 | Vector | Normal Vector | Normal vector for the surface direction at the projected point |
| 5 | Vector | U Direction | The U direction vector at the projected point |
| 6 | Vector | V Direction | The V direction vector at the projected point |

Returned Status

| | |
|---------|---|
| SUCCESS | The point was projected successfully and direction vectors obtained |
| FAILURE | No point or surfaces could be found, or the point was projected to the surface edge after offsetting (try changing Edge Offset Distance). |

Remarks

If Argument 3 is True, three lines will be created. The names of the lines will be based on the point name from Argument 0; [Collection]_[Group]_[Target] Projected [Direction Description], where [Direction Description] is Normal, U-Direction, or V-Direction.

All vectors returned are in the working frame.

The input point is likely generated by laying out points on the surface edge spline. The input point is first offset along the Edge Offset Direction by the Edge Offset Distance, then it is projected to the surface to find the U and V surface directions. This projected point is further projected along the V direction back to the B-Spline. This results in the

returned Projected Point being different than the Point to Project, and the U-direction is only approximate as determined at the offset point projected onto the surface. The Normal vector will be correct relative to the V-direction at the returned point on the B-Spline, but will be approximate relative to the B-Spline in the U-direction.

Construct Points from Cylinder

Constructs points at the endpoints and midpoint of a cylinder's axis.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Cylinder Name | The name of the source cylinder. |
| 1 | Collection Object Name | Group Name | The point group for the resulting points. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The points were created successfully. |
| FAILURE | The source cylinder was not found. |

Remarks

The resulting points will be named "Mid", "End A", and "End B". If the points already exist, they will be sequenced to avoid duplicates.

Construct a Point at Projection of Point onto An Object

Projects a point onto an object at its nearest point.

Input Arguments

| | | | |
|---|------------------------|----------------------|--|
| 0 | Point Name | Point to Project | The name of the point to project. |
| 1 | Collection Object Name | Object Name | The name of the object to which you'd like to project the point. |
| 2 | Point Name | Resulting Point Name | The name of the resulting projected point. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was projected successfully. |
| FAILURE | The source point or object was not found. |

Remarks

This command will project points like “Point To Object” queries, acting as if “ignore edge projections” is off.

If the resulting point already exists, it will be sequenced to avoid duplicates.

Construct Points on Curves Using Max Chordal Deviation

Creates points on the selected curves (b-splines) spaced the curvature parameters, creating a denser distribution with tighter turns along the curve length.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|---|
| 0 | Collection Object Name Ref List | B-Spline List | A list of B-Splines onto which to create the points. |
| 1 | Double | Maximum Chordal Deviation | The maximum distance a chord segment can deviate from the curve |
| 2 | Double | Maximum Trim Edge Angle | Maximum angle a chordal section can reach relative to the curve without adding a point. |
| 3 | Double | Maximum Chord Length | Maximum length a chord can be between points |
| 4 | Collection Object Name | Resulting Group Name | Resulting group name for the created points. |
| 5 | String | Resulting Point Name Prefix | Starting prefix for the created points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one of the B-Splines was found, and the points were created successfully. |
| FAILURE | None of the B-Splines were found. |

Remarks

Set either the Maximum Trimmed Edge Angle (A2) or the Maximum Chord Length (A3) to zero ignore this parameter in the construction operation.

Construct Points at Projection on Surfaces - Parallel to WCF Axis

Projects one or more points to one or more surfaces along one of the working coordinate frame axis directions.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|--|
| 0 | Collection Object Name Ref List | Surface List | The list of surfaces to consider for projection. |
| 1 | Point Name Ref List | Point Names | The list of points to project. |
| 2 | String | Group Name to Contain New Points | Point group to hold the resulting projected points. |
| 3 | String | Point Name Prefix | An optional prefix for the point names. |
| 4 | String | Point Name Suffix | An optional suffix for the point names. |
| 5 | Axis Identifier | Axis | The axis of the working coordinate frame along which to project. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The points were projected successfully. |
| FAILURE | No points or surfaces could be found. |

Remarks

None.

Construct Points at Projection on Surfaces - Radial from WCF Axis

Projects one or more points to one or more surfaces radially away from a working coordinate frame axis.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|---|
| 0 | Collection Object Name Ref List | Surface List | The list of surfaces to consider for projection. |
| 1 | Point Name Ref List | Point Names | The list of points to project. |
| 2 | String | Group Name to Contain New Points | Point group to hold the resulting projected points. |
| 3 | String | Point Name Prefix | An optional prefix for the point names. |
| 4 | String | Point Name Suffix | An optional suffix for the point names. |
| 5 | Axis Identifier | Axis | The axis of the working coordinate frame from which to project. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The points were projected successfully. |
| FAILURE | No points or surfaces could be found. |

Remarks

None.

Construct Points at Projection on Surfaces - Spherical from WCF Origin

Projects one or more points to one or more surfaces spherically from the origin of the working coordinate frame.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|---|
| 0 | Collection Object Name Ref List | Surface List | The list of surfaces to consider for projection. |
| 1 | Point Name Ref List | Point Names | The list of points to project. |
| 2 | String | Group Name to Contain New Points | Point group to hold the resulting projected points. |
| 3 | String | Point Name Prefix | An optional prefix for the point names. |
| 4 | String | Point Name Suffix | An optional suffix for the point names. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The points were projected successfully. |
| FAILURE | No points or surfaces could be found. |

Remarks

None.

Construct Points at Projection On Mesh Along Direction

Constructs a new group by projecting one or more points to a point on a Scan Strip Mesh using the direction of the Z axis of the frame or “Object Providing Direction Reference”.

Input Arguments

| | | | |
|---|------------------------|--------------------------------------|---|
| 0 | Point Name Ref List | Reference Point Names | List of Points used for projection |
| 1 | Collection Object Name | Group Name for Projected Points | The group name for the newly constructed points. |
| 2 | Collection Object Name | Object Providing Direction Reference | Object used to define the direction along which points projected. |
| 3 | Boolean | Bi-directional projection? | True indicates points will be projected in both directions along the reference direction. |
| 4 | Collection Object Name | Mesh Serving As Projection Target | Scan Strip Mesh points will be projected to. |

Return Arguments

| | | | |
|---|---------------------|---------------------------|-------------------------------|
| 5 | Point Name Ref List | Resulting Point Name List | List of newly created Points. |
|---|---------------------|---------------------------|-------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The points were projected successfully. |
| FAILURE | No points or surfaces could be found. |

Remarks

None.

Construct Points Spaced at a Distance on Curves

Creates points on one or more B-Spline curves spaced by a specified distance. The distance is specified along the length of the curve.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|---|
| 0 | Collection Object Name Ref List | B-Spline List | A list of B-Splines onto which to create the points. |
| 1 | Double | Distance Between Points | The distance (along the curve) between each adjacent point. |
| 2 | Collection Object Name | Resultant Group Name | The group to contain the resulting points. |
| 3 | String | Resultant Point Name Prefix | An optional prefix to prepend to each resulting point name. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one of the B-Splines was found, and the points were created successfully. |
| FAILURE | None of the B-Splines were found. |

Remarks

If the specified point group already exists, it will automatically be sequenced to prevent duplicate group names. Point names start with "P0" and increment sequentially.

If the distance between points exceeds the length of the curve, then one point will be created at the beginning of the curve.

Construct Points N-Spaced on Curves

Creates a specified number of points spaced evenly on one or more B-Spline curves.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------------|---|
| 0 | Collection Object Name Ref List | B-Spline List | A list of B-Splines onto which to create the points. |
| 1 | Integer | Number of Evenly Spaced Points | The number of points to create on each curve. |
| 2 | Collection Object Name | Resultant Group Name | The group to contain the resulting points. |
| 3 | String | Resultant Point Name Prefix | An optional prefix to prepend to each resulting point name. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one of the B-Splines was found, and the points were created successfully. |
| FAILURE | None of the B-Splines were found. |

Remarks

If the specified point group already exists, it will automatically be sequenced to prevent duplicate group names. Point names start with “P0” and increment sequentially.

Construct Points on Object's Vertices

Creates points on one or more object's vertices.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Collection Object Name Ref List | Object Name List | A list of objects on which to create vertex points. |
| 1 | Collection Object Name | Resultant Group Name | The group to contain the resulting points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one of the objects was found, and the points were created successfully. |
| FAILURE | None of the objects were found. |

Remarks

None.

Construct Points on Surface(s) by Clicking

Prompts the user to begin clicking to create points on surfaces.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | Group Name for Points | The group name for the resulting points. |
| 1 | String | First Point Name | The name for the first clicked point. Following points will be sequentially named. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the specified point group already exists, it will automatically be sequenced to prevent duplicate group names.

Construct Points From Surface Faces - Runtime Select

Creates points from CAD surfaces selected by the user at runtime. Currently this command is used to define hole center points. It will build a point from a selected cylindrical CAD face at the center of the cylinder end closest to the click point. It will build the center of a sphere and both ends of a selected conical face.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | All possible points were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Construct Points From Surfaces On UV Grid

Creates a point group or set of point groups laid out on surfaces along the surface curves. U and V describe orthogonal directions along a surface face using its particular curve directions.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------------|---|
| 0 | Collection Object Name Ref List | Surface List | List of surfaces to use for point construction |
| 1 | String | UV Point Group Base Name | Base name for the Point Group to be built |
| 2 | Boolean | Make Each Line Separate Group? | When True it splits the resulting points into separate groups for each line. |
| 3 | Integer | Number of U Grids | Number of points to be built in U per face |
| 4 | Integer | Number of V Grids | Number of points to be built in V per face |
| 5 | MP Edge Mode | Edge Point Mode | Selection of edge options for including or excluding points built on surface edges. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | All possible points were created successfully. |
| FAILURE | The surfaces could not be found. |

Remarks

None.

Construct Point at Object Origin

Constructs a point at an object's local origin.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Object Name | The object used as the source for the origin. |
| 1 | Point Name | Resultant Point Name | The name for the resulting point. |

Return Arguments

| | | | |
|---|--------|-----------------------|---|
| 2 | Vector | Vector Representation | The vector representation of the resulting point. |
| 3 | Double | X Value | The X coordinate of the resulting point. |
| 4 | Double | Y Value | The Y coordinate of the resulting point. |
| 5 | Double | Z Value | The Z coordinate of the resulting point. |

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The point was created successfully. |
| FAILURE | The specified object was not found. |

Remarks

Each object has an internal local origin. For example, a circle's local origin is at its center, while a line's local origin is at one end. Depending on the object type, this "origin" can be used as a type of cardinal point for the object.

If the specified point already exists, it will automatically be sequenced to prevent duplicate point names.

Construct Points Shifted in Working Frame

Copies points and translates them by the specified Cartesian offset (expressed in the working frame).

Input Arguments

| | | | |
|---|------------------------|----------------------|--|
| 0 | Point Name Ref List | Original Points | A list of points to copy. |
| 1 | Collection Object Name | Group for New Points | A point group for the newly created points. |
| 2 | Vector | Shift Vector | The vector (in the working frame) describing the translation for the points. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The point was created successfully. |
| PARTIAL SUCCESS | At least one source point (but not all) was found. |
| FAILURE | None of the source points was found. |

Remarks

The newly shifted points are essentially copies of the original points. As such, any points which are measured targets are converted to constructed points, and their offsets are reset to zero.

If the specified points already exist, they will automatically be sequenced to prevent duplicate point names.

Construct Points Cylindrically Shifted

Copies points and translates them by the specified Cylindrical offset (expressed relative to a reference object's local coordinate system).

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | Reference Object Name | Object whose local coordinate system defines the R, theta, and Z directions. |
| 1 | Point Name Ref List | Original Points | A list of the points to copy and shift. |
| 2 | Collection Object Name | Group for New Points | The group to contain the new points. |
| 3 | Double | Radial Shift | The radial amount to shift the points. |
| 4 | Double | Theta Shift (degrees) | The angular amount to shift the points. |
| 5 | Double | Planar Shift | The planar amount to shift the points. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The points were created successfully. |
| PARTIAL SUCCESS | At least one source point (but not all) was found. |
| FAILURE | None of the source points were found, or the reference object was not found. |

Remarks

The newly shifted points are essentially copies of the original points. As such, any points which are measured targets are converted to constructed points, and their offsets are reset to zero.

If the specified points already exist, they will automatically be sequenced to prevent duplicate point names.

Construct Points WildCard Selection

Searches one or more source point groups and copies points matching a search criteria into a specified group.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name Ref List | Groups to Select From | A list of the source point groups. |
| 1 | Point Name | WildCard Selection Names | A point name containing the point selection criteria. |
| 2 | Collection Object Name | Group for New Points | The group to contain the new points. |
| 3 | Boolean | Include prior complete name | Specify whether to include each point's source point group in the name of the point. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | All source point groups were found, and the points were created successfully. |
| PARTIAL SUCCESS | At least one source point group (but not all) was found. |
| FAILURE | No source point groups were found. |

Remarks

The newly shifted points are essentially copies of the original points. As such, any points which are measured targets are converted to constructed points, but their offsets are preserved.

If the resulting points already exist, they will automatically be sequenced to prevent duplicate point names.

If no points match the search criteria, an empty point group will be created.

If “Include prior complete name” is set to TRUE, the new point name will be prefixed with “A_” and appended with an underscore and the original point name.

Enter wildcard values for the collection, point group, and point name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all points from all collections in point groups that start with “s” and whose point names have two digits starting with “1”, the point name defining the selection criteria would be *::s*::1?.

Construct Points Subset with Greatest Spacing

Creates a point group by subsampling a list of points, maintaining the greatest possible spacing between the subsampled points.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Point Name Ref List | Points to Subsample | A list of points to subsample. |
| 1 | Integer | Subset Size | The number of points to subsample. |
| 2 | Collection Object Name | Group for Subset | The group to contain the subsampled points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The points were subsampled successfully. |
| FAILURE | The source points were not found. |

Remarks

None.

Construct Points Layout on Grid

Creates a point group by laying out points on a grid.

Input Arguments

| | | | |
|---|------------------------|--------------|---|
| 0 | Collection Object Name | Group Name | The name of the group to create. |
| 1 | String | Point Prefix | The prefix to apply to each point name. |
| 2 | Double | X Min | The minimum X coordinate of the grid. |
| 3 | Double | X Max | The maximum X coordinate of the grid. |
| 4 | Integer | X Count | The number of points in the grid along the X direction. |
| 2 | Double | Y Min | The minimum Y coordinate of the grid. |
| 3 | Double | Y Max | The maximum Y coordinate of the grid. |
| 4 | Integer | Y Count | The number of points in the grid along the Y direction. |
| 2 | Double | Z Min | The minimum Z coordinate of the grid. |
| 3 | Double | Z Max | The maximum Z coordinate of the grid. |
| 4 | Integer | Z Count | The number of points in the grid along the Z direction. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the specified point group already exists, asterisks will be appended to the group name as necessary to make the name unique.

Construct Points Auto-Correspond 2 groups Proximity

Copies a point group and renames the copied points based on their proximity to a reference group.

Input Arguments

| | | | |
|---|------------------------|--|---|
| 0 | Collection Object Name | Reference group (known point names) | The name of the group containing the reference names. |
| 1 | Collection Object Name | Group to be copied (unknown point names) | The name of the group to be copied. |
| 2 | Double | Auto-correspond same-point tolerance | The proximity that a point needs to be to a reference point to be considered a match. |
| 3 | Collection Object Name | Group to contain matched points | The resulting (copied) group name. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The copied group was created successfully. |
| FAILURE | The source or reference group could not be found. |

Remarks

If the specified point group already exists, asterisks will be appended to the group name as necessary to make the name unique.

Construct Points Auto-Correspond 2 groups Inter-Point Distance

Copies a point group and renames the copied points based on their relative positions compared to a reference group.

Input Arguments

| | | | |
|---|------------------------|--|---|
| 0 | Collection Object Name | Reference group (known point names) | The name of the group containing the reference names. |
| 1 | Collection Object Name | Group to be copied (unknown point names) | The name of the group to be copied. |
| 2 | Double | Auto-correspond same-point tolerance | The maximum difference between a pair of reference points and the same pair of points to rename to be considered a match. |
| 3 | Collection Object Name | Group to contain matched points | The resulting (copied) group name. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The copied group was created successfully. |
| FAILURE | The source or reference group could not be found. |

Remarks

If the specified point group already exists, asterisks will be appended to the group name as necessary to make the name unique.

Average a set of Groups

Computes average points from a series of groups. Points with matching names from different groups are averaged.

Input Arguments

| 0 | Collection Object Name Ref List | Group Names | List of the source groups to average. |
|---|---------------------------------|---|---|
| 1 | Collection Object Name | Resulting Group Name | Name of the resulting group containing the average points. |
| 2 | Double | RMS Tolerance (0.0 for none) | A tolerance for the resulting RMS deviation of the averaged points from their source points. |
| 3 | Double | Maximum Absolute Tolerance (0.0 for none) | A tolerance for the maximum absolute difference between the averaged point and each source point. |
| 4 | Double | Maximum Average Tolerance (0.0 for none) | A tolerance for the maximum average difference between the averaged point and the source points. |

Return Arguments

| | | | |
|---|--------|------------------------|---|
| 5 | Double | RMS Deviation | The actual RMS deviation from the averaged points to their source points. |
| 6 | Double | Max Absolute Deviation | The actual maximum deviation from the averaged points to the source points. |
| 7 | Double | Average Deviation | The actual average deviation between the averaged points and their source points. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The groups were averaged successfully. |
| PARTIAL SUCCESS | The groups were averaged successfully, but at least one of the supplied tolerances failed. |
| FAILURE | None of the source groups could be found, or one or more tolerances failed. |

Remarks

The resulting averaged points inherit the name of the points that created them.

Copy Groups Excluding Obscured Points

Creates a new group in a specified collection containing points from one or more source point groups but which excludes all source points not visible from a specified instrument (because a surface is in the line of sight).

Input Arguments

| | | | |
|---|---------------------------------|---------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument to consider. |
| 1 | Collection Object Name Ref List | Group Names | The list of groups containing the source points. |
| 2 | Collection Name | New Collection Name | The name of the collection into which to place the resulting group. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The group was created successfully. |
| FAILURE | The instrument was not found. |

Remarks

Only surfaces can obscure points. If you want a primitive object to obscure the view to a set of points, it must be converted to a surface first.

Clear Hidden Point Bar Database

This command removes (deletes) all hidden point bar definitions from the job file.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command is used solely for the purpose of cleaning out the existing point rod definitions in preparation for the addition of a new set.

Create Hidden Point Rod

Creates a Hidden Point Bar in the Hidden Point Bar Database.

Input Arguments

| | | | |
|---|--------|--|---|
| 0 | String | Hidden Point Rod Name | Name to be saved with the Rod definition |
| 1 | Double | A to B (Target to Target) Distance | Defines the distance between the two targets on the hidden point bar. |
| 2 | Double | A to C (Target to Tip) Distance | Defines the distance from the Target A point to the tip of the bar. |
| 3 | Double | A to B (Inter-point Tolerance (0.0 for none) | The tolerance for the distance between points A and B. |

Return Arguments

| | | | |
|---|-------|------------------------|--|
| 3 | Index | Hidden Point Rod Index | The index for the newly-defined hidden point bar in the Hidden Point Bar database. |
|---|-------|------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Create Hidden Point

Creates a Hidden Point using a specified hidden point bar from the Hidden Point Bar Database.

Input Arguments

| | | | |
|---|------------|---------------------------|---|
| 0 | Point Name | End A Point Name | Name for the point measured at the Target A position. |
| 1 | Point Name | End B Point Name | Name for the point measured at the Target B position. |
| 2 | Integer | Hidden Point Rod Index | The index of the hidden point bar definition to use (from the Hidden Point Bar Database). |
| 3 | Boolean | Overwrite existing point? | Specify whether to overwrite a point if it already exists. |
| 4 | Point Name | Point Name to Create | The name of the resulting point. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The hidden point was created successfully. |
| PARTIAL SUCCESS | The hidden point was created successfully, but the tolerance for the hidden point bar was violated. |
| FAILURE | One of the endpoints was not found or the specified hidden point rod index was not found. |

Remarks

If “Overwrite existing point?” is set to FALSE, the constructed point will be automatically sequenced to prevent duplicate point names.

Get Hidden Point Rod Index by Name

Returns the index of an existing hidden point rod based upon the specified rod name.

Input Arguments

| | | | |
|---|--------|-----------------------|--|
| 0 | String | Hidden Point Rod Name | Name of the hidden Point Rod to return |
|---|--------|-----------------------|--|

Return Arguments

| | | | |
|---|---------|------------------------|---|
| 1 | Integer | Hidden Point Rod Index | Index of the specified hidden point rod |
|---|---------|------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The hidden point rod was identified successfully. |
| FAILURE | The hidden point rod with the specified name could not be found. |

Remarks

This command can be helpful in identifying the index of an existing rod. The index will change if an existing rod is deleted, where as the name will remain as a part of the rod definition.

Delete Hidden point Rod

This command removes the specified hidden point rod from the Database.

Input Arguments

| | | | |
|---|---------|------------------------|---|
| 0 | Integer | Hidden Point Rod index | Index of the hidden Point Rod to be removed |
|---|---------|------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The hidden point rod was deleted successfully. |
| FAILURE | The hidden point rod with the specified name could not be found. |

Remarks

This command deletes the specified hidden point rod by index. The index of all rods with a greater number will change if an existing rod is deleted. Using Get Hidden Point Point Rod Index by Name may be helpful in ensuring that the correct index and rod are being referenced.

Point Clouds

Construct Point Clouds from Existing Point Group

Creates a point cloud from an existing point group.

Input Arguments

| | | | |
|---|------------------------|------------------|--|
| 0 | Collection Object Name | Point Group Name | The name of the source point group. |
| 1 | Collection Object Name | Cloud Name | The name of the resulting point cloud. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point group was found and the cloud was created successfully. |
| FAILURE | The point group was not found. |

Remarks

If the resulting point cloud already exists, the cloud name is automatically sequenced to avoid duplicate cloud names.

Construct Point Clouds from Existing Cloud Points - Runtime Select

Creates a point cloud from a set of cloud points selected by the user at runtime.

Input Arguments

| | | | |
|---|------------------------|------------|--|
| 0 | Collection Object Name | Cloud Name | The name of the resulting point cloud. |
|---|------------------------|------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The cloud was created successfully. |
| FAILURE | Nothing was selected or the user cancelled the prompt. |

Remarks

None.

Construct Point Clouds from Existing Clouds - Uniform Spacing

Creates a point cloud from a set of source clouds applying a voxel sub-sampling process. This adds a single point per the desired point spacing, saving the closes cloud point to the average per voxel as long as the minimum number of points condition is met.

Input Arguments

| | | | |
|---|---------------------------------|---------------------------------|--|
| 0 | Collection Object Name Ref List | Existing Point Cloud List | The list of existing point clouds to choose from. |
| 1 | Double | Desired Point Spacing | The desired spacing for the resulting points. |
| 2 | Integer | Minimum Points Per Output Point | The minimum number of points to be used as an input for the uniformly spaced output point. |
| 3 | Collection Object Name | New Cloud Name | The name for the resulting cloud. |
| 4 | Boolean | Hide Original Point Clouds | Optionally hides the original clouds after creation. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | None of the source clouds was found. |

Remarks

None.

Construct Point Cloud from Existing Clouds

Creates a point cloud from a set of source clouds.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------|--|
| 0 | Collection Object Name Ref List | Existing Point Cloud List | The list of existing point clouds to choose from. |
| 1 | Collection Object Name | New Cloud Name | The name for the resulting cloud. |
| 2 | Cloud Thinning Options | Cloud Thinning Options | The desired thinning options to apply. |
| 3 | Boolean | Hide Original Point Clouds | Optionally hides the original clouds after creation. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | None of the source clouds was found. |

Remarks

None.

Construct Point Clouds from Visible Cloud Points

Creates a point cloud from a set of source clouds in only the currently visible points are retained. This can be useful when working with clipping planes.

Input Arguments

| | | | |
|---|---------------------------------|----------------|---|
| 0 | Collection Object Name Ref List | Source Clouds | The list of existing clouds to choose from. |
| 3 | Collection Object Name | New Cloud Name | The name for the resulting cloud. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | None of the source clouds was found. |

Remarks

Each source cloud must be visible. It will then compare its own cloud-specific clipping plane configuration for clipping and then be tested against the view clipping plane configuration. The Cloud Thinning Control settings will be ignored and all the points included as if set to 1:1 visibility regardless of the job settings.

Construct Boundary Points from Cloud

Creates a new point cloud composed of the boundary points from the reference cloud. This command emulates right-clicking on a point cloud and building the boundary points from it.

Input Arguments

| | | | |
|---|------------------------|------------------------|--|
| 0 | Collection Object Name | Source Cloud Name | The reference cloud from which to generate the boundary cloud. |
| 3 | Collection Object Name | Destination Cloud Name | The name for the resulting cloud. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | The source cloud was not found. |

Remarks

None.

Construct Cross Section Cloud

Creates a cross section cloud from a reference, direction object and section criteria.

Input Arguments

| | | | |
|----|---------------------------------|---------------------------------|--|
| 0 | Collection Object Name Ref List | Cross Section Cloud Name | Name of the new cross section cloud to build |
| 1 | Boolean | Cylindrical Cross Section Mode? | True builds concentric cylinders and filters to them as apposed to planes |
| 2 | Double | Start Distance | Offset of the first cross section from the reference object |
| 3 | Double | Section Spacing | Distance between cross sections |
| 4 | Double | Proximity Threshold | Proximity definition used to acquire cloud points |
| 5 | Integer | Maximum Section Count | The maximum number of sections to build |
| 6 | Boolean | Limit Cross Section Extent | True limits the filter extent relative to the reference. |
| 7 | Double | Radius Limit | Radial limit to apply with respect to the normal axis of the reference |
| 8 | Boolean | Project to Reference Surface | True projects the filtered points |
| 9 | Collection Object Name | Reference Object | Name of the reference object |
| 10 | Collection Object Name Ref List | Input Clouds | Clouds to consider for filtering purposes |
| 11 | Cloud Thinning Options | Cloud Thinning Settings | Cloud thinning to apply to the input cloud |
| 12 | Boolean | Update Existing Cloud | True update an existing cross section cloud, while false will build a new cloud appending "*" to an existing cloud name. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | None of the source clouds were found. |

Remarks

For more details on cross section clouds refer to the [Point Clouds](#) chapter of the users manual.

Construct Cross Section Cloud - User Select

Creates a cross section cloud from a user selected set of reference planes.

Input Arguments

| | | | |
|----|---------------------------------|------------------------------|---|
| 0 | Collection Object Name Ref List | Cross Section Cloud Name | Name of the new cross section cloud to build |
| 4 | Double | Proximity Threshold | Proximity definition used to acquire cloud points |
| 5 | Integer | Maximum Section Count | The maximum number of sections to build |
| 6 | Boolean | Limit Cross Section Extent | True limits the filter extent relative to the reference. |
| 7 | Double | Radius Limit | Radial limit to apply with respect to the normal axis of the reference |
| 8 | Boolean | Project to Reference Surface | True projects the filtered points |
| 9 | Collection Object Name Ref List | Reference Planes | List of the reference planes to use |
| 10 | Collection Object Name Ref List | Input Clouds | Clouds to consider for filtering purposes |
| 11 | Cloud Thinning Options | Cloud Thinning Settings | Cloud thinning to apply to the input cloud |
| 12 | Boolean | Update Existing Cloud | True update an existing cross section cloud, while false will build a new cloud appending "" to an existing cloud name. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The cloud was created successfully. |
| FAILURE | None of the source clouds were found. |

Remarks

For more details on cross section clouds refer to the *Point Clouds* chapter of the users manual.

Extract Sphere Centers from Point Cloud

Creates a point group by extracting sphere centers from a point cloud generated by a raster-format scanner (Surphaser scanner or FARO Photon scanner, for example).

Input Arguments

| | | | |
|---|------------------------|-----------------------|---|
| 0 | Collection Object Name | Cloud Name | The name of the point cloud in which to search for spheres. |
| 1 | Double | Desired Diameter | The nominal diameter of the spheres. |
| 2 | Double | Extraction Tolerance | The allowable tolerance on the desired diameter. Spheres exceeding this tolerance are not extracted. |
| 3 | Integer | Minimum Point Count | The minimum number of points required for the sphere fit. If less points are detected, the sphere is not extracted. |
| 4 | Collection Object Name | Group Name for Points | The group into which to place the extracted sphere centers. |
| 5 | Boolean | Perform Final Fit | True performs the ASTM fit algorithm |
| 6 | Double | Final Fit Cone Angle | Cone angle to apply |

Return Arguments

| | | | |
|---|---------|----------------------------|---|
| 5 | Integer | Number of Points Extracted | The number of center points extracted from the cloud. |
|---|---------|----------------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The points were extracted successfully. |
| FAILURE | The source cloud was not found. |

Remarks

When the Final Fit Boolean (A5) is set to True a more advance sphere fit is used. This method employs the standard sphere extraction method defined in ASME E3125-17. This method uses the instrument shot line to improve outlier rejection and further refine the location of the sphere center point.

Scale Bars

Construct Scale Bar

Creates a scale bar between two points with a specified nominal length and uncertainty.

Input Arguments

| | | | |
|---|------------------------|--------------------------|---|
| 0 | Collection Object Name | Scale Bar Name | The collection and object name for the new scale bar. |
| 1 | Point Name | Begin Target | The name for the first target of the scale bar. |
| 2 | Point Name | End Target | The name for the second target of the scale bar. |
| 3 | Double | Length | The nominal length of the scale bar. |
| 4 | Double | Uncertainty | The scale bar's uncertainty. |
| 5 | Boolean | Use Relative Tolerances? | True indicates tolerances will be relative to the nominal value |
| 6 | Boolean | Use High Tolerance | True enables a high tolerance |
| 7 | Boolean | Use Low Tolerance | True enables a low tolerance |
| 8 | Double | High Tolerance | The high tolerance value to use |
| 9 | Double | Low Tolerance | The low tolerance value to use |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Enabling relative tolerance means that the tolerance values will be set in relationship to the nominal. Disabling relative tolerance will result in tolerance values being absolute. Therefore a tolerance value of +/-0.005 applied relative to a 64in will result in a 64.005 and 63.995 tolerance. With relative disabled the 64.005 and 63.995 value should be entered for the tolerances

Lines

Construct Line 2 Points

Creates a line between two specified points.

Input Arguments

| | | | |
|---|------------------------|--------------|---|
| 0 | Collection Object Name | Line Name | The name of the resulting line to create. |
| 1 | Point Name | First Point | The name of the first point on the line. |
| 2 | Point Name | Second Point | The name of the second point on the line. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Both points were found and the line was created successfully. |
| FAILURE | At least one of the two points was not found. |

Remarks

The resulting line direction will be from the first point to the second point. It is possible to create a line of zero length, although the analysis of a zero-length line is undefined. If the specified line name already exists, the name will be incremented to avoid duplicates.

Construct Line 2 Points (Vector Notation)

Creates a line between two specified points specified in vector notation.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Line Name | The name of the resulting line to create. |
| 1 | Vector | First Vector | The vector defining the first point on the line. |
| 2 | Vector | Second Vector | The vector defining the second point on the line. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The resulting line direction will be from the first vector to the second vector. A line of zero length can be created, although analysis with a zero-length line is undefined. If the specified line name already exists, the name will be incremented to avoid duplicates.

Construct Line Normal to Object

Creates a line of a specified length normal to an object's internal Z axis.

Input Arguments

| | | | |
|---|------------------------|-------------|--|
| 0 | Collection Object Name | Line Name | The name of the resulting line. |
| 1 | Double | Line Length | The length of the resulting line. |
| 2 | Collection Object Name | Object | Object defining the normal for the resulting line. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The line was created successfully. |
| FAILURE | The source object was not found. |

Remarks

The resulting line direction will be along the specified object's internal Z axis. For example, for planes this is normal to the plane. For lines, this is parallel to the line. For frames, it's along the frame's Z axis.

A line of zero length can be created, although analysis with a zero-length line is undefined. If the specified line name already exists, the name will be incremented to avoid duplicates.

Construct Line - Project Line to Object Reference Plane

Creates a line by projecting a line to a plane.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Line to Create | The name of the resulting line. |
| 1 | Collection Object Name | Line to Project | The source line to project to the plane. |
| 2 | Collection Object Name | Object to project to | The plane to which the source line will be projected. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The line was created successfully. |
| FAILURE | The source line or projection plane was not found. |

Remarks

If the “Object to project to” is not a plane, the results are undefined--but the step will not fail.

A line of zero length can be created, although analysis with a zero-length line is undefined. If the specified line name already exists, the name will be incremented to avoid duplicates.

Construct Line - Normal to Object through Point

Creates a line normal to an object's internal Z-axis and passing through a specified point.

Input Arguments

| | | | |
|---|------------------------|----------------|---|
| 0 | Collection Object Name | Line to Create | The name of the resulting line. |
| 1 | Collection Object Name | Object Name | The object defining the normal for the resulting line. |
| 2 | Point Name | Point Name | The name of the point through which the resulting line should pass. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The line was created successfully. |
| FAILURE | The object defining the normal or the specified point were not found. |

Remarks

If the specified line name already exists, the name will be incremented to avoid duplicates.

The length of the resulting line is a valid but unspecified value.

Using any object other than a plane may result in a valid line, but the direction may be unspecified.

Construct Line 2 Plane Intersection

Creates a line along the intersection of two non-parallel planes.

Input Arguments

| | | | |
|---|------------------------|--------------|---------------------------------|
| 0 | Collection Object Name | Line Name | The name of the resulting line. |
| 1 | Collection Object Name | First Plane | The first plane to intersect. |
| 2 | Collection Object Name | Second Plane | The second plane to intersect. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The line was created successfully. |
| FAILURE | One or both of the source planes was not found, or the planes were parallel. |

Remarks

If the specified line name already exists, the name will be incremented to avoid duplicates.

The planes must not be parallel, or the command will fail.

Construct Lines From Surface Faces-Runtime Select

Creates lines from CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All possible lines were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Construct Line Center of Slot

Creates a line at the center of a slot.

Input Arguments

| | | | |
|---|------------------------|-----------|---------------------------------|
| 0 | Collection Object Name | Line Name | The name of the resulting line. |
| 1 | Collection Object Name | Slot Name | The name of the slot. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The line was created successfully. |
| FAILURE | Slot was not found. |

Remarks

None.

Construct Line From Instrument Shot

Creates a line from an instrument shot line.

Input Arguments

| | | | |
|---|------------------------|-------------------|-------------------------------|
| 0 | Point Name | Point Name | Name of the reference point |
| 1 | Integer | Observation Index | Index of the shot of interest |
| 2 | Collection Object Name | Line Name | Name of the new line to use |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The line was created successfully. |
| FAILURE | The point or observation was not found. |

Remarks

None.

Planes

Construct Plane

Constructs a plane with specified center, normal direction, and bounds.

Input Arguments

| 0 | Collection Object Name | Plane Name | The name of the resulting plane. |
|---|------------------------|---------------------------------------|---|
| 1 | Vector | Plane Center (in working coordinates) | The center of the resulting plane. |
| 2 | Vector | Plane Normal (in working coordinates) | The normal direction of the resulting plane. |
| 3 | Double | Plane Edge Dimension | The length of each edge of the plane as depicted in the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The plane was created successfully. |
| FAILURE | A normal of zero length was specified. |

Remarks

If the specified plane name already exists, the name will be incremented to avoid duplicates.

The plane normal cannot be a vector of zero length. While the plane edge dimension can be zero or negative, this could potentially cause problems with other calculations, so a positive edge dimension should always be specified.

The edge dimension is merely used to draw the plane in the graphical view. When used for analysis, a plane extends infinitely in two dimensions.

Construct Plane, Normal to Object, Through Point

Constructs a plane normal to a specified object and through a specified point.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---|
| 0 | Collection Object Name | Resultant Plane Name | The name of the resulting plane. |
| 1 | Collection Object Name | 'Normal to' Object Name | The object defining the normal for the plane. |
| 2 | Point Name | 'Through' Point Name | The point through which the plane should pass. |
| 3 | Double | Plane Edge Dimension | The length of each edge of the plane as depicted in the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The plane was created successfully. |
| FAILURE | The normal object or through point was not found. |

Remarks

The normal direction is defined by the Z-axis of the object defining the direction (if you were to build the objects frame using Construct>Frame>On Object). However, if a B-Spline is selected as the normal direction of the plane then the normal direct of the b-spline at the selected point (if the point lies on the curve or the projected closes point) will be used to define the plane direction.

If the specified plane name already exists, the name will be incremented to avoid duplicates.

While the plane edge dimension can be zero or negative, this could potentially cause problems with other calculations, so a positive edge dimension should always be specified. The edge dimension is merely used to draw the plane in the graphical view. When used for analysis, a plane extends infinitely in two dimensions.

Construct Planes, Bounding Point Group

Constructs two planes parallel to a reference plane which bound a set of specified points. The provided point group's points all lie between the two resulting planes.

Input Arguments

| | | | |
|---|------------------------|--|--|
| 0 | Collection Object Name | Reference Plane Name | The name of the reference plane defining the resulting plane normals. |
| 1 | Collection Object Name | Group to Bound | The point group whose points will be bounded. |
| 2 | Collection Object Name | Resulting 'High' Plane Name (optional) | A name for the plane constructed on the "high" side of the points. |
| 3 | Collection Object Name | Resulting 'Low' Plane Name (optional) | A name for the plane constructed on the "low" side of the points. |
| 4 | Boolean | Override Target/Point Offsets | Specify whether to override the offsets of the provided points. If yes, the offsets provided in Argument 5 will determine the plane locations. |
| 5 | Double | Offset Value | A specified offset from the measured centers to locate the bounding planes (only applies if Argument 4 is TRUE). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The planes were created successfully. |
| FAILURE | The reference plane or group to bound was not found. |

Remarks

If the specified plane names already exist, they will be incremented to avoid duplicates.

Construct Plane, Bisect 2 Planes

Constructs a plane by bisecting two planes.

Input Arguments

| | | | |
|---|------------------------|----------------------|---------------------------------|
| 0 | Collection Object Name | Resultant Plane Name | The name of the resultant plane |
| 1 | Collection Object Name | First Plane | The name of the first plane. |
| 2 | Collection Object Name | Second Plane | The name of the second plane |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The planes were created successfully. |
| FAILURE | The firsts or second plane was not found. |

Remarks

None.

Shift Plane

Shifts a plane along its normal and/or scales the size of the plane in the graphical view.

Input Arguments

| 0 | Collection Object Name | Plane | The plane to modify. |
|---|------------------------|-----------------------|---|
| 1 | Double | Shift Along Normal | The amount to shift the plane along its normal. Negative values are permitted and will shift the plane in the direction opposite its normal. |
| 2 | Double | Grow Bounds by Factor | A scale factor to apply to modify the size (bounds) of the plane in the graphical view. Values greater than 1 will make the plane larger. Values less than 1 will make the plane smaller. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The plane was modified successfully. |
| FAILURE | The plane was not found. |

Remarks

None.

Construct Planes From Surface Faces-Runtime Select

Creates planes from planar CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | All possible planes were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Circles

Construct Circle

Constructs a circle with a specified center, normal direction, and radius.

Input Arguments

| 0 | Collection Object Name | Circle Name | The name for the resulting Circle. |
|---|------------------------|--|---|
| 1 | Vector | Circle Center (in working coordinates) | A vector defining the center of the resulting circle. |
| 2 | Vector | Circle Normal (in working coordinates) | A vector defining the normal of the resulting circle. |
| 3 | Double | Circle Radius | The radius of the resulting circle. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The circle was created successfully. |
|---------|--------------------------------------|

Remarks

If the specified circle name already exists, the name will be incremented automatically to avoid duplicates.

If a circle name is not provided, a default name of "Circle" will be used.

While a normal length of zero or a zero/negative radius can be used, the results of using a zero or negative radius circle in analysis are undefined.

Construct Circles From Surface Faces-Runtime Select

Creates circles from CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All possible circles were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Construct Circles (Lines) From Surfaces

Creates circles or lines from surfaces that may have no thickness within specified diameter constraints

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name Ref List | Surfaces | The names of the surfaces |
| 1 | Double | Minimum Diameter | The minimum diameter of the circle from surface. |
| 2 | Double | Maximum Diameter | The maximum diameter of the circle from surface. |
| 3 | Double | Tolerance | The toleranc for each circle. |
| 4 | Circle Line Mode Type | Circle Line Mode | Choose between circle or line. |
| 5 | Collection Name | Destination Collection Name | The collection where created geometry will reside. |
| 6 | String | Base Name | The Prefix of the created geometry name. |

Return Arguments

| | | | |
|---|---------------------------------|------------------|--------------------------|
| 7 | Collection Object Name Ref List | Geometry Objects | The resultant geometries |
|---|---------------------------------|------------------|--------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | All possible circles or lines were created successfully. |
| FAILURE | Reference surface could not be found or no geometry found within diameter constraints. |

Remarks

This will construct circles or lines from surfaces that may have no thickness within specified diameter constraints.

Cylinders

Construct Cylinder

Constructs a cylinder with a specified end point, axis, diameter, and length.

Input Arguments

| 0 | Collection Object Name | Cylinder Name | The name for the resulting cylinder. |
|---|------------------------|---|---|
| 1 | Vector | Cylinder End Point (in working coordinates) | The end point of the resulting cylinder. |
| 2 | Vector | Cylinder Axis (in working coordinates) | A vector defining the axis of the resulting cylinder. |
| 3 | Double | Cylinder Diameter | The diameter of the resulting cylinder. |
| 4 | Double | Cylinder Length | The length of the resulting cylinder. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The cylinder was created successfully. |
| FAILURE | An invalid cylinder axis (normal) was provided. |

Remarks

If the specified cylinder name already exists, the name will be incremented automatically to avoid duplicates.

If a cylinder name is not provided, a default name of “Cylinder” will be used.

While a zero or negative length/diameter can be specified, the results during analysis can be undefined.

Construct Cylinder From End Points

Constructs a cylinder with the specified endpoints and diameter.

Input Arguments

| 0 | Collection Object Name | Cylinder Name | The name for the resulting cylinder. |
|---|------------------------|---|--|
| 1 | Vector | Cylinder End Point A (in working coordinates) | One endpoint for the resulting cylinder. |
| 2 | Vector | Cylinder End Point B (in working coordinates) | The other endpoint for the resulting cylinder. |
| 3 | Double | Cylinder Diameter | The diameter of the resulting cylinder. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The cylinder was created successfully. |
|---------|--|

Remarks

If the specified cylinder name already exists, the name will be incremented automatically to avoid duplicates.

If a cylinder name is not provided, a default name of “Cylinder” will be used.

While a zero or negative diameter (or coincident endpoints) can be specified, the results during analysis can be undefined.

Construct Cylinders From Surface Faces-Runtime Select

Creates cylinders from CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All possible cylinders were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Spheres

Construct Sphere

Constructs a sphere with a specified center and radius.

Input Arguments

| | | | |
|---|------------------------|--|-------------------------------------|
| 0 | Collection Object Name | Sphere Name | The name for the resulting sphere. |
| 1 | Vector | Sphere Center (in working coordinates) | The center of the resulting sphere. |
| 2 | Double | Sphere Radius | The radius of the resulting sphere. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The sphere was created successfully. |
|---------|--------------------------------------|

Remarks

If the specified sphere name already exists, the name will be incremented automatically to avoid duplicates.

If a sphere name is not provided, a default name of “Sphere” will be used.

While a zero or negative radius can be specified, the results during analysis can be undefined.

Construct Spheres From Surface Faces-Runtime Select

Creates spheres from CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All possible spheres were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Cones

Construct Cone

Creates a cone.

Input Arguments

| 0 | Collection Object Name | Cone Name | The name for the resulting cone. |
|---|------------------------|---|--|
| 1 | Vector | Cone End Point (in working coordinates) | The coordinates for the end point of the cone, expressed in the working coordinate frame. |
| 2 | Vector | Cone Axis (in working coordinates) | The vector for the cone axis, expressed in the working coordinate frame. This vector need not be normalized. |
| 3 | Double | Cone Length | The length of the resulting cone. |
| 4 | Double | Cone Theta Start | The starting sweep angle for the cone. |
| 5 | Double | Cone Theta Span | The ending sweep angle for the cone. |
| 6 | Double | Cone Included Angle | The included angle of the cone. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Construct Cones From Surface Faces-Runtime Select

Creates cones from CAD surfaces selected by the user at runtime.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All possible cones were created successfully. |
| FAILURE | The user pressed ESC at the prompt. |

Remarks

None.

Ellipsoids

Construct Ellipsoid

Creates a ellipsoid.

Input Arguments

| | | | |
|---|------------------------|----------------------------------|--|
| 0 | Collection Object Name | Ellipsoid Name | The name for the resulting ellipsoid. |
| 1 | Double | X-Axis Length | The length of the ellipsoid in the x-axis. |
| 2 | Double | Y-Axis Length | The length of the ellipsoid in the y-axis. |
| 3 | Double | Z-Axis Length | The length of the ellipsoid in the z-axis. |
| 3 | Double | Magnification | Magnification scaling |
| 3 | Boolean | Uncertainty Ellipsoid? | Uncertainty designation. |
| 4 | Transform | Transform in Working Coordinates | The transform of the ellipsoid. |
| 5 | Color | Ellipse Color | Specify the color of the ellipsoid |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Construct B-Spline From Intersection of Surfaces

Creates one or more B-Splines at the intersection of two surfaces.

Input Arguments

| | | | |
|---|------------------------|-------------------------|--|
| 0 | Collection Object Name | Resulting B-Spline Name | A name for the resulting B-spline (or a prefix, if more than one will be created). |
| 1 | Collection Object Name | First Surface Name | The name of the first surface to intersect. |
| 2 | Collection Object Name | Second Surface Name | The name of the second surface to intersect. |
| 3 | Double | Approximation Tolerance | The maximum allowable difference between the resulting B-spline and the true surface/surface intersection. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The B-Spline(s) were created successfully. |
| FAILURE | One or more of the source surfaces was not found. |

Remarks

If the resulting B-Spline names already exist, the names will be incremented automatically to avoid duplicates.

If more than one B-spline will be created, they will each have a numerical suffix in their name, starting from zero.

If the two surfaces do not intersect, no curves will be created.

B-Splines

Construct B-Spline Fit Options

Builds a Fit Options definition that can be used to construct a B-Spline that passes through a series of 3 or more points.

Input Arguments

| | | | |
|----|-------------------------------|----------------------------|---|
| 0 | Boolean | Open Curve? | True builds and open spline |
| 1 | Boolean | Use Interpolation for Fit? | True Interpolates between points |
| 2 | Integer | Number of Control Points | Number of control points to use in the fit |
| 3 | Integer | Degree of Curve | Minimum degree of curvature to allow in the fit |
| 4 | B-Spline Point Sort Mode Type | Sort Method | Choose from the available options |
| 5 | Boolean | Span Any Gap? | True spans gaps of any size |
| 6 | Double | Termination Gap Length | Gap width to use for termination |
| 7 | Boolean | Ignore Proximate Points? | True ignores proximate points |
| 8 | Double | Proximate Point Threshold | Proximate Point threshold to apply |
| 9 | Boolean | Use Global Tesselations | True applies global tessellation, false uses the following chordal and angular values |
| 10 | Double | Maximum Choral Deviation | Chordal deviation to apply |
| 11 | Double | Maximum Trim Edge Angle | Trim Edge Angle to use |

Return Arguments

| | | | |
|----|----------------------|----------------------|--|
| 12 | B-spline Fit Options | B-Spline Fit Options | The resulting fit options definition for reference |
|----|----------------------|----------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Spline was created successfully. |
| FAILURE | The instrument or one or more of the source points was not found, or less than three points were provided. |

Remarks

If the specified B-Spline name already exists, the name will automatically increment to avoid duplicates.

Construct B-Spline From Points

Constructs a B-Spline that passes through a series of 3 or more points.

Input Arguments

| | | | |
|---|------------------------|-------------------------|--|
| 0 | Collection Object Name | Resulting B-Spline Name | The name for the resulting B-Spline. |
| 1 | B-Spline Fit Options | B-Spline Fit Options | User Settings defining how the curve is to be fit to the specified points. |
| 2 | Point Name Ref List | Point List | The points defining the B-Spline. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Spline was created successfully. |
| FAILURE | The instrument or one or more of the source points was not found, or less than three points were provided. |

Remarks

If the specified B-Spline name already exists, the name will automatically increment to avoid duplicates.

The B-Spline Fit Options can reference a pre-defined set.

Construct B-Spline From Point Sets

Constructs a B-Spline that passes through a series of 3 or more point from within a Point Set.

Input Arguments

| | | | |
|---|------------------------|-------------------------|--|
| 0 | Collection Object Name | Resulting B-Spline Name | The name for the resulting B-Spline. |
| 1 | B-Spline Fit Options | B-Spline Fit Options | User Settings defining how the curve is to be fit to the specified points. |
| 2 | Collection Object Name | Point Set Container | The Point Set used to define the B-Spline. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Spline was created successfully. |
| FAILURE | The Instrument or Point Set could not be found, or no solution could be found. |

Remarks

If the specified B-Spline name already exists, the name will be incremented automatically to avoid duplicates.

Construct B-Spline From Several B-Splines

Constructs a single B-Spline by splicing consecutive B-Splines together.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|---|
| 0 | Collection Object Name | Resulting B-Spline Name | The name for the resulting B-Spline. |
| 1 | Collection Object Name Ref List | B-Spline List | A list of the source B-Splines to connect. |
| 2 | Boolean | Close Resulting B-Spline | Indicate whether the resulting B-Spline should be closed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Spline was created successfully. |
| FAILURE | One or more of the source B-Splines was not found. |

Remarks

If the specified B-Spline name already exists, the name will be incremented automatically to avoid duplicates.

Construct B-Splines From Surfaces

Builds the defining B-splines from a list of surfaces

Input Arguments

| | | | |
|---|---------------------------------|--|--|
| 0 | String | Resulting BSpline Name prefix (optional) | An optional prefix for the resulting B-spline names. |
| 1 | Collection Object Name Ref List | Surface List | A list of the source surfaces to convert. |

Return Arguments

| | | | |
|---|---------------------------------|---------------|--|
| 2 | Collection Object Name Ref List | B-Spline List | A list of the resulting B-splines that were created. |
|---|---------------------------------|---------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The B-Splines were created successfully. |
| FAILURE | One or more of the source surfaces was not found. |

Remarks

If the resulting B-Spline names already exist, the names will be incremented automatically to avoid duplicates.

Construct B-Splines From Lines

Converts one or more lines into B-splines.

Input Arguments

| | | | |
|---|---------------------------------|--|--|
| 0 | String | Resulting BSpline Name prefix (optional) | An optional prefix for the resulting B-spline names. |
| 1 | Collection Object Name Ref List | Line List | A list of the source lines to convert. |

Return Arguments

| | | | |
|---|---------------------------------|---------------|--|
| 2 | Collection Object Name Ref List | B-Spline List | A list of the resulting B-splines that were created. |
|---|---------------------------------|---------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Splines were created successfully. |
| FAILURE | One or more of the source lines was not found. |

Remarks

If the resulting B-Spline names already exist, the names will be incremented automatically to avoid duplicates.

The resulting lines will have the same name as their source line, with the optional prefix.

Construct B-Spline From Intersection of Plane and Surface

Creates one or more B-Splines at the intersection of a plane and surface.

Input Arguments

| | | | |
|---|------------------------|-------------------------|--|
| 0 | Collection Object Name | Resulting B-Spline Name | A name for the resulting B-spline (or a prefix, if more than one will be created). |
| 1 | Collection Object Name | Plane Name | The name of the source plane. |
| 2 | Collection Object Name | Surface Name | The name of the source surface. |
| 3 | Double | Approximation Tolerance | The maximum allowable difference between the resulting B-spline and the true plane/surface intersection. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Splines were created successfully. |
| FAILURE | The source plane or surface was not found. |

Remarks

If the resulting B-Spline names already exist, the names will be incremented automatically to avoid duplicates.

If more than one B-spline will be created, they will each have a numerical suffix in their name, starting from zero.

If the plane and surface do not intersect, no curves will be created.

Construct B-Spline From Intersection of Plane and Mesh

Creates one or more B-Splines at the intersection of a plane and a Mesh.

Input Arguments

| | | | |
|---|------------------------|---|---|
| 0 | Collection Object Name | Resulting B-Spline Name | A name for the resulting B-spline (or a prefix, if more than one will be created). |
| 1 | Collection Object Name | Plane Name | The name of the source plane. |
| 2 | Collection Object Name | Mesh Name | The name of the source surface. |
| 3 | Integer | Delete closed lines whose number of segment is less than this value | The integer entered is the minimum number of segments (or poly-lines) necessary to keep a closed curve. |
| 4 | Integer | Delete unclosed lines whose number of segment is less than this value | The integer entered is the minimum number of segments (or poly-lines) necessary to keep a curve. |
| 5 | Boolean | Create Intersection Points? | True will build a points cloud with the intersection points. |

Return Arguments

| | | | |
|---|---------------------------------|---------------|-------------------------------|
| 6 | Collection Object Name Ref List | B-Spline List | List of newly created curves. |
|---|---------------------------------|---------------|-------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The B-Splines were created successfully. |
| FAILURE | The source plane or surface was not found. |

Remarks

If the resulting B-Spline names already exist, the names will be incremented automatically to avoid duplicates.

If more than one B-spline will be created, they will each have a numerical suffix in their name, starting from zero.

If the mesh and surface do not intersect, no curves will be created.

Surfaces

Construct Surfaces From Objects

Creates surfaces from one or more source objects.

Input Arguments

| | | | |
|---|---------------------------------|---------|--|
| 0 | Collection Object Name Ref List | Objects | The objects from which to create surfaces. |
|---|---------------------------------|---------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The surfaces were created successfully. |
| FAILURE | No source objects were found. |

Remarks

None.

Construct Surface From B-Splines

Creates a surface from a series of at least 4 B-Splines.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|--|
| 0 | Collection Object Name | Resulting Surface Name | A name for the resulting surface. |
| 1 | Collection Object Name Ref List | BSpline List | A list of the source B-Splines to use. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The surface was created successfully. |
| FAILURE | One or more of the source B-Splines was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface From Cylinder

Creates a surface from a cylinder.

Input Arguments

| | | | |
|---|------------------------|------------------------|---|
| 0 | Collection Object Name | Resulting Surface Name | A name for the resulting surface. |
| 1 | Collection Object Name | Cylinder Name | The name of the source cylinder. |
| 2 | Boolean | Internal Cylinder? | Indicates whether the cylinder should have normals facing inward. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was created successfully. |
| FAILURE | The source cylinder was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface From Plane

Creates a surface from a plane.

Input Arguments

| | | | |
|---|------------------------|------------------------|-----------------------------------|
| 0 | Collection Object Name | Resulting Surface Name | A name for the resulting surface. |
| 1 | Collection Object Name | Plane Name | The name of the source plane. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was created successfully. |
| FAILURE | The source plane was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface From Sphere

Creates a surface from a sphere.

Input Arguments

| | | | |
|---|------------------------|------------------------|-----------------------------------|
| 0 | Collection Object Name | Resulting Surface Name | A name for the resulting surface. |
| 1 | Collection Object Name | Sphere Name | The name of the source sphere. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was created successfully. |
| FAILURE | The source sphere was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface From Cone

Creates a surface from a cone.

Input Arguments

| | | | |
|---|------------------------|------------------------|-----------------------------------|
| 0 | Collection Object Name | Resulting Surface Name | A name for the resulting surface. |
| 1 | Collection Object Name | Cone Name | The name of the source cone. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was created successfully. |
| FAILURE | The source cone was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface From a Collection of Surfaces

Creates a surface from a selection of existing surfaces.

Input Arguments

| | | | |
|---|---------------------------------|---------------------------|---|
| 0 | Collection Object Name Ref List | Surfaces to Combine | A list of the surfaces to be combined. |
| 1 | Collection Object Name | Resulting Surface Name | Name to be applied to the newly created surface. |
| 2 | Boolean | Hide Original Surfaces? | True would result in the original surfaces being hidden. |
| 3 | Boolean | Delete Original Surfaces? | True would result in the original surfaces being deleted from the job file. |
| 4 | Boolean | Enable Sewing Tolerance? | True would enable sewing. |
| 5 | Double | Sewing Tolerance | Sewing tolerance used. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was created successfully. |
| FAILURE | The source cone was not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates. Sewing when selected, allows adjacent surface faces which lack tangency (they do not smoothly flow from one surface face to another) will be stitched together in accordance with the sewing tolerance. The tolerance controls the maximum normal deviation from one surface face to the next in which those adjacent surface faces will be stitched.

Construct Surface Fit From Nominal Surfaces and Actual Data

Creates a surface fit to actual data, using nominal surfaces as a “guide” or starting point.

Input Arguments

| | | | |
|---|------------------------|------------------------|---|
| 0 | Collection Object Name | Nominal Surface List | The nominal surface(s). |
| 1 | Point Name Ref List | Actual Data Point List | A list of actual points. |
| 2 | Collection Object Name | Resulting Surface Name | The name for the resulting surface to create. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The surface was created successfully. |
| FAILURE | The nominal surface, or one or more actual points were not found. |

Remarks

If the resulting surface name already exists, the name will be incremented automatically to avoid duplicates.

Construct Surface by Dissecting Surface(s)

Functioning as a Runtime-Select command, this step prompts the user to select either entire surfaces or just surface faces and builds a set of new surfaces from the selected.

Input Arguments

| | | | |
|---|------------------------------|-----------------|--|
| 0 | Surface Dissection Mode Type | Dissection Mode | Specifies whether the selected surfaces should be dissected entirely, or whether the selected faces should be extracted (dissected) from the surfaces. |
|---|------------------------------|-----------------|--|

Return Arguments

| | | | |
|---|---------------------------------|-------------------------|---|
| 1 | Collection Object Name Ref List | Resulting Surfaces List | Returns a list of the newly created surfaces. |
|---|---------------------------------|-------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Construct Surfaces by Dissecting Surfaces from Ref List

This function takes a list of surfaces and breaks them down into new objects build from the surface faces of the reference objects.

Input Arguments

| | | | |
|---|---------------------------------|---------------------|-----------------------------------|
| 0 | Collection Object Name Ref List | Surfaces to Dissect | The original surfaces to dissect. |
|---|---------------------------------|---------------------|-----------------------------------|

Return Arguments

| | | | |
|---|---------------------------------|-------------------------|-------------------------------------|
| 1 | Collection Object Name Ref List | Resulting Surfaces List | A list of the newly built surfaces. |
|---|---------------------------------|-------------------------|-------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Use caution with larger CAD files. This command can build a lot of surfaces. Using Construct Sufaces By Projecting Points can be a handy alternative because it is much more specific to the faces of interest.

Construct Surface From Point Groups

Fits a surface to point groups, ordered in rows on a surface. Equivalent to Construct > Surfaces > From Point Groups.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|--|
| 0 | Collection Object Name Ref List | Group Name List | The list of point groups defining the surface, one for each "row" or cross-section on the surface. |
| 1 | B-Spline Fit Options | B-Spline Fit Options | The options for B-Splining each group of points |
| 2 | Collection Object Name | Resulting Surface Name | The name of the resulting surface. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The surface was created successfully. |
| FAILURE | Less than four point groups were supplied, or one or more groups could not be found. |

Remarks

At least four point groups are needed for this command to succeed.

Construct Surfaces By Projecting Points

This function constructs a surface by projecting the selected points to the selected reference surfaces, identifying individual faces from those surfaces and building a new object from a copy of those surface faces.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|-----------------------------|
| 0 | Collection Object Name Ref List | Projection Target Name List | List of Objects to consider |
| 1 | Point Name Ref List | Point List | List of Points to project |
| 2 | Collection Object Name | Resulting Surface Name | Resulting object name |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface was build successfully. |
| FAILURE | Reference surface could not be found. |

Remarks

None.

Construct Surface by offsetting a surface

Fits a surface to point groups, ordered in rows on a surface. Equivalent to Construct > Surfaces > From Point Groups.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|---|
| 0 | Collection Object Name Ref List | Reference Surface | The original surface that will getting offset. |
| 1 | Double | Surface Offset | The amount to offset the surface. |
| 2 | Boolean | Hide Original Surface? | Hide the reference surface after offset surface is created. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface offset successfully. |
| FAILURE | Reference surface could not be found. |

Remarks

None.

Construct Surface from Annotation Links

Builds a new surface from the list of associated CAD faces referenced in the selected GD&T Annotations.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|--|
| 0 | Collection Object Name Ref List | Annotation List | The GD&T Annotations used to define the CAD face references used for surface construction. |
| 1 | Collection Object Name | Resulting Surface Name | The name used for the newly built surface. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The surface as built successfully |
| FAILURE | Reference surface could not be found. |

Remarks

None.

Construct Geometry From Surfaces

Constructs lines, circles, and cylinders from surface's cylinder faces within specified diameter constraints..

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name Ref List | Surfaces | The surfaces used to create geometry. |
| 1 | Double | Minimum Diameter | The minimum diameter of surface cylinder. |
| 2 | Double | Maximum Diameter | The maximum diameter of surface cylinder. |
| 3 | Geometry Mode Type | Geometry Mode | Choose between line, circle or cylinder. |
| 4 | Collection Object Name | Reference Frame | Designate a frame of reference. |
| 5 | Collection Name | Destination Collection Name | Name of collection where created geometries will reside. |
| 6 | String | Base Name | The prefix of the name of each created geometry. |

Return Arguments

| | | | |
|---|---------------------------------|------------------|--------------------------------------|
| 7 | Collection Object Name Ref List | Geometry Objects | The resultant geometries are created |
|---|---------------------------------|------------------|--------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The geometries are created successfully. |
| FAILURE | Reference surface could not be found or no geometry found within diameter constraints. |

Remarks

This will construct lines, circles, and cylinders from surface cylinder faces within specified diameter constraints.

Polygonized Surfaces

Construct Polygonized Surface from Point Clouds

Creates a polygonized surface (mesh) from a set of input point clouds. This command is the MP equivalent to the *Construct*►*Polygonized Mesh*►*From Point Clouds* menu command.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------|--|
| 0 | Collection Object Name Ref List | Point Cloud List | The list of input point clouds. |
| 1 | Mesh Orientation Type | Mesh Orientation | Specifies whether the current point of view or the current working frame's Z axis is used for orienting the mesh. |
| 2 | Double | Grid Resolution | The resulting resolution for the generated mesh. Smaller resolutions have higher detail but require more dense measurement data. |
| 3 | Collection Object Name | Polygonized Surface Name | The name for the resulting polygonized surface. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The polygonized mesh was created successfully. |
| PARTIAL SUCCESS | One or more point clouds was not found |
| FAILURE | No supplied point clouds were found. |

Remarks

If the specified polygonized surface already exists, a unique name will be selected and inserted into Argument 3.

Frames

Construct Frame with Wizard

Opens the frame wizard with a predefined frame name.

Input Arguments

| 0 | Collection Object Name | New Frame Name | The name for the newly constructed frame. |
|---|------------------------|---------------------|---|
| 1 | Boolean | Wait for Completion | Indicates whether the MP should pause until the frame wizard is closed. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | The wizard was closed or a frame was not created. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates. If the Wait for Completion argument is set to FALSE, the command will always succeed.

Construct Frame

Creates a frame based on a given transform.

Input Arguments

| 0 | Collection Object Name | New Frame Name | The name for the newly constructed frame. |
|---|------------------------|----------------------------------|---|
| 1 | Transform | Transform in Working Coordinates | A transform defining the position and orientation of the new frame. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame on Instrument Base

Creates a frame on an instrument's base.

Input Arguments

| | | | |
|---|------------|-----------------------|--|
| 0 | Inst ID | Instrument ID | The instrument ID number for the instrument. |
| 1 | Frame Name | Frame Name (Optional) | An optional name for the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | The specified instrument was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates. The frame will be created in the active collection. Note that since only an instrument ID (integer index) is supplied, the instrument must be in the active collection at the time the command is executed.

Construct Frame On Object

Creates a frame on an object's local coordinate system.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | Reference Object | The object defining the origin and orientation of the resulting frame. |
| 1 | Collection Object Name | Frame Name (Optional) | An optional name for the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The frame was created successfully. |
| FAILURE | The specified object was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame, 3 Points

Creates a frame defined by three points: an origin, point along an axis, and clocking point.

Input Arguments

| | | | |
|---|-------------------------------|-----------------------|--|
| 0 | 3Pt Frame Construction Method | Construction Method | The method by which the frame will be constructed. |
| 1 | Point Name | Origin Point | The point defining the origin of the frame. |
| 2 | Point Name | Primary Axis Point | A point lying on the primary axis. |
| 3 | Point Name | Secondary Axis Point | A point to which the secondary axis is clocked. |
| 4 | Frame Name | Frame Name (Optional) | The name of the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | One or more of the specified points could not be found or were invalid. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame, at Point, with working Z, and clocked axis

Creates a frame at a specified point whose primary axis is parallel to the working frame's Z axis, and whose secondary (clocked) axis is defined by a specified point.

Input Arguments

| | | | |
|---|-----------------|-----------------------|--|
| 0 | Point Name | Origin Point | The point defining the origin of the frame. |
| 1 | Axis Identifier | Clocked axis | The axis to treat as the secondary (clocked) axis. |
| 2 | Point Name | Clocking Point | The point to which the resulting frame's secondary axis should be clocked. |
| 3 | Frame Name | Frame Name (Optional) | The name of the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | The origin or clocking point was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame, Pick origin and point on X axis - clock Z along working Z

Creates a frame at a specified point whose X axis passes through a point, and whose Z axis is clocked along the working frame's Z axis.

Input Arguments

| | | | |
|---|------------|-----------------------|---|
| 0 | Point Name | Origin Point | The point defining the origin of the frame. |
| 1 | Point Name | Point on X-Axis | The point through which the X-axis should pass. |
| 2 | Frame Name | Frame Name (Optional) | The name of the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | The origin or clocking point was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame, Known Origin, Object Direction, Object Direction

Creates a frame at a specified point whose primary and secondary axes are defined by objects.

Input Arguments

| | | | |
|---|------------------------|-----------------------------------|---|
| 0 | Point Name | Known Point | A known point for the origin (or offset from the origin). |
| 1 | Vector | Known Point Value in New Frame | The coordinate of the known point in the resulting coordinate frame. |
| 2 | Collection Object Name | Primary Axis Object | The object whose axis defines the primary axis of the resulting frame. |
| 3 | Axis Identifier | Primary Axis Defines Which Axis | The axis which is considered primary. |
| 4 | Collection Object Name | Secondary Axis Object | The object whose axis defines the secondary (clocking) axis of the resulting frame. |
| 5 | Axis Identifier | Secondary Axis Defines Which Axis | The axis which is considered secondary. |
| 6 | Collection Object Name | Frame Name (Optional) | The name of the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | The known point or primary/secondary axis object could not be found, or an invalid orientation was given. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame, 3 Planes

Creates a frame based on the orientation of 3 construction planes.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | X Plane | The plane defining the primary (X) axis. |
| 1 | Double | X Value on PLane | The X value of the plane in the resulting frame. |
| 2 | Collection Object Name | Y Plane | The plane defining the secondary (Y) axis. |
| 3 | Double | Y Value on PLane | The Y value of the plane in the resulting frame. |
| 4 | Collection Object Name | Z Plane | The plane defining the tertiary (Z) axis. |
| 5 | Double | Z Value on Plane | The Z value of the plane in the resulting frame. |
| 6 | Collection Object Name | Frame Name (Optional) | The name of the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The frame was created successfully. |
| FAILURE | One or more of the construction planes was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

This command is intended to be used with orthogonal planes. Non-orthogonal planes will give a solution, although the results may not be what you expect.

Construct Frame - Copy And Make Left Handed

Copies a right-handed frame and makes it left-handed, reversing either the X, Y, or Z axis.

Input Arguments

| | | | |
|---|------------------------|-----------------------|---|
| 0 | Collection Object Name | Reference Frame | The frame to copy and make left-handed. |
| 1 | Collection Object Name | Frame Name (Optional) | The name for the new left-handed frame. |
| 2 | Axis Identifier | Axis to reverse | The axis on the reference frame to reverse. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The frame was created successfully. |
| FAILURE | The reference frame was not found. |

Remarks

If the resulting frame name already exists, the name will be incremented automatically to avoid duplicates.

Construct Frame - Average of Other Object Frames

Creates a frame representing the average position and orientation of other object orientations.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------|---|
| 0 | Collection Object Name Ref List | Objects | The list of objects specifying the frames to average. |
| 1 | Collection Object Name | Frame Name (Optional) | The name for the resulting averaged frame. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The frame was created successfully. |
| FAILURE | Less than two source objects were found. |

Remarks

None.

Construct Frame at Robot Link

Creates a frame at a robot linkage, oriented along the linkage.

Input Arguments

| | | | |
|---|------------------------|-----------------|--|
| 0 | Collection Machine ID | Machine ID | The Machine ID of the robot in question. |
| 1 | String | Link Name | The name of the link. |
| 2 | Collection Object Name | Resulting Frame | The name for the resulting frame. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The frame was created successfully. |
| FAILURE | The machine or link was not found. |

Remarks

None.

Construct Frame from Point Measurement Probing Frames

Constructs probing frames (signifying *ijk* vectors) for point measurements that have this information.

Input Arguments

| | | | |
|---|---------------------|-------------|--|
| 0 | Point Name Ref List | Point List | The list of points for which probing frames should be created (if possible). |
| 1 | Boolean | Show Frame? | If TRUE, the frame will be shown. If FALSE, the frame will be hidden. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The frames were created successfully. |
| FAILURE | The specified points do not have probing frames or were not found. |

Remarks

None.

Construct Mirror Cube Frame

Input Arguments

| | | | |
|---|------------------------|--|---|
| 0 | Collection Object Name | Mirror Cube Frame Name | Name of the resulting Mirror Cube Frame |
| 1 | Point Name | Point Name | Reference Point Name |
| 2 | Boolean | Use Current Measurements Marked as Mirror Shots | True enables marked shots |
| 3 | Double | Nominal Cube Face Angle | Nominal Reference angle |

Return Arguments

| | | | |
|---|--------|---------------------|--|
| 4 | Double | Total Angular Error | |
|---|--------|---------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The frames were created successfully. |
| FAILURE | The specified points do not have probing frames or were not found. |

Remarks

None.

Perimeters

Construct Perimeter From Points

Creates a perimeter from a set of points.

Input Arguments

| | | | |
|---|---------------------|--------------------------|--|
| 0 | Perimeter Name | Resulting Perimeter Name | The name of the perimeter to create. |
| 1 | Point Name Ref List | Point List | A list of points that consecutively trace the perimeter. |
| 2 | Boolean | Open Perimeter? | Specify whether the perimeter should be open or closed. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The perimeter was created successfully. |
| FAILURE | One or more of the specified points could not be found, or less than two points were specified. |

Remarks

If the resulting perimeter name already exists, the name will be incremented automatically to avoid duplicates.

Vector Groups

Construct a Vector Group - Group to Group Compare

Creates a vector group comparing two point groups.

Input Arguments

| | | | |
|---|------------------------|---|--|
| 0 | Vector Group Name | Vector Group Name | The name of the vector group to create. |
| 1 | Collection Object Name | Group A | The first point group in the comparison. |
| 2 | Collection Object Name | Group B | The second point group in the comparison. |
| 4 | Double | RMS Deviation Tolerance (0.0 for none) | A tolerance for the RMS deviation of the vector group. |
| 5 | Double | Max Absolute Deviation Tolerance (0.0 for none) | A tolerance for the maximum absolute deviation for the vector group. |
| 6 | Double | Average Deviation Tolerance (0.0 for none) | The tolerance for the average deviation of the vector group. |

Return Arguments

| | | | |
|---|---------|------------------------|--|
| 3 | Integer | Vector Count | The number of vectors created. |
| 7 | Double | RMS Deviation | The actual RMS Deviation of the vector group. |
| 8 | Double | Max Absolute Deviation | The actual maximum absolute deviation of the vector group. |
| 9 | Double | Average Deviation | The actual average deviation of the vector group. |

Returned Status

| | |
|-----------------|---|
| SUCCESS | The vector group was created successfully. |
| PARTIAL SUCCESS | The vector group was created successfully, but one or more specified tolerances failed. |
| FAILURE | One or both point groups could not be found. |

Remarks

If the resulting vector group already exists, the name will be incremented automatically to avoid duplicates.

Construct a Vector Group - Area Profile Check

Compares vectors to a set of nominal vectors. Each sampled vector that lies within a specified radius of a nominal vector is compared. The magnitude difference between the sampled vector and the nominal (sampled minus nominal) is set as the resulting vector magnitude, and the direction is inherited by the sampled vector (keeping in mind that negative magnitudes will flip the vector in the opposite direction). A specified area tolerance is applied to the resulting vectors. The tolerance is specified as a band width. That is, a tolerance of 0.010” implies +/- 0.005”.

Input Arguments

| | | | |
|---|---------------------------------------|-----------------------------|--|
| 0 | Vector Name Ref List | Reference Vectors | A set of reference vector names. Note that these are vector names, not vector group names. |
| 1 | Collection Vector Group Name Ref List | Vector Groups to Check | A set of one or more vector groups that you'd like to check. |
| 2 | Double | Area Radius | The radius within each reference vector to search for a measured vector to check. |
| 3 | Double | Area Tolerance | The width of the tolerance band to apply to the resulting vectors. |
| 4 | Collection Vector Group Name | Resultant Vector Group Name | The name for the resulting vector group. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The vector group was created successfully. |
| PARTIAL SUCCESS | At least one reference vector (but not all) were not found. |
| FAILURE | No reference vectors were found, or no sampled vector groups were found. |

Remarks

If no sampled vectors lie within the specified radius of the nominal vectors, an empty vector group will be created.

Construct a Vector Group From Vector Name Ref List

Creates a vector group from a list of vectors.

Input Arguments

| | | | |
|---|------------------------------|-----------------------------|---|
| 0 | Vector Name Ref List | Vector Name List | The name of the list of vectors that will be used to create the vector group. |
| 1 | Collection Vector Group Name | Resultant Vector Name Group | The name for the vector group to create. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | This command always succeeds. |
| FAILURE | The vector name ref list could not be found. |

Remarks

None.

Construct a Vector in Working Coordinates (Begin/Delta)

Creates a vector in working coordinates. The vector is defined by a starting location and delta value.

Input Arguments

| | | | |
|---|-------------------|---------------------------------|--|
| 0 | Vector Group Name | Vector Group Name | The name of the vector group into which to place the resulting vector, in the active collection. |
| 1 | String | New Vector Name | The name for the vector to create. |
| 2 | Vector | 'Begin' in Working Coordinates. | The location for the beginning of the vector. |
| 3 | Vector | 'Delta' in Working Coordinates. | The delta value for the vector. |
| 4 | Boolean | Is Magnitude Negative | Specifies whether the resulting vector has a negative magnitude. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the specified vector already exists, another will be created with the same name.

Construct a Vector in Working Coordinates (Begin/Direction/Mag.)

Creates a vector in working coordinates. The vector is defined by a starting location, direction, and magnitude.

Input Arguments

| | | | |
|---|-------------------|-------------------------------------|--|
| 0 | Vector Group Name | Vector Group Name | The name of the vector group into which to place the resulting vector, in the active collection. |
| 1 | String | New Vector Name | The name for the vector to create. |
| 2 | Vector | 'Begin' in Working Coordinates. | The location for the beginning of the vector. |
| 3 | Vector | 'Direction' in Working Coordinates. | A vector defining the direction of the vector to create. |
| 4 | Double | Signed Magnitude | The signed magnitude of the resulting vector. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the specified vector already exists, another will be created with the same name.

Construct Vectors WildCard Selection

Creates a set of vectors from one or more source vector groups that match a specified name pattern.

Input Arguments

| | | | |
|---|---------------------------------|------------------------------|---|
| 0 | Collection Object Name Ref List | Vector Groups to Select From | The source vector groups to search. |
| 1 | Vector Name | WildCard Selection Names | A wildcard name for the vectors to match. |
| 2 | Collection Object Name | Vector Group for New Vectors | The name of the vector group for the matching vectors. |
| 3 | Boolean | Include Prior Complete Name | Specify whether the complete name of each vector should be used as the new vector name. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The matching vectors were created successfully. |
| FAILURE | No source vector groups were provided. |

Remarks

Argument 1 can reference a point name. Use *Make a Point Name from Strings* to manually enter the search criteria for the collection, vector group and point name wildcard.

Enter wildcard values for the collection, vector group, and vector name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets []. To find all vectors from all collections in vector groups that start with “s” and whose vector names have two digits starting with “1”, the vector name defining the selection criteria would be `*::s*::1?`.

Construct a Vector Group From a Relationship

Constructs a vector group from a relationship. Equivalent to right-clicking a relationship and selecting Make Vector Group (static).

Input Arguments

| | | | |
|---|------------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the source relationship. |
| 1 | Collection Vector Group Name | Vector Group Name | The name for the resulting vector group. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The resulting vector group was created successfully. |
| FAILURE | The source relationship was not found. |

Remarks

If the destination vector group already exists, the name will be incremented to avoid duplicates.

Make Surface Face List - Runtime Select

Creates a list of surface faces that can be referenced by an Annotation.

Input Arguments

None.

Return Arguments

| | | | |
|---|---------------|------------------------|-----------------------------|
| 0 | Surface Faces | Selected Surface Faces | The returned list of faces. |
|---|---------------|------------------------|-----------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | List of Faces were created successfully. |
| FAILURE | No faces were selected. |

Remarks

None.

Make GD&T Feature Check Annotation

Creates a GD&T Feature Check Annotation.

Input Arguments

| | | | |
|----|---------------------------------|--------------------------------------|--|
| 0 | String | Feature Annotation Name | Name of the new annotation to create |
| 1 | GD&T Feature Type | Feature Type | Pick from the list of available feature check types. |
| 2 | Collection Object Name Ref List | Objects | List of reference objects to check |
| 3 | Relationship Ref List | Geometry Relationships | List of GR-Features to check |
| 4 | Surface Faces | Surface Faces | List of Surface Faces to reference for the check |
| 5 | Boolean | Decompose Multiple Features | True will add a single check annotation to each included object selected |
| 6 | Boolean | Auto Create Diameter Checks? | True builds a diameter check as well as the selected feature check, used with position checks using modifiers. |
| 7 | Boolean | Auto Create Slot Width Checks? | True builds slot width checks, used with position checks using modifiers. |
| 8 | Boolean | Auto Create Slot Length Check? | True builds slot Length checks, used with position checks using modifiers. |
| 9 | String | Datum References | The datums to assign for the Feature Check |
| 10 | String | Tolerance | The tolerance to apply |
| 11 | Boolean | Is Slot? | True indicates the feature is a slot |
| 12 | Boolean | Per unit length/area | True applied per unit length to the evaluation |
| 13 | Boolean | Circular area? (Rectangular default) | True sets the per unit evaluation to use a circular area |
| 14 | Double | Per unit (area) Length distance | Length distance value to use |
| 15 | Double | Per unit (area) Length step over % | Percentage overlap between evaluation zones |
| 16 | Double | Per unit (area) width distance | Width distance value to use |
| 17 | Double | Per unit (area) width step over % | Percentage overlap between evaluation zones |
| 18 | Double | Per unit area circle diameter | Diameter value to use |
| 19 | Double | Per unit area diameter step over | Diametric step overlap |
| 20 | Collection Object Name | Auxiliary Object | Reference Object |
| 21 | Collection Object Name | Auxiliary Geometry Relationship | Reference Geometry Relationship |
| 22 | Boolean | Use Nominal for Dimension Tolerance | True uses the nominal as opposed to a range designation for tolerances |
| 23 | Boolean | Use Reference Object for Nominal | True pulls the nominal value from the selected objects |
| 24 | Double | Nominal Dimension Tolerance | Nominal dimension value to apply |
| 25 | Double | Low Dimension Tolerance | Low tolerance value to apply |
| 26 | Double | High Dimension Tolerance | High tolerance value to apply |
| 27 | GD&T Zone Type | Tolerance Zone Type | Pick a tolerance type to apply |
| 28 | Boolean | Use Projected Tolerance Zone | True applies a projection tolerance zone |
| 29 | Double | Projected Tolerance Zone | Projection value |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Any feature checks annotations created successfully. |
| FAILURE | The specified information was incomplete |

Remarks

This command effectively reproduces the capabilities of the GD&T Toolkit for annotation feature check construction.

Make GD&T Datum Annotation

Creates a GD&T datum annotation

Input Arguments

| | | | |
|---|---------------------------------|---------------------------------|--|
| 0 | String | Datum Name | Name of the datum to construct |
| 1 | Collection Object Name Ref List | Objects | List of reference objects to include |
| 2 | Relationship Ref List | Geometry Relationships | List of GR-Features to include |
| 3 | Surface Faces | Surface Faces | List of Surface Faces to reference for the include |
| 4 | Collection Object Name | Auxiliary Object | Reference Object |
| 5 | Collection Object Name | Auxiliary Geometry Relationship | Reference Geometry Relationship |
| 6 | Boolean | Is Slot? | True indicates the feature is a slot |
| 7 | Boolean | Force Surface Feature | True identifies the selected feature as a surface feature. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Any datum annotations were created successfully. |
| FAILURE | The specified information was incomplete |

Remarks

This command effectively reproduces the capabilities of the GD&T Toolkit for annotation datum construction.

Make Feature Checks

Creates the Feature Checks for a specified collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Collection Name | Collection Name | The collection containing datums/annotations for which the feature checks will be created. |
|---|-----------------|-----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Any feature checks were created successfully. |
| FAILURE | The specified collection does not exist. |

Remarks

None.

Make Annotation Ref List from a Collection

Makes a list of GD&T Annotations in a specified collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Collection Name | Collection Name | The collection containing the Annotations. |
|---|-----------------|-----------------|--|

Return Arguments

| | | | |
|---|------------------------|------------------------|--|
| 1 | Feature Check Ref List | Feature Check Ref List | The list of Annotations in the specified collection. |
|---|------------------------|------------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The list was generated successfully. |
| FAILURE | The specified collection was not found. |

Remarks

None.

Make Annotation Ref List - WildCard Selection

Creates a list of GD&T Annotations based on wildcard selection criteria.

Input Arguments

| | | | |
|---|--------|---------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Feature Check Wildcard Criteria | The wildcard string to specify the Annotations to select. |

Return Arguments

| | | | |
|---|------------------------|-------------------------------------|---|
| 2 | Feature Check Ref List | Resultant Annotation Reference List | The resulting list of Annotations matching the specified wildcard selection criteria. |
|---|------------------------|-------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The matching list of Annotations was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the Annotations in the list is undefined.

Enter wildcard values for the collection and Annotation using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all feature checks that start with "s" from all collections, use a collection criteria of * and an Annotation wildcard criteria of s*.

Make a Feature Check Ref List from a Collection

Makes a list of feature checks in a specified collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|---|
| 0 | Collection Name | Collection Name | The collection containing the feature checks. |
|---|-----------------|-----------------|---|

Return Arguments

| | | | |
|---|------------------------|------------------------|---|
| 1 | Feature Check Ref List | Feature Check Ref List | The list of feature checks in the specified collection. |
|---|------------------------|------------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The list was generated successfully. |
| FAILURE | The specified collection was not found. |

Remarks

None.

Make a Feature Check Reference List - WildCard Selection

Creates a list of feature checks based on wildcard selection criteria.

Input Arguments

| | | | |
|---|--------|---------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Feature Check Wildcard Criteria | The wildcard string to specify the feature checks to select. |

Return Arguments

| | | | |
|---|------------------------|--|--|
| 2 | Feature Check Ref List | Resultant Feature Check Reference List | The resulting list of feature checks matching the specified wildcard selection criteria. |
|---|------------------------|--|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The matching list of feature checks was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the feature checks in the list is undefined.

Enter wildcard values for the collection and feature check using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all feature checks that start with "s" from all collections, use a collection criteria of * and a feature check wildcard criteria of s*.

Delete Feature Checks

Deletes a list of GD&T feature checks.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---------------------------------------|
| 0 | Feature Check Ref List | Feature Check Name List | The list of feature checks to delete. |
|---|------------------------|-------------------------|---------------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The matching list of feature checks was created successfully. |
| PARTIAL SUCCESS | At least one feature check (but not all) was not found. |
| FAILURE | No valid feature check was found, or the list was empty. |

Remarks

None.

Make a Datum Ref List from a Collection

Builds a list of GD&T datums from all datums in a collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|---|
| 0 | Collection Name | Collection Name | The name of the collection to consider. |
|---|-----------------|-----------------|---|

Return Arguments

| | | | |
|---|----------------|----------------|-------------------------------|
| 1 | Datum Ref List | Datum Ref List | The resulting list of datums. |
|---|----------------|----------------|-------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of datums was created successfully. |
| FAILURE | The collection was not found. |

Remarks

None.

Folders

Construct Folder(s)

Constructs one or more folders in the SA tree.

Input Arguments

| | | | |
|---|--------|-------------|---|
| 0 | String | Folder Path | The name of a folder to create, or a path to create a hierarchy of folders. |
|---|--------|-------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

A single folder is created by specifying a name only. Multiple folders in a hierarchy can be created by separating folder names by one or two colons. For example, A::B::C creates a folder “A” with a subfolder “B”, which itself has a subfolder “C”. A:B:C would have the same effect. If a folder already exists, no action will be taken.

Delete Folders by Wildcard

Deletes the set of folders in the SA tree that match the specified search string, including the wildcard characters * and ?.

Input Arguments

| | | | |
|---|---------|----------------------------|--|
| 0 | String | Search String | The wildcard string of folders to remove. The characters * and ? are permitted. |
| 1 | Boolean | Case Sensitive Search | Indicates whether the search should be case-sensitive in identifying matches. |
| 2 | Boolean | Allow Deleting all Folders | If set to <i>FALSE</i> , the command will fail if all folders match the search criteria. |

Return Arguments

| | | | |
|---|---------|-------------|--|
| 3 | Integer | Num Deleted | The number of folders that were successfully deleted. |
| 4 | Integer | Num Failed | The number of folders that matched the search criteria but could not be deleted. |

Returned Status

| | |
|---------|--|
| SUCCESS | The matching folders were successfully deleted. |
| FAILURE | Argument 2 was <i>FALSE</i> and all folders matched the search criteria. |

Remarks

Enter wildcard values for the folder name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To delete all folders that start with **s1** followed by two characters, the search criteria would be **s1??**.

Callouts

Create Vector Callout

Creates a vector callout in a callout view.

Input Arguments

| | | | |
|----|------------------------------|----------------------------------|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Collection Object Name | Vector Group Name | The name of the vector group for the callout. |
| 2 | String | Vector Name | The name of the vector to which to attach the callout. |
| 3 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: 0.75 = 75% from the left edge of the graphical view). |
| 4 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: 0.75 = 75% from the top edge of the graphical view). |
| 5 | Boolean | Show Collection? | Indicates whether the collection should be shown in the callout. |
| 6 | Boolean | Show Vector Group? | Indicates whether the vector group should be shown in the callout. |
| 7 | Boolean | Show Vector Name? | Indicates whether the vector name should be shown in the callout. |
| 8 | Boolean | Show dX? | Indicates whether the dX component should be shown in the callout. |
| 9 | Boolean | Show dY? | Indicates whether the dY component should be shown in the callout. |
| 10 | Boolean | Show dZ? | Indicates whether the dZ component should be shown in the callout. |
| 11 | Boolean | Show dMag? | Indicates whether the dMag component should be shown in the callout. |
| 12 | Boolean | Show Tolerance Color? | Indicates whether the vector tolerance color should be shown in the callout. |
| 13 | Boolean | Show Out of Tolerance Value? | Indicates whether the vector tolerance should be shown in the callout. |
| 14 | Boolean | Show Tolerance Range? | Indicates whether the vector tolerance range should be shown in the callout. |
| 15 | Boolean | Show Vector Color? | Indicates whether the vector color should be shown in the callout. |
| 16 | Boolean | Show Start Point? | Indicates whether the starting coordinate should be shown in the callout. |
| 17 | Boolean | Show End Point? | Indicates whether the ending coordinate should be shown in the callout. |
| 18 | Boolean | Show Units? | Indicates whether to show units in the callout. |
| 19 | String | Additional Note (blank for none) | Additional text to display in the callout (optional). |
| 20 | Boolean | Attach Callout to End Point? | Indicates whether to attach callout to the end point. |
| 21 | Boolean | Use Default Placement | True enables default placement. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The vector callout was successfully created. |
| FAILURE | The supplied vector or callout view was not found. |

Remarks

Argument 21- Choosing to use default placement prevents the application from attempting to choose a logical placement for the callout with respect to the objects in the view.

Create Min/Max Vector Group Callout

Creates a set of vector callouts in a callout view identifying the highest and lowest vector values.

Input Arguments

| | | | |
|----|------------------------------|---------------------------------------|--|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Collection Object Name | Vector Group Name | The name of the vector group for the callout. |
| 2 | Integer | Number of vectors with Highest Mag? | The number of high vectors to identify with callouts |
| 3 | Integer | Number of vectors with Lowest Mag? | The number of low vectors to identify with callouts |
| 4 | Boolean | Show Collection? | Indicates whether the collection should be shown in the callout. |
| 5 | Boolean | Show Vector Group? | Indicates whether the vector group should be shown in the callout. |
| 6 | Boolean | Show Vector Name? | Indicates whether the vector name should be shown in the callout. |
| 7 | Boolean | Show dX? | Indicates whether the dX component should be shown in the callout. |
| 8 | Boolean | Show dY? | Indicates whether the dY component should be shown in the callout. |
| 9 | Boolean | Show dZ? | Indicates whether the dZ component should be shown in the callout. |
| 10 | Boolean | Show dMag? | Indicates whether the dMag component should be shown in the callout. |
| 11 | Boolean | Show Tolerance Color? | Indicates whether the vector tolerance color should be shown in the callout. |
| 12 | Boolean | Tolerance color Blue(+)/Green/Red(-)? | Indicates whether the vector tolerance color Blue /Green/Red should be shown in the callout. |
| 13 | Boolean | Show Out of Tolerance Value? | Indicates whether the vector tolerance should be shown in the callout. |
| 14 | Boolean | Show Tolerance Range? | Indicates whether the vector tolerance range should be shown in the callout. |
| 15 | Boolean | Show Vector Color? | Indicates whether the vector color should be shown in the callout. |
| 16 | Boolean | Show Start Point? | Indicates whether the starting coordinate should be shown in the callout. |
| 17 | Boolean | Show End Point? | Indicates whether the ending coordinate should be shown in the callout. |
| 18 | Boolean | Show Units? | Indicates whether to show units in the callout. |
| 19 | Boolean | Attach Callout to End Point? | Indicates whether to attach callout to the end point. |
| 20 | Boolean | Use Default Placement | True enables default placement. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The vector callout was successfully created. |
| FAILURE | The supplied vector or callout view was not found. |

Remarks

Argument 2- Choosing to use default placement prevents the application from attempting to choose a logical placement for the callout with respect to the objects in the view.

Create Point Callout

Creates a point callout in a callout view.

Input Arguments

| | | | |
|----|------------------------------|---------------------------|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Point Name | Point | The name of the point with which to create a callout. |
| 2 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: 0.75 = 75% from the left edge of the graphical view). |
| 3 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: 0.75 = 75% from the top edge of the graphical view). |
| 4 | Boolean | Show Point Collection? | Choose whether to display the Collection Name |
| 5 | Boolean | Show Point Group? | Choose whether to display the Group Name |
| 6 | Boolean | Show Point Target? | Choose whether to display the Point Name |
| 7 | Boolean | Show X (R)? | Choose whether to display the X (R) value |
| 8 | Boolean | Show Y (Theta)? | Choose whether to display the Y (Theta) value |
| 9 | Boolean | Show Z (Phi)? | Choose whether to display the Z (Phi) value |
| 10 | Boolean | Show Units? | Choose whether to display units |
| 11 | Boolean | Show Ux (Ur)? | Choose whether to display the uncertainty in X |
| 12 | Boolean | Show Uy (Utheta)? | Choose whether to display the uncertainty in Y |
| 13 | Boolean | Show Uz (UPhi)? | Choose whether to display the uncertainty in Z |
| 14 | Boolean | Show UMag? | Choose whether to display the uncertainty magnitude value |
| 15 | Coordinate System Type | Desired Coordinate System | Pick from Cartesian, Spherical (polar), or Cylindrical |
| 16 | Edit Text | Notes (blank for none) | Additional general notes to display (optional). |
| 17 | Boolean | Use Default Placement | True enables default placement. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The callout was successfully created. |
| FAILURE | The supplied point or callout view was not found. |

Remarks

Argument 17- Choosing to use default placement prevents the application from attempting to choose a logical placement for the callout with respect to the objects in the view.

Create Point Comparison Callout

Creates a point comparison callout in a callout view.

Input Arguments

| | | | |
|----|------------------------------|--|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Point Name | First Point | The name of the first point in the comparison. |
| 2 | Point Name | Second Point | The name of the second point in the comparison. |
| 3 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: $0.75 = 75%$ from the left edge of the graphical view). |
| 4 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: $0.75 = 75%$ from the top edge of the graphical view). |
| 5 | Boolean | Show First Point Collection? | Indicates whether the first point's collection should be shown in the callout. |
| 6 | Boolean | Show First Point Group? | Indicates whether the first point's group should be shown in the callout. |
| 7 | Boolean | Show First Point Target? | Indicates whether the first point's target should be shown in the callout. |
| 8 | Boolean | Show First Point Coordinates? | Indicates whether the first point's coordinates should be shown in the callout. |
| 9 | Boolean | Show Second Point Collection? | Indicates whether the second point's collection should be shown in the callout. |
| 10 | Boolean | Show Second Point Group? | Indicates whether the second point's group should be shown in the callout. |
| 11 | Boolean | Show Second Point Target? | Indicates whether the second point's target should be shown in the callout. |
| 12 | Boolean | Show Second Point Coordinates? | Indicates whether the second point's coordinates should be shown in the callout. |
| 13 | Boolean | Show dX? | Indicates whether the dX component should be shown in the callout. |
| 14 | Boolean | Show dY? | Indicates whether the dY component should be shown in the callout. |
| 15 | Boolean | Show dZ? | Indicates whether the dZ component should be shown in the callout. |
| 16 | Boolean | Show dMag? | Indicates whether the dMag component should be shown in the callout. |
| 17 | String | Additional X Comments (blank for none) | Additional text to display for the x component (optional). |
| 18 | String | Additional Y Comments (blank for none) | Additional text to display for the y component (optional). |
| 19 | String | Additional Z Comments (blank for none) | Additional text to display for the z component (optional). |
| 20 | String | Additional Notes (blank for none) | Additional general notes to display (optional). |
| 21 | Boolean | Use Default Placement | True enables default placement. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The callout was successfully created. |
| FAILURE | The supplied points or callout view were not found. |

Remarks

Argument 21- Choosing to use default placement prevents the application from attempting to choose a logical placement for the callout with respect to the objects in the view.

Create Relationship Callout

Creates a relationship callout in a callout view.

Input Arguments

| | | | |
|---|------------------------------|-----------------------------------|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Collection Object Name | Relationship Name | The name of the relationship with which to create a callout. |
| 2 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: 0.75 = 75% from the left edge of the graphical view). |
| 3 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: 0.75 = 75% from the top edge of the graphical view). |
| 4 | String | Additional Notes (blank for none) | Additional general notes to display (optional). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The callout was successfully created. |
| FAILURE | The supplied relationship or callout view was not found. |

Remarks

None.

Create Picture Callout

Creates a relationship callout in a callout view.

Input Arguments

| | | | |
|---|------------------------------|---------------------------------|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Collection Object Name | Picture Name | The name of the picture to include in the callout. |
| 2 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: 0.75 = 75% from the left edge of the graphical view). |
| 3 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: 0.75 = 75% from the top edge of the graphical view). |
| 4 | Integer | Scale Image Percent (10-200) | Optional scale factor to apply to the image. |
| 5 | Collection Object Name | Object for Callout Anchor Point | Anchor point used for the callout |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The callout was successfully created. |
| FAILURE | The supplied picture or callout view was not found. |

Remarks

Note: A2 and A3 arguments define the position of **low left** corner of the picture and defined a position along the horizontal/vertical aspects of the screen relative to the top left corner of the screen. For example, a fraction: 0.75 = 75% or 3/4 of the way across the screen from that starting position. Any X/Y position values close to 0% may cause the picture (depend on the picture's size) to be invisible (out of screen).

By default, if object for anchor point is not defined, the created picture callout is located next to graphical view center.

Create Text Callout

Creates a text callout in a callout view.

Input Arguments

| | | | |
|---|------------------------------|---------------------------------|---|
| 0 | Collection Callout View Name | Destination Callout View | The name of the destination callout view. |
| 1 | Edit Test | Text | Text to display in the Callout. |
| 2 | Double | View X Position | The position along the horizontal direction of the callout view (as a fraction: 0.75 = 75% from the left edge of the graphical view). |
| 3 | Double | View Y Position | The position along the vertical direction of the callout view (as a fraction: 0.75 = 75% from the top edge of the graphical view). |
| 4 | Point Name | Callout Anchor Point (Optional) | Callout leader line anchor point (optional). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The callout was successfully created. |
| FAILURE | The supplied callout view was not found. |

Remarks

None.

Make a Callout View Ref List

Creates an empty callout view reference list.

Input Arguments

None.

Return Arguments

| | | | |
|---|-----------------------|-------------------|--|
| 0 | Callout View Ref List | Callout View List | The empty callout view reference list. |
|---|-----------------------|-------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Callout View Ref List - WildCard Selection

Creates a callout view reference list with callouts that fit the search criteria.

Input Arguments

| | | | |
|---|--------|--------------------------------|----------------------------|
| 0 | String | Collection Wildcard Criteria | Collection search string |
| 1 | String | Callout View Wildcard Criteria | Callout name search string |

Return Arguments

| | | | |
|---|-----------------------|-------------------|--|
| 2 | Callout View Ref List | Callout View List | The resulting callout view reference list. |
|---|-----------------------|-------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Add a Callout View to Callout View Ref List

Adds the selected callout view to a list of callouts.

Input Arguments

| | | | |
|---|------------------------------|-------------------|----------------------------|
| 0 | Collection Callout View Name | Callout View | The callout to be added |
| 1 | Callout View Ref List | Callout View List | The callout list to add to |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The list was returned successfully. |
| FAILURE | The callout was not found. |

Remarks

None.

Sort Callout View Ref List

Provides the means to sort a list of callout views by name in ascending or descending order.

Input Arguments

| | | | |
|---|-----------------------|-------------------|---|
| 0 | Callout View Ref List | Callout View List | The list of callouts to be sorted |
| 1 | Boolean | Case Sensitive? | TRUE means consider case in the sort |
| 2 | Boolean | Ascending Order? | TRUE means use ascending order as apposed to using descending order |

Return Arguments

| | | | |
|---|-----------------------|--------------------------|-------------------------|
| 3 | Callout View Ref List | Sorted Callout View List | Returns the sorted list |
|---|-----------------------|--------------------------|-------------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The list was returned successfully. |
| FAILURE | The callout list could not be found. |

Remarks

None.

Get Number of Callout Views in Callout View Ref List

Returns the number of views included in the specified list.

Input Arguments

| | | | |
|---|-----------------------|-------------------|---------------------------------------|
| 0 | Callout View Ref List | Callout View List | The list of callouts to be considered |
|---|-----------------------|-------------------|---------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|---------------------------|
| 1 | Integer | Total Count | Resulting number of views |
|---|---------|-------------|---------------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The list was returned successfully. |
| FAILURE | The callout list could not be found. |

Remarks

None.

Get i-th Callout View From Callout View Ref List

Returns the specified callout, by index number, from the selected callout view ref list.

Input Arguments

| | | | |
|---|-----------------------|--------------------|---------------------------------------|
| 0 | Callout View Ref List | Callout View List | The list of callouts to be considered |
| 1 | Integer | Callout View Index | Index used for selection |

Return Arguments

| | | | |
|---|------------------------|----------------|---------------------------|
| 2 | Collection Object Name | Resulting Item | The returned callout view |
|---|------------------------|----------------|---------------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The list was returned successfully. |
| FAILURE | The callout list could not be found. |

Remarks

None.

Set Default Callout View Properties

This command defines the default settings for newly built callout views and their callouts by editing the default settings also defined in the Users Options. The same controls are available here.

Input Arguments

| | | | |
|---|-----------|-------------------------------|---|
| 0 | String | Default Callout View Name | The starting Callout View name used as a base and incurmented as new callouts are added |
| 1 | Boolean | Lock View Point? | TRUE means lock viewpoint |
| 2 | Boolean | Recall Working Frame? | TRUE means reset the working frame that was set when the callout was built |
| 3 | Boolean | Recall Visible Layer | TRUE means save the object visibility layer with the callout |
| 4 | Integer | Callout Leader Line Thickness | Pixel width for the line |
| 5 | Color | Leader Line Color | Color for the leader line |
| 6 | Integer | Callout Border Thickness | Pixel width for the callout borders |
| 7 | Color | Callout Border Color | Color for the callout borders |
| 8 | Boolean | Divide Text with Lines? | TRUE adds lines between rows in the callout |
| 9 | Font Type | Font | Font used in the callout by default |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------|
| SUCCESS | Command always succeeds |
|---------|-------------------------|

Remarks

None.

Delete Callout View

Deletes a callout view.

Input Arguments

| | | | |
|---|------------------------------|--------------|---|
| 0 | Collection Callout View Name | Callout View | The name of the callout view to delete. |
|---|------------------------------|--------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The callout view was successfully deleted. |
| FAILURE | The callout view was not found. |

Remarks

None.

Rename Callout View

Renames a callout view.

Input Arguments

| | | | |
|---|------------------------------|----------------------------|---|
| 0 | Collection Callout View Name | Original Callout View Name | The name of the callout view to rename. |
| 1 | Collection Callout View Name | New Callout View Name | The new callout view name. |
| 2 | Boolean | Overwrite if exists? | If true, any existing callout with the new name will be replaced. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The callout view was successfully renamed. |
| FAILURE | The source callout view was not found, or the destination name already exists and argument 2 was FALSE. |

Remarks

None.

Auto Arrange Callout View

Sorts the callouts within the specified callout view. This command performs the optimized solution from the callout right-click menu.

Input Arguments

| | | | |
|---|------------------------------|--------------|---------------------------------------|
| 0 | Collection Callout View Name | Callout View | The name of the callout view to edit. |
|---|------------------------------|--------------|---------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The callout view was successfully updated. |
| FAILURE | The source callout view was not found. |

Remarks

None.

Other MP Types

Make a Boolean

Creates a boolean value.

Input Arguments

| | | | |
|---|---------|---------------|-------------------------------------|
| 0 | Boolean | Initial Value | The value of the boolean to create. |
|---|---------|---------------|-------------------------------------|

Return Arguments

| | | | |
|---|---------|--------|------------------------|
| 1 | Boolean | Result | The resulting boolean. |
|---|---------|--------|------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Integer

Creates an integer value.

Input Arguments

| | | | |
|---|---------|---------------|-------------------------------------|
| 0 | Integer | Initial Value | The value of the integer to create. |
|---|---------|---------------|-------------------------------------|

Return Arguments

| | | | |
|---|---------|--------|------------------------|
| 1 | Integer | Result | The resulting integer. |
|---|---------|--------|------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Integer From String

Converts a string containing an integer value to a string data type.

Input Arguments

| | | | |
|---|--------|--------------|--|
| 0 | String | String Input | The source string containing an integer value. |
|---|--------|--------------|--|

Return Arguments

| | | | |
|---|---------|-------------------|------------------------|
| 1 | Integer | Resultant Integer | The resulting integer. |
|---|---------|-------------------|------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The string was converted successfully to an integer. |
| FAILURE | The source string did not contain a numeric value. |

Remarks

If the source string contains a non-integer numeric value (such as 3.5), only the integer portion will be retained. The numeric value will not be rounded.

Make a Double

Creates a double value.

Input Arguments

| | | | |
|---|--------|---------------|------------------------------------|
| 0 | Double | Initial Value | The value of the double to create. |
|---|--------|---------------|------------------------------------|

Return Arguments

| | | | |
|---|--------|--------|-----------------------|
| 1 | Double | Result | The resulting double. |
|---|--------|--------|-----------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Double From an Integer

Converts a integer value to a double data type.

Input Arguments

| | | | |
|---|---------|---------------|----------------------------------|
| 0 | Integer | Integer Value | The value of the source integer. |
|---|---------|---------------|----------------------------------|

Return Arguments

| | | | |
|---|--------|--------|-----------------------|
| 1 | Double | Result | The resulting double. |
|---|--------|--------|-----------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Since doubles inherently have the capacity to store higher precision (and larger value) numbers, no information will be lost in the conversion from the integer to the double data type.

Make a Double from String

Converts a string containing a numeric value to a double data type.

Input Arguments

| | | | |
|---|--------|--------------|--|
| 0 | String | String Input | The source string containing a double value. |
|---|--------|--------------|--|

Return Arguments

| | | | |
|---|--------|------------------|-----------------------------|
| 1 | Double | Resultant Double | The resulting double value. |
|---|--------|------------------|-----------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The string was converted successfully to a double. |
| FAILURE | The source string did not contain a numeric value. |

Remarks

None.

Make a Double List

Creates a list of doubles.

Input Arguments

| | | | |
|---|-------------|-------------|----------------------|
| 0 | Double List | Double List | The list of doubles. |
|---|-------------|-------------|----------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Add Double to Double List

Adds a double value to an existing list of doubles.

Input Arguments

| | | | |
|---|-------------|-------------|-------------------------------|
| 0 | Double List | Double List | The existing list of doubles. |
| 1 | Double | Value | The value to add to the list. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The double was added successfully. |
| FAILURE | The existing list was not found. |

Remarks

None.

Make a Boolean From an Integer

Converts an integer value to a boolean data type.

Input Arguments

| | | | |
|---|---------|---------------|--------------------------------|
| 0 | Integer | Integer Value | The source integer to convert. |
|---|---------|---------------|--------------------------------|

Return Arguments

| | | | |
|---|---------|--------|------------------------------|
| 1 | Boolean | Result | The resulting boolean value. |
|---|---------|--------|------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Any nonzero value is considered TRUE. Zero is the only value that is considered FALSE.

Make a String

Creates a string.

Input Arguments

| | | | |
|---|--------|-----------------------------|--|
| 0 | String | Base String | The source string. |
| 1 | String | String to Append (optional) | An optional string to append to the base string. |

Return Arguments

| | | | |
|---|--------|------------------|-----------------------------|
| 2 | String | Resultant String | The resulting string value. |
|---|--------|------------------|-----------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make String from Integer

Converts an integer value into a string.

Input Arguments

| | | | |
|---|---------|---------------------------------------|--|
| 0 | Integer | Integer to Convert | The source integer. |
| 1 | Integer | Minimum Digits (will prefix with 0's) | The number of digits to use in the resulting string. |

Return Arguments

| | | | |
|---|--------|------------------|-----------------------------|
| 2 | String | Resultant String | The resulting string value. |
|---|--------|------------------|-----------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make String from Double

Converts a double value to a string data type.

Input Arguments

| | | | |
|---|---------|-------------------|---|
| 0 | Double | Double to convert | The source double value. |
| 1 | Integer | Decimal Precision | The number of decimals to retain in the resulting string. |

Return Arguments

| | | | |
|---|--------|------------------|-----------------------------|
| 2 | Double | Resultant Double | The resulting double value. |
|---|--------|------------------|-----------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The double value will be rounded, or zeroes will be appended as necessary to adhere to the “Decimal Precision” argument.

Make String from Decimal Degrees Angular Value

Converts a decimal degree value to another angle format, returning the result as a string.

Input Arguments

| | | | |
|---|---------------|---------------------------------|--|
| 0 | Double | Angular Value (Decimal Degrees) | The angular value to convert, in decimal degrees. |
| 1 | Angular Units | Output Units | The units to convert to: Decimal Degrees, Degrees:Minutes:Seconds, Degrees:Minutes, Radians, Gons (grad) |
| 2 | Integer | Decimal Precision | The number of decimal places to retain in the end result. |

Return Arguments

| | | | |
|---|--------|------------------|--|
| 3 | String | Resultant String | The resulting string after the conversion. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make an Incremented String

Takes a source string value and increments it by one.

Input Arguments

| | | | |
|---|--------|-------------|---|
| 0 | String | Base String | The base string from which to base the increment. |
|---|--------|-------------|---|

Return Arguments

| | | | |
|---|--------|------------------|-----------------------|
| 1 | String | Resultant String | The resulting string. |
|---|--------|------------------|-----------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the Base String does not end with a numeric value, the number 1 will be appended. Otherwise, the last whole numeric value will be incremented by one.

Make a System String

Creates a string based on the current SA version, filename, or date/time information.

Input Arguments

| | | | |
|---|---------------|--------------------------|---|
| 0 | System String | String Content | The information to carry in the string: SA Version, XIT Filename, MP Filename, MP Filename (Full Path), Date & Time, Date, Date (Short), Time |
| 1 | String | Format String (Optional) | A custom format string (see Remarks below). |

Return Arguments

| | | | |
|---|--------|------------------|-----------------------------|
| 2 | String | Resultant String | The resulting string value. |
|---|--------|------------------|-----------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | This command always succeeds. |
| FAILURE | An invalid user-formatted string was entered. |

Remarks

The optional format string is used if the String Content argument is set to Date & Time, Date, Date (Short), or Time. If the format string is left blank in these cases, the default format is used.

Valid values for the format string include:

- **%a.** Abbreviated weekday name
- **%A.** Full weekday name
- **%b.** Abbreviated month name
- **%B.** Full month name
- **%c.** Date and time representation appropriate for locale
- **%#c.** Long date and time representation, appropriate for the current locale. For example, “Tuesday, March 14, 1995, 12:41:29”.
- **%d.** Day of month as decimal number (01-31)
- **%H.** Hour in 24-hour format (00-23)
- **%I.** Hour in 12-hour format (01-12)
- **%j.** Day of year as decimal number (001-366)
- **%m.** Month as decimal number (01-12)

- **%M.** Minute as decimal number (00-59)
- **%p.** Current locale's AM/PM indicator for 12-hour clock
- **%S.** Second as decimal number (00-59)
- **%U.** Week of year as decimal number, with Sunday as first day of week (00-53)
- **%w.** Weekday as decimal number (0-6; Sunday is 0)
- **%W.** Week of year as decimal number, with Monday as first day of week (00-53)
- **%x.** Date representation for the current locale.
- **%#x.** Long date representation, appropriate to the current locale. For example: "Tuesday, March 14, 1995".
- **%X.** Time representation for the current locale.
- **%y.** Year without century, as decimal number (0-99)
- **%Y.** Year with century, as decimal number.
- **%z, %Z.** Either the time-zone name or time zone abbreviation, depending on registry settings; no characters if time zone is unknown.
- **%%.** Percent sign

Use **%#d, %#H, %#I, %#j, %#m, %#M, %#S, %#U, %#w, %#W, %#y, and %#Y** to remove leading zeroes from the respective numbers (if applicable).

Concatenate Strings

Combines two or more strings into one string.

Input Arguments

| | | | |
|---|-----------------|---------------|--|
| 0 | String Ref List | Input Strings | The source strings to combine together |
|---|-----------------|---------------|--|

Return Arguments

| | | | |
|---|--------|------------------|------------------------------------|
| 1 | String | Resultant String | The resulting concatenated string. |
|---|--------|------------------|------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

An empty item added to the string reference list will result in a newline. This allows you to put following text on the next line.

Make a String Ref List

Creates a reference list (MP array) of strings.

Input Arguments

| | | | |
|---|-----------------|-------------|--|
| 0 | String Ref List | String List | A list of strings to combine into a single list. |
|---|-----------------|-------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Picture Name Ref List - Runtime Select

Makes a list of pictures from a user selection.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display to the user in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|----------------------------------|-------------------|--------------------------------|
| 1 | Collection Picture Name Ref List | Picture Name List | The list of selected pictures. |
|---|----------------------------------|-------------------|--------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The list was returned successfully. |
| FAILURE | The user pressed ESC or cancelled the selection. |

Remarks

None.

Make a String From A String Ref List

Outputs a concatenated string from a string reference list.

Input Arguments

| | | | |
|---|-----------------|-------------|--|
| 0 | String Ref List | String List | The list of strings to combine into a single string. |
|---|-----------------|-------------|--|

Return Arguments

| | | | |
|---|--------|------------------|------------------------------------|
| 1 | String | Resultant String | The resulting concatenated string. |
|---|--------|------------------|------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Strings from a Point Name

Dissects the collection, group, and target name for a point name into separate strings.

Input Arguments

| | | | |
|---|------------|------------|------------------------|
| 0 | Point Name | Point Name | The source point name. |
|---|------------|------------|------------------------|

Return Arguments

| | | | |
|---|------------|------------|--|
| 1 | Collection | Collection | The collection containing the source point. |
| 2 | Group | Group | The point group containing the source point. |
| 3 | Target | Target | The target name of the source point. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Strings from a Collection Object Name

Dissects the collection and object names from a collection object name into strings.

Input Arguments

| | | | |
|---|------------------------|----------------------------------|------------------------------------|
| 0 | Collection Object Name | Resultant Collection Object Name | The source collection object name. |
|---|------------------------|----------------------------------|------------------------------------|

Return Arguments

| | | | |
|---|-------------|-------------|--|
| 1 | String | Collection | The collection containing the source object. |
| 2 | Object | Object | The name of the source object. |
| 3 | Object Type | Object Type | The object type for the source object. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command can also be used to convert a Collection Instrument ID into a collection (string) and an Instrument ID (returned as a string instead of an integer).

Make a Point Name from Strings

Takes three string values (one for the collection, one for the group, and one for the target) and creates a point name data type.

Input Arguments

| | | | |
|---|--------|------------|---|
| 0 | String | Collection | The name of the collection for the resulting point. |
| 1 | String | Group | The name of the group for the resulting point. |
| 2 | String | Target | The name of the target for the resulting point. |

Return Arguments

| | | | |
|---|------------|----------------------|--|
| 3 | Point Name | Resultant Point Name | The resulting point name, as a point name data type. |
|---|------------|----------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the collection name is left blank, then when the point name is used, it will refer to the active collection.

Make a Point Name - Runtime Select

Prompts the user with a custom prompt in the graphical view, and waits for them to select a single point.

Input Arguments

| | | | |
|---|--------|-------------|------------------------------------|
| 0 | String | User Prompt | The prompt to display to the user. |
|---|--------|-------------|------------------------------------|

Return Arguments

| | | | |
|---|------------|----------------------|--|
| 1 | Point Name | Resultant Point Name | The resulting point name for the point selected by the user. |
|---|------------|----------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | A point was selected successfully. |
| FAILURE | The ESC key was pressed while the prompt was displayed. |

Remarks

None.

Make a Point Name - Ensure Unique

Takes a supplied point name and ensures that it is unique, adding asterisks or numeric values as desired.

Input Arguments

| 0 | Point Name | Point Name | The desired name of the point. |
|---|------------|--------------------|--|
| 1 | Boolean | Use Number Suffix? | Indicates whether asterisks should be appended to the target name to make it unique (FALSE) or whether numeric suffixes should be used (TRUE). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Unlike most commands, this command does not return an argument. Instead, it just modifies the input argument as necessary to make it unique. Therefore, other commands should reference argument 0 to obtain the unique point name.

Make a Point Name Ref List

Creates a reference list (MP array) of point names.

Input Arguments

| | | | |
|---|---------------------|-----------------|------------------------------------|
| 0 | Point Name Ref List | Point Name List | The list of point names to create. |
|---|---------------------|-----------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Point Name Ref List From a Group

Creates a reference list (MP array) of point names by getting all points in a specified point group.

Input Arguments

| | | | |
|---|------------------------|------------|---|
| 0 | Collection Object Name | Group Name | The point group containing the source points. |
|---|------------------------|------------|---|

Return Arguments

| | | | |
|---|---------------------|---------------------------|---------------------------|
| 1 | Point Name Ref List | Resultant Point Name List | The resulting point list. |
|---|---------------------|---------------------------|---------------------------|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | A list was selected successfully. |
| FAILURE | The source point group was not found. |

Remarks

None.

Make a Point Name Ref List - Runtime Select

Displays a custom prompt in the graphical view and asks the user to select one or more points, which are returned (after pressing ENTER) as a point name reference list.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|---------------------|---------------------------|---------------------------|
| 1 | Point Name Ref List | Resultant Point Name List | The resulting point list. |
|---|---------------------|---------------------------|---------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | A list was selected successfully. |
| FAILURE | The user pressed ESC prior to completing the selection. |

Remarks

The user must press ENTER to accept the selection, or ESC to cancel it. ESC will result in the command returning failure. Points are added to the list in the order in which they were selected. Selecting multiple points at one time results in ambiguous ordering in the list.

Make a Point Name Ref List - Wildcard Select

Returns a list of points based on wildcard selection criteria.

Input Arguments

| | | | |
|---|--------|------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard criteria for the collection name. |
| 1 | String | Group Name Wildcard Criteria | The wildcard criteria for the group name. |
| 2 | String | Point Name Wildcard Criteria | The wildcard criteria for the point name. |

Return Arguments

| | | | |
|---|---------------------|---------------------------|---------------------------|
| 3 | Point Name Ref List | Resultant Point Name List | The resulting point list. |
|---|---------------------|---------------------------|---------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Enter wildcard values for the collection, point group, and point name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all points that start with **s1** followed by two characters, the point name defining the selection criteria would be ***::*::s1??**.

Append Two Point Name Ref Lists

Combines two point name reference lists into a single list by appending the second list to the first.

Input Arguments

| | | | |
|---|---------------------|-------------------|---------------------------------------|
| 0 | Point Name Ref List | Point Name List A | The first list of points to combine. |
| 1 | Point Name Ref List | Point Name List B | The second list of points to combine. |

Return Arguments

| | | | |
|---|---------------------|--------------------------------|--------------------------|
| 2 | Point Name Ref List | Resultant Point Name List(A+B) | The combined point list. |
|---|---------------------|--------------------------------|--------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The lists were combined successfully. |
| FAILURE | One or both source point lists could not be found. |

Remarks

None.

Subtract Two Point Name Ref Lists

Provides a means to create a list of points missing from list A by subtracting a second list B.

Input Arguments

| | | | |
|---|---------------------|-------------------|--|
| 0 | Point Name Ref List | Point Name List A | The first list of points |
| 1 | Point Name Ref List | Point Name List B | The second list of points to subtract. |

Return Arguments

| | | | |
|---|---------------------|--------------------------------|---------------------------|
| 2 | Point Name Ref List | Resultant Point Name List(A-B) | The resulting point list. |
|---|---------------------|--------------------------------|---------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The lists were subtracted successfully. |
| FAILURE | One or both source point lists could not be found. |

Remarks

None.

Make a Vector Name Ref List From a Vector Group

Creates a list of vector names (as a reference list) from a source vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|--------------------------------------|
| 0 | Collection Object Name | Vector Group Name | The name of the source vector group. |
|---|------------------------|-------------------|--------------------------------------|

Return Arguments

| | | | |
|---|----------------------|----------------------------|-------------------------------------|
| 1 | Vector Name Ref List | Resultant Vector Name List | The resulting list of vector names. |
|---|----------------------|----------------------------|-------------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The list was created successfully. |
| FAILURE | The source vector group could not be found. |

Remarks

None.

Make a Vector Name Ref List - Runtime Select

Displays a custom prompt in the graphical view and asks the user to select a collection, which is returned as a collection name.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|----------------------|----------------------------|-----------------------------|
| 1 | Vector Name Ref List | Resultant Vector Name List | The selected cvector names. |
|---|----------------------|----------------------------|-----------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | A vector(s) was selected successfully. |
| FAILURE | The user pressed ESC prior to selecting vectors. |

Remarks

Pressing ESC before selecting a vector will result in the command returning failure.

Make Vector Names Unique in Vector Group

Creates new names for each vector within a vector group that has a redundant name.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group housing the like vectors. |
|---|------------------------|-------------------|--|

Returned Status

| | |
|---------|--------------------------|
| SUCCESS | Command always succeeds. |
|---------|--------------------------|

Remarks

Each redundant vector name will be changed to have an underscore followed by an incremental number following its original name.

Make a Collection Name - Runtime Select

Displays a custom prompt in the graphical view and asks the user to select a collection, which is returned as a collection name.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|-----------------|---------------------------|-------------------------------|
| 1 | Collection Name | Resultant Collection Name | The selected collection name. |
|---|-----------------|---------------------------|-------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | A collection was selected successfully. |
| FAILURE | The user pressed ESC prior to selecting a collection. |

Remarks

Pressing ESC before selecting a collection will result in the command returning failure.

Make a Collection Item Name from Strings

Builds a Collection Object Name from Strings which can be used to identify any item. An “Item” is a generic term that can apply to anything in the tree other than specific points. This includes things that are not objects such as reports or charts or GD&T annotations for example.

Input Arguments

| | | | |
|---|-------------|-------------|--|
| 0 | String | Collection | The collection to use in the collection object name. |
| 1 | String | Object | The object to use in the collection object name. |
| 2 | Object Type | Object Type | The type for the object. |

Return Arguments

| | | | |
|---|------------------------|----------------------------------|--|
| 3 | Collection Object Name | Resultant Collection Object Name | The resulting collection object name from the command. |
|---|------------------------|----------------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Collection Item Name Reference List - Wildcard Selection

Builds a Collection Item Names which can be used to identify any item. An “Item” is a generic term that can apply to anything in the tree other than specific points. This includes things that are not objects such as reports or charts or GD&T annotations for example. Creates a list of objects through the use of wildcard criteria.

Input Arguments

| | | | |
|---|----------------------|------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Object Wildcard Criteria | The wildcard string to specify the items to select from. |
| 2 | Collection Item Type | Item Type | A filter to only select objects of this type. |

Return Arguments

| | | | |
|---|---------------------------------|---|---|
| 3 | Collection Object Name Ref List | Resultant Collection Object Name Reference List | The resulting list of items matching the specified wildcard selection criteria. |
|---|---------------------------------|---|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The matching list of items was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the objects in the list is undefined.

Enter wildcard values for the collection and object using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all objects that start with “s” from all collections, use a collection criteria of * and an object wildcard criteria of s*.

Make a Collection Object Name from Strings

Builds a Collection Object Name from Strings which can be used to identify an object.

Input Arguments

| | | | |
|---|-------------|-------------|--|
| 0 | String | Collection | The collection to use in the collection object name. |
| 1 | String | Object | The object to use in the collection object name. |
| 2 | Object Type | Object Type | The type for the object. |

Return Arguments

| | | | |
|---|------------------------|----------------------------------|--|
| 3 | Collection Object Name | Resultant Collection Object Name | The resulting collection object name from the command. |
|---|------------------------|----------------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Collection Object Name - Runtime Select

Displays a custom prompt in the graphical view and asks the user to select an object, which is returned as a collection object name.

Input Arguments

| | | | |
|---|-------------|-------------|---|
| 0 | String | User Prompt | The prompt to display in the graphical view. |
| 1 | Object Type | Object Type | The type of object to require the user to select. The user will be unable to select a type that does not match this filter. |

Return Arguments

| | | | |
|---|------------------------|----------------------------------|--------------------------------------|
| 2 | Collection Object Name | Resultant Collection Object Name | The selected collection object name. |
|---|------------------------|----------------------------------|--------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | An object was selected successfully. |
| FAILURE | The user pressed ESC prior to selecting an object. |

Remarks

Pressing ESC before selecting an object will result in the command returning failure.

Using “Any” for the “Object Type” argument will allow all object types to be selectable without restriction.

Point Cloud selection can be refined as follows:

Cloud = Cloud, Scan Stripe Cloud, or Cross Section Cloud

Scan Stripe Cloud = Scan Stripe Cloud or Cross Section Cloud (but not basic clouds)

Cross Section Cloud = allows Cross Section Cloud selection only

Make a Collection Object Name - Ensure Unique

Takes a supplied collection object name and ensures that it is unique, adding asterisks or numeric values as desired.

Input Arguments

| 0 | Collection Object Name | Collection Object Name | The desired name for the object. |
|---|------------------------|------------------------|--|
| 1 | Boolean | Use Number Suffix? | Indicates whether asterisks should be appended to the object name to make it unique (FALSE) or whether numeric suffixes should be used (TRUE). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Unlike most commands, this command does not return an argument. Instead, it just modifies the input argument as necessary to make it unique. Therefore, other commands should reference argument 0 to obtain the unique object name.

Make a Collection Object Name Reference List - Runtime Select

Displays a custom prompt in the graphical view and asks the user to select one or more objects, which is returned as a collection object name reference list.

Input Arguments

| | | | |
|---|-------------|-------------|--|
| 0 | String | User Prompt | The prompt to display in the graphical view. |
| 1 | Object Type | Object Type | The type of objects to require the user to select. The user will be unable to select a type that does not match this filter. |

Return Arguments

| | | | |
|---|---------------------------------|---|---|
| 2 | Collection Object Name Ref List | Resultant Collection Object Name Reference List | The selected list of collection object names. |
|---|---------------------------------|---|---|

Returned Status

| | |
|---------|---|
| SUCCESS | One or more objects were selected successfully. |
| FAILURE | The user pressed ESC prior to completing selection. |

Remarks

Pressing ESC before finishing selection will result in the command returning failure.

Objects will be added to the list in the order in which they were selected. If several objects are selected simultaneously (via F2 or marquee-select) then their order in the list is undefined. Using “Any” for the “Object Type” argument will allow any entity types to be selected.

Make a Collection Object Name Reference List - WildCard Selection

Creates a list of objects (in the form of a collection object name reference list) through the use of wildcard criteria.

Input Arguments

| | | | |
|---|-------------|------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Object Wildcard Criteria | The wildcard string to specify the objects to select from. |
| 2 | Object Type | Object Type | A filter to only select objects of this type. |

Return Arguments

| | | | |
|---|---------------------------------|---|---|
| 3 | Collection Object Name Ref List | Resultant Collection Object Name Reference List | The resulting list of objects matching the specified wildcard selection criteria. |
|---|---------------------------------|---|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The matching list of objects was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the objects in the list is undefined.

Enter wildcard values for the collection and object using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all objects that start with "s" from all collections, use a collection criteria of * and an object wildcard criteria of s*.

Make a Collection Object Name Ref List - By Type

This command creates a list of objects (in the form of a collection object name reference list) based on a specified type.

Input Arguments

| | | | |
|---|-------------|-------------|--------------------------------|
| 0 | String | Collection | The collection to select from. |
| 1 | Object Type | Object Type | The type of object to select. |

Return Arguments

| | | | |
|---|---------------------------------|---------------------------------------|---|
| 2 | Collection Object Name Ref List | Resultant Collection Object Name List | The resulting list of objects matching the specified type and collection. |
|---|---------------------------------|---------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The matching list of objects was created successfully. |
| FAILURE | The specified collection was not found. |

Remarks

This command may return an empty list. The order of the objects in the list is undefined.

Point Cloud selection can be refined as follows:

Cloud = Cloud, Scan Stripe Cloud, or Cross Section Cloud

Scan Stripe Cloud = Scan Stripe Cloud or Cross Section Cloud (but not basic clouds)

Cross Section Cloud = allows Cross Section Cloud selection only

Make a Collection Object Name Ref List - By Type and Color

This command creates a list of objects (in the form of a collection object name reference list) based on a specified type AND object color. It only returns items that match both criteria.

Input Arguments

| | | | |
|---|-------------|--------------|--------------------------------|
| 0 | String | Collection | The collection to select from. |
| 1 | Object Type | Object Type | The type of object to select. |
| 2 | Color | Object Color | The color desired |

Return Arguments

| | | | |
|---|---------------------------------|---------------------------------------|---|
| 2 | Collection Object Name Ref List | Resultant Collection Object Name List | The resulting list of objects matching the specified type and collection. |
|---|---------------------------------|---------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The matching list of objects was created successfully. |
| FAILURE | The specified collection was not found. |

Remarks

This command may return an empty list. The order of the objects in the list is undefined.

Point Cloud selection can be refined as follows:

Cloud = Cloud, Scan Stripe Cloud, or Cross Section Cloud

Scan Stripe Cloud = Scan Stripe Cloud or Cross Section Cloud (but not basic clouds)

Cross Section Cloud = allows Cross Section Cloud selection only

Make a Collection Object Name Ref List

Creates a list of objects (in the form of a collection object name reference list) by specifying the individual objects.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|---|
| 0 | Collection Object Name Ref List | Collection Object Name List | The list of objects to put into the list. |
|---|---------------------------------|-----------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The order of objects in the list is defined by the order in which they're entered in this command.

Append two Collection Object Name Ref Lists

Combines two collection object name reference lists into a single list.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------------|--|
| 0 | Collection Object Name Ref List | Collection Object Name List A | The first collection object name reference list to combine. |
| 1 | Collection Object Name Ref List | Collection Object Name List B | The second collection object name reference list to combine. |

Return Arguments

| | | | |
|---|---------------------------------|--|---|
| 2 | Collection Object Name Ref List | Resultant Collection Object Name List(A+B) | The combined collection object name reference list. |
|---|---------------------------------|--|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The first list is placed before the second in the resulting combined list. The order of objects in the list is maintained.

Add a Collection Object Name to a Ref List

Adds a single object to an existing collection object name reference list.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|-----------------------------------|
| 0 | Collection Object Name Ref List | Collection Object Name List | The list of objects to append to. |
| 1 | Collection Object Name | Collection Object To Add | The object to add to the list. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The object is added to the end of the source list.

Make a Collection Object Name Ref List from all Groups in a Collection

Makes a collection object name reference list from all groups in a specified collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Collection Name | Collection Name | The name of the collection to select from. |
|---|-----------------|-----------------|--|

Return Arguments

| | | | |
|---|---------------------------------|-----------------------------|---|
| 1 | Collection Object Name Ref List | Collection Object Name List | The resulting list of objects (point groups). |
|---|---------------------------------|-----------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The resulting list was created successfully. |
| FAILURE | The specified collection was not found. |

Remarks

The resulting list may be empty. The order of objects in the list is undefined.

Make a Collection Instrument Reference List

Creates an empty Instrument ID Reference List.

Input Arguments

None.

Return Arguments

| | | | |
|---|-----------------------------------|--|--|
| 0 | Collection Instrument ID Ref List | Resultant Collection Instrument Reference List | The empty list, which can be referenced later. |
|---|-----------------------------------|--|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Collection Name and Index from Collection Instrument ID

Returns the Collection Name and index from the Collection Instrument ID.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | The collection Instrument ID to break down |
|---|--------------------------|---------------|--|

Return Arguments

| | | | |
|---|-----------------|------------------|---|
| 1 | Collection Name | Collection Name | The collection name containing the instrument |
| 2 | Integer | Instrument Index | The index of the instrument |

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument name was returned successfully |
| FAILURE | The instrument could not be found. |

Remarks

None.

Get Collection Instrument Reference List Variable

Returns a list of Collection Instrument IDs from the specified variable.

Input Arguments

| | | | |
|---|--------|------|-------------------------------|
| 0 | String | Name | Name of the new list variable |
|---|--------|------|-------------------------------|

Return Arguments

| | | | |
|---|--------------------------------|-------|--------------------------------|
| 1 | Collection Instrument Ref List | Value | List of instruments to include |
|---|--------------------------------|-------|--------------------------------|

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The list was returned Successfully |
| Failure | The variable could not be found |

Remarks

None.

Set Collection Instrument Reference List Variable

Creates or updates an Instrument ID Reference List variable.

Input Arguments

| | | | |
|---|--------------------------------|-------|--------------------------------|
| 0 | String | Name | Name of the new list variable |
| 1 | Collection Instrument Ref List | Value | List of instruments to include |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Add a Collection Instrument to a Ref List

Adds an instrument to an existing list of instruments.

Input Arguments

| | | | |
|---|-----------------------------------|--------------------------------------|------------------------------------|
| 0 | Collection Instrument ID Ref List | Collection Instrument Reference List | Instrument list to add to. |
| 1 | Collection Instrument ID | Collection Instrument To Add | The instrument to add to the list. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was added to the list successfully. |
| FAILURE | The specified instrument to add was not found. |

Remarks

The instrument is added to the end of the list.

Add Collection Instruments to a Ref List - WildCard Selection

Adds an instrument to an existing list of instruments based on wildcard selection criteria.

Input Arguments

| 0 | Collection Instrument ID Ref List | Collection Instrument Reference List | Instrument list to add to. |
|---|-----------------------------------|--------------------------------------|----------------------------------|
| 1 | String | Collection Wildcard Criteria | Wildcard criteria of collection. |
| 2 | String | Instrument Wildcard Criteria | Wildcard criteria of instrument. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was added to the list successfully. |
| FAILURE | The specified instrument to add was not found. |

Remarks

The instrument is added to the end of the list.

Make a Collection Instrument Reference List - Runtime Select

Displays a prompt to the user in the graphical view and asks the user to select one or more instruments, returning that selection as a collection instrument ID reference list.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display to the user in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|-----------------------------------|--|---|
| 1 | Collection Instrument ID Ref List | Resultant Collection Instrument Reference List | The resulting list of selected instruments. |
|---|-----------------------------------|--|---|

Returned Status

| | |
|---------|---|
| SUCCESS | One or more instruments were selected successfully. |
| FAILURE | The user pressed ESC prior to completing selection. |

Remarks

Pressing ESC before finishing selection will result in the command returning failure.

Instruments will be added to the list in the order in which they were selected. If several instruments are selected simultaneously (via F2 or marquee-select) then their order in the list is undefined.

Make a Relationship Reference List-WildCard Selection

Creates a list of relationships based on wildcard selection criteria.

Input Arguments

| | | | |
|---|-------------------|--------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Relationship Wildcard Criteria | The wildcard string to specify the relationships to select. |
| 2 | Relationship Type | Relationship Type Filter | Type of relationships to include in the list |

Return Arguments

| | | | |
|---|-----------------------|---------------------------------------|---|
| 3 | Relationship Ref List | Resultant Relationship Reference List | The resulting list of relationships matching the specified wildcard selection criteria. |
|---|-----------------------|---------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of matching relationships was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the relationships in the list is undefined.

Enter wildcard values for the collection and relationship using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all relationships that start with “s” from all collections, use a collection criteria of * and a relationship wildcard criteria of s*.

Make a Relationship Reference List- Runtime Selection

Creates a list of relationships based on prompted user selection.

Input Arguments

| | | | |
|---|-------------------|--------------------------|---|
| 0 | String | User Prompt | The message displayed to the user during selection. |
| 1 | Relationship Type | Relationship Type Filter | Selection of relationship types used to limit user selection. |

Return Arguments

| | | | |
|---|-----------------------|---------------------------------------|---|
| 2 | Relationship Ref List | Resultant Relationship Reference List | The resulting list of relationships matching the specified wildcard selection criteria. |
|---|-----------------------|---------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of matching relationships was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

None.

Make an Event Reference List-Wildcard Selection

Makes a list of reportable events based on wildcard selection criteria.

Input Arguments

| | | | |
|---|--------|------------------------------|--|
| 0 | String | Collection Wildcard Criteria | The wildcard string to specify the collections to select from. |
| 1 | String | Event Wildcard Criteria | The wildcard string to specify the events to select. |

Return Arguments

| | | | |
|---|----------------|--------------------------------|--|
| 2 | Event Ref List | Resultant Event Reference List | The resulting list of events matching the specified wildcard selection criteria. |
|---|----------------|--------------------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of matching relationships was created successfully. |
| FAILURE | The wildcard criteria strings were invalid. |

Remarks

This command may return an empty list. The order of the events in the list is undefined.

Enter wildcard values for the collection and event using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. Specific characters can also be found using brackets[]. To find all events that start with "s" from all collections, use a collection criteria of * and an event wildcard criteria of s*.

Append two Relationship Ref Lists

Provides a means to combine two relationship reference lists.

Input Arguments

| | | | |
|---|-----------------------|-------------------------------|------------------------------|
| 0 | Relationship Ref List | Relationship Reference List A | First list of relationships |
| 1 | Relationship Ref List | Relationship Reference List B | Second list of relationships |

Return Arguments

| | | | |
|---|-----------------------|---|--------------------------------|
| 2 | Relationship Ref List | Resulting Relationship Reference List (A+B) | Combined list of relationships |
|---|-----------------------|---|--------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | This command always succeeds. |
| FAILURE | Relationships lists could not be found. |

Remarks

None.

Make a Collection Instrument ID from a Collection and an Integer

Makes a collection instrument ID from a collection name and an integer.

Input Arguments

| | | | |
|---|-----------------|-----------------|------------------------------------|
| 0 | Collection Name | Collection Name | The name of the collection. |
| 1 | Integer | Instrument | The numeric ID for the instrument. |

Return Arguments

| | | | |
|---|--------------------------|---------------|--|
| 2 | Collection Instrument ID | Instrument ID | The resulting Collection Instrument ID |
|---|--------------------------|---------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Collection Instrument ID - Runtime Select

Makes a collection instrument ID from a runtime selection of an instrument.

Input Arguments

| | | | |
|---|--------|-------------|------------------------------------|
| 0 | String | User Prompt | The prompt to display to the user. |
|---|--------|-------------|------------------------------------|

Return Arguments

| | | | |
|---|--------------------------|---------------|---|
| 1 | Collection Instrument ID | Instrument ID | The resulting Collection Instrument ID. |
|---|--------------------------|---------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument was returned successfully. |
| FAILURE | The user pressed the ESC key and cancelled the selection. |

Remarks

None.

Make a Collection Vector Group Name Ref List - Runtime Select

Prompts the user to select vector groups, then returns those vector groups as a list.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display to the user in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|---------------------------------------|---|--------------------------------------|
| 1 | Collection Vector Group Name Ref List | Resultant Collection Vector Group Name Reference List | The resulting list of vector groups. |
|---|---------------------------------------|---|--------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The list was returned successfully. |
| FAILURE | The user pressed ESC or cancelled the selection. |

Remarks

None.

Make a Collection Machine ID from a Collection and an Integer

Makes a collection machine ID from a collection name and an integer.

Input Arguments

| | | | |
|---|-----------------|-----------------|-----------------------------|
| 0 | Collection Name | Collection Name | The name of the collection. |
| 1 | Integer | Machine | The Machine ID. |

Return Arguments

| | | | |
|---|-----------------------|------------|--------------------------------------|
| 2 | Collection Machine ID | Machine ID | The resulting collection machine ID. |
|---|-----------------------|------------|--------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Report Ref List from a Collection

Makes a list of all reports in the specified collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|---|
| 0 | Collection Name | Collection Name | The name of the collection in question. |
|---|-----------------|-----------------|---|

Return Arguments

| | | | |
|---|--------------------|-------------|----------------------|
| 1 | SA Report Ref List | Report List | The list of reports. |
|---|--------------------|-------------|----------------------|

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The list was returned successfully. |
| FAILURE | The collection was not found. |

Remarks

None.

Make a Report Ref List - Runtime Select

Prompts the user to select reports from the tree or F2 dialog, and returns the selected reports as a list.

Input Arguments

| | | | |
|---|--------|-------------|--|
| 0 | String | User Prompt | The prompt to display to the user in the graphical view. |
|---|--------|-------------|--|

Return Arguments

| | | | |
|---|--------------------|-------------|-------------------------------|
| 1 | SA Report Ref List | Report List | The list of selected reports. |
|---|--------------------|-------------|-------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The list was returned successfully. |
| FAILURE | The user hit ESC or cancelled the selection. |

Remarks

None.

Make a Picture Name Ref List

Creates an empty picture name reference list.

Input Arguments

None.

Return Arguments

| | | | |
|---|----------------------------------|-------------------|--|
| 0 | Collection Picture Name Ref List | Picture Name List | The empty picture name reference list. |
|---|----------------------------------|-------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Invert Transform

This function has been implemented such that $T_{input} * T_{inverted} = T_{identity}$. It can be useful for placing objects back in a starting location after applying an initial known transform.

Input Arguments

| | | | |
|---|-----------|-----------|---|
| 0 | Transform | Transform | The transform to convert (T_{input}). |
|---|-----------|-----------|---|

Return Arguments

| | | | |
|---|-----------|-------------------|--|
| 1 | Transform | Inverse Transform | The resulting inverse transform ($T_{inverted}$) |
|---|-----------|-------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The Inverse transform was computed successfully |
| FAILURE | The input transform was incorrect |

Remarks

None.

Make a Report Items Ref List

Creates an empty report items reference list.

Input Arguments

None.

Return Arguments

| | | | |
|---|-----------------------|-------------------|--|
| 0 | Report Items Ref List | Report Items List | The empty report items reference list. |
|---|-----------------------|-------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Transform from Doubles (Fixed XYZ)

Creates a transform operator from specified XYZ/RxRyRz values.

Input Arguments

| | | | |
|---|--------|------------|---------------------------------|
| 0 | Double | X | The x value for the transform. |
| 1 | Double | Y | The y value for the transform. |
| 2 | Double | Z | The z value for the transform. |
| 3 | Double | Rx (Roll) | The Rx value for the transform. |
| 4 | Double | Ry (Pitch) | The Ry value for the transform. |
| 5 | Double | Rz (Yaw) | The Rz value for the transform. |

Return Arguments

| | | | |
|---|-----------|---------------------|--|
| 6 | Transform | Resultant Transform | The resulting transform constructed from the elements above. |
|---|-----------|---------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Transform from Doubles (Matrix Elements)

Creates a transform from the 4x4 matrix elements.

Input Arguments

| | | | |
|----|--------|------|--------------------------|
| 0 | Double | r0c0 | Row 0, column 0 element. |
| 1 | Double | r0c1 | Row 0, column 1 element. |
| 2 | Double | r0c2 | Row 0, column 2 element. |
| 3 | Double | r0c3 | Row 0, column 3 element. |
| 4 | Double | r1c0 | Row 1, column 0 element. |
| 5 | Double | r1c1 | Row 1, column 1 element. |
| 6 | Double | r1c2 | Row 1, column 2 element. |
| 7 | Double | r1c3 | Row 1, column 3 element. |
| 8 | Double | r2c0 | Row 2, column 0 element. |
| 9 | Double | r2c1 | Row 2, column 1 element. |
| 10 | Double | r2c2 | Row 2, column 2 element. |
| 11 | Double | r2c3 | Row 2, column 3 element. |
| 12 | Double | r3c0 | Row 3, column 0 element. |
| 13 | Double | r3c1 | Row 3, column 1 element. |
| 14 | Double | r3c2 | Row 3, column 2 element. |
| 15 | Double | r3c3 | Row 3, column 3 element. |

Return Arguments

| | | | |
|----|-----------|---------------------|--|
| 16 | Transform | Resultant Transform | The resulting transform constructed from the elements above. |
|----|-----------|---------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a World Transform Operator (from Transform and Scale)

Builds a world transform operator from a transform and a scale value.

Input Arguments

| | | | |
|---|-----------|-----------|--------------------------|
| 0 | Transform | Transform | The transform to use. |
| 1 | Double | Scale | The scale factor to use. |

Return Arguments

| | | | |
|---|--------------------------|--------------------------|---|
| 2 | World Transform Operator | World Transform Operator | The resulting world transform operator. |
|---|--------------------------|--------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Working Transform of Object (Fixed XYZ)

Obtains the transform of an object (in the working frame), reported using fixed XYZ notation.

Input Arguments

| | | | |
|---|------------------------|-------------|--|
| 0 | Collection Object Name | Object Name | The object whose transform should be obtained. |
|---|------------------------|-------------|--|

Return Arguments

| | | | |
|---|-----------|-----------|--|
| 1 | Transform | Transform | The transform of the specified object. |
|---|-----------|-----------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The object's transform was obtained successfully. |
| FAILURE | The specified object could not be found. |

Remarks

None.

Decompose Transform into Doubles (Fixed XYZ)

Obtains the individual numeric (double) values of a transform, using fixed XYZ format.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|------------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Rx (Roll) | The Rx-value of the transform. |
| 5 | Double | Ry (Pitch) | The Ry-value of the transform. |
| 6 | Double | Rz (Yaw) | The Rz-value of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Vectors (Fixed XYZ)

Obtains the position and orientation vectors from a Fixed XYZ transform value, expressed in the active coordinate frame.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|------------------------|--|
| 1 | Vector | Position in Working | The XYZ position for the transform. |
| 2 | Vector | Orientation in Working | The RxRyRz orientation of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Vectors (Origin and Axes)

Returns the origin and XYZ axes of a given transform.

Input Arguments

| | | | |
|---|-----------|-----------|-----------------------------|
| 0 | Transform | Transform | The transform to decompose. |
|---|-----------|-----------|-----------------------------|

Return Arguments

| | | | |
|---|--------|--------|--|
| 1 | Vector | Origin | The origin of the transform. |
| 2 | Vector | X Axis | The vector of the X axis of the transform. |
| 3 | Vector | Y Axis | The vector of the Y axis of the transform. |
| 4 | Vector | Z Axis | The vector of the Z axis of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This can be used to provide the X/Y/Z axes of any object, for example a frame, when its transform is known.

Decompose Transform into Doubles (Matrix Elements)

Converts a transformation into its individual 4x4 matrix elements.

Input Arguments

| | | | |
|---|-----------|-----------|-----------------------------|
| 0 | Transform | Transform | The transform to decompose. |
|---|-----------|-----------|-----------------------------|

Return Arguments

| | | | |
|----|--------|------|--------------------------|
| 1 | Double | r0c0 | Row 0, column 0 element. |
| 2 | Double | r0c1 | Row 0, column 1 element. |
| 3 | Double | r0c2 | Row 0, column 2 element. |
| 4 | Double | r0c3 | Row 0, column 3 element. |
| 5 | Double | r1c0 | Row 1, column 0 element. |
| 6 | Double | r1c1 | Row 1, column 1 element. |
| 7 | Double | r1c2 | Row 1, column 2 element. |
| 8 | Double | r1c3 | Row 1, column 3 element. |
| 9 | Double | r2c0 | Row 2, column 0 element. |
| 10 | Double | r2c1 | Row 2, column 1 element. |
| 11 | Double | r2c2 | Row 2, column 2 element. |
| 12 | Double | r2c3 | Row 2, column 3 element. |
| 13 | Double | r3c0 | Row 3, column 0 element. |
| 14 | Double | r3c1 | Row 3, column 1 element. |
| 15 | Double | r3c2 | Row 3, column 2 element. |
| 16 | Double | r3c3 | Row 3, column 3 element. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose World Transform Operator into Doubles (Fixed XYZ in World)

Obtains the individual numeric (double) values of a transform, using fixed XYZ format. The transform is expressed relative to the WORLD coordinate frame.

Input Arguments

| | | | |
|---|--------------------------|--------------------------------|---|
| 0 | World Transform Operator | Input World Transform Operator | The transform to decompose (relative to WORLD). |
|---|--------------------------|--------------------------------|---|

Return Arguments

| | | | |
|---|--------|------------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Rx (Roll) | The Rx-value of the transform. |
| 5 | Double | Ry (Pitch) | The Ry-value of the transform. |
| 6 | Double | Rz (Yaw) | The Rz-value of the transform. |
| 7 | Double | Scale | The scale of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Doubles (Euler XYZ)

Obtains the individual numeric (double) values of a transform, using Euler XYZ format.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|----------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Euler Rx | The Rx-value of the transform. |
| 5 | Double | Euler Ry | The Ry-value of the transform. |
| 6 | Double | Euler Rz | The Rz-value of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Doubles (Euler ZYX)

Obtains the individual numeric (double) values of a transform, using Euler ZYX format.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|----------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Euler Rz | The Rz-value of the transform. |
| 5 | Double | Euler Ry | The Ry-value of the transform. |
| 6 | Double | Euler Rx | The Rx-value of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Doubles (Euler ZYZ)

Obtains the individual numeric (double) values of a transform, using Euler ZYZ format.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|----------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Euler Rz | The Rz-value of the transform. |
| 5 | Double | Euler Ry | The Ry-value of the transform. |
| 6 | Double | Euler Rz | The Rz-value of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Transform into Doubles (Euler ZXZ)

Obtains the individual numeric (double) values of a transform, using Euler ZXZ format.

Input Arguments

| | | | |
|---|-----------|-----------------|-----------------------------|
| 0 | Transform | Input Transform | The transform to decompose. |
|---|-----------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|----------|--------------------------------|
| 1 | Double | X | The X-value of the transform. |
| 2 | Double | Y | The Y-value of the transform. |
| 3 | Double | Z | The Z-value of the transform. |
| 4 | Double | Euler Rz | The Rz-value of the transform. |
| 5 | Double | Euler Rx | The Rx-value of the transform. |
| 6 | Double | Euler Rz | The Rz-value of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose World Transform Operator into Vectors (Fixed XYZ in World)

Obtains the position and orientation vectors from a Fixed XYZ transform value, expressed in the World coordinate frame.

Input Arguments

| | | | |
|---|--------------------------|--------------------------------|-----------------------------|
| 0 | World Transform Operator | Input World Transform Operator | The transform to decompose. |
|---|--------------------------|--------------------------------|-----------------------------|

Return Arguments

| | | | |
|---|--------|------------------------|--|
| 1 | Vector | Position in Working | The XYZ position for the transform. |
| 2 | Vector | Orientation in Working | The RxRyRz orientation of the transform. |
| 3 | Double | Scale | The scale of the transform. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make a Vector from Doubles

Creates a vector from 3 values.

Input Arguments

| | | | |
|---|--------|---|-----------------------------|
| 0 | Double | X | The x value for the vector. |
| 1 | Double | Y | The y value for the vector. |
| 2 | Double | Z | The z value for the vector. |

Return Arguments

| | | | |
|---|--------|------------------|---|
| 3 | Vector | Resultant Vector | The resulting vector constructed from the elements above. |
|---|--------|------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Decompose Vector into Doubles

Extracts the three numeric components of a vector into double values.

Input Arguments

| | | | |
|---|--------|--------------|--------------------------|
| 0 | Vector | Input Vector | The vector to decompose. |
|---|--------|--------------|--------------------------|

Return Arguments

| | | | |
|---|--------|---|--------------------------------|
| 1 | Double | X | The X-component of the vector. |
| 2 | Double | Y | The y-component of the vector. |
| 3 | Double | Z | The z-component of the vector. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Split String into Two Strings

Separates a source string into two separate strings based on a supplied character index to define the dividing point.

Input Arguments

| | | | |
|---|---------|--------------------------|--|
| 0 | String | Input String | The string to split. |
| 1 | Integer | Dividing Character Index | The zero-based character of the split. |

Return Arguments

| | | | |
|---|--------|---------------|--------------------------------------|
| 2 | String | First String | The first of two resulting strings. |
| 3 | String | Second String | The second of two resulting strings. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The index is zero-based, meaning it starts at zero. The index specifically defines which character will be the first character in the second string. Therefore, an index of zero results in an empty first string and a second string containing the source string.

Make a Normalized Vector

Normalizes a vector. By definition, the resulting vector has the same direction as the source vector, but has a magnitude of 1.

Input Arguments

| | | | |
|---|--------|--------------|---------------------------------|
| 0 | Vector | Input Vector | The source vector to normalize. |
|---|--------|--------------|---------------------------------|

Return Arguments

| | | | |
|---|--------|------------------|----------------------------------|
| 1 | Vector | Resultant Vector | The resulting normalized vector. |
|---|--------|------------------|----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Convert to Euler Angles from Fixed Angles

Converts a transform expressed in Fixed XYZ form to a transform expressed in Euler form.

Input Arguments

| | | | |
|---|------------------|--------------------------------|--|
| 0 | Transform | Input Transform (Fixed Angles) | The source (Fixed XYZ) transform to convert. |
| 1 | Euler Angle Type | Pick Euler Angle Type | The angle type (XYZ, ZYX, or ZYZ) to convert to. |

Return Arguments

| | | | |
|---|-----------|----------------------------------|------------------------------------|
| 2 | Transform | Output Transform of Euler Angles | The resulting converted transform. |
|---|-----------|----------------------------------|------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Projection Options

Creates projection options for a vector group.

Input Arguments

| 0 | Projection Output Type | Output Type | The type of projection to output. |
|---|------------------------|---|---|
| 1 | Boolean | Ignore Edge Projections | Indicates whether edge projections should be ignored. |
| 2 | Boolean | Probe Offsets - Override Target Values? | Indicates whether the offsets should be overridden. |
| 3 | Double | Probe Offsets - Override Value | The value with which to override probe offsets. |
| 4 | Double | Extra Material Thickness | Extra material thickness to apply, if any. |

Return Arguments

| | | | |
|---|--------------------|----------------------------------|------------------------------------|
| 5 | Projection Options | Output Transform of Euler Angles | The resulting converted transform. |
|---|--------------------|----------------------------------|------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Axis Identifier from String

Converts a string into an axis identifier (for use with a frame construction command, for instance).

Input Arguments

| | | | |
|---|--------|--------------|------------------------------|
| 0 | String | String Input | The input string to convert. |
|---|--------|--------------|------------------------------|

Return Arguments

| | | | |
|---|-----------------|---------------------------|--|
| 1 | Axis Identifier | Resultant Axis Identifier | The resulting axis identifier, for use in a frame construction command (for instance). |
|---|-----------------|---------------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The identifier was successfully created. |
| FAILURE | An invalid input string was supplied. |

Remarks

Allowable inputs are +X axis, -X axis, +Y axis, -Y axis, +Z axis, and -Z axis. The input string is not case-sensitive.

Make UDP Settings

Builds a list of UPD settings for reference.

Input Arguments

| | | | |
|---|---------|---|--|
| 0 | Boolean | Transmit Watch Window Text Over Network | Sets transmit status. |
| 1 | Boolean | Send to Entire Subnet | Sets subnet transmit check box status. |
| 2 | String | Computer Name or IP | Sets the computer name or IP to use. |
| 3 | integer | Port | Port used for broadcast. |

Return Arguments

| | | | |
|---|--------------|-------------------------------|------------------------|
| 4 | UDP Settings | UDP Network Transmit Settings | The resulting settings |
|---|--------------|-------------------------------|------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The identifier was successfully created. |
| FAILURE | An invalid input string was supplied. |

Remarks

None.

This Page Intentionally Left Blank.

8

ANALYSIS OPERATIONS

Set Object Reporting Frame

This command sets an objects reporting frame to a specified frame.

Input Arguments

| 0 | Collection Object Name | Object Name | Object to be edited |
|---|------------------------|--------------|--------------------------------|
| 1 | Collection Object Name | Report Frame | Reporting frame to be assigned |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame as been set. |
| Failure | The object or frame could not be found. |

Remarks

None.

Get Object Reporting Frame

This command sets an objects reporting frame to a specified frame.

Input Arguments

| | | | |
|---|------------------------|-------------|---------------------|
| 0 | Collection Object Name | Object Name | Object to be edited |
|---|------------------------|-------------|---------------------|

Return Arguments

| | | | |
|---|------------------------|--------------|--------------------------------|
| 1 | Collection Object Name | Report Frame | Reporting frame to be assigned |
|---|------------------------|--------------|--------------------------------|

Returned Status

| | |
|---------|-----------------------------------|
| SUCCESS | The reporting frame was returned. |
| Failure | The object could not be found. |

Remarks

None.

Re-Compute Calculated Items

Will recompute targets from shots, hidden points, or relationships..

Input Arguments

| | | | |
|---|---------|--------------------|----------------------------|
| 0 | Boolean | Targets from Shots | Recompute measured points. |
| 1 | Boolean | Hidden Points | Recompute hidden points. |
| 2 | Boolean | Relationships | Recompute relationships. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Number of Collections

Returns the number of collections in the current job file.

Input Arguments

None.

Return Arguments

| | | | |
|---|---------|-------------|----------------------------|
| 0 | Integer | Total Count | The number of collections. |
|---|---------|-------------|----------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get i-th Collection Name

Returns the name of the i-th collection in the current job file.

Input Arguments

| | | | |
|---|---------|------------------|---------------------------|
| 0 | Integer | Collection Index | The collection to return. |
|---|---------|------------------|---------------------------|

Return Arguments

| | | | |
|---|-----------------|----------------|----------------------------------|
| 1 | Collection Name | Resultant Name | The name of the i-th collection. |
|---|-----------------|----------------|----------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The specified collection name was returned successfully. |
| FAILURE | The specified collection does not exist. |

Remarks

Collections are ordered starting from zero as they appear in the tree. The topmost collection is considered collection 0.

Get i-th Report Item From Report Items Ref List

Returns the name of the i-th report item in the reference list.

Input Arguments

| | | | |
|---|-----------------------|-------------------|--|
| 0 | Report Items Ref List | Report Items List | The report item list used as reference |
| 1 | Integer | Report Item Index | The index of the returned report item. |

Return Arguments

| | | | |
|---|------------------------|-------------|---|
| 2 | Collection Object Name | Report Item | The name of the i-th report item in the list. |
|---|------------------------|-------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The specified report item name was returned successfully. |
| FAILURE | The specified report item ref list does not exist. |

Remarks

None.

Rename points based on proximity to reference points

Renames a set of points in a point group based on their proximity to a set of reference points in another point group.

Input Arguments

| | | | |
|---|------------------------|------------------------------|---|
| 0 | Collection Object Name | Reference Group Name | The group containing the reference points to consider. |
| 1 | Collection Object Name | Group To Rename Points | The group containing the points to rename. |
| 2 | Double | Proximity Threshold | A point must be within this distance from a reference point to be considered for renaming. |
| 3 | Boolean | Verify Results? | Indicates whether a dialog should be displayed showing the results of the renaming operation. |
| 4 | Boolean | Rename All Proximate Points? | True will rename all points within the specified proximity threshold. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The specified group was renamed successfully. |
| FAILURE | The reference group or group to rename was not found, or no points were renamed. |

Remarks

A proximity threshold of zero indicates that the points to rename must lie exactly on top of the reference points. If more than one point lies within the proximity threshold to a reference point, then only the closest point will be renamed by default. That can be changed using the *Rename All Proximate Points* option.

Rename points based on inter-point distance to reference points

Renames a set of points in a point group based on their inter-point spacing relative to a reference group.

Input Arguments

| | | | |
|---|------------------------|------------------------|--|
| 0 | Collection Object Name | Reference Group Name | The group containing the reference points to consider. |
| 1 | Collection Object Name | Group To Rename Points | The group containing the points to rename. |
| 2 | Double | Distance Threshold | The difference between a set of points in the reference group and the group to rename must be within this difference in order to be considered for renaming. |
| 3 | Boolean | Verify Results? | Indicates whether a dialog should be displayed showing the results of the renaming operation. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The specified group was renamed successfully. |
| FAILURE | The reference group or group to rename was not found, or no points were renamed. |

Remarks

A proximity threshold of zero indicates that the inter-point distances in the reference group must exactly match the inter-point distances in the group to rename. For best results, the distance threshold should be set as small as possible, while still being larger than the largest expected error between a given reference point and its corresponding point.

Is Object of Type

Determines whether a supplied object matches a given type (circle, sphere, point group, etc).

Input Arguments

| | | | |
|---|------------------------|-------------|-------------------------|
| 0 | Collection Object Name | Object Name | The object to consider. |
| 1 | Object Type | Object Type | The type to consider. |

Return Arguments

| | | | |
|---|---------|-----------|--|
| 2 | Boolean | Resultant | Indicates whether the object is of the given type. |
|---|---------|-----------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The object was compared to the type successfully. |
| FAILURE | The reference object could not be found. |

Remarks

Point Cloud selection can be refined as follows:

Cloud = Cloud, Scan Stripe Cloud, or Cross Section Cloud

Scan Stripe Cloud = Scan Stripe Cloud or Cross Section Cloud (but not basic clouds)

Cross Section Cloud = allows Cross Section Cloud selection only

Get Number of Instruments in Collection Instrument Ref List

Returns the number of instruments in a collection instrument reference list.

Input Arguments

| | | | |
|---|-----------------------------------|--------------------------------------|-----------------------------------|
| 0 | Collection Instrument ID Ref List | Collection Instrument Reference List | The list of instruments to count. |
|---|-----------------------------------|--------------------------------------|-----------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The number of instruments in the list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Number of Point Names in Point Name Ref List

Returns the number of points in a point name reference list.

Input Arguments

| | | | |
|---|---------------------|-----------------|--------------------|
| 0 | Point Name Ref List | Point Name List | The list to count. |
|---|---------------------|-----------------|--------------------|

Return Arguments

| | | | |
|---|---------|-------------|---|
| 1 | Integer | Total Count | The total number of points in the list. |
|---|---------|-------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The points were counted successfully. |
| FAILURE | The supplied point list was not found. |

Remarks

None.

Get Number of Objects in Collection Object Name Ref List

Returns the number of objects in a collection object name reference list.

Input Arguments

| | | | |
|---|---------------------------------|------------------|---|
| 0 | Collection Object Name Ref List | Object Name List | The reference list of collection object names to count. |
|---|---------------------------------|------------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of objects in the list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The objects were counted successfully. |
| FAILURE | The supplied collection object name reference list was not found. |

Remarks

None.

Get Number of Points in Group

Returns the number of points in a point group.

Input Arguments

| | | | |
|---|------------------------|------------|---|
| 0 | Collection Object Name | Group Name | The name of the group containing the points to count. |
|---|------------------------|------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of points in the point group. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The points were counted successfully. |
| FAILURE | The supplied point group was not found. |

Remarks

None.

Get Number of Frames In Frame Set

Returns the number of Frames in a Frame Set.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Frame Set Container | The name of the Frame Set containing the frames to count. |
|---|------------------------|---------------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of frames in the frame set. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The frames were counted successfully. |
| FAILURE | The supplied Frame Set was not found. |

Remarks

None.

Get Number of Points In Point Set

Returns the number of Points in a Point Set.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Point Set Container | The name of the Point Set containing the Points to count. |
|---|------------------------|---------------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of points in the point group. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The points were counted successfully. |
| FAILURE | The supplied Point Set was not found. |

Remarks

None.

Get Number of Strings in String Ref List

Returns the number of strings in a string reference list.

Input Arguments

| | | | |
|---|-----------------|-------------|--|
| 0 | String Ref List | String List | The string reference list containing the strings to count. |
|---|-----------------|-------------|--|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of strings in the list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The strings were counted successfully. |
| FAILURE | The supplied string reference list was not found. |

Remarks

None.

Get Number of Vectors in Vector Group

Returns the number of vectors in a vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group containing the vectors to count. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of vectors in the vector group. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The vectors were counted successfully. |
| FAILURE | The supplied vector group could not be found. |

Remarks

None.

Get Number of Vectors in Vector Name Ref List

Returns the number of vectors in a vector name ref list.

Input Arguments

| | | | |
|---|----------------------|------------------|---|
| 0 | Vector Name Ref List | Vector Name List | The name of the vector name ref list containing the vectors to count. |
|---|----------------------|------------------|---|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The total number of vectors in the vector name ref list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The vectors were counted successfully. |
| FAILURE | The supplied vector name ref list could not be found. |

Remarks

None.

Get Number of characters in a string

Returns the number of characters in a string.

Input Arguments

| | | | |
|---|--------|--------------------|--|
| 0 | String | String in question | The string whose characters will be counted. |
|---|--------|--------------------|--|

Return Arguments

| | | | |
|---|---------|-----------------|---|
| 1 | Integer | Character Count | The number of characters in the specified string. |
|---|---------|-----------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Vector Group Properties

Obtains the following properties of a vector group:

- Total number of vectors
- Number of vectors in tolerance
- Number of vectors out of tolerance
- Percentage of vectors in tolerance
- Percentage of vectors out of tolerance
- Absolute max magnitude
- Absolute min magnitude
- Max magnitude
- Min magnitude
- Standard Deviation
- Standard Deviation Mean Zero
- Average Magnitude
- Average of Absolute Magnitude
- High Tolerance Value
- Low Tolerance Value.

Input Arguments

| | | | |
|---|------------------------|-------------------|----------------------------------|
| 0 | Collection Object Name | Vector Group Name | The vector group to be examined. |
|---|------------------------|-------------------|----------------------------------|

Return Arguments

| | | | |
|----|---------|------------------------------|---|
| 1 | Integer | Total Vectors | Total number of vectors in the vector group. |
| 2 | Integer | Vectors In Tolerance | Number of vectors in tolerance. |
| 3 | Integer | Vectors Out Of Tolerance | Number of vectors out of tolerance. |
| 4 | Double | % Vectors In Tolerance | Percentage of vectors in tolerance. |
| 5 | Double | % Vectors Out Of Tolerance | Percentage of vectors out of tolerance. |
| 6 | Double | Absolute Max Magnitude | Absolute maximum magnitude. |
| 7 | Double | Absolute Min Magnitude | Absolute minimum magnitude. |
| 8 | Double | Max Magnitude | Maximum magnitude. |
| 9 | Double | Min Magnitude | Minimum magnitude. |
| 10 | Double | Standard Deviation | Standard deviation of the vector deviations. |
| 11 | Double | Standard Deviation Mean Zero | Standard deviation of the vectors, centered about zero. |
| 12 | Double | Average Magnitude | Average magnitude of the vectors. |
| 13 | Double | Avg of Abs Magnitude | Average of the absolute magnitudes. |
| 14 | Double | High Tolerance Value | The high tolerance set for the vector group. |

| | | | |
|----|--------|---------------------|---|
| 15 | Double | Low Tolerance Value | The low tolerance set for the vector group. |
|----|--------|---------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The vector properties were obtained successfully. |
| FAILURE | The vector group could not be found. |

Remarks

None.

Auto-Range and Set Vector Group Colorization (Selected)

Sets the colorization and display options for selected vector groups and auto-ranges their colorization saturation limits.

Input Arguments

| | | | |
|---|---------------------------------------|---------------------------------------|---|
| 0 | Collection Vector Group Name Ref List | Vector Groups to be Set | The vector groups whose display and colorization properties should be modified. |
| 1 | Boolean | Treat Individually? | Specifies whether each vector group should be treated individually when applying the high and low saturation limits (from the auto-range operation). See Remarks for details. |
| 2 | Colorization Options | Colorization Options (Uses Mode Only) | Specifies all of the vector group display and colorization options. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The vector group's display/colorization properties were set successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the vector groups could not be found. |
| FAILURE | None of the supplied vector groups could be found. |

Remarks

When the Treat Individually argument is set to TRUE, each supplied vector group's max and min will be calculated and used for the vector group's high and low colorization saturation limits. When set to FALSE, the max and min of the entire list of vector groups will be calculated, and this will be used to apply the high and low saturation limits. Leaving this option set to FALSE ensures that the coloring across all vector groups is equivalent.

Auto-Range and Set Vector Group Colorization (All)

Sets the colorization and display options for all vector groups and auto-ranges their colorization saturation limits.

Input Arguments

| | | | |
|---|----------------------|---------------------------------------|---|
| 0 | Boolean | Treat Individually? | Specifies whether each vector group should be treated individually when applying the high and low saturation limits (from the auto-range operation). See Remarks for details. |
| 1 | Colorization Options | Colorization Options (Uses Mode Only) | Specifies all of the vector group display and colorization options. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

When the Treat Individually argument is set to TRUE, each vector group's max and min will be calculated and used for the vector group's high and low colorization saturation limits. When set to FALSE, the max and min of all vector groups will be calculated, and this will be used to apply the high and low saturation limits to all vector groups. Leaving this option set to FALSE ensures that the coloring across all vector groups is equivalent.

Set Vector Group Colorization Options (Selected)

Sets the colorization and display options for selected vector groups.

Input Arguments

| | | | |
|---|---------------------------------------|-------------------------|---|
| 0 | Collection Vector Group Name Ref List | Vector Groups to be Set | The vector groups whose display and colorization properties should be modified. |
| 1 | Colorization Options | Colorization Options | Specifies all of the vector group display and colorization options. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The vector group's display/colorization properties were set successfully. |
| PARTIAL SUCCESS | At least one (but not all) of the vector groups could not be found. |
| FAILURE | None of the supplied vector groups could be found. |

Remarks

None.

Set Vector Group Colorization Options (All)

Sets the colorization and display options for all vector groups.

Input Arguments

| | | | |
|---|----------------------|----------------------|---|
| 0 | Colorization Options | Colorization Options | Specifies all of the vector group display and colorization options. |
|---|----------------------|----------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Vector Group Colorization Options

Retrieves the colorization and display options for the specified vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group from which the colorization options should be retrieved. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|----------------------|----------------------|---|
| 1 | Colorization Options | Colorization Options | The colorization options on the specified vector group. |
|---|----------------------|----------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The colorization options were retrieved successfully. |
| FAILURE | The vector group was not found. |

Remarks

None.

Create Point Uncertainty Fields

Creates uncertainty fields for a list of points.

Input Arguments

| | | | |
|---|---------------------|-------------------|--|
| 0 | Point Name Ref List | Point Name List | The list of points for which to create uncertainty fields. |
| 1 | Integer | Number of Samples | The number of uncertainty samples to create. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The uncertainty fields were created successfully. |
| FAILURE | The points were not found. |

Remarks

None.

Coordinate

Obtains the coordinate of a specified point in the active coordinate frame, represented in Cartesian coordinates.

Input Arguments

| | | | |
|---|------------|------------|--|
| 0 | Point Name | Point Name | The name of the point whose coordinates should be retrieved. |
|---|------------|------------|--|

Return Arguments

| | | | |
|---|--------|-----------------------|---|
| 1 | Vector | Vector Representation | The coordinates of the specified point, represented in vector (XYZ) format. |
| 2 | Double | X Value | The X-coordinate of the point. |
| 3 | Double | Y Value | The Y-coordinate of the point. |
| 4 | Double | Z Value | The Z-coordinate of the point. |

Returned Status

| | |
|---------|--|
| SUCCESS | The point's coordinates were retrieved successfully. |
| FAILURE | The specified point could not be found. |

Remarks

Note that since this command fails if the point does not exist, this command can be used to determine if a point exists.

Get Point Coordinate (Cylindrical)

Retrieves the coordinate of a specified point in the active coordinate frame, represented in Cylindrical coordinates.

Input Arguments

| | | | |
|---|------------|------------|--|
| 0 | Point Name | Point Name | The name of the point whose coordinates should be retrieved. |
|---|------------|------------|--|

Return Arguments

| | | | |
|---|----------------------------|----------------|-------------------------------|
| 0 | File Path or Embedded File | STEP File Path | The path of the file to save. |
|---|----------------------------|----------------|-------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The point's coordinates were retrieved successfully. |
| FAILURE | The specified point could not be found. |

Remarks

Note that since this command fails if the point does not exist, this command can be used to determine if a point exists.

Get Point Coordinate (Polar)

Obtains the coordinate of a specified point in the active coordinate frame, represented in Polar coordinates.

Input Arguments

| | | | |
|---|------------|------------|--|
| 0 | Point Name | Point Name | The name of the point whose coordinates should be retrieved. |
|---|------------|------------|--|

Return Arguments

| | | | |
|---|--------|-----------------------|---|
| 1 | Vector | Vector Representation | The coordinates of the specified point, represented in vector (XYZ) format. |
| 2 | Double | Radius Value | The radius of the point. |
| 3 | Double | Theta Value | The theta value of the point (in decimal degrees). |
| 4 | Double | Phi Value | The phi value of the point (in decimal degrees). |

Returned Status

| | |
|---------|--|
| SUCCESS | The point's coordinates were retrieved successfully. |
| FAILURE | The specified point could not be found. |

Remarks

Note that since this command fails if the point does not exist, this command can be used to determine if a point exists.

Get Point Properties

Obtains the offset and uncertainty values for a specified point.

Input Arguments

| | | | |
|---|------------|------------|---|
| 0 | Point Name | Point Name | The name of the point whose properties should be retrieved. |
|---|------------|------------|---|

Return Arguments

| | | | |
|---|------------------|--------------------|---|
| 1 | Double | Planar Offset | The planar offset of the point. |
| 2 | Double | Radial Offset | The radial offset of the point. |
| 3 | Double | Ux | The x-component of uncertainty for the point. |
| 4 | Double | Uy | The y-component of uncertainty for the point. |
| 5 | Double | Uz | The z-component of uncertainty for the point. |
| 6 | Double | Umag | The magnitude of uncertainty for the point. |
| 7 | Vector Tolerance | Position Tolerance | The position tolerance of the point. |
| 8 | Vector | Component Weights | The weights for the point. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point's properties were retrieved successfully. |
| FAILURE | The specified point could not be found. |

Remarks

Note that since this command fails if the point does not exist, this command can be used to determine if a point exists.

Get Point Tolerance

Obtains the tolerance value for a specified point.

Input Arguments

| | | | |
|---|------------|------------|--|
| 0 | Point Name | Point Name | The name of the point whose tolerance should be retrieved. |
|---|------------|------------|--|

Return Arguments

| | | | |
|----|------------------|-------------------------|---|
| 1 | Boolean | Use High X Tolerance? | Whether to use high X tolerance. |
| 2 | Double | High X Tolerance | Specify the high X tolerance value. |
| 3 | Boolean | Use High Y Tolerance? | Whether to use high Y tolerance. |
| 4 | Double | High Y Tolerance | Specify the high Y tolerance value. |
| 5 | Boolean | Use High Z Tolerance? | Whether to use high Z tolerance. |
| 6 | Double | High Z Tolerance | Specify the high Z tolerance value. |
| 7 | Boolean | Use High Mag Tolerance? | Whether to use high Mag tolerance. |
| 8 | Double | High Mag Tolerance | Specify the high Mag tolerance value. |
| 9 | Boolean | Use Low X Tolerance? | Whether to use low X tolerance.. |
| 10 | Double | Low X Tolerance | Specify the low X tolerance value. |
| 11 | Boolean | Use Low Y Tolerance? | Whether to use lowY tolerance. |
| 12 | Double | Low Y Tolerance | Specify the low Y tolerance value. |
| 13 | Boolean | Use Low Z Tolerance? | Whether to use low Z tolerance. |
| 14 | Double | Low Z Tolerance | Specify the low Z tolerance value. |
| 15 | Boolean | Use Low Mag Tolerance? | Specify whether to use low Mag tolerance. |
| 16 | Double | Low Mag Tolerance | Specify the low Mag tolerance value. |
| 1 | Vector Tolerance | Vector Tolerance | Specify the vector tolerance value. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point's tolerances were retrieved successfully. |
| FAILURE | The specified point could not be found. |

Remarks

Note that since this command fails if the point does not exist, this command can be used to determine if a point exists.

Set Point Properties

Sets the planar and radial offsets for a set of one or more points.

Input Arguments

| | | | |
|---|---------------------|-----------------|---|
| 0 | Point Name Ref List | Point Name List | The list of points whose offsets should be set. |
| 1 | Double | Planar Offset | The planar offset for the points. |
| 2 | Double | Radial Offset | The radial offset for the points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | All points that exist had their offsets applied accordingly. |
| FAILURE | None of the specified points could be found. |

Remarks

None.

Get Point To Point Distance

Obtains the 3D distance between two points.

Input Arguments

| | | | |
|---|------------|--------------|-------------------------------|
| 0 | Point Name | First Point | The first point to consider. |
| 1 | Point Name | Second Point | The second point to consider. |

Return Arguments

| | | | |
|---|--------|-----------------------|---|
| 2 | Vector | Vector Representation | The vector from the first point to the second point. |
| 3 | Double | X Value | The x-component of the distance from the first point to the second. |
| 4 | Double | Y Value | The y-component of the distance from the first point to the second. |
| 5 | Double | Z Value | The z-component of the distance from the first point to the second. |
| 6 | Double | Magnitude | The magnitude of the distance between the two points. |

Returned Status

| | |
|---------|---|
| SUCCESS | The distance was calculated successfully. |
| FAILURE | One or both points could not be found. |

Remarks

None.

Get Point To Line Distance

Obtains the 3D distance between a point and a line.

Input Arguments

| | | | |
|---|------------------------|-------|------------------------|
| 0 | Point Name | Point | The point to consider. |
| 1 | Collection Object Name | Line | The line to consider. |

Return Arguments

| | | | |
|---|--------|-----------------------|---|
| 2 | Vector | Vector Representation | The vector from the point to the line. |
| 3 | Double | X Value | The x-component from the point to the line. |
| 4 | Double | Y Value | The y-component from the point to the line. |
| 5 | Double | Z Value | The z-component from the point to the line. |
| 6 | Double | Magnitude | The magnitude of the distance from the point to the line. |

Returned Status

| | |
|---------|---|
| SUCCESS | The distance was calculated successfully. |
| FAILURE | The point or line could not be found. |

Remarks

None.

Get i-th Point Name From Point Name Ref List

Returns the name of a point at a specified index in a point name reference list.

Input Arguments

| | | | |
|---|---------------------|------------------|--------------------------------------|
| 0 | Point Name Ref List | Point Name List | The list of points to consider. |
| 1 | Integer | Point Name Index | The index of the point to determine. |

Return Arguments

| | | | |
|---|------------|----------------------|---|
| 2 | Point Name | Resulting Point Name | The full collection:: group:: point name of the point at the specified index. |
| 3 | String | Point Name Only | Returns the point name only. |
| 4 | Vector | Vector in Working | The coordinates of the returned point. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first point in a ref list is at index 0.

Get i-th Point Name From Point Name Ref List (Iterator)

Returns the name of a point at a specified index in a point name reference list. This command is an iterator, which means it is a shortcut to a traditional loop. Iterators proceed sequentially, one-by-one, from a starting index, until the end of the list. For more information on iterators, see Loop Structures in the Fundamental Terms Chapter.

Input Arguments

| | | | |
|---|---------------------|-----------------------------|---|
| 0 | Point Name Ref List | Point Name List | The list of points to consider. |
| 1 | Integer | Point Name Index | The starting index of the point to consider. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to after iterating through the last point in the list. |

Return Arguments

| | | | |
|---|------------|----------------------|--|
| 3 | String | Collection | The collection containing the current point. |
| 4 | String | Group | The point group containing the current point. |
| 5 | String | Target | The point name of the current point. |
| 6 | Point Name | Resulting Point Name | The name of the current point as a Point Name data type. |
| 7 | Vector | Vector in Working | The coordinates of the returned point. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first point in a ref list is at index 0.

Get i-th Object From Collection Object Name Ref List

Returns the name of an object at a specified index in a collection object name reference list.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--------------------------------------|
| 0 | Collection Object Name Ref List | Object Name List | The list of objects to consider. |
| 1 | Integer | Object Index | The index of the object to consider. |

Return Arguments

| | | | |
|---|------------------------|------------------|--|
| 2 | Collection Object Name | Resultant Object | The collection object name of the object at the specified index. |
|---|------------------------|------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The object name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first object in a ref list is at index 0.

Get i-th Object From Collection Object Name Ref List (Iterator)

Returns the name of an object at a specified index in a collection object name reference list. This command is an iterator, which means it is a shortcut to a traditional loop. Iterators proceed sequentially, one-by-one, from a starting index, until the end of the list. For more information on iterators, see Iterators.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name Ref List | Object Name List | The list of objects to consider. |
| 1 | Integer | Object Index | The starting index of the object to consider. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to after iterating through the last object in the list. |

Return Arguments

| | | | |
|---|------------------------|------------------|---|
| 3 | String | Collection | The collection containing the object at the specified index. |
| 4 | String | Object | The name of the object at the specified index. |
| 5 | String | Object Type | The type of the object at the specified index. |
| 6 | Collection Object Name | Resultant Object | The object at the specified index, as a collection object name data type. |

Returned Status

| | |
|---------|---|
| SUCCESS | The object name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first object in a ref list is at index 0.

Get i-th Instrument From Collection Instrument Ref List

Returns the name of an object at a specified index in a collection object name reference list.

Input Arguments

| | | | |
|---|-----------------------------------|--------------------------------------|--|
| 0 | Collection Instrument ID Ref List | Collection Instrument Reference List | The list of instruments to consider. |
| 1 | Integer | Index | The index of the instrument to consider. |

Return Arguments

| | | | |
|---|--------------------------|----------------------|--|
| 2 | Collection Instrument ID | Resulting Instrument | The ID of the instrument at the specified index. |
|---|--------------------------|----------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument ID was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first instrument ID in a ref list is at index 0.

Get i-th Point From Group

Returns the point name and coordinates for the i-th point in a point group.

Input Arguments

| | | | |
|---|------------------------|-------------|-------------------------------------|
| 0 | Collection Object Name | Group Name | The name of the point group. |
| 1 | Integer | Point Index | The index of the point to retrieve. |

Return Arguments

| | | | |
|---|------------|----------------------|---|
| 0 | Point Name | Resulting Point Name | The i-th point from the group in full collection::group:: target format |
| 1 | String | Point Name only | Point name only |
| 2 | Vector | Vector in Working | Point Coordinates in the current frame. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point was returned successfully. |
| FAILURE | An invalid point index or point group was supplied. |

Remarks

Point indices are zero based, so the first point in the group is at index 0.

Get Timestamp for i-th Frame in Frame Set

Returns the timestamp (in decimal seconds) saved in the frame that matches the Frame Set index. This value represents the time from the beginning of the scan.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Frame Set | The name of the Frame Set. |
| 1 | Integer | Frame Set Index | The index of the Frame Set to retrieve. |

Return Arguments

| | | | |
|---|--------|-----------|--------------------------------|
| 0 | Double | Timestamp | The saved time stamp (seconds) |
|---|--------|-----------|--------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The timestamp was returned successfully. |
| FAILURE | An invalid Frame index or Frame Set was supplied. |

Remarks

Frame indices are zero based, so the first frame in the Frame Set has an index of 0.

Get Transform for i-th Frame in Frame Set

Returns the transform saved in the frame that matches the Frame Set index.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Frame Set | The name of the Frame Set. |
| 1 | Integer | Frame Set Index | The index of the Frame Set to retrieve. |

Return Arguments

| | | | |
|---|-----------|----------------------|--|
| 0 | Transform | Transform in Working | Returned transform from the selected frame |
|---|-----------|----------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The transform was returned successfully. |
| FAILURE | An invalid Frame index or Frame Set was supplied. |

Remarks

Frame indices are zero based, so the first frame in the Frame Set has an index of 0.

Get Timestamp for i-th Point in Point Set

Returns the timestamp saved in the point that matches the Point Set index.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Point Set | The name of the Point Set. |
| 1 | Integer | Point Set Index | The index of the Point Set to retrieve. |

Return Arguments

| | | | |
|---|--------|-----------|---|
| 0 | Double | Timestamp | Returned timestamp from the selected point. |
|---|--------|-----------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The timestamp was returned successfully. |
| FAILURE | An invalid point index or Point Set was supplied. |

Remarks

Point indices are zero based, so the first point in the Point Set has an index of 0.

Get Coordinate for i-th Point in Point Set

Returns the name and coordinate saved in the point that matches the Point Set index.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Point Set | The name of the Point Set. |
| 1 | Integer | Point Set Index | The index of the Point Set to retrieve. |

Return Arguments

| | | | |
|---|--------|------------|---|
| 0 | String | Point Name | Returned name from the selected Point. |
| 0 | Double | Timestamp | Returned timestamp from the selected point. |

Returned Status

| | |
|---------|---|
| SUCCESS | The name and coordinate were returned successfully. |
| FAILURE | An invalid point index or Point Set was supplied. |

Remarks

Point indices are zero based, so the first point in the Point Set has an index of 0.

Get i-th Vector From Vector Group

Retrieves the vector information of the i-th vector in a vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group in which the desired vector is located. |
| 1 | Integer | Vector Index | The index of the vector to retrieve. |

Return Arguments

| | | | |
|---|--------|----------------------------|---|
| 2 | String | Vector Name | The name of the retrieved vector. |
| 3 | Vector | Begin in Working | The coordinates of the start (tail) of the vector, in working coordinates. |
| 4 | Vector | End in Working | The coordinates of the end (head) of the vector, in working coordinates. |
| 5 | Vector | Total Delta in Working | The delta from the start to the end of the vector, in working coordinates. |
| 6 | Vector | ijk Unit Vector in Working | The normalized (ijk) delta from the start to the end of the vector, in working coordinates. |
| 7 | Double | Magnitude | The magnitude of the vector. |

Returned Status

| | |
|---------|---|
| SUCCESS | The vector was retrieved successfully. |
| FAILURE | The vector group was not found, or an invalid index was supplied. |

Remarks

None.

Get i-th Vector From Vector Name Ref List

Retrieves the vector information of the i-th vector in a vector name ref list.

Input Arguments

| | | | |
|---|----------------------|------------------|--|
| 0 | Vector Name Ref List | Vector Name List | The name of the vector name ref list in which the desired vector is located. |
| 1 | Integer | Vector Index | The index of the vector to retrieve. |
| | | | |

Return Arguments

| | | | |
|---|------------------------|----------------------------|---|
| 2 | Collection Object Name | Vector Group Name | The vector group where the vector is located. |
| 3 | String | Vector Name | The name of the retrieved vector. |
| 4 | Vector | Begin in Working | The coordinates of the start (tail) of the vector, in working coordinates. |
| 5 | Vector | End in Working | The coordinates of the end (head) of the vector, in working coordinates. |
| 6 | Vector | Total Delta in Working | The delta from the start to the end of the vector, in working coordinates. |
| 7 | Vector | ijk Unit Vector in Working | The normalized (ijk) delta from the start to the end of the vector, in working coordinates. |
| 8 | Double | Magnitude | The magnitude of the vector. |
| 9 | Color | Vector Color | The vector color in the current colorization. |

Returned Status

| | |
|---------|---|
| SUCCESS | The vector was retrieved successfully. |
| FAILURE | The vector name ref list was not found, or an invalid index was supplied. |

Remarks

None.

Get i-th String From String Ref List

Returns the string at the i-th index of a string reference list.

Input Arguments

| | | | |
|---|-----------------|--------------|--|
| 0 | String Ref List | String List | The name of the string reference list. |
| 1 | Integer | String Index | The index of the string to retrieve. |

Return Arguments

| | | | |
|---|--------|------------------|------------------------------------|
| 2 | String | Resultant String | The string at the specified index. |
|---|--------|------------------|------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The string was returned successfully. |
| FAILURE | An invalid reference list or index was supplied. |

Remarks

String indices are zero based, so the first string in a string reference list is at index 0.

Get i-th String From String Ref List (Iterator)

Iterates through a list of strings, returning one string on each iteration.

Input Arguments

| | | | |
|---|-----------------|-----------------------------|--|
| 0 | String Ref List | String List | The name of the string reference list. |
| 1 | Integer | String Index | The starting index from which to iterate in the list. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to upon reaching the end of the list. |

Return Arguments

| | | | |
|---|--------|------------------|---------------------------------|
| 3 | String | Resultant String | The current iteration's string. |
|---|--------|------------------|---------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The string was returned successfully. |
| FAILURE | An invalid reference list or index was supplied. |

Remarks

String indices are zero based, so the first string in a string reference list is at index 0.

Get Number of Pictures in Picture Name Ref List

Returns the string at the i-th index of a string reference list.

Input Arguments

| | | | |
|---|----------------------------------|-------------------|---------------------------------|
| 0 | Collection Picture Name Ref List | Picture Name List | The list of pictures to examine |
|---|----------------------------------|-------------------|---------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|-------------------------------------|
| 1 | Integer | Total Count | The number of pictures in the list. |
|---|---------|-------------|-------------------------------------|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The count was retrieved successfully. |
| FAILURE | An invalid list was supplied. |

Remarks

None.

Get i-th Picture From Picture Name Ref List

Returns the picture at the i-th index of a picture reference list.

Input Arguments

| | | | |
|---|----------------------------------|-------------------|---------------------------------------|
| 0 | Collection Picture Name Ref List | Picture Name List | The list of pictures to examine. |
| 1 | Integer | Picture Index | The index of the picture to retrieve. |

Return Arguments

| | | | |
|---|-------------------------|----------------|---|
| 2 | Collection Picture Name | Resultant Item | The picture at the specified index in the list. |
|---|-------------------------|----------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The picture was returned successfully. |
| FAILURE | An invalid list or index was supplied. |

Remarks

Indices are zero based, so the first picture in a picture name reference list is at index 0.

Add a Picture to Picture Name Ref List

Adds a picture to a list of pictures. The picture is placed at the end of the list.

Input Arguments

| | | | |
|---|----------------------------------|-------------------|--|
| 0 | Collection Picture Name | Picture Name | The name of the picture to add. |
| 1 | Collection Picture Name Ref List | Picture Name List | The picture name reference list to modify. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The picture was added successfully. |
| FAILURE | An invalid list was supplied, or the picture was not found. |

Remarks

None.

Get Number of Reports in Report Ref List

Obtains the number of reports in a report ref list.

Input Arguments

| | | | |
|---|--------------------|-------------|----------------------------------|
| 0 | SA Report Ref List | Report List | The name of the report ref list. |
|---|--------------------|-------------|----------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The number of reports in the ref list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The number of reports was obtained successfully. |
| FAILURE | An invalid report ref list was supplied. |

Remarks

None.

Get i-th Report Report Ref List

Obtains a specific report from a report ref list.

Input Arguments

| | | | |
|---|--------------------|--------------|---|
| 0 | SA Report Ref List | Report List | The name of the report ref list. |
| 1 | Integer | Report Index | the index number of the report in teh ref list. |

Return Arguments

| | | | |
|---|------------------------|----------------|---|
| 2 | Collection Object Name | Resultant Item | The report that was indexed from the report ref list. |
|---|------------------------|----------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The report was obtained successfully. |
| FAILURE | An invalid report list or report index was supplied. |

Remarks

None.

Add a Report to Report Ref List

Adds a report to a list of reports.

Input Arguments

| | | | |
|---|------------------------|-------------|--|
| 0 | Collection Object Name | Report Name | The name of the picture to add. |
| 1 | SA Report Ref List | Report List | The picture name reference list to modify. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was added successfully. |
| FAILURE | An invalid list was supplied, or the report was not found. |

Remarks

None.

Remove i-th Object From Collection Object Name Ref List

Removes the specified object from a list of objects.

Input Arguments

| | | | |
|---|---------------------------------|------------------|--|
| 0 | Collection Object Name Ref List | Object Name List | The list of objects to modify. |
| 1 | Integer | Object Index | The index of the object to remove from the list. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The object was successfully removed from the list. |
| FAILURE | The index was invalid. |

Remarks

Indices are zero based, so the first object in the list is at index 0. The list collapses to fill any remaining holes.

Remove i-th Point Name From Point Name Ref List

Removes a specified point from a list of points.

Input Arguments

| | | | |
|---|---------------------|-----------------|-----------------------------------|
| 0 | Point Name Ref List | Point Name List | The list of points to modify. |
| 1 | Integer | Point Index | The index of the point to remove. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was successfully removed from the list. |
| FAILURE | The index was invalid. |

Remarks

Indices are zero based, so the first point in a point name reference list is at index 0. The list collapses to fill any remaining holes.

Remove i-th String from String Ref List

Removes a string from a list of strings.

Input Arguments

| | | | |
|---|-----------------|--------------|------------------------------------|
| 0 | String Ref List | String List | The list of strings to modify. |
| 1 | Integer | String Index | The index of the string to remove. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The list was modified successfully. |
| FAILURE | An invalid reference list or index was supplied. |

Remarks

String indices are zero based, so the first string in a string reference list is at index 0.

Append to String Ref List

Adds a string to the end of a string reference list.

Input Arguments

| | | | |
|---|-----------------|------------------|--|
| 0 | String Ref List | String List | The name of the string reference list onto which to add. |
| 1 | String | String to Append | The string to add to the end of the list. |

Return Arguments

| | | | |
|---|---------|------------------|---|
| 2 | Integer | New String Index | The index at which the new string is located. |
|---|---------|------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The string was added successfully. |
| FAILURE | An invalid reference list was supplied. |

Remarks

String indices are zero based, so the first string in a string reference list is at index 0.

Get Vector From Vector Group By Name

Retrieves vector information from a vector group by specifying the name of the vector.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group in which the desired vector is located. |
| 1 | String | Vector Name | The name of the vector to retrieve. |

Return Arguments

| | | | |
|---|--------|----------------------------|---|
| 2 | Vector | Begin in Working | The coordinates of the start (tail) of the vector, in working coordinates. |
| 3 | Vector | End in Working | The coordinates of the end (head) of the vector, in working coordinates. |
| 4 | Vector | Total Delta in Working | The delta from the start to the end of the vector, in working coordinates. |
| 5 | Vector | ijk Unit Vector in Working | The normalized (ijk) delta from the start to the end of the vector, in working coordinates. |
| 6 | Double | Magnitude | The magnitude of the vector. |

Returned Status

| | |
|---------|--|
| SUCCESS | The vector was retrieved successfully. |
| FAILURE | The vector group or vector name was not found. |

Remarks

None.

Add a Vector to Vector Name Ref List

RAdds vector to an existing vector name ref list.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group in which the desired vector is located. |
| 1 | String | Vector Name | The name of the vector to retrieve. |
| 2 | Vector Name Ref List | Vector Name List | The name of the vector name ref list in which the new vector will be added.. |

Returned Status

| | |
|---------|---|
| SUCCESS | The vector group and vector name were retrieved successfully, and the vector name ref list was found. |
| FAILURE | The vector group, vector name, or vector name ref list was not found. |

Remarks

None.

Delete i-th Vector From Vector Group

Deletes the vector at a specified index from a vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group in which the vector to delete is located. |
| 1 | Integer | Vector Index | The index of the vector to delete. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The vector was deleted successfully. |
| FAILURE | The vector group was not found, or an invalid index was supplied. |

Remarks

Since this command alters the number of vectors in a group, caution should be exercised when iterating through a vector group and deleting vectors to ensure that you do not attempt to iterate past the end of the vector group. Vector group indices are zero-based, so the first vector in a vector group is at index 0.

Delete Vector by Name

Deletes the vector at a specified index from a vector group.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Vector Group Name | The name of the vector group in which the vector to delete is located. |
| 1 | String | Vector Name | The name of the vector to delete. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The vector was deleted successfully. |
| FAILURE | The vector group or vector name were not found. |

Remarks

None.

Delete Vectors

Deletes a list of vectors.

Input Arguments

| | | | |
|---|----------------------|------------------|--|
| 0 | Vector Name Ref List | Vector Name List | The name of the vector name ref list in which the vector to delete is located. |
|---|----------------------|------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The vector was deleted successfully. |
| FAILURE | The vector was not found. |

Remarks

None.

Get Line Properties

Retrieves the properties of a line.

Input Arguments

| | | | |
|---|------------------------|-----------|----------------------------------|
| 0 | Collection Object Name | Line Name | The name of the line to examine. |
|---|------------------------|-----------|----------------------------------|

Return Arguments

| | | | |
|---|--------|------------------------------------|--|
| 1 | Vector | Begin Coordinate | The coordinate of the start of the line, in the active coordinate frame. |
| 2 | Vector | End Coordinate | The coordinate of the end of the line, in the active coordinate frame. |
| 3 | Vector | Delta Components | The delta from the start to the end of the line, in the active coordinate frame. |
| 4 | Double | Length | The length of the line. |
| 5 | Double | Angle about +X from +Y in YZ plane | The angle of the line about the X axis (from the +Y axis) projected into the YZ plane. |
| 6 | Double | Angle about +Y from +Z in XZ plane | The angle of the line about the Y axis (from the +Z axis) projected into the XZ plane. |
| 7 | Double | Angle about +Z from +X in XY plane | The angle of the line about the Z axis (from the +X axis) projected into the XY plane. |

Returned Status

| | |
|---------|--|
| SUCCESS | The line properties were retrieved successfully. |
| FAILURE | The specified line was not found. |

Remarks

Sign conventions for the projected angles follow the right hand rule.

Get Sphere Properties

Retrieves the properties of a sphere.

Input Arguments

| | | | |
|---|------------------------|-------------|-------------------------------------|
| 0 | Collection Object Name | Sphere Name | The name of the sphere to retrieve. |
|---|------------------------|-------------|-------------------------------------|

Return Arguments

| | | | |
|---|--------|-------------------|---|
| 1 | Vector | Center Coordinate | The coordinate of the center of the sphere, in the active coordinate frame. |
| 2 | Double | Radius | The radius of the sphere. |
| 3 | Double | Diameter | The diameter of the sphere. |

Returned Status

| | |
|---------|--|
| SUCCESS | The sphere properties were retrieved successfully. |
| FAILURE | The specified sphere was not found. |

Remarks

None.

Get Circle Properties

Retrieves the properties of a circle.

Input Arguments

| | | | |
|---|------------------------|-------------|-------------------------------------|
| 0 | Collection Object Name | Circle Name | The name of the circle to retrieve. |
|---|------------------------|-------------|-------------------------------------|

Return Arguments

| | | | |
|---|--------|-------------------|---|
| 1 | Vector | Center Coordinate | The coordinate of the center of the circle, in the active coordinate frame. |
| 2 | Vector | Normal Direction | The direction of the normal of the circle. |
| 3 | Double | Radius | The radius of the circle. |
| 4 | Double | Diameter | The diameter of the circle. |

Returned Status

| | |
|---------|--|
| SUCCESS | The circle properties were retrieved successfully. |
| FAILURE | The specified circle was not found. |

Remarks

None.

Set Circle Properties

Provides the means to edit the properties of an existing circle.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Circle Name | The name of the circle to retrieve. |
| 1 | Vector | Center Coordinate | The coordinate of the center of the circle, in the active coordinate frame. |
| 2 | Vector | Normal Direction | The direction of the normal of the circle. |
| 3 | Double | Radius | The radius of the circle. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The circle properties were set successfully. |
| FAILURE | The specified circle was not found. |

Remarks

This command will not adjust the properties of a Geometry Relationship's measured circle, as the relationship is controlling these properties, but it can be used to modify the nominal circle or a constructed circle.

Get Cylinder Properties

Retrieves the properties of a cylinder.

Input Arguments

| | | | |
|---|------------------------|---------------|---------------------------------------|
| 0 | Collection Object Name | Cylinder Name | The name of the cylinder to retrieve. |
|---|------------------------|---------------|---------------------------------------|

Return Arguments

| | | | |
|---|--------|------------------|---|
| 1 | Vector | Begin Coordinate | The coordinate of the beginning of the cylinder (start of the axis line) in the working coordinate frame. |
| 2 | Vector | End Coordinate | The coordinate of the end of the cylinder (end of the axis line) in the working coordinate frame. |
| 3 | Vector | Axis Direction | A normalized vector representing the direction of the cylinder. |
| 4 | Double | Length | The length (height) of the cylinder. |
| 5 | Double | Radius | The radius of the cylinder. |
| 6 | Double | Diameter | The diameter of the cylinder. |

Returned Status

| | |
|---------|--|
| SUCCESS | The cylinder properties were retrieved successfully. |
| FAILURE | The specified cylinder was not found. |

Remarks

None.

Get Plane Properties

Retrieves the properties of a plane.

Input Arguments

| | | | |
|---|------------------------|------------|------------------------------------|
| 0 | Collection Object Name | Plane Name | The name of the plane to retrieve. |
|---|------------------------|------------|------------------------------------|

Return Arguments

| | | | |
|---|--------|------------------|---|
| 1 | Vector | Normal Direction | The normalized vector direction of the specified plane, in working coordinates. |
| 2 | Vector | Point on Plane | A point on the plane. |
| 3 | Double | D Parameter | The value of D in the plane equation ($Ax + By + Cz + D = 0$). |

Returned Status

| | |
|---------|---|
| SUCCESS | The plane properties were retrieved successfully. |
| FAILURE | The specified plane was not found. |

Remarks

None.

Get Torus Properties

Retrieves the properties of a torus.

Input Arguments

| | | | |
|---|------------------------|------------|------------------------------------|
| 0 | Collection Object Name | Torus Name | The name of the torus to retrieve. |
|---|------------------------|------------|------------------------------------|

Return Arguments

| | | | |
|---|--------|-------------------|---|
| 1 | Vector | Center Coordinate | Center point of the torus ring |
| 2 | Vector | Normal Direction | The normalized vector direction of the specified plane, in working coordinates. |
| 3 | Double | Major Radius | Large radius of the outer ring |
| 4 | Double | Minor Radius | Radius of the inner ring, or each cross section |

Returned Status

| | |
|---------|---|
| SUCCESS | The torus properties were retrieved successfully. |
| FAILURE | The specified plane was not found. |

Remarks

None.

Set Default Colorization Options

Sets the default colorization options for all vector groups created after the command has executed.

Input Arguments

| | | | |
|---|----------------------|----------------------|-----------------------------------|
| 0 | Colorization Options | Colorization Options | The options for the vector group. |
|---|----------------------|----------------------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The vector group colorization options are the same as those available in the vector group properties window (arrow vs. blotches, size, draw out of tolerance only, etc).

Set Vector Group Display Attributes

This command creates set of colorization options that can be applied to one or more vector groups using a Set Vector Group Colorization Options command that references these settings.

Input Arguments

| | | | |
|----|-----------------------|-----------------------------|--|
| 1 | Boolean | Draw Arrowheads? | Indicates whether arrows should be drawn for each vector. |
| 2 | Boolean | Indicate Values? | Indicates whether vector magnitudes should be depicted for each vector. |
| 3 | Double | Vector Magnification | The scale factor to apply to the drawn vector length. |
| 4 | Integer | Vector Width | The width of drawn arrows, in pixels. |
| 5 | Boolean | Draw Color Blotches? | Indicates whether color blotches should be drawn. |
| 6 | Double | Blotch Size | The size of color blotches, in job units. |
| 7 | Boolean | Show Out of Tolerance Only? | Indicates whether in-tolerance vectors should be drawn. |
| 8 | Boolean | Show Color Bar In View? | Indicates whether a vector group's color bar should be depicted in the graphical view. |
| 9 | Boolean | Show Color Bar Percentages? | Indicates whether the percentage of in-tolerance, high, and low vectors should be depicted on the color bar in the graphical view. |
| 10 | Boolean | Show Color Bar Fractions? | Indicates whether the fraction of in-tolerance, high, and low vectors should be depicted on the color bar in the graphical view. |
| 11 | Saturation Limit Type | High Saturation Limit Type | Type of saturation limit, choose from the list. |
| 12 | Double | High Saturation Limit | The magnitude at which vectors reach the color limit on the high end of the color bar. |
| 13 | Saturation Limit Type | High Saturation Limit Type | Type of saturation limit, choose from the list. |
| 14 | Double | Low Saturation Limit | The magnitude at which vectors reach the color limit on the low end of the color bar. |
| 15 | Double | High Tolerance | The high tolerance for the vector group. |
| 16 | Double | Low Tolerance | The low tolerance for the vector group. |
| 17 | Color Range Method | Color Ranging Method | The method to use when coloring vectors (Continuous, Go/No-Go, etc). |
| 18 | Base Color Type | Base High Color | The color to use to depict high values. |
| 19 | Base Color Type | Base Mid Color | The color to use to depict mid-range values. |
| 20 | Base Color Type | Base Low Color | The color to use to depict low values. |
| 21 | Boolean | Draw Tubes? | Indicates whether or not tubes should be drawn for the vector. |
| 22 | Boolean | Render in 2D? | Display Color bar using 2D rendering |

Return Arguments

| | | | |
|---|----------------------|----------------------|---|
| 0 | Colorization Options | Colorization Options | The colorization options resulting from the above settings. |
|---|----------------------|----------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The advantage of this command is that it permits you to set individual vector group display settings programmatically instead of using a dialog (as is the case in the “Set Default Colorization Options” command).

Get Vector Group Display Attributes

Retrieves the vector group display attributes from the colorization options.

Input Arguments

| | | | |
|---|----------------------|----------------------|--------------------------------------|
| 0 | Colorization Options | Colorization Options | The colorization options to examine. |
|---|----------------------|----------------------|--------------------------------------|

Return Arguments

| | | | |
|----|--------------------|-----------------------------|--|
| 1 | Boolean | Draw Arrowheads? | Indicates whether arrows should be drawn for each vector. |
| 2 | Boolean | Indicate Values? | Indicates whether vector magnitudes should be depicted for each vector. |
| 3 | Double | Vector Magnification | The scale factor to apply to the drawn vector length. |
| 4 | Integer | Vector Width | The width of drawn arrows, in pixels. |
| 5 | Boolean | Draw Color Blotches? | Indicates whether color blotches should be drawn. |
| 6 | Double | Blotch Size | The size of color blotches, in job units. |
| 7 | Boolean | Show Out of Tolerance Only? | Indicates whether in-tolerance vectors should be drawn. |
| 8 | Boolean | Show Color Bar In View? | Indicates whether a vector group's color bar should be depicted in the graphical view. |
| 9 | Boolean | Show Color Bar Percentages? | Indicates whether the percentage of in-tolerance, high, and low vectors should be depicted on the color bar in the graphical view. |
| 10 | Boolean | Show Color Bar Fractions? | Indicates whether the fraction of in-tolerance, high, and low vectors should be depicted on the color bar in the graphical view. |
| 11 | Double | High Saturation Limit | The magnitude at which vectors reach the color limit on the high end of the color bar. |
| 12 | Double | Low Saturation Limit | The magnitude at which vectors reach the color limit on the low end of the color bar. |
| 13 | Double | High Tolerance | The high tolerance for the vector group. |
| 14 | Double | Low Tolerance | The low tolerance for the vector group. |
| 15 | Color Range Method | Color Ranging Method | The method to use when coloring vectors (Continuous, Go/No-Go, etc). |
| 16 | Base Color Type | Base High Color | The color to use to depict high values. |
| 17 | Base Color Type | Base Mid Color | The color to use to depict mid-range values. |
| 18 | Base Color Type | Base Low Color | The color to use to depict low values. |
| 19 | Boolean | Draw Tubes? | Indicates whether or not tubes should be drawn for the vector. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Query Groups to Objects

Queries one or more point groups to one or more surfaces, creating a vector group or point group in the process.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Collection Object Name Ref List | Group Name List (Groups to Project) | One or more point groups to be projected. |
| 1 | Collection Object Name Ref List | Object Name List (Objects to Project to) | One or more objects that the selected point groups will be projected to. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting vector group or point group. |
| 3 | Projection Options | Projection Options | The projection settings for the query. |
| 4 | Double | RMS Tolerance (0.0 for none) | An RMS tolerance for the query. |
| 5 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum allowable absolute tolerance for the query. |
| 6 | Boolean | Show Results Dialog? | Shows the query dialog, allowing the user to interact with it. |

Return Arguments

| | | | |
|----|--------|------------------------|--|
| 7 | Double | RMS Deviation | The actual RMS deviation for the query. |
| 8 | Double | Max Absolute Deviation | The actual maximum absolute deviation for the query. |
| 9 | Double | Average Deviation | The average of the query's deviation values. |
| 10 | Double | Standard Deviation | The standard deviation for the query results. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The query was performed successfully and the query results were within the specified tolerance (if applicable). |
| PARTIAL SUCCESS | At least one query was performed successfully, but one or more objects were not found or the query results were out of at least one specified tolerance. |
| FAILURE | No queries were performed successfully. |

Remarks

None.

Query Points to Objects

Queries a list of points to one or more surfaces, creating a vector group or point group in the process.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Point Name Ref List | Point Names | The list of points to query. |
| 1 | Collection Object Name Ref List | Object Name List (Objects to Project to) | One or more objects that the selected point groups will be projected to. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting vector group or point group. |
| 3 | Projection Options | Projection Options | The projection settings for the query. |
| 4 | Double | RMS Tolerance (0.0 for none) | An RMS tolerance for the query. |
| 5 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum allowable absolute tolerance for the query. |
| 6 | Boolean | Show Results Dialog? | Shows the query dialog, allowing the user to interact with it. |

Return Arguments

| | | | |
|----|--------|------------------------|--|
| 7 | Double | RMS Deviation | The actual RMS deviation for the query. |
| 8 | Double | Max Absolute Deviation | The actual maximum absolute deviation for the query. |
| 9 | Double | Average Deviation | The average of the query's deviation values. |
| 10 | Double | Standard Deviation | The standard deviation for the query results. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The query was performed successfully and the query results were within the specified tolerance (if applicable). |
| PARTIAL SUCCESS | At least one query was performed successfully, but one or more objects were not found or the query results were out of at least one specified tolerance. |
| FAILURE | No queries were performed successfully. |

Remarks

None.

Query Points to Single Point

Queries a list of points to a single reference point, creating a vector group in the process.

Input Arguments

| | | | |
|---|---------------------|-------------------------|--|
| 0 | Point Name Ref List | Point Names | The list of points to query. |
| 1 | Point Name | Single Point | The reference point to compare all points against. |
| 2 | Boolean | Show Vector Properties? | Indicates whether the vector group properties window should be displayed at the completion of the command. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The query was performed successfully. |
| FAILURE | No points were found, or the reference point was not found. |

Remarks

None.

Query Points to Circle

Queries one or more point groups to a circle, creating a radial, planar, and combined vector group for the query.

Input Arguments

| | | | |
|---|------------------------|-----------------------------------|---|
| 0 | Collection Object Name | Circle Name | The name of the circle to query to. |
| 1 | Collection Object Name | Point Group Name | The name of the point group to project. |
| 2 | Boolean | Is Inside Measurement | Indicates whether the points represent a measurement on the inside of the circle. |
| 3 | Integer | Auto Scale Vectors to % of Radius | A scale factor to apply to the vectors as a function of the radius of the circle. |
| 4 | Vector Group Name | Vector Group Name for Radial | The name of the vector group representing the radial query. |
| 5 | Vector Group Name | Vector Group Name for Planar | The name of the vector group representing the planar query. |
| 6 | Vector Group Name | Vector Group Name for Combined | The name of the vector group representing the combined query. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The query was performed successfully. |
| FAILURE | The query was not performed successfully. Either the circle or point group could not be found. |

Remarks

None.

Query Point to Objects

Queries a point to the closest of a list of one or more objects, and returns the deviation and object queried to.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Point Name | Point Name | The name of the point to query. |
| 1 | Collection Object Name Ref List | Objects | The list of objects to query to. |
| 2 | Boolean | Ignore Target Offset | Indicates whether offsets should be ignored (zero) or if the stored offsets should be used. |

Return Arguments

| | | | |
|---|------------------------|------------------|---|
| 3 | Double | dX | The x-component of the deviation between the point and object. |
| 4 | Double | dY | The y-component of the deviation between the point and object. |
| 5 | Double | dZ | The z-component of the deviation between the point and object. |
| 6 | Double | dMag | The (signed) magnitude of the deviation between the point and object. |
| 7 | Collection Object Name | Resultant Object | The object to which the point was ultimately queried. |

Returned Status

| | |
|---------|--|
| SUCCESS | The query was performed successfully. |
| FAILURE | The point was not found, or none of the objects was found. |

Remarks

This command will return the deviation between the point and the closest object to that point, just as the manual Query commands do.

This command can be used to easily determine the closest object to a specific point in space.

Query Point to Point Along Curve

Queries a point to another point, giving the distance along the curve between the two points.

Input Arguments

| | | | |
|---|------------------------|-----------|---|
| 0 | Point Name | 1st Point | The name of the first point. |
| 1 | Point Name | 2nd Point | The name of the second point. |
| 2 | Collection Object Name | Curve | The curve defining the query path. The two points should lie on this curve. |

Return Arguments

| | | | |
|---|--------|----------------------|---|
| 3 | Double | Distance Along Curve | The distance along the provided curve between the two points. |
|---|--------|----------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The query was performed successfully. |
| FAILURE | Either point or the curve was not found. |

Remarks

If any point does not lie on the provided curve, the point will be projected to the curve first.

Query Frame to Frame

Calculates the transformation from a reference frame to a corresponding frame, reported in the working coordinate frame.

Input Arguments

| | | | |
|---|------------------------|--------------------------|--|
| 0 | Collection Object Name | Reference Frame Name | The reference (starting) frame. |
| 1 | Collection Object Name | Corresponding Frame Name | The corresponding (destination) frame. |

Return Arguments

| | | | |
|---|--------|------------|---|
| 2 | Double | X | The transform's X value, reported in the working coordinate frame. |
| 3 | Double | Y | The transform's Y value, reported in the working coordinate frame. |
| 4 | Double | Z | The transform's Z value, reported in the working coordinate frame. |
| 5 | Double | Rx (Roll) | The transform's Rx value, reported in the working coordinate frame. |
| 6 | Double | Ry (Pitch) | The transform's Ry value, reported in the working coordinate frame. |
| 7 | Double | Rz (Yaw) | The transform's Rz value, reported in the working coordinate frame. |

Returned Status

| | |
|---------|---|
| SUCCESS | The query was performed successfully. |
| FAILURE | The query was not performed successfully. One or both frames was not found. |

Remarks

None.

Transform Objects - Frame To Frame

Transforms one or more objects based on the 6-DOF transformation from a source frame to a destination frame.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|--|
| 0 | Collection Object Name Ref List | Object Name List | The list of objects to transform. |
| 1 | Collection Object Name | Initial Frame Name | The name of the source frame. |
| 2 | Collection Object Name | Destination Frame Name | The name of the destination frame. |
| 3 | Integer | Number of Steps | The number of frames to use when animating the transformation in the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The transformation was performed successfully. |
| FAILURE | The transformation was not successful. Either no objects to transform could be found, or the source or destination frame could not be found. |

Remarks

The animation steps are for graphical purposes only. To speed up MP execution, leave this value at zero.

Transform Objects by Delta (World Transform Operator)

Transforms one or more objects based on a 6-DOF world transform operator.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|--|
| 0 | Collection Object Name Ref List | Objects to Transform | The list of objects to transform. |
| 1 | World Transform Operator | Delta Transform | The world transform to apply to the objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The transformation was performed successfully. |
| PARTIAL SUCCESS | At least one object (but not all) could not be found. |
| FAILURE | The transformation was not successful because no objects to transform could be found. |

Remarks

None.

Transform Objects by Delta (About Working Frame)

Transforms one or more objects based on a 6-DOF transformation about the active coordinate frame.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Collection Object Name Ref List | Objects to Transform | The list of objects to transform. |
| 1 | Transform | Delta Transform | The transform (in the active coordinate frame) to apply to the objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The transformation was performed successfully. |
| PARTIAL SUCCESS | At least one object (but not all) could not be found. |
| FAILURE | The transformation was not successful because no objects to transform could be found. |

Remarks

None.

Translate Objects by Delta

Translates one or more objects based on a 3-DOF translation in the active coordinate frame.

Input Arguments

| 0 | Collection Object Name Ref List | Objects to Translate | The list of objects to translate. |
|---|---------------------------------|----------------------|--|
| 1 | Vector | Delta Translation | The delta translation vector (in the active coordinate frame) to apply to the objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The translation was performed successfully. |
| PARTIAL SUCCESS | At least one object (but not all) could not be found. |
| FAILURE | The translation was not successful because no objects to translate could be found. |

Remarks

None.

Fit Geometry to Point Group

Fits a geometric shape (line, plane, circle, sphere, cylinder, cone, paraboloid, or ellipse) to the points in a specified point group.

Input Arguments

| | | | |
|---|------------------------|--|---|
| 0 | Geometry Type | Geometry Type | The type of geometry to fit to the points. |
| 1 | Collection Object Name | Group to Fit | The point group to use for fitting. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting geometry. |
| 3 | String | Fit Profile Name | The name of the geometry fit profile to use. |
| 4 | Boolean | Report Deviations | Indicates whether a dialog should be displayed showing the fit results. |
| 5 | Double | Fit Interface Tolerance (-1.0 use profile) | The tolerance to use for the geometry fit. |
| 6 | Boolean | Ignore Out of Tolerance Points | Indicates whether or not points outside of the tolerance specified in Argument 5 should be included in the fit. |
| 7 | Collection Object Name | Starting Condition Geometry (optional) | The name of a like geometry type to use as an initial guess for the geometry fit. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The fit was successful. |
| PARTIAL SUCCESS | The fit was successful, but the tolerance was exceeded. |
| FAILURE | The fit was unsuccessful because the point group or fit profile could not be found. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use a fit tolerance of -1.0 to use the tolerance defined in the specified fit profile or a value of 0.0 to indicate that no tolerance should apply. The optional starting condition geometry is useful in rare cases when a fit algorithm gets “confused” and fails to settle on an acceptable solution. Providing a starting condition geometry can often eliminate this problem. An object providing starting condition geometry is usually not required.

As of 2022.3 the Geometry Type argument [A0] can reference a string. It will accept [Line, Plane, Circle, Slot, Ellipse, Cylinder, Sphere, Cone, and Paraboloid] *Notes - first letter must be capitalized.

Fit Geometry to Point Group Projected to Plane

Fits a geometric shape (line, plane, circle, sphere, cylinder, cone, paraboloid, or ellipse) to a point group that has been projected to a plane. a specified point group.

Input Arguments

| | | | |
|---|------------------------|--|--|
| 0 | Geometry Type | Geometry Type | The type of geometry to fit to the points. |
| 1 | Collection Object Name | Group to Fit | The point group to use for fitting. |
| 2 | Collection Object Name | Plane Name | The name of the plane that is being used for the projection. |
| 3 | Collection Object Name | Resulting Object Name | The name of the resulting geometry. |
| 4 | String | Fit Profile Name | The name of the geometry fit profile to use. |
| 5 | Boolean | Report Deviations | Indicates whether a dialog should be displayed showing the fit results. |
| 6 | Double | Fit Interface Tolerance (-1.0 use profile) | The tolerance to use for the geometry fit. |
| 7 | Boolean | Ignore Out of Tolerance Points | Indicates whether or not points outside the tolerance specified in Argument 6 should be included in the fit. |
| 8 | Collection Object Name | Starting Condition Geometry (optional) | The name of a like geometry type to use as an initial guess for the geometry fit. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The fit was successful. |
| PARTIAL SUCCESS | The fit was successful, but the tolerance was exceeded. |
| FAILURE | The fit was unsuccessful because the point group or fit profile could not be found. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use a fit tolerance of -1.0 to use the tolerance defined in the specified fit profile or a value of 0.0 to indicate that no tolerance should apply. The optional starting condition geometry is useful in rare cases when a fit algorithm gets “confused” and fails to settle on an acceptable solution. Providing a starting condition geometry can often eliminate this problem. An object providing starting condition geometry is usually not required.

Fit Geometry to Points

Fits a geometric shape (line, plane, circle, sphere, cylinder, cone, paraboloid, or ellipse) to a list of points.

Input Arguments

| | | | |
|---|------------------------|--|---|
| 0 | Geometry Type | Geometry Type | The type of geometry to fit to the points. |
| 1 | Point Name Ref List | Points to Fit | The points to use in the fit. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting geometry. |
| 3 | String | Fit Profile Name | The name of the geometry fit profile to use. |
| 4 | Boolean | Report Deviations | Indicates whether a dialog should be displayed showing the fit results. |
| 5 | Double | Fit Interface Tolerance (-1.0 use profile) | The tolerance to use for the geometry fit. |
| 6 | Boolean | Ignore Out of Tolerance Points | Indicates whether or not points outside of the tolerance specified in Argument 5 should be included in the fit. |
| 7 | Collection Object Name | Starting Condition Geometry (optional) | The name of a like geometry type to use as an initial guess for the geometry fit. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The fit was successful. |
| PARTIAL SUCCESS | The fit was successful, but the tolerance was exceeded. |
| FAILURE | The fit was unsuccessful because the points or fit profile could not be found. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use a fit tolerance of -1.0 to use the tolerance defined in the specified fit profile or a value of 0.0 to indicate that no tolerance should apply. The optional starting condition geometry is useful in rare cases when a fit algorithm gets “confused” and fails to settle on an acceptable solution. Providing a starting condition geometry can often eliminate this problem. An object providing starting condition geometry is usually not required.

As of 2022.3 the Geometry Type argument [A0] can reference a string. It will accept [Line, Plane, Circle, Slot, Ellipse, Cylinder, Sphere, Cone, and Paraboloid] *Notes - first letter must be capitalized.

Import Geometry Fit Profiles

Imports geometry fit profiles into the current job file.

Input Arguments

| 0 | File Path or Embedded File | Geometry Fit Profiles File Path | The path to the geometry fit profile (.gfp) file. |
|---|----------------------------|------------------------------------|---|
| 1 | Boolean | Overwrite Profiles with Same Name? | Indicates whether existing profiles whose name matches an imported profile should be overwritten. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The file import was successful. |
| FAILURE | The fit profile file could not be found. |

Remarks

Geometry fit profiles can be exported via the [User Options > Analysis](#) tab by clicking the [Geometry Fit Profiles](#) button. If argument 1 is FALSE, imported profiles will have asterisks appended to their names if a like-named profile already exists in the job file.

Best Fit Transformation - Group to Group

Generates the 6-DOF transform required to best-fit one point group to another (scale is fixed at 1.0).

Input Arguments

| | | | |
|----|----------------------------|--|--|
| 0 | Collection Object Name | Reference Group | The group to fit to. |
| 1 | Collection Object Name | Corresponding Group | The group to fit. |
| 2 | Boolean | Show Interface | Indicates whether the best-fit interface should be displayed for the fit. |
| 3 | Double | RMS Tolerance (0.0 for none) | An RMS tolerance for the fit. |
| 4 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum absolute tolerance for the fit. |
| 5 | Boolean | Allow Scale | Indicates if scaling should be allowed. |
| 6 | Boolean | Allow X | Indicates if the X degree of freedom is allowed. |
| 7 | Boolean | Allow Y | Indicates if the Y degree of freedom is allowed. |
| 8 | Boolean | Allow Z | Indicates if the Z degree of freedom is allowed. |
| 9 | Boolean | Allow Rx | Indicates if the Rx rotational degree of freedom is allowed. |
| 10 | Boolean | Allow Ry | Indicates if the Ry rotational degree of freedom is allowed. |
| 11 | Boolean | Allow Rz | Indicates if the Rz rotational degree of freedom is allowed. |
| 12 | Boolean | Lock Degrees of Freedom | True enables locking of degrees of freedom |
| 13 | Boolean | Generate Event | True enables event generation |
| 14 | File Path or Embedded File | File Path for CSV Text Report (requires Show Interface = TRUE) | The path for a CSV text report to create as a result of the fit. This will only create the file if the interface is shown. |

Return Arguments

| | | | |
|----|--------------------------|----------------------------|--|
| 15 | Transform | Transform in Working | The resulting transform in working coordinates. |
| 16 | World Transform Operator | Optimum Transform | The resulting transform represented as a world transform operator. |
| 17 | Double | RMS Deviation | The actual RMS deviation of the fit. |
| 18 | Double | Maximum Absolute Deviation | The actual maximum absolute deviation of the fit. |
| 19 | Integer | Number of Unknowns | The number of unknowns |
| 20 | Integer | Number of Equations | The number of equations used |
| 21 | Double | Robustness | The solution robustness |

Returned Status

| | |
|---------|---|
| SUCCESS | The fit was successful. |
| FAILURE | One or both groups could not be found, or a tolerance was violated. |

Remarks

A tolerance of 0.0 implies that a tolerance should not be applied.

Monitoring the solution robustness factor is a great way to identify the mathematical stability of the solution.

Compute Group to Group Orientation (Rx, Ry, Rz)

Calculates the Fixed XYZ rotations to fit one point group to another.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Reference Group | The group to fit to. |
| 1 | Collection Object Name | Corresponding Group | The group to fit. |
| 2 | Double | Rx | The Fixed X angle to rotate to get from the corresponding group to the reference group. |
| 3 | Double | Ry | The fixed Y angle to rotate to get from the corresponding group to the reference group. |
| 4 | Double | Rz | The fixed Z angle to rotate to get from the corresponding group to the reference group. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The rotations were calculated successfully. |
| FAILURE | The reference group or corresponding group could not be found. |

Remarks

None.

Temperature Compensate a group

Copies a point group and scales the copy based on a temperature change, scaling origin, and coefficient of thermal expansion (CTE).

Input Arguments

| | | | |
|---|------------------------|-----------------------------------|--|
| 0 | Collection Object Name | Original Group | The group to compensate. |
| 1 | Frame Name | Scaling Origin (coordinate frame) | A coordinate frame defining the origin about which the scale should be applied. |
| 2 | Double | Material CTE (1/Deg F) | The coefficient of thermal expansion (CTE) to use when calculating the scaling factor. |
| 3 | Double | Initial Temperature (F) | The starting temperature to use (temperature to scale from). |
| 4 | Double | Final Temperature (F) | The final temperature to use (temperature to scale to). |
| 5 | Collection Object Name | Scaled Group Name | The name for the compensated point group. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The compensation was performed successfully. |
| FAILURE | The original point group could not be found. |

Remarks

Note that all parameters are entered with respect to degrees Fahrenheit (°F).

Query Clouds to Surface

Queries one or more point clouds to a surface, creating point groups or vector groups in the process.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Collection Object Name Ref List | Cloud Names | The point clouds to query to the surface. |
| 1 | Collection Object Name | Filter Surface Name | The surface to query the point clouds to. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting point group or vector group. |
| 3 | Projection Options | Projection Options | The projection options for the query. |
| 4 | Double | Proximity | Cloud points outside of this proximity value are excluded from the query. |
| 5 | Integer | Skip Factor | Subsampling for the cloud points. For example, 3 implies that the query should consider every third point. |
| 6 | Double | RMS Tolerance (0.0 for none) | An RMS tolerance for the query. |
| 7 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum absolute tolerance for the query. |

Return Arguments

| | | | |
|---|--------|----------------------------|---|
| 8 | Double | RMS Deviation | The resulting RMS deviation for the query. |
| 9 | Double | Maximum Absolute Deviation | The resulting maximum absolute deviation for the query. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The query was performed successfully. |
| PARTIAL SUCCESS | The query was performed, but either a tolerance was violated or one or more (but not all) point clouds could not be found. |
| FAILURE | The query was not successful. |

Remarks

Use a tolerance value of 0.0 to ignore that tolerance.

Query Clouds to Objects

Queries one or more point clouds to one or more objects, creating point groups or vector groups in the process.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Collection Object Name Ref List | Cloud Names | The point clouds to query to the objects. |
| 1 | Collection Object Name Ref List | Object Names | The objects to which the point clouds should be queried. |
| 2 | Collection Object Name | Resulting Object Name | The name of the resulting point group or vector group. |
| 3 | Projection Options | Projection Options | The projection options for the query. |
| 4 | Double | Proximity | Cloud points outside of this proximity value are excluded from the query. |
| 5 | Integer | Skip Factor | Subsampling for the cloud points. For example, 3 implies that the query should consider every third point. |
| 6 | Double | RMS Tolerance (0.0 for none) | An RMS tolerance for the query. |
| 7 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum absolute tolerance for the query. |

Return Arguments

| | | | |
|---|--------|----------------------------|---|
| 8 | Double | RMS Deviation | The resulting RMS deviation for the query. |
| 9 | Double | Maximum Absolute Deviation | The resulting maximum absolute deviation for the query. |

Returned Status

| | |
|-----------------|---|
| SUCCESS | The query was performed successfully. |
| PARTIAL SUCCESS | The query was performed, but either a tolerance was violated or one or more (but not all) point clouds or objects could not be found. |
| FAILURE | The query was not successful. |

Get Measurement Weather Data

Obtains the weather metadata stored with a measured point.

Input Arguments

| | | | |
|---|------------|------------|---|
| 0 | Point Name | Point Name | The point name from which the temperature metadata should be extracted. |
|---|------------|------------|---|

Return Arguments

| | | | |
|---|--------|---------------------|---------------------------------|
| 1 | Double | Temperature (deg F) | The temperature (in degrees F). |
| 2 | Double | Pressure (in. Hg) | The pressure (in. Hg). |
| 3 | Double | Humidity (% RH) | The relative humidity (in %). |

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The data was extracted successfully. |
| FAILURE | The point could not be found. |

Remarks

This command will succeed even if the specified point has no weather metadata. In that case, the values will contain garbage.

Get Measurement Auxiliary Data

Retrieves the auxiliary data stored with a measured point.

Input Arguments

| | | | |
|---|------------|----------------|---|
| 0 | Point Name | Point Name | The point name from which the auxiliary data should be extracted. |
| 1 | String | Auxiliary Name | The string identifier for the auxiliary data. |

Return Arguments

| | | | |
|---|--------|-------|------------------------|
| 2 | Double | Value | The auxiliary value. |
| 3 | String | Units | The units of the data. |

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The data was extracted successfully. |
| FAILURE | The point could not be found. |

Remarks

The auxiliary name is instrument dependent. This name can be found in the tree under the point after a real measurement is taken with a specific device that sends auxiliary data.

Set Measurement Auxiliary Data

Sets the auxiliary data stored with a measured point.

Input Arguments

| | | | |
|---|------------|----------------|---|
| 0 | Point Name | Point Name | The point name from which the auxiliary data should be extracted. |
| 1 | String | Auxiliary Name | The string identifier for the auxiliary data. |
| 2 | Double | Value | Auxiliary value to be saved with the point |
| 3 | Sting | Units | Units value to record |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------|
| SUCCESS | The data was set successfully. |
| FAILURE | The point could not be found. |

Remarks

Auxiliary data is saved on a per name basis. Entering a different value with the same name will overwrite the previous value, while new names will create additional entries.

Get Measurement Info Data

Obtains the metadata stored with a measured point as a string.

Input Arguments

| | | | |
|---|------------|------------|---|
| 0 | Point Name | Point Name | The point name from which the metadata should be extracted. |
|---|------------|------------|---|

Return Arguments

| | | | |
|---|--------|-----------|---------------------------|
| 1 | String | Info Data | The measurement metadata. |
|---|--------|-----------|---------------------------|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The data was extracted successfully. |
| FAILURE | The point could not be found. |

Remarks

If a point has more than one observation, then the metadata will be extracted from the first observation.

Get B-Spline Properties

Retrieves the properties of a B-Spline.

Input Arguments

| | | | |
|---|------------------------|---------------|--------------------------------------|
| 0 | Collection Object Name | B-Spline Name | The name of the B-Spline to examine. |
|---|------------------------|---------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|----------------|--|
| 1 | Integer | Degree | The degree of the curve. |
| 2 | Integer | Knots | The number of knots in the curve. |
| 3 | Integer | Control Points | The number of control points in the curve. |
| 4 | Double | Range Min | The minimum parameterization range of the curve. |
| 5 | Double | Range Max | The maximum parameterization range of the curve. |
| 6 | Double | Length | The length of the curve. |

Returned Status

| | |
|---------|--|
| SUCCESS | The properties were obtained successfully. |
| FAILURE | The B-Spline could not be found. |

Remarks

None.

Get Cone Properties

Retrieves the properties of a cone.

Input Arguments

| | | | |
|---|------------------------|-----------|----------------------------------|
| 0 | Collection Object Name | Cone Name | The name of the cone to examine. |
|---|------------------------|-----------|----------------------------------|

Return Arguments

| | | | |
|---|--------|---|---|
| 1 | Vector | Cone End Point (in working coordinates) | The location of the apex of the cone. |
| 2 | Vector | Cone Axis (in working coordinates) | A vector describing the axis direction of the cone. |
| 3 | Double | Cone Length | The length of the cone. |
| 4 | Double | Cone Theta Start | The start sweep angle for the cone. |
| 5 | Double | Cone Theta Span | The end sweep angle for the cone. |
| 6 | Double | Cone Included Angle | The included angle of the cone. |

Returned Status

| | |
|---------|--|
| SUCCESS | The properties were obtained successfully. |
| FAILURE | The cone could not be found. |

Remarks

None.

Get Slot Properties

Retrieves the properties of a slot.

Input Arguments

| | | | |
|---|------------------------|-----------|----------------------------------|
| 0 | Collection Object Name | Slot Name | The name of the slot to examine. |
|---|------------------------|-----------|----------------------------------|

Return Arguments

| | | | |
|---|-----------|---|--|
| 1 | Transform | Slot Transform (in working coordinates) | The location of the slot in working coordinates. |
| 2 | Vector | Slot Position (in working coordinates) | A vector describing the slot's center position. |
| 3 | Vector | Slot Orientation (in working coordinates) | A normal vector describing the slots orientation. |
| 4 | Double | Slot Length | The length of the slot. |
| 5 | Double | Slot Width | The width of the slot. |
| 6 | Boolean | Round Slot type | True indicates that the slot is round, false indicates the slot is square. |

Returned Status

| | |
|---------|--|
| SUCCESS | The properties were obtained successfully. |
| FAILURE | The slot could not be found. |

Remarks

None.

Raster Scan Edge Inspection

Uses ordered (raster) scan data, a surface, and B-Splines defining surface edges to identify edge points from within point clouds. It then creates vectors and determines if the points meet a specified tolerance.

Input Arguments

| | | | |
|---|---------------------------------|---|---|
| 0 | Collection Object Name Ref List | Cloud Names | The names of the clouds to consider. |
| 1 | Collection Object Name | Edge Surface Name | The surface containing the edges to evaluate. |
| 2 | Collection Object Name Ref List | BSpline Edge List | The name of one or more B-Splines along the edge of the surface in argument 1. |
| 3 | Collection Object Name | Prefix for Output Groups | A prefix to apply to the names of all resulting groups from the command. |
| 4 | Double | Tolerance | A tolerance value to meet. |
| 5 | Integer | Minimum Number of "Good" Points per Unit Length | The minimum number of points required per unit B-Spline length required to pass the evaluation. |
| 6 | Double | Maximum Percentage of "Bad" Points (0-100) | The maximum percentage of points that can exceed the tolerance and still be considered passing. |

Return Arguments

| | | | |
|---|--------|----------------|--|
| 7 | String | Summary Result | A text summary of the result of the command. |
|---|--------|----------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The evaluation was performed successfully. |
| FAILURE | One or more required objects was not found. |

Remarks

This command finds cloud points close to the specified edge, and creates four point groups: One for points in the tolerance, one for outside the tolerance, one for outside the tolerance and above, and one for outside the tolerance and below. Vector groups are the created projecting the "above" and "below" points to the nominal edge.

New Raster Scan Edge Inspection

Uses ordered (raster) scan data, a surface, and a B-Spline defining surface edges to identify edge points from within point clouds. It then creates vectors and determines if the points meet a specified tolerance.

Input Arguments

| | | | |
|----|---------------------------------|---|---|
| 0 | Collection Object Name Ref List | Edge Cloud Names | The names of the clouds to consider. |
| 1 | Collection Object Name | Edge Surface Name | The surface containing the edges to evaluate. |
| 2 | Collection Object Name | Edge BSpline Name | The name of a B-Spline along the edge of the surface in argument 1. |
| 3 | Collection Object Name | Output Prefix | A prefix to apply to the names of all resulting groups from the command. |
| 4 | Double | Inspection Increment | The defined distance along the edge at which to sample the cloud data. |
| 5 | Double | Proximity Filter Distance | Points outside this distance are unused. |
| 6 | Double | Edge Bias Value | Affects material thickness at the edges. |
| 7 | Double | Error Tolerance | The allowable tolerance for errors at the edges. |
| 8 | Boolean | Use Cosine Projection Method | Indicates whether the cloud points are projected back to the surface before the distance from the spline is computed. |
| 9 | Integer | Minimum Number of Edge Points per segment | The minimum number of points required for each segment of the edge. |
| 10 | File Path or Embedded File | Intermediate Calculation Results File(optional) | An optional file in which to store a log of the intermediate results of the calculations. |

Return Arguments

| | | | |
|----|--------|----------------|--|
| 11 | String | Summary Result | A text summary of the result of the command. |
|----|--------|----------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The evaluation was performed successfully. |
| FAILURE | One or more required objects was not found. |

Remarks

This command finds cloud points close to the specified edge, and creates four point groups: One for points in the tolerance, one for outside the tolerance, one for outside the tolerance and above, and one for outside the tolerance and below. Vector groups are the created projecting the “above” and “below” points to the nominal edge.

Mushroom Target Hole Inspection

Calculates a hole center using points measured with a mushroom target.

Input Arguments

| | | | |
|---|------------------------|--|--|
| 0 | String | Name Prefix for Intermediate Constructions | A prefix to use for intermediate construction geometry. |
| 1 | Collection Object Name | Sphere Points Group Name | The group containing the points measured with a mushroom target. |
| 2 | Double | Sphere Target Radius | The radius of the mushroom target. |
| 3 | Collection Object Name | Target Contact Plane | The plane at the top of the hole in which the target is resting. |
| 4 | Point Name | Point To Create at Hole | The name of the point to create at the center of the hole. |

Return Arguments

| | | | |
|---|--------|----------------------|--|
| 5 | Double | Sphere Fit RMS Error | The resulting RMS error of the sphere fit. |
| 6 | Double | Sphere Fit Max Error | The resulting maximum error of the sphere fit. |

Returned Status

| | |
|---------|---|
| SUCCESS | The hole point was created successfully. |
| FAILURE | One or more required items was not found. |

Remarks

When measuring, place the probe on a mushroom target and set the mushroom head into the hole. Measure a series of points while rotating the mushroom target to different positions while continuing to rest in the hole. A sphere center will be calculated from these points, which will then be projected to the target contact plane.

Sphere Axis Check

Compares the center of a set of spherical measured points to an axis.

Input Arguments

| | | | |
|---|------------------------|----------------------------------|--|
| 0 | Collection Object Name | Sphere Points Group Name | The name of the point group containing the measured points. |
| 1 | Double | Sphere Target Radius | The radius of the mushroom target/sphere rod. |
| 2 | Point Name | Point To Create at Sphere Center | The name of the point to create at the center of the sphere. |
| 5 | Collection Object Name | Line defining the axis | The line defining the axis to compare against. |

Return Arguments

| | | | |
|----|--------|-----------------------|---|
| 3 | Double | Sphere Fit RMS Error | The resulting RMS error of the sphere fit. |
| 4 | Double | Sphere Fit Max Error | The resulting maximum error of the sphere fit. |
| 6 | Vector | Vector Representation | The vector from the measured center to the closest point on the axis. |
| 7 | Double | X Value | The x-component of the vector from argument 6. |
| 8 | Double | Y Value | The y-component of the vector from argument 6. |
| 9 | Double | Z Value | The z-component of the vector from argument 6. |
| 10 | Double | Magnitude | The magnitude of the vector from argument 6. |

Returned Status

| | |
|---------|---|
| SUCCESS | The center point and axis were compared successfully. |
| FAILURE | One or more required objects was not found. |

Remarks

One example use of this would be to compare a set of points measured with a mushroom target to a hole axis.

Patch Normal Shift - Point

Projects a point to a plane fit through another point group.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|---|
| 0 | Collection Object Name | Plane Points Group Name | The group containing the points that define the plane to project to. |
| 1 | Point Name | Point to Shift | The point you want to shift to the plane. |
| 2 | Point Name | Resulting Point Name | The name of the shifted point. |
| 3 | Double | Additional Material Thickness | Offsets away from the plane by a specified amount to account for additional material thickness. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The shifted point was created successfully. |
| FAILURE | The plane points or point to shift were not found. |

Remarks

None.

Patch Normal Shift - Hole / Pin

Constructs a point at the center of a circle defined by a point group, then projects that point to a plane defined by another point group, optionally offsetting to account for material thickness.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|--|
| 0 | Collection Object Name | Plane Points Group Name | The group containing the points that define the plane to project to. |
| 1 | Collection Object Name | Perimeter Points Group Name | The group containing the points that define the circumference of the pin or hole. |
| 2 | Point Name | Resulting Point Name | The name of the shifted center point. |
| 3 | Double | Additional Material Thickness | Offsets the center point away from the plane by a specified amount to account for additional material thickness. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The shifted point was created successfully. |
| FAILURE | The plane or perimeter points were not found. |

Remarks

None.

Angle Between Two Planes' normals

Calculates the angle between two planes.

Input Arguments

| | | | |
|---|------------------------|--------------------------------|--|
| 0 | Collection Object Name | Plane A | The first plane. |
| 1 | Collection Object Name | Plane B | The second plane. |
| 2 | Double | Nominal Angle | A nominal angle (only used to determine if the tolerance is satisfied, if used). |
| 3 | Double | Angle Tolerance (0.0 for none) | The allowable symmetrical tolerance for the angle, plus or minus from the nominal angle. |

Return Arguments

| | | | |
|---|--------|-------|---|
| 4 | Double | Angle | The actual angle between the two planes |
|---|--------|-------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The plane angle was calculated successfully, and the tolerance (if specified) was not exceeded. |
| FAILURE | One or both planes were not found, or the calculated angle exceeded the tolerance (if specified). |

Remarks

To use a tolerance, specify a number other than zero for argument 3. To ignore the tolerance, leave argument 3 at 0.0.

Angle Between Line and Plane

Calculates the angle between a line and a plane.

Input Arguments

| | | | |
|---|------------------------|--------------------------------|--|
| 0 | Collection Object Name | Selected Line | The line. |
| 1 | Collection Object Name | Selected Plane | The plane. |
| 2 | Double | Nominal Angle | A nominal angle (only used to determine if the tolerance is satisfied, if used). |
| 3 | Double | Angle Tolerance (0.0 for none) | The allowable symmetrical tolerance for the angle, plus or minus from the nominal angle. |

Return Arguments

| | | | |
|---|--------|-------|--|
| 4 | Double | Angle | The actual angle between the line and plane. |
|---|--------|-------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The angle was calculated successfully, and the tolerance (if specified) was not exceeded. |
| FAILURE | The line/plane or both were not found, or the calculated angle exceeded the tolerance (if specified). |

Remarks

To use a tolerance, specify a number other than zero for argument 3. To ignore the tolerance, leave argument 3 at 0.0.

Angle Between Two Lines

Calculates the angle between two lines.

Input Arguments

| | | | |
|---|------------------------|--------------------------------|--|
| 0 | Collection Object Name | Line 1 | The first line. |
| 1 | Collection Object Name | Line 2 | The second line. |
| 2 | Double | Nominal Angle | A nominal angle (only used to determine if the tolerance is satisfied, if used). |
| 3 | Double | Angle Tolerance (0.0 for none) | The allowable symmetrical tolerance for the angle, plus or minus from the nominal angle. |

Return Arguments

| | | | |
|---|--------|-------|---|
| 4 | Double | Angle | The actual angle between the two lines. |
|---|--------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The angle was calculated successfully, and the tolerance (if specified) was not exceeded. |
| FAILURE | One or both lines were not found, or the calculated angle exceeded the tolerance (if specified). |

Remarks

To use a tolerance, specify a number other than zero for argument 3. To ignore the tolerance, leave argument 3 at 0.0.

Group To Surface Fit

Calculates the transform required to fit a point group to a specified surface.

Input Arguments

| | | | |
|---|------------------------|--|--|
| 0 | Collection Object Name | Group to Fit | The point group to fit to the surface. |
| 1 | Collection Object Name | Surface | The surface to fit to. |
| 2 | Boolean | Do Conventional Fit | Specifies whether a conventional fit is used, or whether the Direct fit is used. |
| 3 | Double | RMS Tolerance (0.0 for none) | An optional tolerance for the RMS of the fit. |
| 4 | Double | Maximum Absolute Tolerance(0.0 for none) | An optional maximum absolute error allowable for the fit. |

Return Arguments

| | | | |
|---|--------------------------|----------------------------|---|
| 5 | World Transform Operator | Optimum Transform | The ideal transform for the fit. |
| 6 | Double | RMS Deviation | The actual RMS error |
| 7 | Double | Maximum Absolute Deviation | The largest actual deviation between a point and the surface. |

Returned Status

| | |
|---------|---|
| SUCCESS | The fit was calculated successfully. |
| FAILURE | The group or surface was not found, or the fit exceeded the specified tolerances. |

Remarks

When “Do Conventional Fit” is set to TRUE, the standard fit optimization is performed, however this optimization is susceptible to getting trapped in a local minimum, terminating the optimization prematurely. When false, the “Direct ” optimization method is used, which is a brute force method that more frequently reaches a final solution without getting trapped in a local minimum--although it usually takes significantly longer to calculate.

Reverse Surface Normals

Reverses the normals of specified surfaces.

Input Arguments

| | | | |
|---|---------------------------------|--------------|----------------------------------|
| 0 | Collection Object Name Ref List | Surface List | The list of surfaces to reverse. |
|---|---------------------------------|--------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The surfaces were reversed successfully. |
| PARTIAL SUCCESS | One or more surfaces (but not all) could not found. |
| FAILURE | None of the specified surfaces could be found. |

Remarks

None.

Reverse B-Splines

Reverses the normals of specified B-Splines.

Input Arguments

| | | | |
|---|---------------------------------|---------------|-----------------------------------|
| 0 | Collection Object Name Ref List | B-Spline List | The list of B-Splines to reverse. |
|---|---------------------------------|---------------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The B-Splines were reversed successfully. |
| PARTIAL SUCCESS | One or more B-Splines (but not all) could not found. |
| FAILURE | None of the specified B-Splines could be found. |

Remarks

None.

Get Surface Physical Stats

Obtains the volume/area of a supplied surface.

Input Arguments

| | | | |
|---|------------------------|--------------|--|
| 0 | Collection Object Name | Surface Name | The name of the surface to examine. |
| 1 | Double | Volume | The volume of the surface (if a closed volume). |
| 2 | Double | Area | The total surface area of the specified surface. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The stats were obtained successfully. |
| FAILURE | The surface was not found. |

Remarks

None.

Set Inward Positive Normal

Sets the normal direction for spheres, cylinder, paraboloids, and cones.

Input Arguments

| | | | |
|---|------------------------|-----------------|--|
| 0 | Collection Object Name | Object Name | The name of the sphere, cylinder, paraboloid, or cone to modify. |
| 1 | Boolean | Inward Postive? | Indicates whether the inside of the object should be the positive side of the surface. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The normal was modified successfully. |
| FAILURE | The object was not found. |

Remarks

None.

Sort Point Group in Database

Sorts the specified point group in the tree (and the underlying database) based on the specified criteria.

Input Arguments

| | | | |
|---|------------------------|--------------------------|---|
| 0 | Collection Object Name | Point Group | The name of the point group to sort. |
| 1 | Point Sort Options | Point Group Sort Options | The criteria and order by which to sort the points. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The group was sorted successfully. |
| FAILURE | The group was not found. |

Remarks

This is the MP equivalent of the manual Sort in Database menu item. Points can be sorted by point name, Cartesian/Cylindric/Spherical coordinate, or database order. Ascending or descending sorts can be performed.

Sort Vectors

Accepts a vector list and returns a sorted list based upon your sort criteria. The “Granularity” can be used to open up the sequential sort process. The primary coordinate is defined first but with a larger granularity set, you can bin that sorting and then sort all the data within the bins by the secondary coordinate, etc.

Input Arguments

| | | | |
|---|------------------------|----------------------------------|---|
| 0 | Vector Name Ref List | Source Vectors | The list of vectors to sort |
| 1 | Sort Method Type | Sort Method | The criteria used to sort the points. |
| 2 | Coordinate System Type | Coordinate System | Selection for Coordinate System such as Cartesian or Polar... |
| 3 | Coordinate Sort Type | Primary Sort Coordinate | Sort coordinate selection XYZ, R, Theta, etc. |
| 4 | Coordinate Sort Type | Secondary Sort Coordinate | Sort coordinate selection XYZ, R, Theta, etc. |
| 5 | Coordinate Sort Type | Tertiary Sort Coordinate | Sort coordinate selection XYZ, R, Theta, etc |
| 6 | Double | Primary Coordinate Granularity | Granularity of Primary Coordinate |
| 7 | Double | Secondary Coordinate Granularity | Granularity of Secondary Coordinate |
| 8 | Double | Tertiary Coordinate Granularity | Granularity of Tertiary Coordinate |
| 9 | Boolean | Ascending? | True returns the best fit first |

Return Arguments

| | | | |
|----|----------------------|----------------|--|
| 10 | Vector Name Ref List | Sorted Vectors | The list of vectors ordered by the specified criteria. |
|----|----------------------|----------------|--|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The Vectors were sorted successfully. |
| FAILURE | The Vectors was not found. |

Remarks

None.

Reverse Plane Normals

Reverses the normals for one or more planes.

Input Arguments

| | | | |
|---|---------------------------------|------------|------------------------------|
| 0 | Collection Object Name Ref List | Plane List | A list of planes to reverse. |
|---|---------------------------------|------------|------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one plane in the list was found. |
| FAILURE | None of the planes in the list were found. |

Remarks

None.

Get Double List Max/Min

Retrieves the maximum and minimum values from a list of doubles.

Input Arguments

| | | | |
|---|-------------|-------------|---------------------------------|
| 0 | Double List | Double List | The list of doubles to analyze. |
|---|-------------|-------------|---------------------------------|

Return Arguments

| | | | |
|---|--------|-----|--------------------------------|
| 1 | Double | Max | The maximum value in the list. |
| 2 | Double | Min | The minimum value in the list. |

Returned Status

| | |
|---------|--|
| SUCCESS | At least one plane in the list was found. |
| FAILURE | None of the planes in the list were found. |

Remarks

None.

Geometry Fit Profiles

Make Line Fit Profile

Creates fit profile for a line.

Input Arguments

| | | | |
|---|---------|----------------------------------|---|
| 0 | String | Fit Profile Name | Name of fit profile. |
| 4 | Boolean | Reverse Normal Vector after fit? | Reverses the normal vector. |
| 5 | Boolean | Make Cardinal Points? | Create cardinal points from plane. |
| 6 | Boolean | Cardinal Pt. 1: Point A? | Create cardinal point at the origin of the line. |
| 7 | Boolean | Cardinal Pt. 2: Point B? | Create cardinal point at the end of the line. |
| 7 | Boolean | Cardinal Pt. 3: Mid Point? | Create cardinal point at the mid point of the line. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The line fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

None.

Make Plane Fit Profile

Creates fit profile for plane.

Input Arguments

| | | | |
|---|---------------------------------|---|---|
| 0 | String | Fit Profile Name | Name of fit profile. |
| 1 | Measured Side for Planar Offset | Measured Side for Planar Offset | The side of the plane that is to be measured. |
| 2 | Double | Override Planar Offset (-1.0 use current) | The amount to use if overriding planar offset. |
| 3 | Normal Direction | Planar Offset Direction | The probing direction of the plane. |
| 4 | Boolean | Reverse Normal Vector after fit? | Reverses the normal vector. |
| 5 | Boolean | Make Cardinal Points? | Create cardinal points from plane. |
| 6 | Boolean | Cardinal Pt. 1: Centroid? | Create cardinal point at the centroid of the plane. |
| 7 | Boolean | Cardinal Pt. 2: Point on Normal? | Create a point on normal. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The plane fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Circle Fit Profile

Creates fit profile for circle.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|----|---------------------------------|---|--|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the circle that is to be measured. |
| 2 | Double | Override radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Measured Side for Planar Offset | Measured Side for Planar Offset | The side of the circle that is to be measured. |
| 4 | Double | Override Planar Offset (-1.0 use current) | The amount to use if overriding planar offset. |
| 5 | Normal Direction | Planar Offset Direction | The planar probing direction. |
| 6 | Double | Lock Radius (-1.0 do not lock) | The value in which to lock the radius. |
| 7 | Computation Technique | Circle Computation Technique | The way in which the circle fit will be computed. |
| 8 | Boolean | Reverse Normal Vector after fit? | Reverse the normal vector. |
| 9 | Boolean | Make Cardinal Points? | Creates cardinal points from circle. |
| 10 | Boolean | Cardinal Pt. 1: Center? | Create a cardinal point at the center of the circle. |
| 11 | Boolean | Cardinal Pt. 2: Point on Normal? | Create a point on normal. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The circle fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Slot Fit Profile

Creates fit profile for slot.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|----|---------------------------------|---|--|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the circle that is to be measured. |
| 2 | Double | Override radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Measured Side for Planar Offset | Measured Side for Planar Offset | The side of the slot that is to be measured. |
| 4 | Double | Override Planar Offset (-1.0 use current) | The amount to use if overriding planar offset. |
| 5 | Normal Direction | Planar Offset Direction | The planar probing direction. |
| 6 | Slot Type | Slot Type | Whether slot is round or square. |
| 7 | Computation Technique | Circle Computation Technique | The way in which the slot fit will be computed. |
| 8 | Boolean | Reverse Normal Vector after fit? | Reverse the normal vector. |
| 9 | Boolean | Make Cardinal Points? | Creates cardinal points from slot. |
| 10 | Boolean | Cardinal Pt. 1: Center? | Create a cardinal point at the center of the slot. |
| 11 | Boolean | Cardinal Pt. 2: Point on Normal? | Create a point on normal. |
| 12 | Boolean | Cardinal Pt. 3: Centerline Pt. 1? | Create a cardinal point along slot centerline. |
| 13 | Boolean | Cardinal Pt. 4: Centerline Pt. 2? | Create a cardinal point along slot centerline. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The slot fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Ellipse Fit Profile

Creates fit profile for ellipse.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|----|---------------------------------|---|--|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the ellipse that is to be measured radially. |
| 2 | Double | Override radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Measured Side for Planar Offset | Measured Side for Planar Offset | The side of the ellipse that is to be measured planar. |
| 4 | Double | Override Planar Offset (-1.0 use current) | The amount to use if overriding planar offset. |
| 5 | Normal Direction | Planar Offset Direction | The planar probing direction. |
| 6 | Boolean | Reverse Normal Vector after fit? | Reverse the normal vector. |
| 7 | Boolean | Make Cardinal Points? | Creates cardinal points from ellipse. |
| 8 | Boolean | Cardinal Pt. 1: Center? | Create a cardinal point at the center of the ellipse. |
| 9 | Boolean | Cardinal Pt. 2: Point on Normal? | Create a point on normal. |
| 10 | Boolean | Cardinal Pt. 3: Centerline Pt. 1? | Create a cardinal point along ellipse centerline. |
| 11 | Boolean | Cardinal Pt. 4: Centerline Pt. 2? | Create a cardinal point along ellipse centerline. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The ellipse fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Sphere Fit Profile

Creates a fit profile for a sphere that can be applied as needed.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | String | Fit Profile Name | Name of fit profile. |
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the sphere that is to be measured. |
| 2 | Double | Override Radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Double | Lock Radius (-1.0 do not lock) | The value in which to lock the radius. |
| 4 | Boolean | Make Cardinal Points? | Create cardinal points from the sphere. |
| 5 | Boolean | Cardinal Pt. 1: Center? | Create cardinal point at the center of the sphere. |
| 6 | Computation Mode | Computation Mode | Select the computation mode needed for the fit. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The sphere fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Cone Fit Profile

Creates fit profile for cone.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|---|---------------------------------|---|--|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the cone that is to be measured. |
| 2 | Double | Override Radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Double | Lock Angle in degrees (-1.0 do not lock) | The value in which to lock the radius. |
| 4 | Boolean | Make Cardinal Points? | Create cardinal points from cone. |
| 5 | Boolean | Cardinal Pt. 1: Vertex? | Create cardinal point at the vertex of the cone. |
| 6 | Boolean | Cardinal Pt. 2: Point on Axis? | Create a cardinal point on the cone axis. |
| 7 | Boolean | Cardinal Pt. 3: Cut Point on Axis? | Create a cardinal point along cone axis on top center of truncated cone. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The cone fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Paraboloid Fit Profile

Creates fit profile for paraboloid.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|---|---------------------------------|---|--|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the paraboloid that is to be measured. |
| 2 | Double | Override Radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Double | Lock Focal Length (-1.0 do not lock) | The value in which to lock the focal length. |
| 4 | Degree of Freedom | Degree of Freedom | Lock focus or vertex location. |
| 5 | Boolean | Make Cardinal Points? | Create cardinal points from paraboloid. |
| 6 | Boolean | Cardinal Pt. 1: Vertex? | Create cardinal point at the vertex of the paraboloid. |
| 7 | Boolean | Cardinal Pt. 2: Focal Point? | Create a cardinal point on paraboloid focal point. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The paraboloid fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Make Cylinder Fit Profile

Creates fit profile for cone.

Input Arguments

| 0 | String | Fit Profile Name | Name of fit profile. |
|---|---------------------------------|---|---|
| 1 | Measured Side for Radial Offset | Measured Side for Radial Offset | The side of the sphere that is to be measured. |
| 2 | Double | Override Radial Offset (-1.0 use current) | The amount to use if overriding radial offset. |
| 3 | Double | Lock Radius (-1.0 do not lock) | The value in which to lock the radius. |
| 4 | Fit Method | Locked Radius Fit Method | The fit method used to lock the radius. |
| 5 | Computation Technique | Cylinder Computation Technique | The way in which the cylinder fit will be computed. |
| 6 | Boolean | Make Cardinal Points? | Create cardinal points from cylinder. |
| 7 | Boolean | Cardinal Pt. 1: Begin Pt? | Create cardinal point at the beginning of the cylinder. |
| 8 | Boolean | Cardinal Pt. 2: End Pt? | Create a cardinal point at the end of the cylinder. |
| 9 | Boolean | Cardinal Pt. 3: Center? | Created a cardinal point at the center of the cylinder. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The cylinder fit profile was created successfully. |
| FAILURE | The fit profile could not be created. |

Remarks

Leave the fit profile name blank in order to use the default fit profile for the specified geometry type. Use an offset of -1.0 to use the offset defined in the object properties or a value of 0.0 to indicate that no offset should apply.

Set Geometry Relationship Fit Profile

Applies a fit profile's defined settings to an existing geometry relationship.

Input Arguments

| | | | |
|---|-----------------------|-------------------------------|--|
| 0 | Geometry Type | Geometry Type | Selection for the type of geometry to be used. |
| 1 | Relationship Ref List | Relationship Ref List | List of the relationships to be edited. |
| 2 | String | Fit Profile Name | Name of the fit profile to be applied. |
| 3 | Boolean | Apply Cardinal Point Settings | True applies the cardinal point settings from the fit profile. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The fit profile was applied successfully. |
| FAILURE | The fit profile or relationships could not be found or are the of the wrong type. |

Remarks

None.

GD&T Operations

Get Number of Feature Checks in Feature Check Ref List

Returns the number of feature checks in a feature check reference list.

Input Arguments

| | | | |
|---|------------------------|--------------------|--------------------------------------|
| 0 | Feature Check Ref List | Feature Check List | The list of feature checks to count. |
|---|------------------------|--------------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|---|
| 1 | Integer | Total Count | The number of feature checks in the list. |
|---|---------|-------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Feature Check Datum References

Returns the datum references from a specified feature check. This includes the datum names and objects used to define the datums.

Input Arguments

| | | | |
|---|------------------------|---------------|-----------------------|
| 0 | Collection Object Name | Feature Check | The check to consider |
|---|------------------------|---------------|-----------------------|

Return Arguments

| | | | |
|---|---------------------------------|------------------------------------|--|
| 1 | String | Datum 1 Reference String | Name of the 1st datum |
| 2 | Surface Faces | Datum 1 CAD Faces | List of CAD face indexes referenced |
| 3 | Collection Object Name Ref List | Datum 1 SA Objects | List of SA objects referenced |
| 4 | Collection Object Name Ref List | Datum 1 Aux SA Objects | List of offset or auxiliary objects referenced |
| 5 | Collection Object Name Ref List | Datum 1 Geometry Relationships | List of GR-Features referenced |
| 6 | Collection Object Name Ref List | Datum 1 Aux Geometry Relationships | List of offset or auxiliary GR-Features |
| 1 | String | Datum 2 Reference String | Name of the 2nd datum |
| 2 | Surface Faces | Datum 2 CAD Faces | List of CAD face indexes referenced |
| 3 | Collection Object Name Ref List | Datum 2 SA Objects | List of SA objects referenced |
| 4 | Collection Object Name Ref List | Datum 2 Aux SA Objects | List of offset or auxiliary objects referenced |
| 5 | Collection Object Name Ref List | Datum 2 Geometry Relationships | List of GR-Features referenced |
| 6 | Collection Object Name Ref List | Datum 2 Aux Geometry Relationships | List of offset or auxiliary GR-Features |
| 1 | String | Datum 3 Reference String | Name of the 3rd datum |
| 2 | Surface Faces | Datum 3 CAD Faces | List of CAD face indexes referenced |
| 3 | Collection Object Name Ref List | Datum 3 SA Objects | List of SA objects referenced |
| 4 | Collection Object Name Ref List | Datum 3 Aux SA Objects | List of offset or auxiliary objects referenced |
| 5 | Collection Object Name Ref List | Datum 3 Geometry Relationships | List of GR-Features referenced |
| 6 | Collection Object Name Ref List | Datum 3 Aux Geometry Relationships | List of offset or auxiliary GR-Features |

Returned Status

| | |
|---------|--|
| SUCCESS | This feature check references were returned successfully |
| FAILURE | The feature check could not be found |

Remarks

None.

Get i-th Feature Check From Feature Check Ref List

Returns the name of a feature check at a specified index in a feature check reference list.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---------------------------------|
| 0 | Feature Check Ref List | Feature Check Name List | The list of Feature Checks. |
| 1 | Integer | Feature Check Index | The index of the feature check. |

Return Arguments

| | | | |
|---|------------------------|----------------|---|
| 2 | Collection Object Name | Resultant Item | The name of the feature check at the specified index. |
|---|------------------------|----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The feature check name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first feature check in a ref list is at index 0.

Get i-th Feature Check From Feature Check Ref List (Iterator)

Returns the name of a feature check at a specified index in a feature check reference list. This command is an iterator, which means it is a shortcut to a traditional loop. Iterators proceed sequentially, one-by-one, from a starting index, until the end of the list.

Input Arguments

| | | | |
|---|------------------------|-----------------------------|---|
| 0 | Feature Check Ref List | Reference List | The list of feature checks. |
| 1 | Integer | Feature Check Index | The feature check index at which to start. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to after iterating through the last feature check in the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|--|
| 3 | String | Collection | The collection containing the feature check at the specified index. |
| 4 | String | Feature Check | The name of the feature check at the specified index. |
| 5 | Collection Object Name | Resultant Item | The feature check at the specified index, as a collection object name data type. |

Returned Status

| | |
|---------|---|
| SUCCESS | The feature check name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Get Number of Datums in Datum Ref List

Returns the number of datums in a list of datums.

Input Arguments

| | | | |
|---|----------------|----------------|---------------------|
| 0 | Datum Ref List | Datum Ref List | The list of datums. |
|---|----------------|----------------|---------------------|

Return Arguments

| | | | |
|---|---------|-------------|-----------------------------------|
| 1 | Integer | Total Count | The number of datums in the list. |
|---|---------|-------------|-----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The number of datums was determined successfully. |
| FAILURE | The supplied list was invalid. |

Remarks

None.

Set Feature Check Reporting Options

Sets the reporting options for a specific feature check.

Input Arguments

| 0 | Collection Object Name | Feature Check | The name of the intended feature check. |
|---|-------------------------|-------------------------------------|---|
| 1 | Boolean | Show Feature Control Frame Summary? | True includes the standard Feature Control Frame image in the report |
| 2 | Boolean | Include Title? | True includes the Check Title |
| 3 | Boolean | Show Datum and Tolerance Summary? | True includes the Datum and Tolerance Summary Table in the report. |
| 4 | Boolean | Show Feature Summary? | True includes the Feature Summary Table in the report. |
| 5 | Vector Creation Trigger | Vector Creation | Selection will enable, disable, or set vector creation to only be triggered for failed checks |
| 6 | Boolean | Show Point Details Summary? | True includes the Point Details Summary Table in the report. |

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting options for the feature check was set successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Get Feature Check Reporting Options

Returns the reporting options for a specific feature check.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Feature Check | The name of the intended feature check. |
|---|------------------------|---------------|---|

Return Arguments

| | | | |
|---|-------------------------|-------------------------------------|--|
| 1 | Boolean | Show Feature Control Frame Summary? | True includes the standard Feature Control Frame image in the report |
| 2 | Boolean | Include Title? | True includes the Check Title |
| 3 | Boolean | Show Datum and Tolerance Summary? | True includes the Datum and Tolerance Summary Table in the report. |
| 4 | Boolean | Show Feature Summary? | True includes the Feature Summary Table in the report. |
| 5 | Vector Creation Trigger | Vector Creation | Returns Disable, Always or Only on Fail |
| 6 | Boolean | Show Point Details Summary? | True includes the Point Details Summary Table in the report. |

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting options for the feature check was retrieved successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Get i-th Datum From Datum Ref List

Returns the datum at the specified index in a list of datums.

Input Arguments

| | | | |
|---|----------------|----------------|------------------------|
| 0 | Datum Ref List | Datum Ref List | The list of datums. |
| 1 | Integer | Datum Index | The index in the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|----------------------|
| 2 | Collection Object Name | Resultant Item | The resulting datum. |
|---|------------------------|----------------|----------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The datum was retrieved successfully. |
| FAILURE | The supplied list was invalid, or the index was invalid. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Get i-th Datum From Datum Ref List (Iterator)

Iterates through a list of datums, returning a datum on each iteration.

Input Arguments

| | | | |
|---|----------------|-----------------------------|---|
| 0 | Datum Ref List | Datum Ref List | The list of datums. |
| 1 | Integer | Datum Index | The index at which to start iterating through the list. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to at the end of the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|----------------------|
| 3 | Collection Object Name | Resultant Item | The resulting datum. |
|---|------------------------|----------------|----------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The datum was retrieved successfully. |
| FAILURE | The supplied list or index was invalid. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Get i-th Annotation From Annotation Ref List

Returns the annotation at the specified index in a list of datums.

Input Arguments

| | | | |
|---|---------------------|---------------------|-------------------------|
| 0 | Annotation Ref List | Annotation Ref List | The list of annotation. |
| 1 | Integer | Annotation Index | The index in the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|---------------------------|
| 2 | Collection Object Name | Resultant Item | The resulting annotation. |
|---|------------------------|----------------|---------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The annotation was retrieved successfully. |
| FAILURE | The supplied list was invalid, or the index was invalid. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Get i-th Annotation From Annotation Ref List (Iterator)

Iterates through a list of annotations, returning an annotation on each iteration.

Input Arguments

| | | | |
|---|---------------------|-----------------------------|---|
| 0 | Annotation Ref List | Annotation Ref List | The list of annotations. |
| 1 | Integer | Annotation Index | The index at which to start iterating through the list. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to at the end of the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|---------------------------|
| 3 | Collection Object Name | Resultant Item | The resulting annotation. |
|---|------------------------|----------------|---------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The annotation was retrieved successfully. |
| FAILURE | The supplied list or index was invalid. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Get Datum Measurements

Retrieves a list of the points/clouds associated with a given datum.

Input Arguments

| | | | |
|---|------------------------|-------|-----------------------|
| 0 | Collection Object Name | Datum | The datum to examine. |
|---|------------------------|-------|-----------------------|

Return Arguments

| | | | |
|---|---------------------------------|-------------|---|
| 1 | Point Name Ref List | Point Names | The list of points associated with the datum. |
| 2 | Collection Object Name Ref List | Cloud Names | The list of clouds associated with the datum. |

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The datum was examined successfully. |
| FAILURE | The datum was not found. |

Remarks

None.

Set Datum Measurements

Associates a list of points/clouds with a given datum.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------------|---|
| 0 | Collection Object Name | Datum | The datum of interest. |
| 1 | Point Name Ref List | Point Names | The list of points to associate with the datum. |
| 2 | Collection Object Name Ref List | Cloud Names | The list of clouds to associate with the datum. |
| 3 | Boolean | Replace Existing Measurements? | If TRUE, existing associated measurements will be cleared out before making the assignment. |

Return Arguments

| | | | |
|---|---------------------------------|-------------|---|
| 1 | Point Name Ref List | Point Names | The list of points associated with the datum. |
| 2 | Collection Object Name Ref List | Cloud Names | The list of clouds associated with the datum. |

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The datum was examined successfully. |
| FAILURE | The datum was not found. |

Remarks

None.

Get Feature Check Measurements

Retrieves the point and point cloud measurements associated with a specified feature check.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Feature Check | The name of the feature check to examine. |
|---|------------------------|---------------|---|

Return Arguments

| | | | |
|---|---------------------------------|-------------|---|
| 1 | Point Name Ref List | Point Names | The list of any point measurements associated with the feature check. |
| 2 | Collection Object Name Ref List | Cloud Names | The list of any point cloud measurements associated with the feature check. |

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were retrieved successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Set Feature Check Measurements

Assigns point and/or point cloud measurements to a specified feature check.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------------|--|
| 0 | Collection Object Name | Feature Check | The name of the feature check to modify. |
| 1 | Point Name Ref List | Point Names | The list of any point measurements to associate with the feature check. |
| 2 | Collection Object Name Ref List | Cloud Names | The list of any point clouds to associate with the feature check. |
| 3 | Boolean | Replace Existing Measurements? | If TRUE, existing associated measurements will be cleared. If FALSE, the specified measurements will be added to the set of existing measurements. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were associated successfully. |
| FAILURE | The feature check, point, or clouds could not be found. |

Remarks

None.

Get Feature Check Cylinder Eval Options

Retrieves the Actual Diameter Override value (and whether it's enabled) from a feature check's cylinder evaluation options.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Feature Check | The name of the feature check to examine. |
|---|------------------------|---------------|---|

Return Arguments

| | | | |
|---|---------|---------------------------------|--|
| 1 | Boolean | Enable Actual Diameter Override | Indicates if the "Actual Diameter Override" checkbox is checked. |
| 2 | Double | Actual Diameter Override | Indicates the value for the actual diameter override. |

Returned Status

| | |
|---------|--|
| SUCCESS | The feature check was examined successfully. |
| FAILURE | The feature check could not be found. |

Remarks

None.

Set Feature Check Cylinder Eval Options

Sets the Actual Diameter Override value (and whether it's enabled) from a feature check's cylinder evaluation options.

Input Arguments

| 0 | Collection Object Name | Feature Check | The name of the feature check to set. |
|---|------------------------|---------------------------------|--|
| 1 | Boolean | Enable Actual Diameter Override | Determines whether the "Actual Diameter Override" option is enabled. |
| 2 | Double | Actual Diameter Override | Sets the value for the actual diameter override. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The feature check was set successfully. |
| FAILURE | The feature check could not be found. |

Remarks

None.

Feature Inspection Auto Filter

Auto-filters and associates a set of points, groups, and clouds to a set of datums and feature checks in the active collection.

Input Arguments

| | | | |
|----|---------------------------------|-------------------------------------|---|
| 0 | Point Name Ref List | Point Names | A list of points to filter. |
| 1 | Collection Object Name Ref List | Group Names | A list of groups to filter. |
| 2 | Collection Object Name Ref List | Cloud Names | A list of clouds to filter. |
| 3 | Double | Surface Offset | The maximum allowable offset from the surface when filtering. |
| 4 | Double | Edge Offset | The minimum allowable offset from any edge when filtering. |
| 5 | Offset Direction Type | Offset Direction | BOTH filters points on both sides of the surface. POSITIVE ONLY filters and associates only points on the positive side of a surface. NEGATIVE ONLY filters and associates only points on the negative side of a surface. |
| 6 | Boolean | Enforce Max Pts per Face in Output? | If TRUE, no more than the maximum number of points specified in argument 7 will be associated. If FALSE, all points will be used. |
| 7 | Integer | Max Pts per Face | The maximum number of points allowed to be filtered to a surface, if argument 6 is TRUE. |
| 8 | Feature Check Ref List | Feature Check Name List | A list of feature checks to include in the filter. |
| 9 | Boolean | Include Datums? | Indicates whether or not datums should be included in the filter. |
| 10 | Boolean | Create Cloud for Each Datum/Check | Indicates whether a separate cloud should be created for each individual datum or feature check. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were filtered and associated successfully. |
| FAILURE | The points, groups, or clouds could not be found. |

Remarks

None.

Evaluate Feature Check

Evaluates a GD&T Feature Check.

Input Arguments

| | | | |
|---|------------------------|---------------------|--|
| 0 | Collection Object Name | Feature Check | The name of the Feature Check to evaluate. |
| 1 | Boolean | Perform Evaluation? | Specify whether or not to evaluate check. |

Return Arguments

| | | | |
|----|--------------------------|---------------------------------------|---|
| 2 | Boolean | Check Evaluated? | Indicates whether the check evaluated successfully. |
| 3 | String | Check Result | The result of the feature check, as a string. |
| 4 | Boolean | Non-unique Result? | Indicates whether the solution is unique or is one of several possible solutions. |
| 5 | Double | Measured Deviation (Upper) | The deviation on the upper side of nominal. |
| 6 | Double | Distance Out of Tolerance (Upper) | The amount out of tolerance on the upper side of nominal. |
| 7 | World Transform Operator | Eval Delta Transform (Upper) | The upper transform applied to the measurements at evaluation. |
| 8 | Double | Measured Deviation (Lower) | The deviation on the lower side of nominal. |
| 9 | Double | Distance Out of Tolerance (Lower) | The amount out of tolerance on the lower side of nominal. |
| 10 | World Transform Operator | Eval Delta Transform (Lower) | The lower transform applied to the measurements at evaluation. |
| 11 | String | Check Type | The type of feature check. |
| 12 | String | Tolerance Type | The type of the tolerance. |
| 13 | Double | Tolerance, Simple | The tolerance (if a simple tolerance). |
| 14 | Double | Tolerance, Composite (Upper) | The composite upper tolerance (if applicable). |
| 15 | Double | Tolerance, Composite (Lower) | The composite lower tolerance (if applicable). |
| 16 | Double | Tolerance, Range (Min) | The minimum range tolerance (if applicable). |
| 17 | Double | Tolerance, Range (Max) | The maximum range tolerance (if applicable). |
| 18 | Double | Tolerance, NominalPlusMinus (Nominal) | The nominal value in a nominal/plus/minus tolerance (if applicable). |
| 19 | Double | Tolerance, NominalPlusMinus (Minus) | The "minus" value in a nominal/plus/minus tolerance (if applicable). |
| 20 | Double | Tolerance, NominalPlusMinus (Plus) | The "plus" value in a nominal/plus/minus tolerance (if applicable). |

Returned Status

| | |
|---------|---|
| SUCCESS | The command was performed successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Evaluate Feature Checks

Evaluates a list of GD&T Feature Checks.

Input Arguments

| | | | |
|---|------------------------|--------------------|--|
| 0 | Feature Check Ref List | Feature Check List | The list of feature checks to evaluate |
|---|------------------------|--------------------|--|

Return Arguments

| | | | |
|---|---------|------------------|---|
| 1 | Integer | Total Passed | Indicates the number of feature checks that passed. |
| 2 | Integer | Total Failed | Indicates the number of feature checks that failed. |
| 3 | Integer | Total Incomplete | Indicates the number of feature checks that were determined to be incomplete. |

Returned Status

| | |
|-----------------|---|
| SUCCESS | All feature checks were found and passed. |
| PARTIAL SUCCESS | At least one feature check was not found or was incomplete. |
| FAILURE | No feature checks were supplied to the command. |

Remarks

None.

Start/Stop Feature Check Trapping

Starts or stops trapping of measurements from a live instrument to a feature check.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|--|
| 0 | Collection Object Name | Feature Check | The feature check in question. |
| 1 | Collection Instrument ID | Instrument ID | The live instrument for which to start/stop trapping. |
| 2 | Boolean | Start Trapping (FALSE = Stop) | Indicates whether trapping should be started or stopped. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Trapping started/stopped successfully. |
| FAILURE | The feature check was not found, or the collection instrument ID was invalid. |

Remarks

None.

Enable/Disable Datum Alignment for Feature Check

Enables or disables GD&T alignment for a feature check evaluation. The feature check is evaluated using the existing alignment at runtime.

Input Arguments

| 0 | Collection Object Name | Feature Check | The feature check to modify. |
|---|------------------------|---------------------------------|--|
| 1 | Boolean | Enable Datum Alignment? | If TRUE, datum alignment is used. Otherwise, the existing alignment is used. |
| 2 | Boolean | Enable Custom Initial Alignment | If TRUE, the check will use the custom initial alignment specified. |
| 3 | Collection Object Name | Alignment | Name of the saved alignment. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Datum Alignment was enabled/disabled successfully. |
| FAILURE | The feature check was not found. |

Remarks

Refer to the GD&T Chapter of the Users Manual for more information on Enable/Disable controls for feature checks.

Datum Alignment

Performs a datum alignment based on a specified feature check.

Input Arguments

| | | | |
|---|-----------------------------------|---------------------|---|
| 0 | Collection Object Name | Feature Check | The name of the Feature Check containing the desired alignment. |
| 1 | Collection Object Name Ref List | Objects to Move | A list of objects to move in the alignment. |
| 2 | Collection Instrument ID Ref List | Instruments to Move | A list of instruments to move in the alignment. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The alignment was performed successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Get GD&T Options

Retrieves the GD&T analysis options for the current job file.

Input Arguments

None.

Return Arguments

| | | | |
|---|------------------------------------|-------------------------------|--|
| 0 | Boolean | Use High Points | Indicates whether GD&T Evaluation is using high points in its datum alignment. |
| 1 | GD&T Options Distance Between Mode | Distance Between Mode | Indicates whether distance between checks are set to use centroid or Min/Max values. |
| 2 | GD&T Options Check Validator Type | Check Pre-Eval Validator Type | Indicates which spec is used for pre-evaluation validation of GD&T checks (None, ASME 1994, ASME 2009, ISO 1983, or ISO 2004). |
| 3 | Boolean | Create Actual Features | Indicates if actual features are set to be created. |
| 4 | Boolean | Create Solved Points | Indicates if solved points are set to be created. |
| 5 | Double | Cross Section Criteria | The distance threshold along the primary axis for points to be considered part of the same cross section for GD&T analysis. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set GD&T Options

Sets the GD&T analysis options for the current job file.

Input Arguments

| | | | |
|---|------------------------------------|-------------------------------|---|
| 0 | Boolean | Use High Points | Indicates whether GD&T Evaluation should use high points in its datum alignment. |
| 1 | GD&T Options Distance Between Mode | Distance Between Mode | Indicates whether distance between checks should use centroid or Min/Max values. |
| 2 | GD&T Options Check Validator Type | Check Pre-Eval Validator Type | Indicates which spec should be used for pre-evaluation validation of GD&T checks (None, ASME 1994, ASME 2009, ISO 1983, or ISO 2004). |
| 3 | Boolean | Create Actual Features | Indicates if actual features should be created. |
| 4 | Boolean | Create Solved Points | Indicates if solved points should be created. |
| 5 | Double | Cross Section Criteria | The distance threshold along the primary axis for points to be considered part of the same cross section for GD&T analysis. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Refresh Datums/Feature Checks from Annotations

Refreshes all datums and feature checks from the annotations in the specified collection.

Input Arguments

| | | | |
|---|-----------------|------------|--|
| 0 | Collection Name | Collection | The collection containing the annotations. |
|---|-----------------|------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The datums/feature checks were updated successfully. |
| FAILURE | The collection was not found. |

Remarks

None.

Set Feature Check Reporting Frame

Sets the reporting frame for a specific feature check.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Feature Check | The name of the intended feature check. |
| 1 | Collection Object Name | Reporting Frame | The name of the reporting frame to set. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame for the feature check was set successfully. |
| FAILURE | The feature check or reporting frame were not found. |

Remarks

None.

Get Feature Check Reporting Frame

Gets the reporting frame for a specific feature check.

Input Arguments

| | | | |
|---|------------------------|---------------|---|
| 0 | Collection Object Name | Feature Check | The name of the intended feature check. |
|---|------------------------|---------------|---|

Return Arguments

| | | | |
|---|------------------------|-----------------|-----------------------------------|
| 1 | Collection Object Name | Reporting Frame | The name of the reporting frame.. |
|---|------------------------|-----------------|-----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame for the feature check was retrieved successfully. |
| FAILURE | The feature check was not found. |

Remarks

None.

Relationship Operations

Generate Geometry Relationship Summary

Builds a geometry relationship summary table from the selected relationships.

Input Arguments

| | | | |
|---|-----------------------|--------------------|--|
| 0 | Relationship Ref List | Relationship List | The list of relationships to include in the table. |
| 1 | String | Summary Table Name | Name of to use for the summary table. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | This command always succeeds. |
| FAILURE | Specified relationships could not be found. |

Remarks

None.

Get Number of Relationships in Relationship Ref List

Returns the number of relationships in a relationship reference list.

Input Arguments

| | | | |
|---|-----------------------|-------------------|-------------------------------------|
| 0 | Relationship Ref List | Relationship List | The list of relationships to count. |
|---|-----------------------|-------------------|-------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|--|
| 1 | Integer | Total Count | The number of relationships in the list. |
|---|---------|-------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get i-th Relationship From Relationship Ref List

Returns the name of a relationship at a specified index in a relationship reference list.

Input Arguments

| | | | |
|---|-----------------------|------------------------|--------------------------------|
| 0 | Relationship Ref List | Relationship Name List | The list of relationships. |
| 1 | Integer | Relationship Index | The index of the relationship. |

Return Arguments

| | | | |
|---|------------------------|----------------|--|
| 2 | Collection Object Name | Resultant Item | The name of the relationship at the specified index. |
|---|------------------------|----------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first relationship in a ref list is at index 0.

Get i-th Relationship From Relationship Ref List (Iterator)

Returns the name of a relationship at a specified index in a relationship reference list. This command is an iterator, which means it is a shortcut to a traditional loop. Iterators proceed sequentially, one-by-one, from a starting index, until the end of the list.

Input Arguments

| | | | |
|---|-----------------------|-----------------------------|--|
| 0 | Relationship Ref List | Reference List | The list of relationships to consider. |
| 1 | Integer | Relationship Index | The starting index of the relationship to consider. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to after iterating through the last relationship in the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|---|
| 3 | String | Collection | The collection containing the relationship at the specified index. |
| 4 | String | Relationship | The name of the relationship at the specified index. |
| 5 | Collection Object Name | Resultant Item | The relationship at the specified index, as a collection object name data type. |

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship name was determined successfully. |
| FAILURE | The index supplied was beyond the bounds of the list. |

Remarks

Ref list indices are zero based, so the first item in a ref list is at index 0.

Sort Relationship Ref List

Returns a list of relationships sorted based upon the selection criteria.

Input Arguments

| | | | |
|---|-----------------------|-----------------------|--|
| 0 | Relationship Ref List | Relationship Ref List | The list of relationships to sort. |
| 1 | Boolean | Case Sensitive? | Use case sensitive sorting |
| 2 | Boolean | Ascending Order | True indicates ascending order, false returns in descending order. |

Return Arguments

| | | | |
|---|-----------------------|------------------------------|-----------------------------------|
| 3 | Relationship Ref List | Sorted Relationship Ref List | The sorted list of relationships. |
|---|-----------------------|------------------------------|-----------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship list was sorted successfully. |
| FAILURE | This command always succeeds. |

Remarks

This command does not change the order of relationships in the tree. It simply reorganizes the reference list for more logical processing.

Get Relationship Reporting Frame

Gets the reporting frame for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|------------------------|-----------------|---|
| 1 | Collection Object Name | Reporting Frame | The name of the reporting frame for the relationship. |
|---|------------------------|-----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame was retrieved successfully. |
| FAILURE | The specified relationship or reporting frame could not be found. |

Remarks

None.

Show/Hide Relationship Report

Shows or hides the relationship report for a specified collection.

Input Arguments

| 0 | Collection Name | Collection Name | The name of the collection. |
|---|-----------------|--------------------------|---|
| 1 | Boolean | Show Relationship Report | Indicates whether the relationship should be shown or hidden. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The report was shown/hidden successfully. |
| FAILURE | The collection could not be found. |

Remarks

None.

Show/Hide Relationship Watch

Shows or hides a watch window for the specified relationship.

Input Arguments

| | | | |
|---|------------------------|--------------------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
| 1 | Boolean | Show Relationship Watch | Show the relationship watch window. |
| 2 | Collection Object Name | Relationship Watch Window Properties | The name of the watch window template to be used. |
| 3 | Integer | Window Top Left X Position | Enter the value of the top left X position of the watch window. |
| 4 | Integer | Window Top Left Y Position | Enter the value of the top left Y position of the watch window. |
| 5 | Integer | Window Width | Enter the value of watch window width. |
| 6 | Integer | Window Height | Enter the value of the watch window height. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The watch window was shown/hidden successfully. |
| FAILURE | The relationship could not be found. |

Remarks

None.

Relationship Watch Window Template

Creates a template for a relationship watch window.

Input Arguments

| | | | |
|----|------------------------|-------------------------------|--|
| 0 | Collection Object Name | Watch Window Template Name | The name for the new watch window template. |
| 1 | Integer | Linear Precision | The decimal precision of the objects distance. |
| 2 | Integer | Angular Precision | The decimal precision of the objects angle. |
| 3 | Font Type | Font | The font style of the text |
| 4 | Color | Text Color | The color of the text in the watch window |
| 5 | Color | Background Color | Color of background in watch window. |
| 6 | Color | Highlight Color | Color of highlight in watch window. |
| 7 | Boolean | Show Deviation X (Rx)? | Display X deviation. |
| 8 | Boolean | Show Deviation Y (Ry)? | Display Y deviation. |
| 9 | Boolean | Show Deviation Z (Rz)? | Display Z deviation. |
| 10 | Boolean | Show Deviation Mag? | Display magnitude of the deviation. |
| 11 | UDP Settings | UDP Network Transmit Settings | Set transmission speed. |
| 12 | Boolean | Transparent Background? | The background will become transparent. |
| 13 | Boolean | Hide Units? | The units will not be shown. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Point to Point Relationship

Creates a point to point relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Point Name | First Point Name | The name for the first point in the relationship. |
| 2 | Point Name | Second Point Name | The name for the second point in the relationship. |
| 3 | Vector Tolerance | Tolerance | The component-based tolerances for the relationship. |
| 4 | Vector Constraint | Constraint | Component-based constraints for the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or both points could not be found. |

Remarks

None.

Make Frame to Frame Relationship

Creates a point to point relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | First Frame Name | The name for the first frame in the relationship. |
| 2 | Collection Object Name | Second Frame Name | The name for the second frame in the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or both frames could not be found. |

Remarks

None.

Make Points to Objects Relationship

Creates a points to objects relationship.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Point Name Ref List | Points in Relationship | The list of points to include in the relationship. |
| 2 | Collection Object Name Ref List | Objects in Relationship | The list of objects to include in the relationship. |
| 3 | Projection Options | Projection Options | The projection options for the relationship. |
| 4 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or both points or objects could not be found. |

Remarks

None.

Make Points to Points Relationship

Creates a points to points relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
|---|------------------------|-----------------------------|---|
| 1 | Point Name Ref List | Nominal Points | The list of nominal or reference points to include in the relationship. |
| 2 | Point Name Ref List | Measured Points | The list of measured points to include in the relationship. |
| 3 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |
| 4 | Vector Tolerance | Tolerance | Optional vector tolerance to apply |
| 5 | Vector Constraint | Constraint | Optional vector constraint to apply |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or both points or objects could not be found. |

Remarks

None.

Make Groups to Objects Relationship

Creates a groups to objects relationship.

Input Arguments

| | | | |
|---|---------------------------------|------------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name Ref List | Point Groups in Relationship | The list of point groups to include in the relationship. |
| 2 | Collection Object Name Ref List | Objects in Relationship | The list of objects to include in the relationship. |
| 3 | Projection Options | Projection Options | The projection options for the relationship. |
| 4 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or more objects or groups could not be found. |

Remarks

None.

Make Object to Object Direction Relationship

Creates an object to object direction relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | First Object in Relationship | The first object to include in the relationship. |
| 2 | Collection Object Name | Second Object in Relationship | The second object to include in the relationship. |
| 3 | Double | Nominal Angle | The nominal angle to set in the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The first or second object could not be found. |

Remarks

None.

Get Geom Relationship Point List

Gets the point list of a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|-------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
|---|------------------------|-------------------|-------------------------------|

Return Arguments

| | | | |
|---|----------------------|----------------|---|
| 1 | Point Name Ref LList | All Points | Complete list of all points. |
| 2 | Point Name Ref LList | Used Points | List of points used in the relationship. |
| 3 | Point Name Ref LList | Ignored Points | List of the ignored points in the relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The point list was retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Make Point Clouds to Objects Relationship

Creates a point clouds to objects relationship.

Input Arguments

| | | | |
|---|---------------------------------|------------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name Ref List | Point Clouds in Relationship | The point clouds to include in the relationship. |
| 2 | Collection Object Name Ref List | Objects in Relationship | The objects to include in the relationship. |
| 3 | Projection Options | Projection Options | The projection options for the resulting relationship. |
| 4 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or more clouds or objects could not be found. |

Remarks

None.

Enable All Cloud Cross Sections

Displays all cloud cross sections.

Input Arguments

| | | | |
|---|------------------------|--------------------------|--------------------------------------|
| 0 | Collection Object Name | Cross Section Cloud Name | The name of the cross section cloud. |
|---|------------------------|--------------------------|--------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | All cross section clouds were displayed successfully. |
| FAILURE | The specified cross section cloud name was not found. |

Remarks

None.

Enable/Disable Cloud Cross Sections

Toggles display visibility of specified cloud cross sections.

Input Arguments

| | | | |
|---|------------------------|----------------------------------|--|
| 0 | Collection Object Name | Cross Section Cloud Name | The name of the cross section cloud. |
| 1 | Integer | Cross Section ID | The identifying number of the desired cross section. |
| 2 | Boolean | Enable (True) / Disable (False)? | Specifies whether to show or hide cross section. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Cloud cross sections were toggled successfully. |
| FAILURE | The specified cross section cloud name and/or ID was not found. |

Remarks

None.

Enable Single Cloud Cross Section

Displays visibility of specified cloud cross sections.

Input Arguments

| | | | |
|---|------------------------|--------------------------|--|
| 0 | Collection Object Name | Cross Section Cloud Name | The name of the cross section cloud. |
| 1 | Integer | Cross Section ID | The identifying number of the desired cross section. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Single cloud cross section was displayed successfully. |
| FAILURE | The specified cross section cloud name and/or ID was not found. |

Remarks

None.

Get Number of Cross Sections in Cross Section Cloud

Toggles display visibility of specified cloud cross sections.

Input Arguments

| | | | |
|---|------------------------|--------------------------|--------------------------------------|
| 0 | Collection Object Name | Cross Section Cloud Name | The name of the cross section cloud. |
|---|------------------------|--------------------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|---------------------|--|
| 1 | Integer | Cross Section Count | The number of cross sections within the cross section cloud. |
|---|---------|---------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | Number of cross sections retrieved successfully. |
| FAILURE | The specified cross section cloud name was not found. |

Remarks

None.

Make Group to Group Relationship

Creates a group to group relationship.

Input Arguments

| | | | |
|---|------------------------|-----------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | First Group Name | The first point group to include in the relationship. |
| 2 | Collection Object Name | Second Group Name | The second point group to include in the relationship. |
| 3 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |
| 4 | Vector Tolerance | Tolerance | The tolerance for the relationship. |
| 5 | Vector Constraint | Constraint | The constraints for the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | One or both point groups could not be found. |

Remarks

As is the case when manually creating a group to group relationship, corresponding points must have like names in order to be included in the relationship.

Make Group to Nominal Group Relationship

Creates a group to nominal group relationship.

Input Arguments

| | | | |
|----|------------------------|------------------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | First Group Name | The first point group to include in the relationship. |
| 2 | Collection Object Name | Second Group Name | The second point group to include in the relationship. |
| 3 | Boolean | Auto Update a Vector Group? | Indicates whether an auto-updating vector group should be created. |
| 4 | Boolean | Use Closest Point? | Enable Closest Point renaming during trapping? |
| 5 | Boolean | Display Closest Point Watch Window | Display watch window when trapping is initiated? |
| 6 | Boolean | User View Zooming With Proximity? | Use View Zooming with Proximity during trapping? |
| 7 | Boolean | Ignore Points Beyond Threshold? | Exclude point measurements that fall beyond the proximity threshold when trapping |
| 8 | Double | Proximity Threshold? | Distance to nominal used for point exclusion |
| 9 | Vector Tolerance | Tolerance | The tolerance for the relationship. |
| 10 | Vector Constraint | Constraint | The constraints for the relationship. |
| 11 | Double | Fit Weight | Fit Weight used in relationship optimizations |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | Nominal groups could not be found. |

Remarks

As is the case when manually creating a group to group relationship, corresponding points must have like names in order to be included in the relationship.

Make Average Point Relationship

Creates a relationship using an average point.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Point Name Ref List | Points in Relationship | The list of points to include in the relationship. |
| 2 | Point Name | Average Point Name (Optional) | The name of the average point. |
| 3 | Point Name | Nominal Point Name (Optional) | The name of the nominal pont. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The list of points could not be found. |

Remarks

None.

Make Geometry Fit Only Relationship

Creates a geometry comparison relationship of type “Fit Only”.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|------------------------------------|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name Ref List | Point Groups to Fit | The list of point groups to fit. |
| 2 | Geometry Type | Geometry Type | The typ of geometry to fit. |
| 3 | Collection Object Name | Resulting Object Name (Optional) | The name of the resulting object. |
| 4 | String | Fit Profile Name (Optional) | Name of the fit profile if used. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The relationship was created successfully. |
| PARTIAL SUCCESS | One or more point groups (but not all) were not found, but the relationship was still created. |
| FAILURE | The specified relationship already exists, or none of the point groups could be found. |

Remarks

As of 2022.3 the Geometry Type argument [A2] can reference a string. It will accept [Line, Plane, Circle, Slot, Ellipse, Cylinder, Sphere, Cone, and Paraboloid] *Notes - first letter must be capitalized.

Make Geometry Fit and Compare to Nominal Relationship

Creates a geometry comparison relationship of type “Fit and Compare to Nominal”.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|------------------------------------|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | Nominal Geometry | Nominal geometry to be used. |
| 2 | Collection Object Name Ref List | Point Groups to Fit | The list of point groups to fit. |
| 3 | Collection Object Name | Resulting Object Name (Optional) | The name of the resulting object. |
| 4 | String | Fit Profile Name (Optional) | Name of the fit profile if used. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The relationship was created successfully. |
| PARTIAL SUCCESS | One or more point groups (but not all) were not found, but the relationship was still created. |
| FAILURE | The specified relationship already exists, or the nominal geometry or point groups could not be found. |

Remarks

None.

Make Geometry Compare Only Relationship

Creates a geometry comparison relationship of type “Compare Only”.

Input Arguments

| | | | |
|---|------------------------|-------------------|------------------------------------|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Collection Object Name | Nominal Geometry | The geometry to compare to. |
| 2 | Collection Object Name | Measured Geometry | The geometry to compare. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the nominal or measured geometry could not be found. |

Remarks

The nominal geometry and measured geometry must be of the same type. (For example, if one is a plane, the other must be a plane).

Make Dynamic Point Relationship

Creates a dynamic point relationship using the selected construction mode and reference geometry.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Dynamic Point Mode | Construction Mode | Selected method for construction |
| 2 | Collection Object Name | First Reference Geometry | First reference used by the construction method |
| 3 | Collection Object Name | Second Reference Geometry | Second reference object used for construction |
| 4 | Collection Object Name | Third Reference Geometry | Third reference object used for construction |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference geometry could not be found. |

Remarks

Dynamic Point construction currently supports:

- Intersection of a line and a plane
- Intersection of a Cylinder and Plane, returning the axis intersection point
- Intersection of a Cone and Plane, returning the axis intersection point
- Intersection of 3 planes
- Mid-Point of Perpendicular to 2 lines (intersection of 2 lines allowing for imperfection)

Make Dynamic Line Relationship

Creates a dynamic line relationship using the selected construction mode and reference geometry.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Dynamic Line Mode | Construction Mode | Selected method for construction |
| 2 | Collection Object Name | First Reference Geometry | First reference used by the construction method |
| 3 | Collection Object Name | Second Reference Geometry | Second reference object used for construction |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference geometry could not be found. |

Remarks

Dynamic Line construction currently supports:

- Cone Axis
- Cylinder Axis
- Intersection of Two Planes
- Bisect Two Lines
- Slot Center line Along Length

Make Dynamic Plane Relationship

Creates a dynamic plane relationship using the selected construction mode and reference geometry.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Dynamic Plane Mode | Construction Mode | Selected method for construction |
| 2 | Collection Object Name | First Reference Geometry | First reference used by the construction method |
| 3 | Collection Object Name | Second Reference Geometry | Second reference object used for construction |
| 4 | Double | Offset Plane Offset | The distance to offset the dynamic plane from the reference |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference geometry could not be found. |

Remarks

Dynamic plane construction currently supports:

- Two Cones Intersection - Hold Normal to Best-Fit Plane
- Two Cones Intersection - Hold Normal to First Cone Axis
- Two Cones Intersection - Hold Normal to Second Cone Axis
- Cone and Cylinder Intersection - Hold Normal to Best-Fit Plane
- Cone and Cylinder Intersection - Hold Normal to Cone Axis
- Cone and Cylinder Intersection - Hold Normal to Cylinder Axis
- Offset Plane from Plane

Make Dynamic Circle Relationship

Creates a dynamic circle relationship using the selected construction mode and reference geometry.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Dynamic Circle Mode | Construction Mode | Selected method for construction |
| 2 | Collection Object Name | First Reference Geometry | First reference used by the construction method |
| 3 | Collection Object Name | Second Reference Geometry | Second reference object used for construction |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference geometry could not be found. |

Remarks

Dynamic circle construction currently supports:

- Cylinder and Plane Intersection - Hold Plane Normal
- Cylinder and Plane Intersection - Hold Cylinder Normal
- Cone and Plane Intersection - Hold Plane Normal
- Cone and Plane Intersection - Hold Cone Normal
- Cone and Cylinder Intersection
- Sphere and Plane Intersection
- Intersection of two Cones

Make Dynamic Ellipse Relationship

Creates a dynamic ellipse relationship using the selected construction mode and reference geometry.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name for the new relationship. |
| 1 | Dynamic Ellipse Mode | Construction Mode | Selected method for construction |
| 2 | Collection Object Name | First Reference Geometry | First reference used by the construction method |
| 3 | Collection Object Name | Second Reference Geometry | Second reference object used for construction |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference geometry could not be found. |

Remarks

Dynamic circle construction currently supports:

- Cylinder and Plane Intersection
- Cone and Plane Intersection

Make Vector Group To Vector Group Relationship

Creates a relationship that computes the difference between two reference vector groups. This can be used to compare sequential surface measurements or for shim and gap analysis.

Input Arguments

| | | | |
|---|------------------------|------------------------------------|--|
| 0 | Collection Object Name | New VG to VG Relationship | The name for the new relationship. |
| 1 | Collection Object Name | Reference Vector Group | The geometry to compare to. |
| 2 | Collection Object Name | Corresponding Vector Group | The geometry to compare. |
| 3 | Boolean | Set Opposing Vector Group Polarity | Set True for "Opposing" direction and False for "Same Direction" analysis. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship was created successfully. |
| FAILURE | The specified relationship already exists, or the reference or corresponding vector group could not be found. |

Remarks

For further configuration see the following commands:

- "Set Vector Group To Vector Group Cylindrical Zone"
- "Set Vector Group To Vector Group Fit Weights"
- "Set Vector Group To Vector Group Fit Gradient Factor"
- "Set Vector Group To Vector Group Relative Polarity"

Set Vector Group To Vector Group Cylindrical Zone

Used to update the proximity setting properties of an existing Vector Group to Vector Group Relationship.

Input Arguments

| 0 | Collection Object Name | New VG to VG Relationship | The name for the new relationship. |
|---|------------------------|---------------------------|--|
| 1 | Double | Radial Offset | Radial offset to consider |
| 2 | Double | Minimum Axial Offset | Minimum or negative offset to consider normal to the surface |
| 3 | Double | Maximum Axial Offset | Maximum or positive offset to consider normal to the surface |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was updated successfully. |
| FAILURE | Relationship could not be found. |

Remarks

None.

Set Vector Group To Vector Group Fit Weights

Used to update the Fit setting properties of an existing Vector Group to Vector Group Relationship.

Input Arguments

| | | | |
|---|------------------------|---------------------------|--|
| 0 | Collection Object Name | New VG to VG Relationship | The name for the new relationship. |
| 1 | Double | Minimum Gap | Minimum Gap distance threshold |
| 2 | Double | Minimum Gap Fit Weight | Minimum gap fit weight to apply |
| 3 | Double | Maximum Gap | Maximum Gap distance threshold |
| 1 | Double | Maximum Gap Fit Weight | Maximum gap fit weight to apply |
| 2 | Double | Nominal Gap | Nominal or optimum Gap |
| 3 | Double | Nominal Gap Fit Weight | Nominal or optimum gap fit weight to apply |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was updated successfully. |
| FAILURE | Relationship could not be found. |

Remarks

None.

Set Vector Group To Vector Group Fit Gradient Factor

Used to update the Fit setting properties of an existing Vector Group to Vector Group Relationship.

Input Arguments

| | | | |
|---|------------------------|---------------------------|------------------------------------|
| 0 | Collection Object Name | New VG to VG Relationship | The name for the new relationship. |
| 1 | Double | Fit Gradient Factor | Fit gradient factor to apply |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was updated successfully. |
| FAILURE | Relationship could not be found. |

Remarks

The Fit Gradient Factor is used to determine the rate of change between the fit weights used at the minimum/maximum ranges and the nominal fit weight. Larger values correspond to a sharper transition. For more information refer to the Relationships chapter of the users manual.

Set Vector Group To Vector Group Fit Weights

Used to update the Fit setting properties of an existing Vector Group to Vector Group Relationship.

Input Arguments

| 0 | Collection Object Name | New VG to VG Relationship | The name for the new relationship. |
|---|------------------------|------------------------------------|--|
| 1 | Boolean | Set Opposing Vector Group Polarity | Set True for "Opposing" direction and False for "Same Direction" analysis. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was updated successfully. |
| FAILURE | Relationship could not be found. |

Remarks

Opposing Direction would be used for applications such as a shim and gap process where mating surfaces have been measured. Same Direction would be applicable for two sets of measurements on the same surface where the relative change between the measurements is desired.

Delete Relationship

Deletes a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to delete. |
|---|------------------------|-------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was deleted successfully. |
| FAILURE | The specified relationship was not found. |

Remarks

None.

Set Optimization Options

Edits the job's optimization parameters. This edits the optimization settings for optimization such as Do Relationship Fit and Move Collections by Minimizing Relationships.

Input Arguments

| | | | |
|---|---------|-----------------------------------|--|
| 0 | Integer | Max Number of Step Size Reduction | This number (5-10) determines the number of optimization step size reductions used in ing for a stable solution. |
|---|---------|-----------------------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | This command sets a global variable and always succeeds |
|---------|---|

Remarks

The Max Number of Step Size Reductions argument controls how fine grained the optimization process is. A very small shift (large number of step reductions) can potentially settle on a better solution but will require more time to complete. A value of 5 is the historic SA default and a value of 10 will use the smallest steps in finding a solution. This will increase the time but may or may not produce a better result.

Set Optimization Perturbation Parameters

Edits the job's perturbation parameters within the optimization parameters section of the users options. This edits the optimization settings for Length, Angular and Damping.

Input Arguments

| | | | |
|---|--------|----------------------|----------------------------------|
| 0 | Double | Length Perturbation | |
| 1 | Double | Angular Perturbation | |
| 2 | Double | Damping | Enter a value between 0.1 and 3. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | This command sets a global variable and always succeeds |
|---------|---|

Remarks

None.

Do Relationship Fit

Performs a relationship fit (minimization).

Input Arguments

| | | | |
|---|-----------------------------------|-------------------------------------|---|
| 0 | Collection Name | Collection Containing Relationships | The collection containing the relationships to minimize. |
| 1 | Collection Object Name Ref List | Objects to Move | The list of objects to transform. |
| 2 | Collection Instrument ID Ref List | Instruments to Move | The list of instruments to transform. |
| 3 | Boolean | Perform 'Direct' | Indicates whether standard optimization or a Direct optimization should be performed. |
| 4 | Fit Degree-Of-Freedom Options | Motion to allow | The allowable degrees of freedom for the fit. |
| 5 | Boolean | Use Fit Dialog | Indicates whether the fit dialog should be displayed to the user during execution. |

Return Arguments

| | | | |
|---|-----------|---------------------|---|
| 6 | Transform | Resulting Transform | The transform generated by the relationship minimization. |
|---|-----------|---------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was deleted successfully. |
| FAILURE | The specified relationship was not found. |

Remarks

The actual transformation is applied to the moving objects and instruments, but the transform is returned in order to perform additional transforms.

Move Collections by Minimizing Relationships

Performs a relationship fit (minimization) moving the specified collections.

Input Arguments

| | | | |
|---|-------------------------------|---------------------------|---|
| 0 | String Ref List | Collections To Move | A list of the collections to move . |
| 1 | Relationship Ref List | Relationships to Minimize | List of relationships to use . |
| 2 | Boolean | Perform "Direct" | Controls if direct is used. |
| 3 | Fit Degree-Of-Freedom Options | Motion to allow | Sets the DoF to use in the optimization. |
| 4 | Boolean | Use Fit Dialog | Controls if the fit dialog is displayed. |
| 5 | Double | Convergence Threshold | The convergence threshold defines when the optimization is considered complete. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship optimization was completed successfully. |
| FAILURE | The specified relationship was not found. |

Remarks

The relationship list sets the use status for the relationships in the dialog if displayed. Currently degrees of freedom set here in (A3) set the same condition for all specified collections.

The *Convergence Threshold* is used to determine when to stop the optimization process. If an optimization iteration "n" fails to improve the square of the objective function (obj) by the square of convergence threshold amount (CT), then the optimization termination criteria is satisfied. So termination occurs when

$$obj_{n-1}^2 - obj_n^2 < CT^2$$

Some optimizations may satisfy their relationships' tolerances many iterations before the objective function improvement is sufficiently small enough compared to the default convergence threshold. In these cases, the convergence threshold can be increased to reduce the time it takes to optimize. This threshold should be evaluated against the precision required in moving the collections.

Get General Relationship Statistics

Obtains the RMS, max deviation, and signed maximum and minimum deviations for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to examine. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|---------|-----------------------|--|
| 1 | Double | Max Deviation | The maximum deviation value of the relationship. |
| 2 | Double | RMS | The RMS error of the relationship. |
| 3 | Boolean | Has Signed Deviation? | Indicates whether the deviations are signed. |
| 4 | Double | Signed Max Deviation | The signed maximum deviation. |
| 5 | Double | Signed Min Deviation | The signed minimum deviation. |

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship values were obtained successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

Max = the highest Z dev

RMS = RMS of the individual X, Y, Z components

All the points contribute three values that are submitted to a composite 1D vector of length $3 * N$ (N = number of points). The “AbsMax” and “RMS” are developed from this 1D data set.

The “RMS” results as calculated provides the objective function output for relationship minimization. This is actually a very convenient way to manage degree of freedom constraints for minimization operations -- unfortunately, the output it produces is inconsistent with the sort you would expect from a vector group style representation.

Get Points to Objects Relationship Statistics

Obtains relevant statistics from a points to objects relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to examine. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|---------|------------------------------|--|
| 1 | Double | Absolute Max Deviation | The absolute maximum deviation of the relationship, returned as a positive number. |
| 2 | Double | Max Deviation | The maximum deviation (highest) value of the relationship. |
| 3 | Double | Min Deviation | The smallest deviation (most negative) value of the relationship. |
| 4 | Double | RMS | The RMS error of the relationship. |
| 5 | Integer | # of Candidate Points | The raw number of points in the relationship. |
| 6 | Integer | # of Points Sampled | The number of points in the relationship (after subsampling). |
| 7 | Integer | # of Points Rejected | The number of points rejected from a relationship based on Outlier Rejection. |
| 8 | Integer | # of Points Used | The number of points used for the calculation of the relationship. |
| 9 | Integer | # of Points Out of Tolerance | The number of points in the relationship that exceed the specified tolerance. |

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship values were obtained successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Start/Stop Relationship Trapping

Starts or stops relationship trapping for measurements from a live instrument.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship in question. |
| 1 | Collection Instrument ID | Instrument ID | The live instrument to start or stop trapping. |
| 2 | Boolean | Start Trapping (FALSE = Stop) | Indicates whether trapping should be started or stopped. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Trapping began successfully. |
| FAILURE | The specified relationship could not be found, or the Collection Instrument ID was invalid. |

Remarks

None.

Get Point to Point Relationship Statistics

Obtains relevant statistics from a points to points relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to examine. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|------------------------|-----------------|----------------------------|
| 1 | Double | Delta X | Displacement in X |
| 2 | Double | Delta Y | Displacement in Y |
| 3 | Double | Delta Z | Displacement in Z |
| 4 | Double | Delta Magnitude | Direct linear displacement |
| 5 | Collection Object Name | Reference Frame | Reporting Frame. |

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship values were obtained successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Associated Data

Associates data with a specified relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
|---|---------------------------------|-------------------------|---|
| 1 | Point Name Ref List | Individual Points | The individual points to associate with the relationship. |
| 2 | Collection Object Name Ref List | Point Groups | A list of point groups to associate with the relationship. |
| 3 | Collection Object Name Ref List | Point Clouds | A list of point clouds to associate with the relationship. |
| 4 | Collection Object Name Ref List | Objects | A list of objects to associate with the relationship. |
| 5 | Boolean | Ignore Empty Arguments? | If TRUE, empty arguments to the command will not trigger a failure. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The data was associated successfully. |
| FAILURE | The specified relationship, points, or objects could not be found. |

Remarks

None.

Get Relationship Associated Data

Retrieves a relationship's type and associated data.

Input Arguments

| | | | |
|---|------------------------|-------------------|-------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
| 1 | String | Relationship Type | The type of the relationship. |

Return Arguments

| | | | |
|---|---------------------------------|-------------------|--|
| 2 | Point Name Ref List | Individual Points | The individual points associated with the relationship. |
| 3 | Collection Object Name Ref List | Point Groups | A list of point groups associated with the relationship. |
| 4 | Collection Object Name Ref List | Point Clouds | A list of point clouds associated with the relationship. |
| 5 | Collection Object Name Ref List | Objects | A list of objects associated with the relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The information was retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Points to Points Relationship Associated Data

Sets the associated data

Input Arguments

| | | | |
|---|------------------------|------------------------|--------------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
| 1 | Point Name Ref List | Nominal Points | List of nominal reference points |
| 2 | Point Name Ref List | Actual Points | List of actual points for comparison |
| 3 | Boolean | Ignore Empty Arguments | True allows empty arguments |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The associated data was set successfully |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Points to Points Relationship Associated Data

Retrieves associated data from a Points to Points relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|-------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship. |
|---|------------------------|-------------------|-------------------------------|

Return Arguments

| | | | |
|---|---------------------|----------------|--|
| 2 | Point Name Ref List | Nominal Points | The nominal points associated with the relationship. |
| 3 | Point Name Ref List | Actual Points | The actual points associated with the relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The information was retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Auto Filter Clouds to Nominal Geometry 3D

This function filters cloud data based upon proximity to the nominal features, building new clouds, and associating those clouds with the corresponding relationships.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|--|
| 0 | Relationship Ref List | Auto Filter Target Relationships | The compare to nominal relationships used for the filter process. |
| 1 | Collection object Name Ref List | Clouds | The point clouds considered as part of the filter process. |
| 2 | Cloud Thinning Options | Cloud Thinning Settings | The cloud thinning parameters to use |
| 3 | Filter Proximity Settings 3D | Filter Proximity Settings 3D | The proximity filter settings to be used in selecting points and building the associated clouds. |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The filter process completed successfully |
| FAILURE | The relationships or the clouds specified could not be found |

Remarks

For more information look in the Clouds chapter of the users manual under Feature Extraction from Point Clouds.

Auto Filter Clouds to Nominal Geometry 2D

This function filters cloud data based upon proximity to the nominal features, building new clouds, and associating those clouds with the corresponding relationships. The filter process only considers data in 2D, looking at the holes within the planar cloud data.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------------|--|
| 0 | Relationship Ref List | Auto Filter Target Relationships | The compare to nominal relationships used for the filter process. |
| 1 | Collection object Name Ref List | Clouds | The point clouds considered as part of the filter process. |
| 2 | Cloud Thinning Options | Cloud Thinning Settings | The cloud thinning parameters to use |
| 3 | Filter Proximity Settings 3D | Filter Proximity Settings 2D | The proximity filter settings to be used in selecting points and building the associated clouds. |
| 4 | Double | Geometry Extraction Tolerance | The tolerance value to use |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The filter process completed successfully |
| FAILURE | The relationships or the clouds specified could not be found |

Remarks

For more information look in the Clouds chapter of the users manual under Feature Extraction from Point Clouds.

Auto Filter Points to Nominal Geometry 3D

This function filters point data based upon proximity to the nominal features and associating those points with the corresponding relationships. It works for Geometry Relationships and Points to Surface Face relationships.

Input Arguments

| | | | |
|---|------------------------------|----------------------------------|---|
| 0 | Relationship Ref List | Auto Filter Target Relationships | The compare to nominal relationships used for the filter process. |
| 1 | Point Name Ref List | Points | The points to consider for the filter process. |
| 2 | Filter Proximity Settings 3D | Filter Proximity Settings 3D | The proximity filter settings to be used in selecting points and building the associations. |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The filter process completed successfully |
| FAILURE | The relationships or the points specified could not be found |

Remarks

For more information look in the Clouds chapter of the users manual under Feature Extraction from Point Clouds, the interface is described here and the filter section also applies to the point filter process.

Auto Filter Points/Groups/Clouds to Surface Faces

Filters a set of individual points, groups, and/or clouds to a set of surface faces, just like the manual “Auto Filter to Faces” command.

Input Arguments

| | | | |
|----|---------------------------------|-------------------------------------|--|
| 0 | Point Name Ref List | Points | A list of individual points to filter. |
| 1 | Collection Object Name Ref List | Groups | A list of groups to filter. |
| 2 | Collection Object Name Ref List | Clouds | A list of clouds to filter. |
| 3 | Double | Surface Offset | A distance from the surface beyond which points are excluded from the filter. |
| 4 | Double | Edge Offset | A distance from surface edges inside of which points are excluded from the filter. |
| 5 | Offset Direction Type | Offset Direction | Specifies the side(s) of the surfaces to include in the filter. |
| 6 | Boolean | Enforce Max Pts per Face in Output? | Indicates whether an upper limit should be placed on the number of points allowed to be associated with a given surface. |
| 7 | Integer | Max Pts per Face | The maximum number of points to include on a single face (applies if argument 6 is true). |
| 8 | Collection Object Name Ref List | Surfaces | A list of surfaces to include in the filter. |
| 9 | Cloud Thinning Options | Cloud Thinning Settings | Settings for cloud thinning, if clouds are provided as the source points. |
| 10 | String | Output Cloud Base Name | The Name of the output cloud |
| 11 | Boolean | Use Face IDs for Suffix | Whether to use the Face IDs for the suffix of the output cloud names. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The points were filtered successfully. |
| FAILURE | The points or surfaces were not found. |

Remarks

Auto Filter Points/Groups/Clouds to Surface Faces can take some time to process but should be used when edges are of concern.

Filter Clouds to Surfaces is an alternative that offers the ability to use an asymmetric distance such as +.1 to +.5 and is much faster, but it ignores edges entirely. This means points beyond an edge are also picked up as long as they are within the specified proximity.

Create Points to Objects Map

Builds a map of objects within a proximity distance to points that can be used as a lookup table for accelerated processing.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------|-------------------------------|
| 0 | Point Name Ref List | Points | Points to process |
| 1 | Collection Object Name Ref List | Groups | Groups to process |
| 2 | Collection Object Name Ref List | Objects | Objects to consider |
| 3 | Double | Proximity Tolerance | Distance threshold |
| 4 | String | Points to Objects Map Name | Name of the newly created map |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The analysis completed successfully |
| FAILURE | Some of the points or objects could not be found. |

Remarks

None.

Get Objects From Points to Objects Map

Returns a list of objects from the specified *Create Points to Objects Map* return. Builds a map of objects within a proximity distance to points that can be used as a lookup table for accelerated processing.

Input Arguments

| | | | |
|---|------------|----------------------------|--|
| 0 | String | Points to Objects Map Name | Name of the Map to use. |
| 1 | Point Name | Point | Point to use as a reference in the Map |

Return Arguments

| | | | |
|---|---------------------------------|---------|--|
| 2 | Collection Object Name Ref List | Objects | Objects within proximity to the reference point. |
|---|---------------------------------|---------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The analysis completed successfully |
| FAILURE | Specified Map or point could not be found. |

Remarks

None.

Compute Geometry Relationship Uncertainties

This command can be used to trigger an uncertainty analysis for a GR-Feature. The point uncertainties need to be computed before the feature uncertainty can be computed (see Create Point Uncertainty Fields).

Input Arguments

| 0 | Collection Object Name | Relationship Name | Relationship to adjust |
|---|------------------------|-------------------|--|
| 1 | Boolean | Display Results | True displays the analysis results dialog. |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The uncertainty analysis completed successfully |
| FAILURE | The relationships could not be found or the uncertainty couldn't be computed |

Remarks

The uncertainty for circles with projection planes is not currently supported. GR-Features are set to a dormant status when the uncertainty is computed to prevent further recomputation.

Pipe Relationships

Make Pipe Fitting Relationship

Builds a new Pipe Fitting Relationship for use in pipe cutting operations.

Input Arguments

| | | | |
|---|------------------------|----------------------|-------------------------------------|
| 0 | Collection Object Name | Relationship Name | Relationship Name |
| 1 | Collection Object Name | Pipe 1 - Object Name | Name of the first reference object |
| 2 | Collection Object Name | Pipe 2 - Object Name | Name of the second reference object |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was built successfully |
| FAILURE | The name is invalid or one of the objects could not be found |

Remarks

For more information on Pipe relationships refer to the section within the Relationships chapter of the SA Users Manual.

Set Pipe Relationship Weights

This command is used to configure the weighting factors used in calculating the cut operation in a pipe relationship and its weighting as part of an optimization.

Input Arguments

| 0 | Collection Object Name | Relationship Name | Name of the relationship to edit. |
|---|------------------------|--------------------------|---|
| 1 | Double | Overall Weight | The weight as a whole, which determines the influence in the overall optimization scheme. |
| 2 | Double | Axis Offset | The weight of minimizing the mutual perpendicular midpoint between the two pipe segments (i.e. how "coaxial" the two pipe segments are). |
| 3 | Double | Axis Alignment | The importance of the angle between the two pipe segments. Larger values will force the pipe segments to be more parallel. |
| 4 | Double | Center Pull | The importance of overlapping the center of each cut region (along the axis direction). |
| 5 | Double | Out of Material Weight | An additional weight when out of material. A value of zero implies that the solution does not care if there is sufficient material for the weld. |
| 6 | Double | Out of Material - Offset | A value that controls "how bad" the solution gets when there is insufficient material for the weld. |
| 7 | Boolean | Constrain Region at OD | If checked, ensures that the cut angle at the outer diameter does not cause the cut plane to exceed the defined region. If unchecked, this restriction is ignored and a cut will remain available. |
| 8 | Boolean | Constrain ID/OD overlap | If enabled, the inner diameter/outer diameter values for the pipes are used to ensure that there is sufficient material based on the angle of alignment. If unchecked, this restriction is ignored and a cut will remain available. |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship weights were set successfully |
| FAILURE | Relationship could not be found. |

Remarks

None.

Set Pipe Relationship Segment Properties

Modifies the Pipe segment definitions in a specified Pipe Fitting relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | Name of the relationship to edit. |
|---|------------------------|-------------------------|---|
| 1 | Double | Pipe 1 - Inner Diameter | ID definition for section 1 |
| 2 | Double | Pipe 1 - Outer Diameter | OD definition for section 1 |
| 3 | Double | Pipe 1 - Cut Begin | Starting cut position relative to the reference object origin |
| 4 | Double | Pipe 1 - Cut End | End cut position relative to the reference object origin |
| 5 | Double | Pipe 2 - Inner Diameter | ID definition for section 2 |
| 6 | Double | Pipe 2 - Outer Diameter | OD definition for section 2 |
| 7 | Double | Pipe 2 - Cut Begin | Starting cut position relative to the reference object origin |
| 8 | Double | Pipe 2 - Cut End | End cut position relative to the reference object origin |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship was edited successfully |
| FAILURE | Relationship could not be found. |

Remarks

None.

Get Pipe Relationship Properties

Returned the properties of the specified Pipe Fitting Relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---|
| 0 | Collection Object Name | Relationship Name | Name of the relationship to consider |
| 1 | Collection Object Name | Pipe 1 - Object Name | Name of the reference object defining segment 1 |
| 2 | Double | Pipe 1 - Inner Diameter | ID definition for section 1 |
| 3 | Double | Pipe 1 - Outer Diameter | OD definition for section 1 |
| 4 | Double | Pipe 1 - Cut Begin | Starting cut position relative to the reference object origin |
| 5 | Double | Pipe 1 - Cut End | End cut position relative to the reference object origin |
| 1 | Collection Object Name | Pipe 2 - Object Name | Name of the reference object defining segment 2 |
| 2 | Double | Pipe 2 - Inner Diameter | ID definition for section 2 |
| 3 | Double | Pipe 2 - Outer Diameter | OD definition for section 2 |
| 4 | Double | Pipe 2 - Cut Begin | Starting cut position relative to the reference object origin |
| 5 | Double | Pipe 2 - Cut End | End cut position relative to the reference object origin |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship properties were returned successfully |
| FAILURE | Relationship could not be found. |

Remarks

None.

Get Pipe Relationship Weights

This command is used to configure the weighting factors used in calculating the cut operation in a pipe relationship and its weighting as part of an optimization.

Input Arguments

| | | | |
|---|------------------------|-------------------|-----------------------------------|
| 0 | Collection Object Name | Relationship Name | Name of the relationship to edit. |
|---|------------------------|-------------------|-----------------------------------|

Return Arguments

| | | | |
|---|---------|--------------------------|---|
| 1 | Double | Overall Weight | The weight as a whole, which determines the influence in the overall optimization scheme. |
| 2 | Double | Axis Offset | The weight of minimizing the mutual perpendicular midpoint between the two pipe segments (i.e. how "coaxial" the two pipe segments are). |
| 3 | Double | Axis Alignment | The importance of the angle between the two pipe segments. Larger values will force the pipe segments to be more parallel. |
| 4 | Double | Center Pull | The importance of overlapping the center of each cut region (along the axis direction). |
| 5 | Double | Out of Material Weight | An additional weight when out of material. A value of zero implies that the solution does not care if there is sufficient material for the weld. |
| 6 | Double | Out of Material - Offset | A value that controls "how bad" the solution gets when there is insufficient material for the weld. |
| 7 | Boolean | Constrain Region at OD | If checked, ensures that the cut angle at the outer diameter does not cause the cut plane to exceed the defined region. If unchecked, this restriction is ignored and a cut will remain available. |
| 8 | Boolean | Constrain ID/OD overlap | If enabled, the inner diameter/outer diameter values for the pipes are used to ensure that there is sufficient material based on the angle of alignment. If unchecked, this restriction is ignored and a cut will remain available. |

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship weights were returned successfully. |
| FAILURE | Relationship could not be found. |

Remarks

None.

Get Pipe Relationship Cut Status

Returns the current cut status for each end of the specified pipe relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|-------------------|
| 0 | Collection Object Name | Relationship Name | Relationship Name |
|---|------------------------|-------------------|-------------------|

Return Arguments

| | | | |
|---|---------|-------------------------|-----------------------------------|
| 1 | Boolean | Pipe 1 - Cut Available? | True indicates a cut is available |
| 2 | Boolean | Pipe 1 - Cut Active? | True indicates the cut is active |
| 3 | Boolean | Pipe 2 - Cut Available? | True indicates a cut is available |
| 4 | Boolean | Pipe 2- Cut Active? | True indicates the cut is active |

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship properties were returned successfully. |
| FAILURE | The relationship could not be found. |

Remarks

Cut status depends on the placement and orientation of the pipe segments and the constraint settings in the Pipe to Pipe Relationship Weights dialog.

Make pipe Relationship Cut

Provides a means to activate a cut and return the cut frame.

Input Arguments

| | | | |
|---|------------------------|-----------------------|---|
| 0 | Collection Object Name | Relationship Name | Relationship Name |
| 1 | Boolean | Pipe 1 - Make Cut | True - Sets the cut to active. |
| 2 | Boolean | Pipe 1 - Create Frame | True - builds a frame to define the cut |
| 3 | Collection Object Name | Pipe 1 - Frame Name | Name to use for the newly created frame |
| 1 | Boolean | Pipe 2 - Make Cut | True - Sets the cut to active. |
| 2 | Boolean | Pipe 2 - Create Frame | True - builds a frame to define the cut |
| 3 | Collection Object Name | Pipe 2 - Frame Name | Name to use for the newly created frame |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The pipe relationship was edited successfully |
| FAILURE | The relationship could not be found or a name is invalid. |

Remarks

Typical workflow is as follows: The first cut is set to active for pipe 1 and a frame is generated using *Make Pipe Relationship Cut*. This frame is used to orient and guide the first cutting operation. After the cut has been made the cut end is then measured and the command *Pipe Relationship Force Cut to Frame* is used to finalize the cut. This process is then repeated for pipe 2.

Pipe Relationship Force Cut to Frame

Input Arguments

| | | | |
|---|------------------------|----------------------------|---|
| 0 | Collection Object Name | Relationship Name | Relationship Name |
| 1 | Boolean | Pipe 1- Force Cut to Frame | True - forces the cut to the specified frame |
| 2 | Collection Object Name | Pipe 1 - Frame Name | Name of the frame to force the pipe 1 cut to. |
| 3 | Boolean | Pipe 1- Force Cut to Frame | True - forces the cut to the specified frame |
| 4 | Collection Object Name | Pipe 1 - Frame Name | Name of the frame to force the pipe 1 cut to. |

Return Arguments

None

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship updated successfully |
| FAILURE | The relationship or one of the frames could not be found. |

Remarks

Typical workflow is as follows: The first cut is set to active for pipe 1 and a frame is generated using *Make Pipe Relationship Cut*. This frame is used to orient and guide the first cutting operation. After the cut has been made the cut end is then measured and the command *Pipe Relationship Force Cut to Frame* is used to finalize the cut. This process is then repeated for pipe 2.

Relationship Attributes

Set Relationship Dormant Status

Sets the dormant status for a specified relationship. When set to dormant the relationship does not recompute until the status is reverted. This provides an easy solution for frequent frame changes or optimization processes that would be otherwise hampered by the continual recomputation process.

Input Arguments

| | | | |
|---|-----------------------|----------------|--|
| 0 | Relationship Ref List | Relationships | The name of the relationship reference list. |
| 1 | Boolean | Dormant Status | Specified dormant status. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dormant status was set successfully. |
| FAILURE | The specified relationship(s) could not be found. |

Remarks

None.

Set Relationship Weights Normalized

Normalizes the weighting of all relationships in a collection.

Input Arguments

| | | | |
|---|-----------------|---|---|
| 0 | Collection Name | Collection Name | The name of the collection containing the relationships to normalize. |
| 1 | Boolean | Normalize on equation count AND tolerance width | Specifies whether both the equation count (number of points) and tolerance width will be used for normalizing relationship weights. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship weights were normalized successfully. |
| FAILURE | The specified collection could not be found. |

Remarks

When Argument 1 is set to TRUE, the relationships will be normalized on equation count only--all relationships will have the same weight regardless of how many measurements exist in each relationship. When Argument 1 is set to FALSE, collections with smaller tolerance widths are given more weight than those with large tolerance widths. This command will succeed even if the specified collection contains no relationships.

Get Relationship Type

Retrieves the type of relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to examine. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|--------|-------------------|---|
| 1 | String | Relationship Type | The type of the specified relationship. |
|---|--------|-------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The relationship type was retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

This command returns relationships of the following types:

Geometry Relationships:

Average Point Relationship, Line Geometry Relationship, Plane Geometry Relationship, Circle Geometry Relationship, Slot Geometry Relationship, Ellipse Geometry Relationship, Sphere Geometry Relationship, Cylinder Geometry Relationship, Cone Geometry Relationship, Paraboloid Geometry Relationship

Dynamic Intersection Relationships:

Dynamic Point Relationship, Line Geometry Relationship, Plane Geometry Relationship, Circle Geometry Relationship, Ellipse Geometry Relationship

Special Function Relationships:

Group to Group Relationship, Frame to Nominal Frame Relationship

Other Relationships:

Point to Point Relationship, Point to Object Relationship, Points to Objects Relationship, Points to Surface Faces Relationship, Point Clouds to Objects Relationship, Group to Group Relationship, Frame to Frame Relationship, Object to Object Direction Relationship

Set Relationship Weighting

Sets a specific weight on a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Double | Weight | The weight value to apply to the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship weight was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Relationship Weighting

Retrieves the weighting value on a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to examine. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|--------|--------|--|
| 1 | Double | Weight | The weighting value of the specified relationship. |
|---|--------|--------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship weight was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Sub Sampling Options

Sets the sub-sampling options for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Use every i-th point | Indicates whether sub-sampling should be used. |
| 2 | Integer | i value | Specifies the i value to use (if Argument 1 is TRUE). |
| 3 | Boolean | Use no more than n points | Indicates whether an upper limit will be placed on the number of points to be used in a relationship. |
| 4 | Integer | n value | The maximum number of points to use in the relationship (if Argument 3 is TRUE). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The options were set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Relationship Sub Sampling Options

Gets the sub-sampling options for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship with options. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|---------|---------------------------|--|
| 1 | Boolean | Use every i-th point | Indicates the sub-sampling that was used. |
| 2 | Integer | i value | Specifies the i value that was set. |
| 3 | Boolean | Use no more than n points | Indicates the upper limit placed on the number of points used in a relationship. |
| 4 | Integer | n value | The maximum number of points to use in the relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The options were retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None

Set Relationship Reporting Frame

Sets the reporting frame for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Collection Object Name | Reporting Frame | The new reporting frame for the relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame was set successfully. |
| FAILURE | The specified relationship or reporting frame could not be found. |

Remarks

None.

Get Relationship Reporting Frame

Returns the reporting frame for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|------------------------|-----------------|---|
| 1 | Collection Object Name | Reporting Frame | The new reporting frame for the relationship. |
|---|------------------------|-----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The reporting frame was returned successfully. |
| FAILURE | The specified relationship or reporting frame could not be found. |

Remarks

None.

Set Geom Relationship Criteria

Sets the status of a particular criteria in the specified relationship

Input Arguments

| | | | |
|---|---------------------------------|---------------------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | String | Criteria | String name of the criteria to be set |
| 2 | Boolean | Show in Report | Set to True indicates included or checked. |
| 3 | Tolerance Options (Scalar Type) | Tolerance Options | High and Low tolerance setting |
| 4 | Double | Optimization: Delta Weight | The weight set for the criteria to be used in an optimization such as move objects by minimizing relationships. |
| 5 | Double | Optimization: Out of Tolerance Weight | The weight to apply in an optimization when out of tolerance. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The criteria was set successfully. |
| FAILURE | The specified relationship or criteria could not be found. |

Remarks

This command is string specific. Take care to specify the criteria by name for the particular relationship used. To verify, open the relationships properties. A list is also available in the remarks under [Get Geom Relationship Criteria](#).

Get Geom Relationship Criteria

This command returns the available values for a given criteria, specified by name, for a specified geometry relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the geometry relationship to examine. |
| 1 | String | Criteria | The name of the criteria (see remarks below). |

Return Arguments

| | | | |
|----|---------|---------------------------------------|---|
| 2 | Double | Nominal | The nominal value. |
| 3 | Double | Measured | The measured value. |
| 4 | Double | Delta | The difference between the measured and nominal values. |
| 5 | Double | Low Tolerance | The low tolerance. |
| 6 | Double | High Tolerance | The high tolerance. |
| 7 | Double | Optimization: Delta Weight | The weight set for the criteria to be used in an optimization such as move objects by minimizing relationships. |
| 8 | Double | Optimization: Out of Tolerance Weight | The weight to apply in an optimization when out of tolerance. |
| 9 | String | Is within tolerance? | String defining tolerance condition (see below) |
| 10 | Boolean | Has Uncertainty? | True indicate has uncertainty computed |
| 11 | Double | Uncertainty | Uncertainty value |

Returned Status

| | |
|---------|--|
| SUCCESS | The relationship and criteria were retrieved successfully. |
| FAILURE | The specified geometry relationship or criteria was not found. |

Remarks

Values that do not apply will be populated with values of zero. The available criteria strings are as follows:

| <i>Ave Point</i> | <i>Line</i> | <i>Plane</i> | <i>Circle</i> | <i>Slot</i> | <i>Ellipse</i> |
|------------------|-------------|--------------|---------------|-------------|----------------|
| X | Length | Flatness | Diameter | Length | Centroid X |
| Y | Linearity | Centroid X | Radius | Width | Centroid Y |
| Z | Origin X | Centroid Y | Circularity | Centroid X | Centroid Z |
| Mag XYZ | Origin Y | Centroid Z | X | Centroid Y | Major Radius |
| Max | Origin Z | I | Y | Centroid Z | Minor Radius |
| RMS | I | J | Z | I | Major Diameter |
| | J | K | I | J | Minor Diameter |
| | K | Rx from Y | J | K | I |

| <i>Ave Point</i> | <i>Line</i> | <i>Plane</i> | <i>Circle</i> | <i>Slot</i> | <i>Ellipse</i> |
|------------------|--------------------|------------------|---------------|---------------|----------------|
| | Rx from Y | Ry from Z | K | Rx from Y | J |
| | Ry from Z | Rz from X | Rx from Y | Ry from Z | K |
| | Rz from X | Angle Between | Ry from Z | Rz from X | Rx from Y |
| | Angle Between | Avg Dist Between | Rz from X | Mag XYZ | Ry from Z |
| | Mutual Perp. Dist. | RMS | Mag XYZ | Mag XY | Rz from X |
| | RMS | | Mag XY | Angle Between | Mag XYZ |
| | | | Angle Between | RMS | Mag XY |
| | | | RMS | | Angle Between |
| | | | | | RMS |

| <i>Sphere</i> | <i>Cylinder</i> | <i>Cone</i> | <i>Paraboloid</i> | <i>Frame</i> |
|---------------|--------------------|---------------------|-------------------|--------------|
| Diameter | Diameter | Height | Focal Length | X |
| Radius | Radius | Small Base Radius | Focus X | Y |
| X | Length | Small Base Diameter | Focus Y | Z |
| Y | Cylindricity | Large Base Radius | Focus Z | Rx |
| Z | Origin X | Large Base Diameter | Directrix A | Ry |
| Mag XYZ | Origin Y | Included Angle | Directrix B | Rz |
| Sphericity | Origin Z | Vertex X | Directrix C | Total Angle |
| RMS | I | Vertex Y | Directrix D | Mag XYZ |
| | J | Vertex Z | Vertex X | |
| | K | I | Vertex Y | |
| | Rx from Y | J | Vertex Z | |
| | Ry from Z | K | I | |
| | Rz from X | Rx from Y | J | |
| | Angle Between | Ry from Z | K | |
| | Mutual Perp. Dist. | Rz from X | Rx from Y | |
| | RMS | Mag XYZ | Ry from Z | |
| | | Angle Between | Rz from X | |
| | | RMS | Focal Mag | |
| | | | Vertex Mag | |
| | | | RMS | |

Tolerance Condition String [A9]:

“N/A” if relationship doesn’t set tolerance for the specified criteria or doesn’t have data association yet

“Pass” if a measured value is within tolerance range;

<number> if the measured value is out of tolerance. It shows out-of-tolerance value.

Set Geom Relationship Nominal Geometry

Sets the name of the nominal geometry driven by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|--------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Compare to Nominal | True - sets the specified object as the nominal False - removes any preset nominal |
| 3 | Collection Object Name | Nominal Geometry | The name of the nominal object used. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The nominal was set successfully. |
| FAILURE | The specified relationship or criteria could not be found. |

Remarks

This command converts a Fit only relationship to a Fit and Compare relationship and sets the name of the nominal geometry used. It can also be used to turn off the nominal comparison.

Get Geom Relationship Nominal Geometry

Gets the name of the nominal geometry referenced by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
|---|------------------------|-------------------|------------------------------|

Return Arguments

| | | | |
|---|------------------------|------------------|----------------------------------|
| 1 | Collection Object Name | Nominal Geometry | The name of the nominal geometry |
|---|------------------------|------------------|----------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The nominal was set successfully. |
| FAILURE | The specified relationship or criteria could not be found. |

Remarks

This command returns the name of the nominal geometry referenced by the relationship specified.

Set Geom Relationship Measured Geometry

Sets the name of the measured geometry associated with the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
| 1 | Collection Object Name | Measured Geometry | Geometry to associate with the relationship. |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The measured geometry was associated successfully |
| FAILURE | The specified relationship or geometry could not be found. |

Remarks

Added in 2022, this command effectively converts a geometry relationship into a Comparison Only geometry relationship. Refer to the Relationships chapter of the User Manual for more information.

Get Geom Relationship Measured Geometry

Gets the name of the measured geometry driven by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
|---|------------------------|-------------------|------------------------------|

Return Arguments

| | | | |
|---|------------------------|-------------------|-----------------------------------|
| 1 | Collection Object Name | Measured Geometry | The name of the measured geometry |
|---|------------------------|-------------------|-----------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The measured geometry was returned successfully |
| FAILURE | The specified relationship could not be found. |

Remarks

This command returns the name of the measured geometry referenced by the relationship specified.

Set Geom Relationship Nominal Avg Point

Provides a means to control the nominal point used for comparison by the specified average point relationship.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
| 1 | Boolean | Compare to Nominal? | True - sets the specified point as the nominal False - removes any preset nominal |
| 2 | Point Name | Nominal Average Point | Name of the nominal point to apply |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The nominal point was set successfully. |
| FAILURE | The specified relationship or point could not be found. |

Remarks

None.

Get Geom Relationship Nominal Avg Point

Gets the name of the nominal point used for comparison by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
|---|------------------------|-------------------|------------------------------|

Return Arguments

| | | | |
|---|------------|-----------------------|-------------------------------|
| 1 | Point Name | Nominal Average Point | The name of the Nominal Point |
|---|------------|-----------------------|-------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The nominal point was retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

This command returns the name of the name of the nominal point referenced by the relationship specified.

Get Geom Relationship Measured Avg Point

Gets the name of the measured point computed by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|------------------------------|
| 0 | Collection Object Name | Relationship Name | The name of the relationship |
|---|------------------------|-------------------|------------------------------|

Return Arguments

| | | | |
|---|------------|------------------------|--------------------------------|
| 1 | Point Name | Measured Average Point | The name of the Measured Point |
|---|------------|------------------------|--------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The measured point was returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

This command returns the name of the name of the measured point referenced by the relationship specified.

Set Geom Relationship Projection Plane

Sets the status and name of the projection plane used by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-----------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Project to Plane? | Sets the projection plane status on/off |
| 3 | Collection Object Name | Projection Plane Name | The name of the plane used. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The projection plane status was set successfully. |
| FAILURE | The specified relationship or plane could not be found. |

Remarks

None.

Get Geom Relationship Projection Plane

Returns the status and name of the projection plane used by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 1 | Collection Object Name | Projection Plane Name | The name of the Projection Plane Assigned, if available. |
|---|------------------------|-----------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The projection plane was returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Geom Relationship Cardinal Points

Sets the status and name of the cardinal points built by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|---|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Create Cardinal Pts when Fitting? | Sets the cardinal point status on/off |
| 2 | Boolean | Prefix Cardinal Pts name with Rel Name? | Controls point name prefix |
| 3 | Boolean | Cardinal Pts Group Name | The name of the destination point group |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The cardinal point status was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

If the argument 2 is false and the points with the same name already exist in the specified group the names will automatically have a "*" appended.

Get Geom Relationship Cardinal Points

Gets the names of the cardinal points computed by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|---------------------|--------------------------|--|
| 1 | Point Name Ref List | Cardinal Point Name List | List of the names of the cardinal points |
|---|---------------------|--------------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The cardinal points were returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Geom Relationship Point List

Gets the names of the points associated with a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|---------------------|----------------|--|
| 1 | Point Name Ref List | All Points | List of the names of the associated points |
| 2 | Point Name Ref List | Used Points | List of the names of the points actively used in a fit |
| 3 | Point Name Ref List | Ignored Points | List of points associated but ignored |

Returned Status

| | |
|---------|--|
| SUCCESS | The points were returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

The Point List of a relationship can be directly used for outlier rejection and this command makes it clear which points were used and which were not.

Geom Relationship Ignore Input Points

This command applies a filter to the associated points, moving them into the ignored points list for a specified relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|---|---------------------------|--|
| 1 | Outlier Rejection Options (Scalar Type) | Outlier Rejection Options | The thresholds to use in selecting points to ignore. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The points were returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

The Point List of a relationship can be directly used for outlier rejection and this command moves points from the associated list to the associated but ignored list.

Geom Relationship Reuse Ignored Input Points

This command resets the associated points, moving any ignored points into the used category for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The points were returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

The Point List of a relationship can be directly used for outlier rejection and this command returns excluded points to list of points to use for evaluation.

Get Geom Relationship Auto Vectors

This command returns the current status and names of the auto vectors created by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|------------------------|--------------------------------------|---|
| 1 | Boolean | Auto Vectors Nominal(AVN) - Enabled? | True indicates that the AVN vectors have been created and are being controlled by the relationship. |
| 2 | Collection Object Name | Auto Vectors Nominal(AVN) - Name | Name of the associated Nominal Geometry to Points (AVN) vector group. |
| 3 | Boolean | Auto Vectors Nominal(AVF) - Enabled? | True indicates that the AVF vectors have been created and are being controlled by the relationship. |
| 4 | Collection Object Name | Auto Vectors Nominal(AVF) - Name | Name of the associated Fit Geometry to Points (AVF) vector group. |
| 5 | String | Points Type | Current configuration for the AVN output |

Returned Status

| | |
|---------|--|
| SUCCESS | The points were returned successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

The Points Type [A5] returns the configuration which might include “Cardinal Points” (which is short for: Nominal to Fit Cardinal Points), “Input Points”, or “Nominal Cardinal Points”

Set Geom Relationship Auto Vectors Nominal (AVN)

Sets the status of the nominal comparison auto vectors built by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Create Auto Vectors AVN | Sets the auto vector status on/off |
| 2 | Input Type | Points Type | Sets nominal points used |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The auto vector status was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Auto Vectors Fit (AVF)

Sets the status of the fit auto vectors built by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Create Auto Vectors AVF | Sets the auto vector status on/off |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The auto vector status was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Voxel Cloud Display

Provides the ability to enable/disable Voxel Cloud display in Cloud to Object relationships by providing the controls used in the relationship properties used for controlling voxel cloud display.

Input Arguments

| | | | |
|---|------------------------|--|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Boolean | Enable Voxel Cloud Display? | TRUE turns on Voxel display |
| 2 | Double | Voxel Size (-1.0 autodetect) | This controls the size of the voxel volume to consider for analysis |
| 3 | Integer | Min Pts Count Per Voxel | Voxels with fewer than the specified number of points will be ignored |
| 4 | Double | Voxel Rendering Diameter % (-1.0 fast) | The diameter of the voxel blotch displayed as a percentage of the voxel size. -1 sets the display to fast or shown as cloud points |
| 5 | Surface Analysis Mode | Surface Analysis Mode | Select the analysis mode to use |
| 6 | Colorization Options | Colorization Options | Defines the block colorization options to use when in Relationship Analysis mode |
| 7 | Boolean | Show Color Bar in View? | TRUE displays the color bar |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The auto vector status was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Desired Meas Count

Sets the status of the measurement count used by the specified relationship.

Input Arguments

| | | | |
|---|------------------------|---------------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Integer | Desired Measurement Count | Measurement count used |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurement count was set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

Measurement Count is used for automatic progression through an inspection list. When the limit it reached trapping will either stop or move to the next entry in the list. By default the measurement count is off (set to an integer of 0). To set an measurement count specify a positive number. To evaluate the relationship but not set it to trapping you can specify an integer of -1.

Make Vector Tolerance

Creates a vector tolerance that can be fed into another command, such as Make Point to Point Relationship.

Input Arguments

| | | | |
|----|---------|------------------------|---|
| 0 | Boolean | Use High X Tolerance | Indicates if a high tolerance should be applied to the X component. |
| 1 | Double | High X Tolerance | The high tolerance on the X component. |
| 2 | Boolean | Use High Y Tolerance | Indicates if a high tolerance should be applied to the Y component. |
| 3 | Double | High Y Tolerance | The high tolerance on the Y component. |
| 4 | Boolean | Use High Z Tolerance | Indicates if a high tolerance should be applied to the Z component. |
| 5 | Double | High Z Tolerance | The high tolerance on the Z component. |
| 6 | Boolean | Use High Mag Tolerance | Indicates if a high tolerance should be applied to the magnitude. |
| 7 | Double | High Mag Tolerance | The high tolerance on the magnitude. |
| 8 | Boolean | Use Low X Tolerance | Indicates if a low tolerance should be applied to the X component. |
| 9 | Double | Low X Tolerance | The low tolerance on the X component. |
| 10 | Boolean | Use Low Y Tolerance | Indicates if a low tolerance should be applied to the Y component. |
| 11 | Double | Low Y Tolerance | The low tolerance on the Y component. |
| 12 | Boolean | Use Low Z Tolerance | Indicates if a low tolerance should be applied to the Z component. |
| 13 | Double | Low Z Tolerance | The low tolerance on the Z component. |
| 14 | Boolean | Use Low Mag Tolerance | Indicates if a low tolerance should be applied to the magnitude. |
| 15 | Double | Low Mag Tolerance | The low tolerance on the magnitude. |

Return Arguments

| | | | |
|----|------------------|----------------------------|---------------------------------|
| 16 | Vector Tolerance | Resultant Vector Tolerance | The resulting vector tolerance. |
|----|------------------|----------------------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Vector Fit Constraint

Creates a vector fit constraint that can be fed into another command, such as Make Point to Point Relationship.

Input Arguments

| | | | |
|----|---------|--------------------|---|
| 0 | Boolean | Use High X Limit | Indicates if a high limit should be applied to the X component. |
| 1 | Double | High X Limit | The high limit on the X component. |
| 2 | Boolean | Use High Y Limit | Indicates if a high limit should be applied to the Y component. |
| 3 | Double | High Y Limit | The high limit on the Y component. |
| 4 | Boolean | Use High Z Limit | Indicates if a high limit should be applied to the Z component. |
| 5 | Double | High Z Limit | The high limit on the Z component. |
| 6 | Boolean | Use High Mag Limit | Indicates if a high limit should be applied to the magnitude. |
| 7 | Double | High Mag Limit | The high limit on the magnitude. |
| 8 | Boolean | Use Low X Limit | Indicates if a low limit should be applied to the X component. |
| 9 | Double | Low X Limit | The low limit on the X component. |
| 10 | Boolean | Use Low Y Limit | Indicates if a low limit should be applied to the Y component. |
| 11 | Double | Low Y Limit | The low limit on the Y component. |
| 12 | Boolean | Use Low Z Limit | Indicates if a low limit should be applied to the Z component. |
| 13 | Double | Low Z Limit | The low limit on the Z component. |
| 14 | Boolean | Use Low Mag Limit | Indicates if a low limit should be applied to the magnitude. |
| 15 | Double | Low Mag Limit | The low limit on the magnitude. |

Return Arguments

| | | | |
|----|-------------------|-----------------------------|----------------------------------|
| 16 | Vector Constraint | Resultant Vector Constraint | The resulting vector constraint. |
|----|-------------------|-----------------------------|----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Relationship Position Fit Constraints (Vector Type)

Sets positional constraints on a relationship.

Input Arguments

| | | | |
|---|------------------------|----------------------------|--------------------------------------|
| 0 | Collection Object Name | Relationship Name | The relationship to modify. |
| 1 | Vector Constraint | Position Vector Constraint | The positional constraints to apply. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The constraints were set successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Set Relationship Orientation Fit Constraints (Vector Type)

Sets orientation constraints on a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|--------------------------------------|
| 0 | Collection Object Name | Relationship Name | The relationship to modify. |
| 1 | Vector Constraint | Orientation Vector Constraint | The rotational constraints to apply. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The constraints were set successfully. |
| FAILURE | The relationship was not found. |

Remarks

None. **Set Relationship Tolerance (Vector Type)**

Sets tolerance of a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---------------------------------------|
| 0 | Collection Object Name | Relationship Name | The relationship to modify. |
| 1 | Vector Tolerance | Vector Tolerance | The tolerance to apply to the vector. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The tolerance were set successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Get Relationship Tolerance (Vector Type)

Gets tolerance of a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The relationship in which to retrieve the information. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|----|------------------|------------------------|---|
| 1 | Boolean | Use High X Tolerance? | Indicates if a high tolerance should be applied to the X component. |
| 2 | Double | High X Tolerance | The high tolerance on the X component. |
| 3 | Boolean | Use High Y | Indicates if a high tolerance should be applied to the Y component. |
| 4 | Double | High Y Tolerance | The high tolerance on the Y component. |
| 5 | Boolean | Use High Z Tolerance | Indicates if a high tolerance should be applied to the Z component. |
| 6 | Double | High Z Tolerance | The high tolerance on the Z component. |
| 7 | Boolean | Use High Mag Tolerance | Indicates if a high tolerance should be applied to the magnitude. |
| 8 | Double | High Mag Tolerance | The high tolerance on the magnitude. |
| 9 | Boolean | Use Low X Tolerance? | Indicates if a low tolerance should be applied to the X component. |
| 10 | Double | Low X Tolerance | The low tolerance on the X component. |
| 11 | Boolean | Use Low Y Tolerance? | Indicates if a low tolerance should be applied to the Y component. |
| 12 | Double | Low Y Tolerance | The low tolerance on the Y component. |
| 13 | Boolean | Use Low Z Tolerance? | Indicates if a low tolerance should be applied to the Z component. |
| 14 | Double | Low Z Tolerance | The low tolerance on the Z component. |
| 15 | Boolean | Use Low Mag Tolerance? | Indicates if a low tolerance should be applied to the magnitude. |
| 16 | Double | Low Mag Tolerance | The low tolerance on the magnitude. |
| 17 | Vector Tolerance | Vector Tolerance | The low tolerance on the magnitude. |

Returned Status

| | |
|---------|--|
| SUCCESS | The tolerance were retrieved successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Set Relationship Projection Options

Sets projection options of a relationship.

Input Arguments

| | | | |
|---|------------------------|--------------------|----------------------------------|
| 0 | Collection Object Name | Relationship Name | The relationship to modify. |
| 1 | Projection Options | Projection Options | The projection options to apply. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The projection options were set successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Get Relationship Projection Options

Gets projection options of a relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The relationship in which to retrieve the information. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|--------------------|---|--|
| 1 | Boolean | Ignore Edge Projections? | Whether edge projections ignored. |
| 2 | Boolean | Probe Offsets - Override Target Values? | Whether probe offsets are overridden with target values. |
| 3 | Double | Probe Offsets - Override Values | Probe offset override values. |
| 4 | Boolean | Add Extra Material? | Whether extra material was added. |
| 5 | Double | Extra Material Thickness | The extra material thickness if applicable. |
| 6 | Projection Options | Projection Options | The high tolerance on the Z component. |

Returned Status

| | |
|---------|---|
| SUCCESS | The projection options were retrieved successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Relationship Attributes (Scalar Types)

Set Relationship Outlier Rejection (Scalar Type)

Sets the outlier rejection for a specified relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|---|---------------------------|---|
| 1 | Outlier Rejection Options (Scalar Type) | Outlier Rejection Options | The new outlier rejection options for the specified relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The outlier rejection options were set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Relationship Outlier Rejection (Scalar Type)

Gets the outlier rejection for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|---|---------------------------|---|
| 1 | Boolean | Use High Limit? | Indicate whether to use a high limit. |
| 2 | Double | High Limit | Specify the high limit if used. |
| 3 | Boolean | Use Low Limit? | Indicate whether to use a low limit. |
| 4 | Double | Low Limit | Specify the low limit if used. |
| 5 | Outlier Rejection Options (Scalar Type) | Outlier Rejection Options | The new outlier rejection options for the specified relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The outlier rejection options were retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Rejection (Scalar Type)

Gets the outlier rejection for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship with information. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|---------|---------------------------|--|
| 1 | Boolean | Use every i-th point | Indicates the sub-sampling that was used. |
| 2 | Integer | i value | Specifies the i value that was set. |
| 3 | Boolean | Use no more than n points | Indicates the upper limit placed on the number of points used in a relationship. |
| 4 | Integer | n value | The maximum number of points to use in the relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The outlier rejection options were retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Relationship Fit Constraints (Scalar Type)

Sets the fit constraints for a specified relationship.

Input Arguments

| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|--------------------------------------|------------------------|--|
| 1 | Fit Constraint Options (Scalar Type) | Fit Constraint Options | The new fit constraint options for the specified relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The fit constraint options were set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Relationship Fit Constraints (Scalar Type)

Retrieves the fit constraints for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
|---|------------------------|-------------------|---|

Return Arguments

| | | | |
|---|--------------------------------------|------------------------|--|
| 1 | Fit Constraint Options (Scalar Type) | Fit Constraint Options | The new fit constraint options for the specified relationship. |
|---|--------------------------------------|------------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The fit constraint options were retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Object to Object Direction Relationship Fit Constraints

Sets the fit constraints for a specified object to object direction relationship.

Input Arguments

| | | | |
|---|--------------------------------------|---|--|
| 0 | Collection Object Name | Relationship Name | The name of the object to object direction relationship to modify. |
| 1 | Fit Constraint Options (Scalar Type) | Angle Between Vectors Fit Constraints | The new angular fit constraints for the specified object to object direction relationship. |
| 2 | Fit Constraint Options (Scalar Type) | Mutual Perpendicular Length Fit Constraints | The new mutual perpendicular length fit constraints for the specified object to object direction relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The fit constraint options were set successfully. |
| FAILURE | The specified object to object direction relationship could not be found. |

Remarks

None.

Set Relationship Tolerance (Scalar Type)

Sets the tolerance for a specified relationship.

Input Arguments

| | | | |
|---|---------------------------------|-------------------|---|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Tolerance Options (Scalar Type) | Tolerance Options | The new tolerance options for the specified relationship. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The tolerances were set successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Get Relationship Tolerance (Scalar Type)

Gets the tolerance for a specified relationship.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to get information. |
|---|------------------------|-------------------|--|

Return Arguments

| | | | |
|---|---------------------------------|---------------------|---|
| 1 | Boolean | Use High Tolerance? | Whether to use the high tolerance. |
| 2 | Double | High Tolerance | The high tolerance value. |
| 3 | Boolean | Use Low Tolerance? | Whether to use the low tolerance. |
| 4 | Double | Low Tolerance | The low tolerance value. |
| 5 | Tolerance Options (Scalar Type) | Tolerance Options | The new tolerance options for the specified relationship. |

Returned Status

| | |
|---------|--|
| SUCCESS | The tolerances were retrieved successfully. |
| FAILURE | The specified relationship could not be found. |

Remarks

None.

Set Object to Object Direction Relationship Tolerance

Sets the tolerance for a specified object to object direction relationship.

Input Arguments

| | | | |
|---|---------------------------------|--|--|
| 0 | Collection Object Name | Relationship Name | The name of the object to object direction relationship to modify. |
| 1 | Tolerance Options (Scalar Type) | Angle Between Vectors Tolerances | The new tolerances for the angle between vectors. |
| 2 | Tolerance Options (Scalar Type) | Mutual Perpendicular Length Tolerances | The new tolerances for the mutual perpendicular length. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The tolerances were set successfully. |
| FAILURE | The specified object to object direction relationship could not be found. |

Remarks

None.

Make Scalar Tolerance

Makes a scalar tolerance.

Input Arguments

| | | | |
|---|---------|--------------------|---|
| 0 | Boolean | Use High Tolerance | Indicates if the high tolerance should be active. |
| 1 | Double | High Tolerance | The high tolerance value. |
| 2 | Boolean | Use Low Tolerance | Indicates if the low tolerance should be active. |
| 3 | Double | Low Tolerance | The low tolerance value. |

Return Arguments

| | | | |
|---|---------------------------------|-----------------------------|----------------------------------|
| 4 | Tolerance Options (Scalar Type) | Resultant Tolerance Options | The resulting tolerance options. |
|---|---------------------------------|-----------------------------|----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Scalar Fit Constraint

Makes a scalar fit constraint.

Input Arguments

| | | | |
|---|---------|----------------|---|
| 0 | Boolean | Use High Limit | Indicates if the high limit should be active. |
| 1 | Double | High Limit | The upper limit for the fit constraint. |
| 2 | Boolean | Use Low Limit | Indicates if the low limit should be active. |
| 3 | Double | Low Limit | The lower limit for the fit constraint. |

Return Arguments

| | | | |
|---|--------------------------------------|------------------------------|-----------------------------------|
| 4 | Fit Constraint Options (Scalar Type) | Resultant Constraint Options | The resulting constraint options. |
|---|--------------------------------------|------------------------------|-----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Symmetric Scalar Tolerance

Makes a symmetric tolerance.

Input Arguments

| | | | |
|---|--------|---------------|--------------------------------|
| 0 | Double | +/- Tolerance | The symmetric tolerance value. |
|---|--------|---------------|--------------------------------|

Return Arguments

| | | | |
|---|---------------------------------|-----------------------------|----------------------------------|
| 1 | Tolerance Options (Scalar Type) | Resultant Tolerance Options | The resulting tolerance options. |
|---|---------------------------------|-----------------------------|----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Outlier Rejection Options

Makes outlier rejection options.

Input Arguments

| | | | |
|---|---------|----------------|---|
| 0 | Boolean | Use High Limit | Indicates whether a high outlier rejection limit should be applied. |
| 1 | Double | High Limit | The high limit to use, if Argument 0 is TRUE. |
| 2 | Boolean | Use Low Limit | Indicates whether a low outlier rejection limit should be applied. |
| 3 | Double | Low Limit | The low limit to use, if Argument 2 is TRUE. |

Return Arguments

| | | | |
|---|---|-------------------------------------|--|
| 4 | Outlier Rejection Options (Scalar Type) | Resultant Outlier Rejection Options | The resulting outlier rejection options. |
|---|---|-------------------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Symmetric Outlier Rejection Options

Makes symmetric outlier rejection options.

Input Arguments

| | | | |
|---|--------|-----------|---|
| 0 | Double | +/- Limit | The high/low symmetric limit for outlier rejection. |
|---|--------|-----------|---|

Return Arguments

| | | | |
|---|---|-------------------------------------|--|
| 1 | Outlier Rejection Options (Scalar Type) | Resultant Outlier Rejection Options | The resulting symmetric outlier rejection options. |
|---|---|-------------------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Events

Get Number of Events in Event Ref List

Obtains the number of events in an event reference list.

Input Arguments

| | | | |
|---|----------------|------------|--------------------------------------|
| 0 | Event Ref List | Event List | The event reference list to examine. |
|---|----------------|------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|-----------------------------------|
| 1 | Integer | Total Count | The number of events in the list. |
|---|---------|-------------|-----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get i-th Event From Event Ref List

Retrieves the event at the specified index in an event reference list.

Input Arguments

| | | | |
|---|----------------|-------------|--------------------------------------|
| 0 | Event Ref List | Event List | The event reference list to examine. |
| 1 | Integer | Event Index | The zero-based index into the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|----------------------|
| 2 | Collection Object Name | Resultant Item | The resulting event. |
|---|------------------------|----------------|----------------------|

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The event was retrieved successfully. |
| FAILURE | An invalid index was provided. |

Remarks

None.

Get i-th Event From Event Ref List (Iterator)

Iterates through an event reference list, retrieving successive events from the list.

Input Arguments

| | | | |
|---|----------------|-----------------------------|--|
| 0 | Event Ref List | Reference List | The event reference list to examine. |
| 1 | Integer | Event Index | The index at which to start in the list. |
| 2 | Step ID | Step to Jump at End of List | The step to jump to upon reaching the end of the list. |

Return Arguments

| | | | |
|---|------------------------|----------------|--|
| 3 | String | Collection | The collection containing the current event. |
| 4 | String | Event | The name of the current event. |
| 5 | Collection Object Name | Resultant Item | The current event as a collection object name. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Rename Event

Renames an event.

Input Arguments

| | | | |
|---|------------------------|----------------------|--|
| 0 | Collection Object Name | Original Event Name | The name of the event to rename. |
| 1 | Collection Object Name | New Event Name | The new name for the event. |
| 2 | Boolean | Overwrite if exists? | Indicates whether an existing event should be overridden if the new event name already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The event was renamed successfully. |
| FAILURE | The source event could not be found. |

Remarks

None.

Delete Event

Deletes an event from the job.

Input Arguments

| | | | |
|---|------------------------|------------|----------------------------------|
| 0 | Collection Object Name | Event Name | The name of the event to delete. |
|---|------------------------|------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The event was deleted successfully. |
| FAILURE | The event could not be found. |

Remarks

None.

Cloud Filters

Filter Clouds to Plane

Filters one or more clouds to a plane by proximity and outputs a new point group or cloud. Cloud points within the proximity are kept, while cloud points outside of the proximity are removed.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Collection Object Name | Filter Plane's Name | The name of the plane to filter to. |
| 2 | Collection Object Name | Output Group Name | The name for the resulting point group or point cloud (as specified in Argument 5). |
| 3 | Double | Proximity | The proximity inside which cloud points remain. |
| 4 | Offset Direction Type | Allowable Offset Dir | Specify whether the proximity implies to just the positive or negative side of the plane, or whether it applies to both sides of the plane. |
| 5 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds or filter plane could not be found. |

Remarks

Offsets are retained and match the source points in the new group or point cloud.

Filter Clouds to Group

Filters one or more clouds to a point group by proximity and outputs a new point group. Cloud points within the proximity are kept, while cloud points outside of the proximity are removed.

Input Arguments

| | | | |
|---|---------------------------------|--|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Collection Object Name | Filter Group's Name | The name of the group to filter to. |
| 2 | Collection Object Name | Output Group Name | The name for the resulting point group. |
| 3 | Double | Proximity (0 for Closest Point only) | The proximity to each filter group point in which cloud points are kept. If set to zero, only the closest cloud point to each filter point is kept. |
| 4 | Integer | Maximum Number of Points (0 for Unlimited) | The maximum number of cloud points to filter to each filter point. If set to zero, there is no limit. |
| 5 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds or filter group could not be found. |

Remarks

None.

Filter Clouds to Surface

Filters one or more clouds to a surface by proximity and outputs a new point group. Cloud points within the proximity are kept, while cloud points outside of the proximity are removed.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Collection Object Name | Filter Surface's Name | The name of the surface to filter to. |
| 2 | Collection Object Name | Output Group Name | The name for the resulting point group. |
| 3 | Double | Low Proximity | The minimum proximity value. Points above this value (and below the high proximity) will be kept. |
| 4 | Double | High Proximity | The maximum proximity value. Points below this value (and above the low proximity) will be kept. |
| 5 | Integer | Skip Factor | A subsampling value to use. For example, if set to 5, only every 5th point will be considered. |
| 6 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds or filter group could not be found. |

Remarks

Filter Clouds to Surfaces offers the ability to use an asymmetric distance such as +.1 to +.5 but it ignores edges entirely. This means points beyond an edge are also picked up as long as they are within the specified proximity.

Auto Filter Points/Groups/Clouds to Surface Faces is an alternative that take longer to process, only offsets single proximity value above, below or in both directions but respects edges. This command should be used when edges are of concern.

Filter Clouds to BSplines

Filters one or more clouds to one or more B-Splines by proximity and outputs a new point group. Cloud points between the minimum and maximum proximity to a B-Spline are kept, while cloud points outside of the specified range are removed.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Collection Object Name Ref List | Filter BSpline Names | The names of the B-Splines to filter to. |
| 2 | Collection Object Name | Output Group Name | The name for the resulting point group. |
| 3 | Double | Minimum Proximity | The minimum distance for which points should be retained. |
| 4 | Double | Maximum Proximity | The maximum distance for which points should be retained. |
| 5 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds or splines (but not all) could not be found. |
| FAILURE | The clouds or filter splines could not be found. |

Remarks

None.

Filter Clouds to Line Segment

Filters one or more clouds to a line segment by proximity and outputs a new point group. Cloud points between the minimum and maximum proximity to a B-Spline are kept, while cloud points outside of the specified range are removed.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Point Name | First Line End Point | The coordinate of one end of the line. |
| 2 | Point Name | Second Line End Point | The coordinate of the other end of the line. |
| 3 | Collection Object Name | Output Group Name | The name for the resulting point group. |
| 4 | Double | Minimum Proximity | The minimum distance for which points should be retained. |
| 5 | Double | Maximum Proximity | The maximum distance for which points should be retained. |
| 6 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds could not be found. |

Remarks

None.

Filter Clouds to Vector Groups - Resolve Points

Filters one or more clouds to one or more vector groups and outputs a new point group. Cloud points between the minimum and maximum proximity to a vector (radially)--and no more than the maximum distance from the beginning (tail end) of a vector--will be averaged to a new position (along the vector axis) and will be located on each vector's axis.

Input Arguments

| | | | |
|---|---------------------------------|------------------------------------|--|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Collection Object Name Ref List | Vector Group Names | The names of the vector groups to filter to. |
| 2 | Collection Object Name | Output Group Name | The name for the resulting point group. |
| 3 | Double | Minimum Proximity | The minimum distance for which points should be retained. |
| 4 | Double | Maximum Proximity | The maximum distance for which points should be retained. |
| 5 | Double | Maximum Distance From Vector Begin | The maximum distance from the tail end of a vector to a cloud point in order for that point to be considered. |
| 6 | Integer | Minimum number of required points | The minimum number of cloud points required to satisfy the proximity requirements (Arguments 3-5) in order for a filtered point to be created. |
| 7 | Output Type | Output Type | Specify whether point clouds or point groups should be created. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds or vector groups (but not all) could not be found. |
| FAILURE | The clouds or vector groups could not be found. |

Remarks

If the "Minimum number of required points" argument is not satisfied for a given vector, no point will be created on that vector.

RGB Cloud Point Filter

Sets the cloud point display filtering based upon the selected RGB and Grey scale component thresholds

Input Arguments

| | | | |
|----|---------------------------------|-----------------------|--|
| 0 | String | Filter Name | Saved name for the filter |
| 1 | Collection Object Name Ref List | Clouds to be Filtered | List of clouds to adjust |
| 2 | Boolean | Red Enabled | True enables red filtering |
| 3 | Boolean | Red High Enabled | True enables the upper limit threshold |
| 4 | Integer | Red High Threshold | The threshold value above which are hidden |
| 5 | Boolean | Red Low Enabled | True enables the upper lower threshold |
| 6 | Integer | Red Low Threshold | The threshold value below which are hidden |
| 7 | Boolean | Green Enabled | True enables red filtering |
| 8 | Boolean | Green High Enabled | True enables the upper limit threshold |
| 9 | Integer | Green High Threshold | The threshold value above which are hidden |
| 10 | Boolean | Green Low Enabled | True enables the upper lower threshold |
| 11 | Integer | Green Low Threshold | The threshold value below which are hidden |
| 12 | Boolean | Blue Enabled | True enables red filtering |
| 13 | Boolean | Blue High Enabled | True enables the upper limit threshold |
| 14 | Integer | Blue High Threshold | The threshold value above which are hidden |
| 15 | Boolean | Blue Low Enabled | True enables the upper lower threshold |
| 16 | Integer | Blue Low Threshold | The threshold value below which are hidden |
| 17 | Boolean | Gray Enabled | True enables red filtering |
| 18 | Boolean | Gray High Enabled | True enables the upper limit threshold |
| 19 | Integer | Gray High Threshold | The threshold value above which are hidden |
| 20 | Boolean | Gray Low Enabled | True enables the upper lower threshold |
| 21 | Integer | Gray Low Threshold | The threshold value below which are hidden |
| 22 | RGB Filter Mode Type | RGB Filter Operation | Filter operation control to apply |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds could not be found. |

Remarks

For more details refer to the [Color \(Intensity\) Mode](#) section of the Users Manual in the chapter on Clouds.

Delete Cloud Points by Radial Distance from Points

Applies sphere filters to one or more clouds. All cloud points inside or outside a specified radius of one or more supplied center points will be removed.

Input Arguments

| | | | |
|---|---------------------------------|---------------|--|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Point Name Ref List | Points | One or more center points to filter to. |
| 2 | Double | Radius | The radius to use for the sphere filter. |
| 3 | Boolean | Delete Inside | Indicate whether points inside the radius should be deleted, or points outside the radius should be deleted. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds or points (but not all) could not be found. |
| FAILURE | The clouds or points could not be found. |

Remarks

None.

Delete Cloud Points by X Y Z Range

Applies a box filter to one or more clouds. Removes all cloud points inside or outside a specified box (in the working coordinate frame).

Input Arguments

| | | | |
|---|---------------------------------|---------------|---|
| 0 | Collection Object Name Ref List | Cloud Names | One or more point clouds to filter. |
| 1 | Optional Double | X Min | The minimum X value to use. |
| 2 | Optional Double | X Max | The maximum X value to use. |
| 3 | Optional Double | Y Min | The minimum Y value to use. |
| 4 | Optional Double | Y Max | The maximum Y value to use. |
| 5 | Optional Double | Z Min | The minimum Z value to use. |
| 6 | Optional Double | Z Max | The maximum Z value to use. |
| 7 | Boolean | Delete Inside | Specify whether to delete points inside the box or outside the box. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The cloud(s) were filtered successfully. |
| PARTIAL SUCCESS | One or more clouds (but not all) could not be found. |
| FAILURE | The clouds could not be found. |

Remarks

If one of the range values is set to “Ignore”, then that value is considered to be infinite. For example, if X Max is set to “Ignore”, then all values from X Min to positive infinity are considered to be “in the box”.

Scale Bars

Get Scale Bar Stats

Obtains the nominal length, actual length, and deviation for an existing scale bar.

Input Arguments

| | | | |
|---|------------------------|----------------|--|
| 0 | Collection Object Name | Scale Bar Name | The name of the scale bar to analyze. |
| 1 | Double | Nominal Length | The nominal length of the scale bar. |
| 2 | Double | Actual Length | The measured length of the scale bar. |
| 3 | Double | Deviation | The deviation between the measured length and nominal length of the scale bar. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The check was performed successfully. |
| FAILURE | Either or both points were not found, or the length exceeded the allowable tolerance. |

Remarks

If the measured length is shorter than the nominal length, the deviation will be reported as a negative value.

Scale Bar Check

Performs a temperature-compensated scale bar check.

Input Arguments

| | | | |
|---|------------|-------------------------|--|
| 0 | Point Name | ScaleBar Point A | The point representing one end of the scale bar. |
| 1 | Point Name | ScaleBar Point B | The point representing the other end of the scale bar. |
| 2 | Double | Current Temperature (F) | The ambient temperature (temperature of the scale bar), in degrees Fahrenheit. |
| 3 | Double | Length of Bar at 68F | The length of the scale bar at 68 degrees Fahrenheit. |
| 4 | Double | Material CTE (PPM/F) | The Coefficient of Thermal Expansion of the scale bar, in Parts per Million per degree Fahrenheit. |
| 5 | Double | Tolerance | The allowable tolerance for the scale bar's length (from nominal). |

Return Arguments

| | | | |
|---|--------|------------------|--|
| 6 | Double | Deviation at 68F | The deviation between the measured scale bar and the nominal scale bar, compensated to a 68 degree Fahrenheit value. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The check was performed successfully. |
| FAILURE | Either or both points were not found, or the length exceeded the allowable tolerance. |

Remarks

None.

Delete Scale Bar

Deletes a scale bar.

Input Arguments

| | | | |
|---|------------------------|----------------|---|
| 0 | Collection Object Name | Scale Bar Name | The collection and name of the scale bar. |
|---|------------------------|----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scale bar was deleted successfully. |
| FAILURE | The specified scale bar could not be found. |

Remarks

None.

Reverse Engineering

Send Points to Geomagic

Sends one or more point groups to Geomagic (Power3).

Input Arguments

| | | | |
|---|---------------------------------|------------------------|---|
| 0 | Collection Object Name Ref List | Point Group(s) to Send | The list of point groups to send to Geomagic. |
|---|---------------------------------|------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The groups were sent successfully. |
| FAILURE | One or more groups were not found, or the connection to Geomagic could not be established. |

Remarks

This command requires the Power3 package.

Send Clouds to Geomagic

Sends one or more point clouds to Geomagic (Power3).

Input Arguments

| | | | |
|---|---------------------------------|------------------------|---|
| 0 | Collection Object Name Ref List | Point Cloud(s) to Send | The list of point clouds to send to Geomagic. |
|---|---------------------------------|------------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The clouds were sent successfully. |
| FAILURE | One or more clouds were not found, or the connection to Geomagic could not be established. |

Remarks

This command requires the Power3 package.

Generate/Regenerate Coarse Mesh

Input Arguments

| | | | |
|---|------------------------|--------------------------------|----------------------------------|
| 0 | Collection Object Name | Source Cloud | The cloud used to build the mesh |
| 1 | Collection Object Name | Output Mesh Name | Name of the mesh to build |
| 2 | Double | Deviation Error (0.0 for none) | Deviation Error to use |
| 3 | Double | Accuracy | Accuracy setting to apply |
| 4 | Double | Minimum Average Distance | Minimum Average Distance |
| 5 | Integer | Hole Options (0..2) | Hole Options |
| 6 | Double | Maximum Triangle Edge Length | Maximum Triangle Edge Length |
| 7 | Integer | Optimization Structure Options | Optimization Structure Options |
| 8 | Boolean | Reverse Normal Vectors | Choose to Reverse Normal Vectors |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mesh was built successfully. |
| FAILURE | Cloud was not found, or the mesh could not be built. |

Remarks

DeviationError - Deviation error (tolerance) to respect

Accuracy - Noise reduction strategy.

MiniAverageDist - =0: No noise reduction. >0: Mini average distance between points. The function project a “grid” of this size on the shape to mesh and takes ONLY the best points in each grid element. <0: Greatest level of detail. The function tries first to choose the best points as if MiniAverageDist>0. Then, if the tolerance is not reached, the other points (suspected to be noisy) can also be chosen.

OptionHole - 0: Try to detect all the holes, 1: Detect the outside border and fill the inside holes. 2: Try to close the polyhedron (watertight mesh).

MaximumTriangleEdgeLength - Max length of triangle to fill holes; -1 if no limit.

Optimization Structure - Optimized structure option to choose the most relevant points: bit 0: use the normal table if it exists and its size is lower than MiniAverageDist, bit 1: If true and there is scanning direction inside the cloud, we put the scanning direction of points of the cloud inside vertex of the resulting mesh.

Generate General Mesh

Builds a mesh using the general mesh function using the referenced point cloud data.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------------------|--|
| 0 | Collection Object Name | Output Mesh Name | Name of the mesh to build |
| 1 | Collection Object Name Ref list | Cloud to Mesh | List of clouds to use for the mesh |
| 2 | Double | Maximum Triangle Size | Resolution of the mesh |
| 3 | Double | Smallest Hole Diameter | Smallest hole to detect |
| 4 | Boolean | Finalize | True builds a finalized mesh |
| 5 | Boolean | Use Scan Direction For Point Normal | True enables the use of the normal transforms saved within a Scan Stripe Cloud |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mesh was built successfully. |
| FAILURE | One of the clouds was not found, or the mesh could not be built. |

Remarks

None.

Mesh Volume

Returns the volume of a mesh above or below a specified reference plane.

Input Arguments

| | | | |
|---|---------------------------------|-------|--|
| 0 | Collection Object Name | Mesh | Name of the mesh to consider |
| 1 | Collection Object Name Ref list | Plane | Reference plane used for the computation |

Return Arguments

| | | | |
|---|--------|-------|---------------------------------|
| 2 | Double | Above | Computed volume above the plane |
| 3 | Double | Below | Computed volume below the plane |

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The volume was computed successfully. |
| FAILURE | The mesh or plane were not found. |

Remarks

If the mesh is complete and fully encloses a volume, and is not intersected by the reference plane, the full enclosed volume will be returned either above or below based upon the position of the mesh with respect to the plane.

If the plane bisects an enclosed volume the volume above and below the plane will be reported.

If the mesh does not represent an enclosed volume, then the volume reported will represent the volume between the mesh and the projected triangles on the reference plane.

Dimensions

Create Point to Point Dimension

Creates a dimension between two points.

Input Arguments

| | | | |
|---|------------|--------------|------------------------------------|
| 0 | Point Name | First Point | The first point in the dimension. |
| 1 | Point Name | Second Point | The second point in the dimension. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension was created successfully. |
| FAILURE | One or both points could not be found. |

Remarks

None.

Create Point to Object Dimension

Creates a dimension between an object and a point.

Input Arguments

| | | | |
|---|------------------------|--------|-------------------------------------|
| 0 | Point Name | Point | The point to use in the dimension. |
| 1 | Collection Object Name | Object | The object to use in the dimension. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension was created successfully. |
| FAILURE | The point and/or object could not be found. |

Remarks

None.

Create Object to Object Dimension

Creates a dimension between two objects.

Input Arguments

| | | | |
|---|------------------------|---------------|--|
| 0 | Collection Object Name | First Object | The first object to use in the dimension. |
| 1 | Collection Object Name | Second Object | The second object to use in the dimension. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension was created successfully. |
| FAILURE | One or both objects could not be found. |

Remarks

None.

Create Diameter Dimension

Creates a diametrical dimension.

Input Arguments

| | | | |
|---|------------------------|--------|---|
| 0 | Collection Object Name | Object | The circle, sphere, or cylinder to dimension. |
|---|------------------------|--------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The dimension was created successfully. |
| FAILURE | The object could not be found or was not of the proper type. |

Remarks

None.

Create Radius Dimension

Creates a radial dimension between two objects.

Input Arguments

| | | | |
|---|------------------------|--------|--|
| 0 | Collection Object Name | Object | The circle, sphere or cylinder to dimension. |
|---|------------------------|--------|--|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The dimension was created successfully. |
| FAILURE | The object could not be found or was not of the proper type. |

Remarks

None.

Set Point to Point Dimension Properties

Sets dimension properties for point to point dimension.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |
| 2 | Collection Object Name | Reference Frame Name | The frame in which the dimension will be displayed. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

Set Point to Object Dimension Properties

Sets dimension properties for point to object dimension.

Input Arguments

| | | | |
|---|------------------------|----------------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

Set Object to Object Dimension Properties

Sets dimension properties for object to object dimension.

Input Arguments

| | | | |
|---|------------------------|----------------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

Set Diameter Dimension Properties

Sets dimension properties for diameter dimension.

Input Arguments

| | | | |
|---|------------------------|----------------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

Set Radius Dimension Properties

Sets dimension properties for radius dimension.

Input Arguments

| | | | |
|---|------------------------|----------------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

Set Common Properties to Dimensions

Sets common properties for dimensions. These include properties common to all dimension types including text size, placement and color.

Input Arguments

| | | | |
|---|----------------------|----------------------|--|
| 0 | Dimension Ref List | Dimension List | The name of the dimension ref list. |
| 1 | Dimension Properties | Dimension Properties | Set the properties. |
| 2 | Boolean | Show? (Hide = False) | Provides a means to control the visibility of all the selected dimensions. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension properties were set successfully. |
| FAILURE | The dimensions could not be found. |

Remarks

None.

Make a Dimension Ref List from a Collection

Creates a reference list of dimensions from a collection.

Input Arguments

| | | | |
|---|-----------------|-----------------|-----------------------------|
| 0 | Collection Name | Collection Name | The name of the collection. |
|---|-----------------|-----------------|-----------------------------|

Return Arguments

| | | | |
|---|--------------------|--------------------|---|
| 1 | Dimension Ref List | Dimension Ref List | The dimension ref list created from collection. |
|---|--------------------|--------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The dimension ref list was created successfully. |
| FAILURE | The collection and/or the dimension ref list could not be found. |

Remarks

None.

Make a Dimension Ref List- WildCard Selection

Creates a reference list of dimensions using a wildcard word , building a list of dimensions with names that match the string criteria.

Input Arguments

| | | | |
|---|--------|------------------------------|------------------------------------|
| 0 | String | Collection WildCard Criteria | The name of the collection to for. |
| 1 | String | Dimension WildCard Criteria | The name of the dimension to for. |

Return Arguments

| | | | |
|---|--------------------|--------------------|---|
| 1 | Dimension Ref List | Dimension Ref List | The dimension ref list created from collection. |
|---|--------------------|--------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The dimension ref list was created successfully. |
| FAILURE | The collection and/or the dimension ref list could not be found. |

Remarks

None.

Get Number of Dimensions in Dimension Ref List

Counts the number of dimensions in the dimension reference list.

Input Arguments

| | | | |
|---|--------------------|-----------------|-------------------------------------|
| 0 | Dimension Ref List | Dimensions List | The name of the dimension ref list. |
|---|--------------------|-----------------|-------------------------------------|

Return Arguments

| | | | |
|---|---------|-------------|---|
| 1 | Integer | Total Count | The number of dimensions in the ref list. |
|---|---------|-------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The total amount of dimensions were counted successfully. |
| FAILURE | The dimension ref list could not be found. |

Remarks

None.

Add a Dimension to Dimension Ref List

Adds another dimension to the dimension reference list.

Input Arguments

| | | | |
|---|------------------------|----------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
| 1 | Dimension Ref List | Dimension List | The dimension ref list. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The dimension was added to the ref list successfully. |
| FAILURE | The dimension and/or the dimension ref list could not be found. |

Remarks

None.

Get i-th Dimension From Dimension Ref List

Returns the specified dimension, using an index, from the dimension reference list.

Input Arguments

| | | | |
|---|--------------------|-----------------|--|
| 0 | Dimension Ref List | Dimensions List | The name of the dimension ref list. |
| 1 | Integer | Dimension Index | The number of the dimension in the dimension ref list. |

Return Arguments

| | | | |
|---|------------------------|----------------|--|
| 2 | Collection Object Name | Resultant Item | The dimension from the dimension ref list. |
|---|------------------------|----------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | Gets the correct dimension from the dimension ref list successfully. |
| FAILURE | The dimension ref list could not be found. |

Remarks

None.

Get i-th Dimension From Dimension Ref List (Iterator)

Returns the specified dimension, using an index, from the dimension reference list. The “Iterator” step has a built in counter and its index will increment automatically each time the step is executed.

Input Arguments

| | | | |
|---|--------------------|-----------------------------|---|
| 0 | Dimension Ref List | Dimensions List | The name of the dimension ref list. |
| 1 | Integer | Dimension Index | The number of the dimension in the dimension ref list. |
| 2 | Step ID | Step to Jump At End of List | The step to jump to when the index exceeds the number of dimensions in the list |

Return Arguments

| | | | |
|---|------------------------|----------------|--|
| 2 | Collection Object Name | Resultant Item | The dimension from the dimension ref list. |
|---|------------------------|----------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | Gets the correct dimension from the dimension ref list successfully. |
| FAILURE | The dimension ref list could not be found. |

Remarks

None.

Get Dimension Value

Retrieves a value from a dimension.

Input Arguments

| | | | |
|---|------------------------|----------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
|---|------------------------|----------------|----------------------------|

Return Arguments

| | | | |
|---|---------|------------------------|--|
| 1 | Double | Dimensions Value | The value of the dimension. |
| 2 | Boolean | Nominal Value Enabled | True indicates that the nominal is enabled. |
| 3 | Boolean | High Tolerance Enabled | True indicates High Tolerance is enabled |
| 4 | Boolean | Low Tolerance Enabled | True indicates Low Tolerance is enabled |
| 5 | Double | Nominal Value | The Nominal value set |
| 6 | Double | High Tolerance | The High Tolerance set, this includes the Nominal + High Tolerance |
| 7 | Double | Low Tolerance | The Low Tolerance set, this includes the Nominal - Low Tolerance |

Returned Status

| | |
|---------|--|
| SUCCESS | Gets the correct dimension value successfully. |
| FAILURE | The dimension could not be found. |

Remarks

The high and low tolerance values returned in this step can always be assumed to be the full extents of the tolerance range. For example, a tolerance set as 1.5” +/-0.01 will report a high tolerance of 1.51 and a low tolerance of 1.49 regardless of the nominal value assignment.

Delete Dimension

Deletes a dimension.

Input Arguments

| | | | |
|---|------------------------|----------------|----------------------------|
| 0 | Collection Object Name | Dimension Name | The name of the dimension. |
|---|------------------------|----------------|----------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | Deletes the dimension successfully. |
| FAILURE | The dimension could not be found. |

Remarks

None.

9

REPORTING OPERATIONS

Set Report Tag Value From String

Sets the value of a tagged field on a report to a string value.

Input Arguments

| | | | |
|---|--------|-----------|---|
| 0 | String | Tag Name | The tag of the field to modify. |
| 1 | String | Tag Value | The string value to assign to the tagged field. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | The field tag was not found. |

Remarks

None.

Set Report Tag Value From Integer

Sets the value of a tagged field on a report to an integer value.

Input Arguments

| | | | |
|---|---------|-----------|--|
| 0 | String | Tag Name | The tag of the field to modify. |
| 1 | Integer | Tag Value | The integer value to assign to the tagged field. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | The field tag was not found. |

Remarks

None.

Set Report Tag Value From Double

Sets the value of a tagged field on a report to a double value.

Input Arguments

| | | | |
|---|--------|-----------|---|
| 0 | String | Tag Name | The tag of the field to modify. |
| 1 | Double | Tag Value | The double value to assign to the tagged field. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | The field tag was not found. |

Remarks

None.

Get Report Tag Value

Obtains the value of a tagged field on a report.

Input Arguments

| | | | |
|---|--------|----------|--|
| 0 | String | Tag Name | The tag of the field containing the value to retrieve. |
|---|--------|----------|--|

Return Arguments

| | | | |
|---|--------|----------------------|---|
| 1 | String | Tag Value As String | The value of the field expressed as a string. |
| 2 | String | Tag Value As Integer | The value of the field expressed as an integer. |
| 3 | String | Tag Value As Double | The value of the field expressed as a double. |

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was obtained successfully. |
| FAILURE | The field tag was not found. |

Remarks

If the field contains a string, the integer and double return arguments will contain zeroes.

Get Defined Report Tags

Obtains a list of the report tags defined in a job file.

Input Arguments

None.

Return Arguments

| | | | |
|---|-----------------|--------------|---|
| 1 | String Ref List | Defined Tags | Returns a list of the report tag strings defined in the job file. |
|---|-----------------|--------------|---|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was obtained successfully. |
| FAILURE | This command always succeeds. |

Remarks

The returned String list will include both the default system tags and those defined by the user.

Remove Report Tag

Removed a defined report tag from the job file.

Input Arguments

| | | | |
|---|--------|----------|---------------------------------------|
| 0 | String | Tag Name | The name of the report tag to remove. |
|---|--------|----------|---------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was removed successfully. |
| FAILURE | The report tag could not be removed. |

Remarks

Using Get Defined Report Tags will return a list of tags on the system. Some of the tags such as page number are system tags that cannot be removed. These include:

<<Page #>>

<<Date/Time>>

<<Filename>>

<<Filename Short>>

<<SA Version>>

<<Units>>

<<Working Frame>>

Set Report Options for Object

Opens the report options dialog for the specified object.

Input Arguments

| | | | |
|---|------------------------|--------|--|
| 0 | Collection Object Name | Object | The object for which the report options should be displayed. |
|---|------------------------|--------|--|

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The report options were opened successfully. |
| PARTIAL SUCCESS | Report options are not available for the specified object. |
| FAILURE | The object was not found. |

Remarks

None.

Set Vector Group Report Options

Sets the report options for a vector group.

Input Arguments

| | | | |
|---|------------------------|----------------|--|
| 0 | Collection Object Name | Vector Group | The name of the vector group to modify. |
| 1 | Report Output Options | Report Options | The standard vector group report options to set. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The report options were set successfully. |
| FAILURE | The vector group was not found. |

Remarks

None.

Set Relationship Report Options

Sets the report options for a relationship of type Group to Group, Points to Objects, Clouds to Objects, and Point to Point.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Relationship Name | The name of the relationship to modify. |
| 1 | Report Output Options | Report Options | The standard relationship report options to set. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The report options were set successfully. |
| FAILURE | The relationship was not found. |

Remarks

None.

Set Point Delta Report Options

Creates a set of report options that can be assigned to a relationship or vector group via Set Vector Group Report Options or Set Relationship Report Options.

Input Arguments

| | | | |
|----|----------------------------|-------------------------------|--|
| 1 | Coordinate System Type | Coordinate System | The type of coordinate system to use (Cartesian, Cylindric, or Polar[sic]). |
| 2 | Boolean | Summary Table? | Indicates whether the summary table should be displayed. |
| 3 | Report Details Line Format | Details Table | The line format for the report. |
| 4 | Boolean | Point A? | Indicates whether Point A should be displayed. |
| 5 | Boolean | Point B? | Indicates whether Point B should be displayed. |
| 6 | Boolean | Delta? | Indicates whether the delta should be displayed. |
| 7 | Boolean | Mag? | Indicates whether the magnitude should be displayed. |
| 8 | Boolean | Component 1? | Indicates whether the first component value (X or R) should be displayed. |
| 9 | Boolean | Component 2? | Indicates whether the second component value (Y or θ) should be displayed. |
| 10 | Boolean | Component 3? | Indicates whether the third component value (Z or Φ) should be displayed. |
| 11 | Boolean | Show Tolerance Fields? | Indicates whether tolerance fields should be shown. |
| 12 | Boolean | Colorize In Tolerance Fields? | Indicates whether in-tolerance fields should be colored green. |
| 13 | Boolean | Sort by Point Names? | Indicates whether points should be sorted by name. |

Return Arguments

| | | | |
|---|-----------------------|----------------|-------------------------------|
| 0 | Report Output Options | Report Options | The resulting report options. |
|---|-----------------------|----------------|-------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Point Group Report Options

Sets the reporting options for a point group.

Input Arguments

| 0 | Collection Object Name | Point Group | Point Group to modify |
|----|------------------------|--------------------------------|---|
| 1 | Coordinate System Type | Coordinate System | Pick from Cartesian, Cylindrical or Polar |
| 2 | Boolean | Show X Component | True - display X |
| 3 | Boolean | Show Y Component | True - display Y |
| 4 | Boolean | Show Z Component | True - display Z |
| 5 | Boolean | Show Offsets | True - display offsets |
| 6 | Boolean | Show Uncertainty | True- display uncertainty |
| 7 | Boolean | Show Notes | True - display notes |
| 8 | Boolean | Show Measurements | True - display measurements |
| 9 | Boolean | Show Measurement Details | True - display measurement details |
| 10 | Boolean | Show PointingError/Worst Angle | True- display pointing error / worst angle |
| 11 | Boolean | Sort by Point Names | True - sorts group by point name |
| 12 | Boolean | Make Default | True- Sets these settings as default |
| 13 | Boolean | Apply to All | True - applies these settings to all existing point groups. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | Settings were applied successfully |
| FAILURE | The Point Group could not be found. |

Remarks

None.

Set Scale for Picture

Sets a scale factor associated with a specified picture so that it appears with a desired size when added to a report.

Input Arguments

| | | | |
|---|-------------------------|--------------|---------------------------------------|
| 0 | Collection Picture Name | Picture Name | The name of the picture to modify. |
| 1 | Double | Scale | The new scale factor for the picture. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The scale factor was set successfully. |
| FAILURE | The picture was not found. |

Remarks

None.

Rename Picture

Renames a picture.

Input Arguments

| | | | |
|---|-------------------------|-----------------------|---|
| 0 | Collection Picture Name | Original Picture Name | The name of the picture to rename. |
| 1 | Collection Picture Name | New Picture Name | The new name for the picture. |
| 2 | Boolean | Overwrite if exists? | Indicates whether the existing picture should be replaced if the destination name already exists. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The picture was renamed successfully. |
| FAILURE | The picture was not found, or the destination name already exists and Argument 2 was set to FALSE. |

Remarks

None.

Delete Picture

Deletes a picture.

Input Arguments

| | | | |
|---|-------------------------|--------------|------------------------------------|
| 0 | Collection Picture Name | Picture Name | The name of the picture to delete. |
|---|-------------------------|--------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The picture was deleted successfully. |
| FAILURE | The picture was not found. |

Remarks

None.

Combine SA Reports

Combines separate SA reports into a single report.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--|
| 0 | SA Report Ref List | SA Reports to Combine | The list of SA reports to combine. |
| 1 | Collection Object Name | Output SA Report Name | The name for the resulting SA report. |
| 2 | Boolean | Show Report? | Indicates whether the new combined report should be displayed. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command does not verify that the source SA reports exist.

If the destination SA Report already exists, a new one will be created with an asterisk appended to the name (to make the name unique).

Quick Report

Generates a Quick Report for the specified object.

Input Arguments

| | | | |
|---|------------------------|------------------------|--|
| 0 | Collection Object Name | Item Name | The item to report. |
| 1 | String | Report Name (optional) | The name for the quick report. |
| 2 | Boolean | Open Report? | Indicates whether the quick report should be opened or just added to the tree. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The report was generated successfully. |
| FAILURE | The item was not found or does not support Quick Reports. |

Remarks

None.

Close All Reports

Closes all open report windows.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Define Report Template

Creates an SA Report Template.

Input Arguments

| | | | |
|---|---------------------------------|---------------------------------------|---|
| 0 | Collection Object Name | Report Template Name | The name for the SA Report Template. |
| 1 | String Ref List | Title | A list of strings for the report title. Each string item appears on its own line in the report. |
| 2 | Report View Options | Graphical View Options | Specifies whether a view will be included on the report. |
| 3 | Collection Object Name Ref List | Items to Report | A list of objects, tables, and/or feature checks to include in the report. |
| 4 | Relationship Ref List | Relationships To Report | A list of desired relationships to include in the report. |
| 5 | Event Ref List | Events To Report | A list of events to include in the report. |
| 6 | Report Output Options | Report Output Options | Specifies the options and formatting of the resulting report. |
| 7 | Report Page Settings | Report Page Settings (SA Report Only) | The page settings for the report. |
| 8 | Boolean | Generate Now? | Indicates whether the report should be immediately generated or not. |
| 9 | Boolean | Show Generated Report? | Indicates whether the report should be displayed if argument 8 is set to true. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make New SA Report

Creates a new blank SA report.

Input Arguments

| | | | |
|---|------------------------|-------------------------------|--|
| 0 | Collection Object Name | New SA Report Name | The name for the new SA report. |
| 1 | Collection Object Name | SA Report Template (optional) | The name of a template from which to generate the new SA report. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Append Items to SA Report

Adds items (objects, relationships, tables, pictures, etc) to an existing SA report.

Input Arguments

| | | | |
|---|---------------------------------|-----------------|---|
| 0 | Collection Object Name | Report Name | The name of the existing SA report. |
| 1 | Collection Object Name Ref List | Items To Report | The list of items to add to the report. |
| 2 | Boolean | Show Report? | Indicates whether the report should be displayed after the items are added. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one of the supplied items was successfully added to the report. |
| FAILURE | The report was not found, or none of the specified items could be found. |

Remarks

None.

Add Item to SA Report at Location

Adds an item to an existing SA report at a specific position on a specific page.

Input Arguments

| | | | |
|---|------------------------|---------------------|---|
| 0 | Collection Object Name | Report Name | The name of the existing SA report. |
| 1 | Collection Object Name | Item Name | The item to add to the report. |
| 2 | Integer | Page Number | The destination page number for the item. This is a 1-based value, so the first page is page 1. |
| 3 | Double | Horizontal Location | The distance (in page units) from the left end of the printable page. |
| 4 | Double | Vertical Location | The distance (in page units) from the top of the printable page. |
| 5 | Boolean | Show Report? | If TRUE, the report window will be displayed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The item was added successfully. |
| FAILURE | The report or item was not found, or the location was invalid. |

Remarks

None.

Output SA Report to PDF

Exports the specified SA report to a PDF file.

Input Arguments

| | | | |
|---|----------------------------|-------------|--|
| 0 | Collection Object Name | Report Name | The name of the existing SA report. |
| 1 | File Path or Embedded File | File Name | The name of the resulting PDF file to create. |
| 2 | Boolean | Show PDF? | Indicates whether the PDF should be opened upon successful creation. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was successfully exported to a PDF file. |
| FAILURE | The report was not found, the filename was not valid, or another error occurred. |

Remarks

None.

Output SA Report to Excel

Exports the specified SA report to an Excel file.

Input Arguments

| | | | |
|---|----------------------------|-------------|--|
| 0 | Collection Object Name | Report Name | The name of the existing SA report. |
| 1 | File Path or Embedded File | File Name | The name of the resulting Excel file to create. |
| 2 | Boolean | Show File? | Indicates whether the Excel spreadsheet should be opened upon successful creation. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was successfully exported to an Excel file. |
| FAILURE | The report was not found, the filename was not valid, or another error occurred. |

Remarks

None.

Delete SA Report Template

Deletes an SA Report Template from the tree.

Input Arguments

| | | | |
|---|------------------------|----------------------|---|
| 0 | Collection Object Name | Report Template Name | The name of the SA Report Template to delete. |
|---|------------------------|----------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The template was deleted successfully. |
| FAILURE | The template was not found. |

Remarks

None.

Delete SA Report

Deletes an SA Report from the tree.

Input Arguments

| | | | |
|---|------------------------|-------------|--------------------------------------|
| 0 | Collection Object Name | Report Name | The name of the SA Report to delete. |
|---|------------------------|-------------|--------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The report was deleted successfully. |
| FAILURE | The report was not found. |

Remarks

None.

Delete SA Doc

Deletes an SA Doc from the tree.

Input Arguments

| | | | |
|---|------------------------|----------|-----------------------------------|
| 0 | Collection Object Name | Doc Name | The name of the SA Doc to delete. |
|---|------------------------|----------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The SA Doc was deleted successfully. |
| FAILURE | The SA Doc was not found. |

Remarks

None.

Generate/Update Templated Report

Generates or updates a report from an SA Report Template.

Input Arguments

| | | | |
|---|------------------------|-----------------|---|
| 0 | Collection Object Name | Report Template | The name of the SA Report Template to use for generating/updating the report. |
|---|------------------------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was generated/updated successfully. |
| FAILURE | The report template was not found. |

Remarks

None.

Make Report Graphical View Options

Makes an item of MP type “Report View Options” that can be used later when specifying view options for a report.

Input Arguments

| | | | |
|---|---------|------------------------------------|--|
| 0 | Boolean | No View? | Indicates whether the graphical view should be visible at all. |
| 1 | Boolean | Use Current View? | Indicates whether the current view should be used. |
| 2 | Boolean | Use Callout View? | Indicates whether a callout view should be used. |
| 3 | String | Callout Collection Name (optional) | The name of the callout view's collection to use (if specified). |
| 4 | String | Callout View Name (optional) | The name of the callout view to use (if specified). |

Return Arguments

| | | | |
|---|---------------------|----------------------------------|---|
| 5 | Report View Options | Resulting Graphical View Options | The resulting Report View Options item. |
|---|---------------------|----------------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Make Report Output Options

Makes an item of MP type “Report Output Options” that can be used later when specifying output options for a report.

Input Arguments

| | | | |
|---|----------------------------|--|---|
| 0 | Boolean | No Output? | Indicates whether the report should be output at all. |
| 1 | Boolean | Output to Embedded SA Dynamic Report? | Indicates whether a report should be sent to an embedded SA Dynamic Report. |
| 2 | Boolean | Output to Embedded SA Doc (rtf)? | Indicates whether a report should be sent to an embedded SA Doc. |
| 3 | String | Embedded File Collection Name (optional) | If specified, the collection into which to place the embedded report. |
| 4 | String | Embedded File Name (optional) | If specified, the filename to use for the embedded report. |
| 5 | Boolean | Output to External PDF File | Indicates whether a report should be sent to an external PDF file. |
| 6 | Boolean | Output to External RTF file | Indicates whether a report should be sent to an external RTF file. |
| 7 | File Path or Embedded File | Output File Name (optional) | If specified, the filename to use for the external report. |

Return Arguments

| | | | |
|---|-----------------------|---------------------------------|---|
| 8 | Report Output Options | Resulting Report Output Options | The resulting Report Output Options item. |
|---|-----------------------|---------------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Insert Page Break

Adds a page break to an MS Office Report.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | A page break was successfully added. |
| FAILURE | An MS Office Report is not currently open. |

Remarks

None.

Generate Custom HTML Report

Generates an HTML-formatted report, replacing specified keywords in a template file with specified values.

Input Arguments

| | | | |
|---|-------------------------------|--------------------------|--|
| 0 | File Path or Embedded File | HTML Template File | The path to the template HTML file to use for the report. |
| 1 | File Path or Embedded File | HTML Output File | The path for the final HTML report. |
| 2 | HTML Keyword Association List | Keyword Association List | A list of symbol-value pairs. When symbols appear in the HTML template, they will be replaced by the values in the final report. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was successfully generated. |
| FAILURE | The template file could not be found. |

Remarks

Create a template by generating any standard HTML-formatted file. Then add symbols to the HTML file for later replacement.

Generate Standard HTML Report

Generates a standard format HTML report for everything in the file.

Input Arguments

| | | | |
|---|----------------------------|-------------------|---------------------------------------|
| 0 | File Path or Embedded File | HTML Output File | The path for the final HTML report. |
| 1 | Integer | Decimal Precision | The decimal precision for the report. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Capture Screen to File (BMP/JPG/PNG/GIF/TIFF)

Captures a screenshot and saves it to an image file.

Input Arguments

| | | | |
|---|----------------------------|-----------------|-------------------------------------|
| 0 | File Path or Embedded File | File to save to | The path for the output image file. |
|---|----------------------------|-----------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Capture Current View

Takes a snapshot of the current primary graphical view and saves it to the tree.

Input Arguments

| | | | |
|---|-------------------------|--------------|-------------------------------------|
| 0 | Collection Picture Name | Picture Name | The name for the resulting picture. |
|---|-------------------------|--------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Save Current View (BMP/JPG/PNG/GIF/TIFF)

Saves the graphical view to an image file.

Input Arguments

| 0 | File Path or Embedded File | File to save to | The path for the output image file. |
|---|----------------------------|--|---|
| 1 | Double | Render Scale Factor (1.0 uses window size) | A scale for the rendered image size. A value of 2.0 creates a rendered image at twice the resolution of the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Create Chart from Vector Group

Creates a Bullseye, Run, or Moving Range chart from a vector group.

Input Arguments

| | | | |
|---|-------------------|--------------------------------|---|
| 0 | Chart Name | New Chart Name | The name for the new chart. |
| 1 | Vector Group Name | Vector Group Name | The name of the vector group from which to create the chart. |
| 2 | Chart Type | Chart Type | The type of chart (Bullseye, Run, or Moving Range). |
| 3 | Vector Component | Data Set to Chart | The component (X, Y, Z, or Mag) to graph on the primary axis. |
| 4 | Vector Component | Aux Data Set to Chart | The component (X, Y, Z, or Mag) to graph on the secondary axis. |
| 5 | Chart Name | Template Chart Name (optional) | The name of the chart template to use (if applicable). |
| 6 | Boolean | Show Interface? | Indicates whether the chart interface should be displayed to the user during execution. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The chart was created successfully. |
| FAILURE | The specified vector group was not found. |

Remarks

None.

Save Chart to JPEG File

Saves a chart to an image file.

Input Arguments

| | | | |
|---|----------------------------|-----------------|--|
| 0 | Chart Name | Chart to Save | The chart to save to an image file. |
| 1 | File Path or Embedded File | File to save to | The name of the image file to save to. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The chart image was saved successfully. |
| FAILURE | The specified chart was not found. |

Remarks

None.

Delete Chart

Deletes an Chart from the tree.

Input Arguments

| | | | |
|---|------------------------|------------|----------------------------------|
| 0 | Collection Object Name | Chart Name | The name of the Chart to delete. |
|---|------------------------|------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The Chart was deleted successfully. |
| FAILURE | The Chart was not found. |

Remarks

None.

Make Utility Chart

Creates a custom chart (Figure 9-1) which contains a custom polygonal zone. A point can be plotted, and the command will indicate if the point is inside the zone. The resulting chart is stored as an image in the tree.

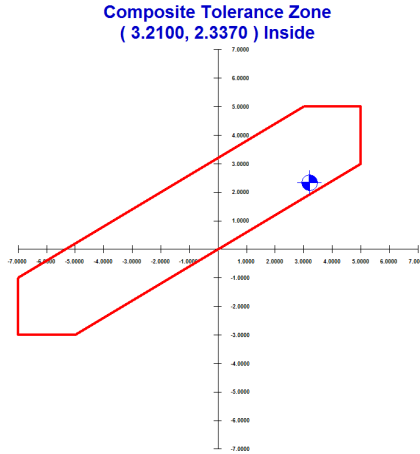


Figure 9-1. A sample Utility Chart.

Input Arguments

| | | | |
|---|----------------------------|---------------------------|---|
| 0 | File Path or Embedded File | ASCII File Path | The path to the ASCII configuration file containing the chart parameters (see Remarks below). |
| 1 | String | Chart Title Override | A title for the chart that overrides the title provided in the ASCII configuration file. |
| 2 | Collection Picture Name | Output Picture Name | The name for the resulting image for the chart to be created. |
| 3 | Boolean | Show Chart Dialog? | Indicates whether the chart is displayed once created. |
| 4 | Boolean | Plot Additional XY Value? | Indicates whether an additional XY value provided in the arguments should be plotted on the chart (instead of the point defined in the ASCII configuration file). |
| 5 | Double | X Value | The X value for the additional value to plot. |
| 6 | Double | Y Value | The Y value for the additional value to plot. |

Return Arguments

| | | | |
|---|---------|------------------|---|
| 7 | Boolean | Is Point Inside? | Indicates whether the plotted point is inside the perimeter provided in the ASCII configuration file. |
|---|---------|------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The ASCII configuration file should be formatted as indicated below. The following file generated the chart that appears in Figure 9-1.

```
<SetColor>,,
255,0,0
''
<SetLineWeight>,,
4,,
''
''
<Polygon>,,
3,5,
5,5,
5,3,
-5,-3,
-7,-3,
-7,-1,
''
<SetColor>,,
0,0,255
''
<SetLineWeight>,,
1,,
''
<Point>,,
3.21,1.227,
''
<Title>,,
Composite Tolerance Zone
```

Notify User Integer

Displays an integer and optional leading text to the user.

Input Arguments

| | | | |
|---|-----------|-----------------|---|
| 0 | String | Leading Text | The optional leading text to display in front of the reported integer value. |
| 1 | Font Type | Font | The font in which to display the text. |
| 2 | Integer | Reported Value | The integer to report to the user. |
| 3 | Integer | Display Timeout | If set to a number >0 the dialog will close automatically after the specified number of seconds |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Notify User Double

Displays a double and optional leading text to the user.

Input Arguments

| | | | |
|---|-----------|-------------------|---|
| 0 | String | Leading Text | The optional leading text to display in front of the reported double value. |
| 1 | Font Type | Font | The font in which to display the text. |
| 2 | Double | Reported Value | The double value to report to the user. |
| 3 | Integer | Decimal Precision | The number of decimal places to use to report the double to the user. |
| 3 | Integer | Display Timeout | If set to a number >0 the dialog will close automatically after the specified number of seconds |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Notify User Text Array

Displays multi-line text to the user.

Input Arguments

| | | | |
|---|-----------|--------------------------|---|
| 0 | String | Notification Text | The text to display to the user. |
| 1 | Font Type | Font | The font in which to display the text. |
| 2 | Boolean | Auto expand to fit text? | True automatically increase the size of the display dialog based upon text size. |
| 3 | Integer | Display Timeout | If set to a number >0 the dialog will close automatically after the specified number of seconds |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The Auto Expand to fit function (A2) is helpful if you have a large amount of text or large font sizes and want to make sure that an operator does not have to touch the screen to adjust the dialog display. Otherwise a scroll bar may appear and text may be directly visible.

Notify User HTML

Displays a window containing HTML content to the user.

Input Arguments

| 0 | File Path or Embedded File | HTML File | The HTML file to display to the user. |
|---|----------------------------|---|---|
| 1 | Integer | Step to jump to if Canceled (-1 will fail Step on Cancel) | The step to jump to if the user cancels the dialog. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

It is possible to enter an HTTP link as the HTML file--in which case the user can navigate a web page as if in a web browser.

HTML Display Board

Displays an HTML page that can remain open during an MP operation and be updated to display status changes as needed.

Input Arguments

| | | | |
|---|----------------------------|------------|---------------------------------------|
| 0 | File Path or Embedded File | HTML File | The HTML file to display to the user. |
| 1 | Boolean | Show Board | True opens the specified HTML file |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

There is also a *Close HTML Display Board* which can be used to close the HTML page dynamically. However, in a tight loop the *Show Board* argument should be used to reduce system resource use.

Close HTML Display Board

Closes the current HTML Display Board, if open.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

There is also a *Close HTML Display Board* which can be used to close the HTML page dynamically. However, in a tight loop the *Show Board* argument should be used to reduce system resource use.

Report Bar

Add Objects to Report Bar

Adds the specified objects (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|---------------------------------|-----------------|--|
| 0 | Collection Object Name Ref List | Object(s) | The list of objects (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The objects were added to the Report Bar successfully. |
| PARTIAL SUCCESS | At least one (but not all) objects were not found. |
| FAILURE | The specified objects were not found. |

Remarks

None.

Add Callout Views to Report Bar

Adds the specified callout views (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|-----------------------|-----------------|--|
| 0 | Callout View Ref List | Callout View(s) | The list of callout views (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The callout views were added to the Report Bar successfully. |
| PARTIAL SUCCESS | At least one (but not all) callout views were not found. |
| FAILURE | The specified callout views were not found. |

Remarks

None.

Add Charts to Report Bar

Adds the specified charts (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|----------------|-----------------|--|
| 0 | Chart Ref List | Chart(s) | The list of charts (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The charts were added to the Report Bar successfully. |
| PARTIAL SUCCESS | At least one (but not all) charts were not found. |
| FAILURE | The specified charts were not found. |

Remarks

None.

Add Datums to Report Bar

Adds the specified datums (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|----------------|-----------------|--|
| 0 | Datum Ref List | Datum(s) | The list of datums (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The datums were added to the Report Bar successfully. |
| PARTIAL SUCCESS | At least one (but not all) datums were not found. |
| FAILURE | The specified datums were not found. |

Remarks

None.

Add Dimensions to Report Bar

Adds the specified dimensions (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|--------------------|-----------------|--|
| 0 | Dimension Ref List | Dimension(s) | The list of dimensions (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The dimensions were added to the Report Bar successfully. |
| PARTIAL SUCCESS | At least one (but not all) dimensions were not found. |
| FAILURE | The specified dimensions were not found. |

Remarks

None.

Add Events to Report Bar

Adds the specified events (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|----------------|-----------------|--|
| 0 | Event Ref List | Event(s) | The list of events (in order) to add to the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one event was added to the Report Bar successfully. |
| FAILURE | No specified events were found. |

Remarks

None.

Add Feature Checks to Report Bar

Adds the specified feature checks (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|------------------------|------------------|--|
| 0 | Feature Check Ref List | Feature Check(s) | The list of feature checks (in order) to add to the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one feature check was added to the Report Bar successfully. |
| FAILURE | No specified feature check was found. |

Remarks

None.

Add Pictures to Report Bar

Adds the specified pictures (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|----------------------------------|-----------------|--|
| 0 | Collection Picture Name Ref List | Picture(s) | The list of pictures (in order) to add as tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one picture was added to the Report Bar successfully. |
| FAILURE | No specified pictures were found. |

Remarks

None.

Add Relationships to Report Bar

Adds the specified relationships (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|------------------------|-----------------|--|
| 0 | Relationships Ref List | Relationship(s) | The list of relationships (in order) to add to the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | At least one relationship was added to the Report Bar successfully. |
| FAILURE | No specified relationships were found. |

Remarks

None.

Add Scale Bars to Report Bar

Adds the specified scale bars (in order) as tabs in the Report Bar.

Input Arguments

| | | | |
|---|--------------------|-----------------|--|
| 0 | Scale Bar Ref List | Scale Bar(s) | The list of scale bars (in order) to add to the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one scale bar was added to the Report Bar successfully. |
| FAILURE | No specified scale bars were found. |

Remarks

None.

Add Custom Tables to Report Bar

Adds one or more custom tables to the report bar, with each table appearing in a separate tab.

Input Arguments

| | | | |
|---|------------------------------|---------------------------|--|
| 0 | Custom Report Table Ref List | Custom Table(s) To Report | The list of tables to add as separate tabs in the Report Bar. |
| 1 | Boolean | Clear Existing? | Specifies whether any existing tabs should be cleared from the Report Bar first. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | At least one table was added to the Report Bar successfully. |
| FAILURE | No specified tables were found. |

Remarks

None.

Generate Quick Report from Tab Order

Generates a Quick Report from the current Report Bar tab order.

Input Arguments

| | | | |
|---|---------|------------------------|---|
| 0 | String | Report Name (optional) | The name for the generated report. |
| 1 | Boolean | Open Report? | Specifies whether the generated report should be displayed upon creation. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If a report name is not specified, the report will be created with an empty name in the tree.

Set Report Bar Visibility

Sets the visibility state for the Report Bar.

Input Arguments

| | | | |
|---|---------|------------------|---|
| 0 | Boolean | Show Report Bar? | Specifies whether the Report Bar should be displayed in the SA interface. |
|---|---------|------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Refresh Report Bar

Refreshes the information displayed in the Report Bar.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Custom Report Tables

Make Custom Table

Creates a custom table in the tree for later addition to a report.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Table Name | The name of the table to create. |
| 1 | Integer | Decimal Precision | The decimal precision to use in the table. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the specified table already exists, a unique name will be used for the table by adding asterisks as necessary. Argument 0 will then hold the new unique name for the table.

Clear Custom Table

Clears a custom table and removes all rows and columns.

Input Arguments

| | | | |
|---|------------------------|------------|---------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to clear. |
|---|------------------------|------------|---------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The table was cleared successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Delete Custom Table

Deletes a custom table.

Input Arguments

| | | | |
|---|------------------------|------------|----------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to delete. |
|---|------------------------|------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The table was deleted successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Get Custom Table Cell String

Retrieves a string from the specified table cell.

Input Arguments

| | | | |
|---|------------------------|------------|-------------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to read from. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |

Return Arguments

| | | | |
|---|--------|-------|--|
| 3 | String | Value | The string stored at the specified cell. |
|---|--------|-------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The string was retrieved successfully. |
| FAILURE | The specified table or cell was not found. |

Remarks

None.

Get Custom Table Cell Double

Retrieves a double from the specified table cell.

Input Arguments

| | | | |
|---|------------------------|------------|-------------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to read from. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |

Return Arguments

| | | | |
|---|--------|-------|--|
| 3 | Double | Value | The double stored at the specified cell. |
|---|--------|-------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The double was retrieved successfully. |
| FAILURE | The specified table or cell was not found. |

Remarks

None.

Set Custom Table Title

Sets the title for a custom table.

Input Arguments

| | | | |
|---|------------------------|--------------|--|
| 0 | Collection Object Name | Table Name | The name of the table whose title should be set. |
| 1 | String | Title Line 1 | The first line for the title. |
| 2 | String | Title Line 2 | The second line for the title (optional). |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The title was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Set Custom Table Header Row

Sets a header row for a custom table that spans the full width of the table.

Input Arguments

| | | | |
|---|------------------------|-------------|---|
| 0 | Collection Object Name | Table Name | The name of the table to which a header cell should be added. |
| 1 | Integer | Row | The row for the header cell. |
| 3 | String | Header Text | The text to place in the header cell. |
| 4 | Alignment Type | Alignment | The alignment for the text in the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

The header will be repeated on successive pages of the table when viewed in an SA Report.

Multiple consecutive rows of headers are allowed.

If a row of table headers is added to a table that is not contiguous with a previous header, that row becomes the new header for the remainder of the table.

Set Custom Table Header Cell

Sets a header for one or more consecutive columns in a table.

Input Arguments

| | | | |
|---|------------------------|-------------|---|
| 0 | Collection Object Name | Table Name | The name of the table to which a header cell should be added. |
| 1 | Integer | Row | The row for the header cell. |
| 2 | Integer | Column | The column for the header cell. |
| 3 | String | Header Text | The text to place in the header cell. |
| 4 | Alignment Type | Alignment | The alignment for the text in the cell. |
| 5 | Integer | Span | The number of columns the cell should span. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

Unlike Set Custom Table Header Row, this command is intended to be used to define the header for a single column and the width of the text is considered in determining the resulting width of the column. The step can also be used to span the entire width of the table. To do so enter (-1) in the A5 (Span).

The header will be repeated on successive pages of the table when viewed in an SA Report.

Multiple consecutive rows of headers are allowed.

If a row of table headers is added to a table that is not contiguous with a previous header, that row becomes the new header for the remainder of the table.

Set Custom Table Cell String

Places the specified string into a custom table.

Input Arguments

| | | | |
|---|------------------------|------------|--|
| 0 | Collection Object Name | Table Name | The name of the table to which the cell should be added. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |
| 3 | String | Value | The text to place in the cell. |
| 4 | Alignment Type | Alignment | The alignment for the text in the cell. |
| 5 | Integer | Span | The number of columns the cell should span. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Set Custom Table Cell Double

Places the specified double into a custom table.

Input Arguments

| | | | |
|---|------------------------|-------------------|--|
| 0 | Collection Object Name | Table Name | The name of the table to which the cell should be added. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |
| 3 | Double | Value | The number to place in the cell. |
| 4 | Alignment Type | Alignment | The alignment for the value in the cell. |
| 5 | Integer | Span | The number of columns the cell should span. |
| 6 | Integer | Decimal Precision | The number of decimal places in which the number should be reported. A value of -1 indicates that the value should inherit the decimal precision of the enclosing table. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Set Custom Table Cell Color

Sets a cell's foreground and background color in a custom table.

Input Arguments

| | | | |
|---|------------------------|-----------------------|--------------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to modify. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |
| 3 | Color | Foreground Color Name | The color for the text. |
| 4 | Color | Background Color Name | The color for the cell's background. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table was not found. |

Remarks

None.

Set Custom Table Cell Font

Sets the font to be used in a table cell.

Input Arguments

| | | | |
|---|------------------------|------------|----------------------------------|
| 0 | Collection Object Name | Table Name | The name of the table to modify. |
| 1 | Integer | Row | The row for the cell. |
| 2 | Integer | Column | The column for the cell. |
| 3 | Font Type | Font | The font to use in the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The cell was set successfully. |
| FAILURE | The specified table or cell was not found. |

Remarks

None.

Add Custom Table to SA Report

Adds a custom table to an existing SA report.

Input Arguments

| | | | |
|---|------------------------|--------------|--|
| 0 | Collection Object Name | Table Name | The name of the table to add. |
| 1 | Collection Object Name | Report Name | The name of the SA Report to which the table should be added. |
| 2 | Boolean | Show Report? | Determines whether the report should be displayed once the table is added. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The table was added successfully. |
| FAILURE | The specified table or report was not found. |

Remarks

None.

10 EXCEL DIRECT CONNECT

Open Workbook File

Opens a hidden Excel workbook (.XLS/.XLSX) for editing.

Input Arguments

| 0 | File Path or Embedded File | Workbook File Path | The path to the Excel workbook file to open. |
|---|----------------------------|---------------------|--|
| 1 | Boolean | Verify File Exists? | True will check if the file exists, if not a new file will be created. |

Return Arguments

| 2 | Integer | Workbook Handle | A handle to the opened workbook. |
|---|---------|-----------------|----------------------------------|
|---|---------|-----------------|----------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The workbook was opened successfully. |
| FAILURE | The workbook was not found, or Microsoft Excel is not installed on the system. |

Remarks

The workbook handle returned by this command (argument 1) should be referenced by other commands that use this workbook. Ensure that you close the workbook when finished working with it. Otherwise, a hidden instance of Microsoft Excel will continue executing in the background. (It can be terminated using the Windows Task Manager).

When opened, the workbook is stored in the active SA file. Executing a “New SA File” MP command after opening a workbook will cause changes to be lost and the reference to the workbook to become invalid.

Step Status Test can be used on the *Open Workbook File* step if the *Verify File Exists?* argument is TRUE. If FALSE, the command always succeeds. In either case the workbook will be opened. The difference is whether you are explicitly requiring a prior instance of the file to exist prior to opening it or if you are content to start a new workbook.

Embedded files can be worked with directly using the *Existing Embedded File* method or an embedded file can be referenced using the convention “<collection>::<File>” . When the file name is parsed (either direct entry or provided by reference), it will always parse the name and attempt to find the embedded file in the tree. If it is successful, then it will use the embedded file for specified operation – otherwise it will transition to a disk-based file.

Set Workbook Address

Sets the selected cell on an Excel spreadsheet and defines cursor behavior when reading from and writing to the spreadsheet.

Input Arguments

| | | | |
|----|----------------------------|---|--|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Workbook Address Mode Type | Addressing Mode | Indicates how you would like to indicate the new selected cell (see Remarks). |
| 2 | String | Absolute Position Worksheet Name | The name of the worksheet you would like to be active (applies to "Absolute Position" addressing mode only). |
| 3 | String | Absolute Position Column (A, B, C, ...) | The column containing the cell you would like to select (applies to "Absolute Position" addressing mode only). |
| 4 | Integer | Absolute Position Row (1, 2, 3, ...) | The row containing the cell you would like to select (applies to "Absolute Position" addressing mode only). |
| 5 | Move Direction Type | Relative Move Direction | The direction in which to move the cell cursor (relative to the currently-selected cell). Applies to the "Relative Move" addressing mode only. |
| 6 | Integer | Relative Move # Cells | The number of cells to move in the relative move direction, relative to the currently-selected cell). Applies to the "Relative Move" addressing mode only. |
| 7 | String | Named Cell/Range in Workbook | The name of a cell or range in a workbook to move to (applies to "Named Cell/Range in Workbook" addressing mode only). |
| 8 | Write Mode Type | Write Mode | Indicate whether you would like to insert new cells when writing, or overwrite the cell's contents (see Remarks). |
| 9 | Move Direction Type | Auto Move Direction | The direction that the cell cursor should automatically move after a cell is read or written to. |
| 10 | Integer | Auto Move # Cells | The number of cells the cursor should automatically move by after a cell is read or written to. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The cell was selected successfully. |
| FAILURE | The workbook handle was invalid, or the named cell/range was not found. |

Remarks

There are four addressing modes:

- No Position Change allows you to change settings (such as Auto Move Direction) without changing the active cell selection.
- Absolute Position allows you to specify a specific cell to select (for example, Sheet 1, cell C5).
- Relative Move allows you to move the “cell cursor” relative to the currently selected cell (for example, move 5 columns to the right of the currently selected cell).
- Named Cell/Range in Workbook allows you to move directly to a named cell or range in a workbook.

If a worksheet name is specified that does not already exist, a new worksheet with that name will be created.

Insert write mode shifts the current cell (and those below it) downward in order to add the data. *Overwrite* mode replaces the cell's existing contents.

Anytime data is read from a cell or written to a cell, an “auto move” (see argument 9) takes place. Therefore, after reading or writing to cell C8, the selected cell will be C9 (assuming the Move Direction Type is Down and the Auto Move # Cells argument is 1).

Get Workbook Address

Returns the selected cell, write mode, and auto move direction for an Excel spreadsheet.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

| | | | |
|---|---------------------|---|--|
| 1 | String | Absolute Position Worksheet Name | The name of the active worksheet. |
| 2 | String | Absolute Position Column (A, B, C, ...) | The currently selected column. |
| 3 | String | Absolute Position Row (1, 2, 3, ...) | The currently selected row. |
| 4 | Write Mode Type | Write Mode | The current write mode for the spreadsheet. |
| 5 | Move Direction Type | Auto Move Direction | The current auto-move direction. |
| 6 | Integer | Auto Move # Cells | The current number of cells being used with auto-move. |
| | | | |

Returned Status

| | |
|---------|--|
| SUCCESS | The information was returned successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Set Clear After Insert

Clears the formatting and other properties of a cell being written to when using the *Insert* write mode.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Boolean | Clear Comments | Indicates if comments should be cleared from the cell. |
| 2 | Boolean | Clear Formulas | Indicates if formulae should be cleared from the cell. |
| 3 | Boolean | Clear Formats | Indicates if formatting should be cleared from the cell. |
| 4 | Boolean | Clear Notes | Indicates if notes should be cleared from the cell. |
| 5 | Boolean | Clear Outline | Indicates if outlines should be cleared from the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The insert behavior was set appropriately. |
| FAILURE | The workbook handle was invalid. |

Remarks

When the write mode is set to *Insert*, the existing formats and other cell properties remain in the cell being written. Often, it is desired to clear out the existing formatting, formulas, and other properties of the cell such that the newly written cell is in a “default” configuration. This command makes that possible.

Run Macro

Runs a macro stored in an Excel workbook.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | String | Macro Name | The name of the macro to run. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The macro was run successfully. |
| FAILURE | The workbook handle was invalid, or the macro was not found. |

Remarks

The MP will pause until the macro completes.

Save

Saves the specified Excel workbook.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The workbook was saved successfully. |
| FAILURE | The workbook handle was invalid, or there was a file system error. |

Remarks

None.

Close

Closes (and optionally saves) the Excel workbook.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Boolean | Save? | Indicates whether the workbook should be saved prior to being closed. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The workbook was closed successfully. |
| FAILURE | The workbook handle was invalid, or there was a file system error. |

Remarks

None.

Write

Write Integer

Writes an integer to the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Integer | Data to Write | The integer to write to the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The cell was written to successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Write Double

Writes a double to the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Double | Data to Write | The double to write to the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The cell was written to successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Write String

Writes a string to the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | String | Data to Write | The string to write to the cell. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The cell was written to successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Write Variables

Writes one or more variables to a series of cells.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The workbook was written to successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

If the selected cell at the time that this command is executed has a defined name, and if there is a stored variable matching that name, its value will be written to the cell. The cursor will then automatically advance to the next cell and this process will repeat until the defined name for a cell no longer matches a variable name.

Write Picture

Places a picture stored in the SA tree into an Excel workbook.

Input Arguments

| | | | |
|---|-------------------------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
| 1 | Collection Picture Name | Picture Name | The picture (in the SA tree) to write to the workbook. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The picture was written successfully. |
| FAILURE | The workbook handle was invalid, or the specified picture was not found. |

Remarks

The picture will be placed at the current cursor location in the Excel workbook.

Read

Read Integer

Reads an integer from the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The cell was read successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Read Double

Reads a double from the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The cell was read successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Read String

Reads a string from the selected cell.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The cell was read successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

None.

Read Variables

Reads one or more variables from a series of cells.

Input Arguments

| | | | |
|---|---------|-----------------|---|
| 0 | Integer | Workbook Handle | The handle for the workbook returned from the Open Workbook File command. |
|---|---------|-----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The workbook was read to successfully. |
| FAILURE | The workbook handle was invalid. |

Remarks

If the selected cell at the time that this command is executed has a defined name, then a variable will be created matching that name, and its value will match that of the cell contents. The selected cell is then advanced according to the Auto Move Direction and the next value is read in, if it also has a defined name. Values are usually read in as string variables.

11 MS OFFICE REPORTING OPERATIONS

Initialize Office Report

Initializes an MS Office Report to prepare it for writing.

Input Arguments

| | | | |
|---|----------------------------|------------------------------|--|
| 0 | File Path or Embedded File | Report File Path | The path for the newly created MS Word file. |
| 1 | String | Title for Document | A title for the report. |
| 2 | File Path or Embedded File | Template File Path(optional) | The path to a template file (if desired). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was initialized successfully. |
| FAILURE | A template file was specified but not found. |

Remarks

Ensure that you close the report when finished working with it. Otherwise, a hidden instance of Microsoft Word will continue executing in the background. (It can be terminated using the Windows Task Manager).

Insert Section Break

Adds a section break to the current position in an MS Office Report.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The section break was added successfully. |
| FAILURE | An MS Office report is not currently active. |

Remarks

None.

Set Page Orientation

Sets the page orientation for an MS Office report to portrait or landscape.

Input Arguments

| | | | |
|---|---------|-----------|---|
| 0 | Boolean | Portrait? | Indicates whether the page orientation should be portrait or landscape. |
|---|---------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The page orientation was set successfully. |
| FAILURE | An MS Office report is not currently active. |

Remarks

None.

Add Section Heading to Report

Adds a section heading to an MS Office report.

Input Arguments

| | | | |
|---|---------------|--------------------------|--|
| 0 | String | Section Heading | The heading to use for the section. |
| 1 | Word Headings | Heading Level Designator | The heading level as defined in Microsoft Word. May be Heading 1, 2, 3, or 4. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The section heading was added successfully. |
| FAILURE | An MS Office report is not currently active. |

Remarks

None.

Add Objects to Report

Adds one or more object's reportable information to an MS Office report.

Input Arguments

| | | | |
|---|---------------------------------|--------------------------------------|-----------------------------------|
| 0 | Collection Object Name Ref List | Object Name List (Objects to Report) | The objects to add to the report. |
|---|---------------------------------|--------------------------------------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | All object information was added successfully. |
| PARTIAL SUCCESS | One or more (but not all) objects was not successfully added. |
| FAILURE | No objects were successfully added, or no MS Office report is active. |

Remarks

None.

Insert Graphics from file

Adds an image to an MS Office report.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------------|--|
| 0 | String | Caption for Figure | A text caption to place underneath the image. |
| 1 | Integer | Percentage of Page Width (10-100) | The percentage of the page width that should be occupied by the image. |
| 2 | File Path or Embedded File | File to Add | The path to the image file to add. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The image was added successfully. |
| FAILURE | The image was not found or an MS Office report is not active. |

Remarks

None.

Add Graphics View to Report

Adds an image of the current graphical view to an MS Office report.

Input Arguments

| | | | |
|---|---------|-----------------------------------|--|
| 0 | String | Caption for Figure | A text caption to place underneath the image. |
| 1 | Integer | Percentage of Page Width (10-100) | The percentage of the page width that should be occupied by the image. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The image was added successfully. |
| FAILURE | An MS Office report is not active. |

Remarks

None.

Add User Input Notes to Report

Displays a text dialog asking for user input, then adds the text to an MS Office report.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The notes were added successfully. |
| FAILURE | An MS Office report is not active, or the user hit the Cancelor Close button. |

Remarks

None.

Add Preset Notes to Report

Adds specified text to an MS Office report.

Input Arguments

| | | | |
|---|-----------|-------------|--------------------------------|
| 0 | Edit Text | Text to Add | The text to add to the report. |
|---|-----------|-------------|--------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The text was added successfully. |
| FAILURE | An MS Office report is not active. |

Remarks

None.

Make Report Table

Adds a custom table to an MS Office report.

Input Arguments

| | | | |
|---|--------------|-------|--|
| 0 | Report Table | Table | The parameters for the table to add (see remarks). |
|---|--------------|-------|--|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The table was added successfully. |
| FAILURE | An MS Office report is not active. |

Remarks

The report options dialog contains the following:

- **Caption.** Text to display underneath the table.
- **Use Column Width Ratios in Table.** The relative widths of the columns in the dialog will match the relative widths of the columns in the actual report table. Use the Row/Column spinners to define the number of rows and columns in the table, then right-click a cell to add text, a reference, or an image to a cell.

Close Office Report

Closes and saves an MS Office report.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The report was closed successfully. |
| FAILURE | An MS Office report is not active. |

Remarks

None.

Save Office Report as RTF

Saves an active MS Office report as a Rich Text File (RTF).

Input Arguments

| | | | |
|---|----------------------------|----------|--------------------------------------|
| 0 | File Path or Embedded File | RTF File | The path for the RTF file to create. |
|---|----------------------------|----------|--------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The report was saved successfully. |
| FAILURE | An MS Office report is not active or there was a file system problem. |

Remarks

None.

Add SADoc From File (RTF)

Adds an SADoc (RTF) file to an MS Office report.

Input Arguments

| | | | |
|---|----------------------------|----------|---|
| 0 | File Path or Embedded File | RTF File | The path for the RTF SADoc file to add to the report. |
|---|----------------------------|----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The report was added successfully. |
| FAILURE | An MS Office report is not active or the SADoc file was not found. |

Remarks

None.

12 INSTRUMENT OPERATIONS

Get Last Instrument Index

Returns the instrument index of the most recently added instrument.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Integer | Instrument ID | The instrument ID as an integer. |
| 1 | Collection Instrument ID | Instrument ID | The instrument's collection instrument ID. |

Returned Status

| | |
|---------|--|
| SUCCESS | The ID was obtained successfully. |
| FAILURE | No instruments were found in the file. |

Remarks

None.

Rename Instrument

Renames an instrument in the tree.

Input Arguments

| | | | |
|---|--------------------------|---------------|-------------------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the instrument to rename. |
| 1 | String | New Name | The new name for the instrument. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was renamed successfully. |
| FAILURE | The instrument was not found. |

Remarks

None.

Get Instrument ID from Name

Returns an instrument ID by name.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the instrument of interest. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|--------------------------|---------------|--|
| 1 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
|---|--------------------------|---------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument ID was obtained successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be located in the active collection.

Get Instrument Model

Retrieves the name of an instrument as displayed in the “Add Instrument” menu.

Input Arguments

| | | | |
|---|--------------------------|---------------|----------------------------|
| 0 | Collection Instrument ID | Instrument ID | The instrument to look up. |
|---|--------------------------|---------------|----------------------------|

Return Arguments

| | | | |
|---|--------|-------|---------------------------------------|
| 1 | String | Name | The name of the specified instrument. |
| 2 | String | Model | The model of the specified instrument |

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The name was obtained successfully. |
| FAILURE | The instrument was not found. |

Remarks

None.

Move Instrument to Another Collection

Moves an instrument to another collection.

Input Arguments

| | | | |
|---|--------------------------|-----------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to move. |
| 1 | Collection Name | Collection Name | The name of the collection to move the instrument into. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was moved successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be located in the active collection.

If the specified collection does not already exist, it will be created for you.

Save Instrument Configuration

Exports the configuration of an instrument interface to a file. For laser trackers, this consists of all of the measurement profile parameters, and is equivalent to clicking the Export button in the Manage Measurement Profiles window of the tracker interface.

Input Arguments

| | | | |
|---|----------------------------|--------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | File Path or Embedded File | Configuration File | A filename to use for the instrument's configuration file. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The configuration was saved successfully. |
| FAILURE | The instrument was not found or there was a file system error. |

Remarks

The instrument must be located in the active collection.

Load Instrument Configuration

Loads the configuration for an instrument interface, which for laser tracker consists of the custom measurement profile parameter settings. This is equivalent to clicking the Import button in the Manage Measurement Profiles window of the tracker interface.

Input Arguments

| | | | |
|---|----------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | File Path or Embedded File | Configuration File | The filename for the instrument's configuration file. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The configuration was loaded successfully. |
| FAILURE | The instrument or file was not found. |

Remarks

The instrument must be located in the active collection. Typically the loaded configuration file has a .MSP extension.

Point At Target

Points an instrument at the specified target. (Only applies for “pointable” instruments such as laser trackers, laser radars, etc.).

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Target ID | The name of the point to point at. |
| 2 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML file to use for prompted instructions, if desired. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument successfully pointed at the point. |
| FAILURE | The point was not found, or the instrument could not be pointed. |

Remarks

None.

Measure Single Point Here

Measures a point at the current probe position.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Target ID | The name of the point to measure. |
| 2 | Boolean | Measure Immediately | Indicate whether the measurement should be taken immediately, or whether the user should be prompted to take a measurement. |
| 3 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML prompt that can be displayed to the user before measurement. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument successfully measured the point. |
| FAILURE | The instrument could not be found, or the measurement failed. |

Remarks

The instrument must be located in the active collection.

Get Current Instrument Position Update

Returns the position of the last instrument update position received by SA.

Input Arguments

| | | | |
|---|--------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Export Frame Mode | Reporting Frame | Select between reference frames |
| 2 | Boolean | Polar Coordinates? | Choose the display method for the point coordinate, either polar or Cartesian |

Return Arguments

| | | | |
|---|--------|-------------------------|---|
| 3 | Double | X / R | Resulting X / R Value Position |
| 4 | Double | Y / Theta (Degrees) | Resulting Y / Theta Position |
| 5 | Double | Z / Phi (Degrees) | Resulting Z / Phi Position |
| 6 | Double | Time Since Update (Sec) | Time in seconds from the last received position updated from the instrument |
| 7 | String | TimeStamp (Approximate) | Approximate system date and time of last position update. |

Returned Status

| | |
|---------|--|
| SUCCESS | The instruments last point location was returned successfully. |
| FAILURE | The instrument could not be found, or no position was obtained |

Remarks

This command, like a watch window, will report the last reflector position received by SA. It does not trigger a measurement from the instrument. The time duration is an approximate time, roughly accurate to the second and should not be confused with the precise controller times recorded in the measurement details of some instruments. This is for relative reference only and was added with longer term remote monitoring applications in mind.

'Build' Target

Guides a user through measuring a point by displaying deviations from a nominal point.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Output Target Name | The name for the measured point. |
| 2 | Point Name | Nominal Point | The "nominal" point to compare against. |
| 3 | Vector Tolerance | Tolerance | Tolerance values for each component (dx, dy, dz, dmag). |
| 4 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML file that can be displayed to the user as a prompt. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The point was successfully measured in tolerance. |
| FAILURE | The point was not measured, or the tolerance was exceeded. |

Remarks

The instrument must be located in the active collection.

Measure Existing Single Point

Points an instrument at a point, locks onto a target, then measures a point. This only applies to instruments that can be pointed.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Existing Target ID | The name of the existing point to measure. |
| 2 | Collection Object Name | Group name for new point | The group into which to place the new point. |
| 3 | Boolean | Measure Immediately | Indicate whether the point should be measured immediately, or the user should have control to initiate measurement. |
| 4 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML file that can be displayed to the user as a prompt. |

Return Arguments

| | | | |
|---|------------|----------------------|---|
| 5 | Point Name | Resulting Point Name | The name of the resulting measured point. |
|---|------------|----------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The point was successfully measured. |
| FAILURE | The existing point could not be found, or the measurement was not successful. |

Remarks

The instrument must be located in the active collection.

The measured point will inherit the target name of the nominal point (but with a different group name).

Measure Existing Single Point (Manual Guide)

Points an instrument at a point, locks onto a target, then measures a point. If not measured immediately, the user is provided with controls for releasing the motors and steering the head toward the desired point. This only applies to instruments that can be pointed.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Existing Target ID | The name of the existing point to measure. |
| 2 | Collection Object Name | Group name for new point | The group into which to place the new point. |
| 3 | Boolean | Measure Immediately | Indicate whether the point should be measured immediately, or the user should have control to initiate measurement. |
| 4 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML file that can be displayed to the user as a prompt. |

Return Arguments

| | | | |
|---|------------|----------------------|---|
| 5 | Point Name | Resulting Point Name | The name of the resulting measured point. |
|---|------------|----------------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The point was successfully measured. |
| FAILURE | The existing point could not be found, or the measurement was not successful. |

Remarks

The instrument must be located in the active collection.

The measured point will inherit the target name of the nominal point (but with a different group name).

Measure Existing Single Point and Compare

Points an instrument at a point, locks onto a target, then measures a point. The deviation between the nominal point and the measured point is calculated automatically as return arguments.

Input Arguments

| | | | |
|---|----------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Existing Target ID | The name of the existing point to measure. |
| 2 | Collection Object Name | Group name for new point | The group into which to place the new point. |
| 3 | Boolean | Measure Immediately | Indicate whether the point should be measured immediately, or the user should have control to initiate measurement. |
| 4 | File Path or Embedded File | HTML Prompt File (optional) | An optional HTML file that can be displayed to the user as a prompt. |
| 5 | Double | Tolerance (0.0 for none) | A tolerance on the distance between the existing and measured points. |

Return Arguments

| | | | |
|----|------------|-----------------------|---|
| 6 | Vector | Vector Representation | The vector between the measured and existing points. |
| 7 | Double | X Value | The x deviation between the measured and existing points. |
| 8 | Double | Y Value | The y deviation between the measured and existing points. |
| 9 | Double | Z Value | The z deviation between the measured and existing points. |
| 10 | Double | Magnitude | The magnitude between the measured and existing points. |
| 11 | Point Name | Resulting Point Name | The name of the measured point. |

Returned Status

| | |
|---------|---|
| SUCCESS | The point was successfully measured and was in tolerance (if applicable). |
| FAILURE | The existing point could not be found, the measurement was not successful, or the tolerance was exceeded. |

Remarks

The instrument must be located in the active collection.

The measured point will inherit the target name of the nominal point (but with a different group name).

Stop Active Measurement Mode

Exits any active measurement mode.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
|---|--------------------------|---------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The measurement mode was stopped. |
| FAILURE | The instrument could not be found. |

Remarks

None.

Set Probe Offset Frame Online (Measure Raw Frame)

Enable an offset frame definition for a single or multi-face 6D probe. The will edit the offset frame saved directly with the probe, not any of the individual measurement profiles. This command will trigger a measurement of a raw frame and apply the current offset of the selected offset frame to the target definition.

Input Arguments

| | | | |
|---|--------------------------|--------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | Instrument to be edited |
| 1 | String | Probe Name | Name of the probe definition to edit |
| 2 | Integer | Face ID | Face ID to edit |
| 3 | String | Measurement Profile Name | Name of the Measurement Profile to use for the offset frame measurement. |
| 4 | Double | Timeout in Seconds | Maximum duration to wait for a measurement to be taken |
| 5 | Collection Object Name | Offset Frame | Selected offset frame which defines the transform to apply with respect to the raw measured frame. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The offset frame was applied |
| FAILURE | The instrument or target could not be found. |

Remarks

None.

Set Probe Offset Frame Offline (Select Previously Measured Frame)

Enables an offset frame definition for a single or multi-face 6D probe by selecting the measured and offset frames directly, without measuring. This will apply an offset frame to the target definition based upon their relative transform.

Input Arguments

| | | | |
|---|--------------------------|--------------------|--|
| 0 | Collection Instrument ID | Instrument ID | Instrument to be edited |
| 1 | String | Probe Name | Name of the probe definition to edit |
| 2 | Integer | Face ID | Face ID to edit |
| 3 | Collection Object Name | Raw Measured Frame | Selected Measured Raw frame defining the probes current position |
| 4 | Collection Object Name | Offset Frame | Selected offset frame which defines the transform to apply with respect to the raw measured frame. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The offset frame was applied |
| FAILURE | The instrument or target could not be found. |

Remarks

None.

Enable/Disable Frame Set Scan Mode

Sets SA data storage mode such that 6D Scans are recorded in Frame Sets rather than as individual frames.

Input Arguments

| | | | |
|---|---------|-----------------------|-------------------------------------|
| 0 | Boolean | Enable Frame Set Mode | True turns on Frame Set acquisition |
|---|---------|-----------------------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Enable/Disable Point Set Scan Mode

Sets SA data storage mode for the selected instrument such that point scans are recorded in Point Sets rather than as individual points within a point group.

Input Arguments

| | | | |
|---|--------------------------|-----------------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument to set in Point Set Scan Mode |
| 1 | Boolean | Enable Frame Set Mode | True turns on Frame Set acquisition |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
| FAILURE | The instrument was not found. |

Remarks

Point Sets or a more efficient data structure than individual points in that points are saved within a single object in the tree, the meta data from the first measured point is saved in the Point Set properties and each point index includes a point name, position, and timestamp when available.

Add New Instrument

Adds a new instrument to the current job file.

Input Arguments

| | | | |
|---|------------|-----------------|--------------------------------|
| 0 | Inst. Type | Instrument Type | The type of instrument to add. |
|---|------------|-----------------|--------------------------------|

Return Arguments

| | | | |
|---|--------------------------|---------------------------|--|
| 1 | Collection Instrument ID | Instrument Added (result) | The instrument ID of the added instrument. |
|---|--------------------------|---------------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Delete Instrument

Deletes an instrument from the current job file.

Input Arguments

| 0 | Collection Instrument ID | Instrument ID | The instrument to be deleted. |
|---|--------------------------|-------------------------|--|
| 1 | Boolean | Prompt user to confirm? | Indicate whether the user should be required to confirm the instrument's deletion. |
| 2 | Boolean | Keep resulting points? | Indicate whether the instrument's measured points should be kept. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was deleted successfully. |
| FAILURE | The instrument was not found. |

Remarks

None.

Delete Measurements

Deletes observations from a point originating from a specific instrument, but optionally keeps the point itself.

Input Arguments

| | | | |
|---|--------------------------|---|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument whose measurements should be deleted. |
| 1 | Point Name | Point Name | The name of the point containing the observations to delete. |
| 2 | Boolean | Delete point if no measurements remain? | If TRUE, the point will be deleted if it has no remaining measurements from any other instruments. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were deleted successfully. |
| FAILURE | The instrument or target was not found. |

Remarks

This command is typically used when you have observations to a specific point from multiple instruments, and you only want to delete the observations from a specific instrument.

Delete Measurement Observation

Deletes observation from a point originating from a specific instrument, but optionally keeps the point itself.

Input Arguments

| | | | |
|---|------------|---|--|
| 0 | Point Name | Point Name | The name of the point containing the observations to delete. |
| 1 | Integer | Observation Index | The index of the observation to delete. |
| 2 | Boolean | Delete point if no measurements remain? | If TRUE, the point will be deleted if it has no remaining measurements from any other instruments. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were deleted successfully. |
| FAILURE | The instrument or target was not found. |

Remarks

This command is typically used when you have observations to a specific point from multiple instruments, and you only want to delete the observation from a specific instrument.

Move Measurement Observation

Moves an observation from a point originating from a specific instrument to another point.

Input Arguments

| | | | |
|---|------------|---|--|
| 0 | Point Name | Source Point Name | The name of the point containing the observations to be moved. |
| 1 | Integer | Observation Index | The index of the observation to move. |
| 2 | Boolean | Delete point if no measurements remain? | If TRUE, the point will be deleted if it has no remaining measurements from any other instruments. |
| 3 | Point Name | Destination Point Name | The name of the point which will receive the observation. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were deleted successfully. |
| FAILURE | The instrument or target was not found. |

Remarks

This command is typically used when you have observations to a specific point from multiple instruments, and you only want to move the observation from a specific instrument from the measured point to a different point.

Initiate Servo-Guide

Guides the user through measurement of one or more points. Audio feedback indicates whether the probe tip is within a specified tolerance, and the graphical view actively zooms and pans to keep the probe tip and goal point in the view. A translucent red sphere surrounds the goal point until the probe is within the specified tolerance, at which time it turns green. The user initiates measurement in this mode.

Input Arguments

| | | | |
|---|--------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name Ref List | Nominal Points | A list of the nominal points to measure. |
| 2 | String | Group name suffix | A suffix to attach to the nominal point's group names. |
| 3 | String | Target name suffix | A suffix to attach to the nominal point's target names. |
| 4 | Double | Tolerance | A tolerance for the points. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mode was completed successfully. |
| FAILURE | The nominal points were not found, or an error occurred. |

Remarks

None.

Watch Point to Point

Displays a Point to Point watch window.

Input Arguments

| | | | |
|---|--------------------------|----------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Reference Point | The name of the point to watch. |
| 2 | Double | Tolerance | A tolerance to use for the window. |
| 3 | String | Measurement Mode | The name of the measurement mode to initiate. |
| 4 | Boolean | Pause MP Until Closed | Indicates whether the MP should be paused on this step until the watch window is manually closed. |
| 5 | Integer | Window Top Left X Position | The top left X position of the watch window. |
| 6 | Integer | Window Top Left Y Position | The top left Y position of the watch window. |
| 7 | Integer | Window Width | Enter the value of the width of the watch window. |
| 8 | Integer | Window Height | Enter the value of the height of the watch window. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mode was completed successfully. |
| FAILURE | The reference point, measurement profile, or instrument was not found. |

Remarks

None.

Watch Point to Objects

Displays a Point to Objects watch window.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Collection Object Name Ref List | Objects to Consider | A list of objects to watch. |
| 2 | Projection Options | Projection Options | Options for the watch window projection. |
| 3 | Double | Tolerance | A tolerance for the comparison. |
| 4 | String | Measurement Mode | The text name of the desired measurement mode to use. |
| 5 | Boolean | Pause MP Until Closed | Indicate whether the MP should pause on this command until the watch window is closed. |
| 6 | Integer | Window Top Left X Position | The top left X position of the watch window. |
| 7 | Integer | Window Top Left Y Position | The top left Y position of the watch window. |
| 8 | Integer | Window Width | Enter the value of the width of the watch window. |
| 9 | Integer | Window Height | Enter the value of the height of the watch window. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The mode was completed successfully. |
| PARTIAL SUCCESS | One ore more objects, but not all, could not be found. |
| FAILURE | The reference objects, measurement profile, or instrument was not found. |

Remarks

None.

Watch Point to Edge

Displays a special compensated point watch window that first compensates for the offsets of the active target relative to a primary object and then reports the distance from this compensated point to a reference, such as a hole's edge.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Collection Object Name Ref List | Projection Reference Objects | Objects used for offset compensation. |
| 1 | Collection Object Name Ref List | Measurement Reference Objects | Objects used for distance measurements |
| 2 | Projection Options | Projection Options | Options for the watch window projection. |
| 3 | Double | Tolerance | A tolerance for the comparison. |
| 4 | String | Measurement Mode | The text name of the desired measurement mode to use. |
| 5 | Boolean | Pause MP Until Closed | Indicate whether the MP should pause on this command until the watch window is closed. |
| 6 | Integer | Window Top Left X Position | The top left X position of the watch window. |
| 7 | Integer | Window Top Left Y Position | The top left Y position of the watch window. |
| 8 | Integer | Window Width | Enter the value of the width of the watch window. |
| 9 | Integer | Window Height | Enter the value of the height of the watch window. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The mode was completed successfully. |
| PARTIAL SUCCESS | One ore more objects, but not all, could not be found. |
| FAILURE | The reference objects, measurement profile, or instrument was not found. |

Remarks

None.

Watch Closest Point

Displays a Closest Point watch window.

Input Arguments

| | | | |
|---|---------------------------------|----------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Collection Object Name Ref List | Groups to Consider | A list of point groups to watch. |
| 2 | Double | Tolerance | A tolerance for the comparison. |
| 3 | String | Measurement Mode | The text name of the desired measurement mode to use. |
| 4 | Boolean | Pause MP Until Closed | Indicate whether the MP should pause on this command until the watch window is closed. |
| 5 | Integer | Window Top Left X Position | The top left X position of the watch window. |
| 6 | Integer | Window Top Left Y Position | The top left Y position of the watch window. |
| 7 | Integer | Window Width | Enter the value of the width of the watch window. |
| 8 | Integer | Window Height | Enter the value of the height of the watch window. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The mode was completed successfully. |
| PARTIAL SUCCESS | One or more--but not all--point groups could not be found. |
| FAILURE | The reference groups, measurement profile, or instrument was not found. |

Remarks

None.

Watch Instrument

Displays an Instrument's Point watch window.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The collection instrument ID of the instrument to watch. |
| 1 | Boolean | Pause MP Until Closed | Indicate whether the MP should pause on this command until the watch window is closed. |
| 2 | Collection Object Name | 3 DOF Watch Window Properties | The name of the watch window template if one is used. |
| 3 | Integer | Window Top Left X Position | The top left X position of the watch window. |
| 4 | Integer | Window Top Left Y Position | The top left Y position of the watch window. |
| 5 | Integer | Window Width | Enter the value of the width of the watch window. |
| 6 | Integer | Window Height | Enter the value of the height of the watch window. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The mode was completed successfully. |
| FAILURE | The instrument could not be found. |

Remarks

None.

Watch Window Template 3D

Creates a watch window template.

Input Arguments

| | | | |
|----|------------------------|---------------------------------------|--|
| 0 | Collection Object Name | Watch Window Template Name | The Name of the template to be created. |
| 1 | Integer | Linear Precision | Decimal precision of the linear distances. |
| 2 | Integer | Angular Precision | Decimal precision of the angle. |
| 3 | Font Type | Font | The font type to be used. |
| 4 | Color | Text Color | The color of the text to be used. |
| 5 | Color | Background Color | The background color to be used. |
| 6 | Color | Highlight Color | The highlight color to be used. |
| 7 | Boolean | Show Deviation 1? | Whether to show deviation 1. |
| 8 | Boolean | Show Deviation 2? | Whether to show deviation 2. |
| 9 | Boolean | Show Deviation 3? | Whether to show deviation 3. |
| 10 | Boolean | Show Deviation Mag? | Whether to show the total deviation. |
| 11 | Coordinate System Type | Coordinate System | The coordinate system to be used. |
| 12 | UDP Settings | UDP Network Transmit Settings | Network transmit settings to be used. |
| 13 | Boolean | Report in Working Frame? | The working frame to report to. |
| 14 | Collection Object Name | Reference Frame | The reference frame to be used. |
| 15 | Vector Tolerance | Vector Component Tolerances | Select parameters for the vector tolerance, the magnitude values can also be used for high and low point to object tolerances. |
| 16 | Double | Magnitude Only Tolerance | Single value tolerance used if vector tolerances are not set. |
| 17 | Boolean | Transparent Background? | Whether to make background transparent. |
| 18 | Boolean | Hide Units? | Whether to hide units. |
| 19 | Boolean | Show Graphical Guide? | True - displays the graphical guide window. |
| 20 | Point List mode | Projection Options (Point to Point) | Pick direction (Normal to Actual or Actual to Normal) |
| 21 | Projection Options | Projection Options (Point to Objects) | Select desired projection display. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Watch Point to Point With View Zooming

Creates a watch window between two points with zooming.

Input Arguments

| | | | |
|---|--------------------------|---------------------------|---|
| 0 | Collection Instrument ID | Instrument's ID | The name of the instrument. |
| 1 | Point Name | Reference Point | The Name of the reference point. |
| 2 | Boolean | Update(TRUE),Close(FALSE) | Set TRUE for update and FALSE to close. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None

Start Theodolite Interface

Starts the Theodolite Manager interface using the most recent successful connection settings.

Input Arguments

| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument to start. |
|---|--------------------------|--|---|
| 1 | String | Theodolite Type (Must match Theodolite Manager Add Instrument type). | The type of theodolite, as specified in the theodolite manager connection dialog. (For example, "LEICA T1800"). |
| 2 | Integer | Com Port | The COM port to use for connection. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The interface was started successfully. |
| FAILURE | The instrument could not be found, the type string or COM port is invalid, or the interface is already running. |

Remarks

Starting in 2022.2 defaults have been added to the Add instrument dialog to choose between Theodolite Manager or the Total Station interface. The selected default interface is saved with the instrument model and will be called when this command is run.

Start Instrument Interface

Starts an instrument's interface.

Input Arguments

| | | | |
|---|--------------------------|-----------------------------------|---|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument to start. |
| 1 | Boolean | Initialize at Startup | Indicate whether the instrument should perform its initialization routine upon startup. |
| 2 | String | Device IP Address (optional) | The IP address of the instrument to start up. |
| 3 | Integer | Interface Type (0=default) | The type of interface to start. |
| 4 | Boolean | Run in Simulation | If TRUE, the interface will be immediately started in simulation mode. |
| 5 | Boolean | Allow Start w/o Init Requirements | Allows initialization requirements to be bypassed ** |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The interface was completed successfully. |
| FAILURE | The instrument could not be found, or the interface is already running. |

Remarks

A2 - If the IP address is left blank, then the IP address used the last time the instrument was started (on a given computer) will be used.

Laser Trackers

A1 - The allowable interface types include: 0 (laser tracker), 1 (Leica Automation Interface Control interface), or 2 (Leica T-Scan).

A5 - This is added for Leica AT40x models. When set TRUE, this removes the requirement at startup for the tracker to be locked on to a target in order to successfully initialize (a hardware requirement). This allows you to automate the process of starting an un-initialized AT40x tracker by pointing and locking on a target after starting the interface in order to initialize it.

Total Stations

A1 - The allowable interface types include: 0 (Total Stations), 1 (Theodolite Manager).

Stop Instrument Interface

Stops an instrument's interface.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument interface to stop. |
|---|--------------------------|-----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The interface was stopped successfully. |
| FAILURE | The instrument could not be found. |

Remarks

None.

Activate/Deactivate Instrument Toolbar

Activates and deactivates the instrument toolbar.

Input Arguments

| | | | |
|---|--------------------------|---------------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument toolbar to activate or deactivate. |
| 1 | Boolean | Deactivate Toolbar? | Deactivate Toolbar. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument toolbar was deactivated successfully. |
| FAILURE | The instrument could not be found. |

Remarks

None.

Verify Instrument Connection

Verifies that an instrument still has an active connection with SA.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument to verify. |
|---|--------------------------|-----------------|--|

Return Arguments

| | | | |
|---|---------|------------|---|
| 1 | Boolean | Connected? | Indicates whether the instrument was detected as still connected. |
|---|---------|------------|---|

Returned Status

| | |
|-----------------|--|
| SUCCESS | The check executed successfully and the instrument was connected. |
| Partial Success | The check executed successfully but the instrument was not connected. |
| FAILURE | The instrument could not be found or the verification failed for another reason. |

Remarks

None.

Configure and Measure

Sets an instrument's target name and measurement mode, then initiates measurement. The interface will check to see if the instrument is busy before the command executes, and if so, waits 5 seconds before checking again. If still busy, the command fails without any action having been taken.

Input Arguments

| | | | |
|---|--------------------------|---------------------|---|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Target Name | The name of the target to measure. |
| 2 | String | Measurement Mode | The name of the measurement mode to use. |
| 3 | Boolean | Measure Immediately | Indicate whether measurement should be performed immediately, or whether the user should initiate measurement. |
| 4 | Boolean | Wait for Completion | Indicate whether the MP should pause until the measurement is complete, or whether it can continue executing while the measurement is occurring. |
| 5 | Double | Timeout in Seconds | If Measure Immediately is set to FALSE and Wait for Completion is set to TRUE, the step will fail if a measurement is not received from the instrument within this time period (laser trackers only). |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurement was successful. |
| FAILURE | The measurement was not successful, the measurement mode was not found, the timeout was exceeded, the instrument was not responding, or the instrument was not found. |

Remarks

If "Measure Immediately" is set to TRUE, the command will for a valid distance for trackers before attempting measurement, and fail if a valid lock is not obtained. For some instruments (listed below), you can pass special strings to obtain additional behavior:

| Laser Trackers | |
|--|---|
| Enter the name of the Measurement Profile to use | Any Measurement Profile can be triggered by this command. |

| Portable CMM Arms | |
|-------------------|--|
| Discrete | Single Discrete Point measurement mode. |
| Stream | Scan points per user option setting (spatial or temporal measurement). |
| Patch | Measure patch, or projected point. |

| Portable CMM Arms | |
|-------------------|--|
| Pin | Measure pin, or outside circle with projection plane. |
| Hole | Measure hole, or inside circle with projection plane. |
| Slot | Measure slot, or two inside circles with projection plane. |
| Line | Measure line per user option setting (two point, averaged, or edge). |
| Circle | Measure circle, inside, outside, or on face, with no planar offset. |
| Plane | Measure Plane |
| Sphere | Measure Sphere. |
| Section | Measure Cross Sections (multiple if cross value not equal to zero). |
| Frame | Measure frames (origin at probe center, using probe orientation). |
| Batch | Perform Guided Measurement (invoked from SA with Batch of pts). |
| Scanner | If available, use installed line scanner to measure cloud points. |
| Average | Single averaged point. |
| Geom Trigger | Measure across array of planar geometry triggers. |

| Nikon Laser Radar | |
|---|---|
| Enter the name of the Target Profile to use | Any Target Profile can be triggered by name such as [Tooling Ball], when entered as the Measurement Mode in Argument 2. |

| Theodolite Manager | |
|--------------------|--|
| Record | Triggers a discrete measurement (set the Fast, Standard or Precise if desired using an <i>Instrument Operational Check</i>) |

| Surphaser | |
|-----------------------|--|
| [Saved Parameter Set] | The "Point Name" argument will set the Collection and Cloud names, as well as the group name for found targets, and the voxel cloud name (if set to send). The "Measurement Mode" argument specifies the Saved Parameter Set (measurement profile). If the profile is not found, the command will fail if the User Interaction Mode is set to Silent. Otherwise, you'll be asked if you want to use the current settings. If "Measure Immediately" is false, the command will simply set the profile selected if it is found. The "Timeout in Seconds" is ignored, since scan time can vary quite a lot, depending on scan parameters. |

Measure

Initiates measurement.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument in question. |
|---|--------------------------|-----------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurement was successful. |
| FAILURE | The measurement was not successful or the instrument was not found. |

Remarks

None.

Show/Hide Instrument Interface

Shows/hides and minimizes/restores an instrument interface.

Input Arguments

| | | | |
|---|--------------------------|---------------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument in question. |
| 1 | Boolean | Minimize Interface? | Indicate whether the interface should be minimized or restored. |
| 2 | Boolean | Hide Interface? | If TRUE, the interface will be completely hidden. Set to FALSE in a following command to show the interface again. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The interface was restored or minimized successfully. |
| FAILURE | The instrument was not found or was not connected to an interface. |

Remarks

None.

Dock Instrument Interface

Shows/hides and minimizes/restores an instrument interface.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Boolean | Dock Interface? | Indicates whether or not the interface should be docked to the Instrument Docking Bar. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The interface was docked or undocked successfully. |
| FAILURE | The instrument was not found, or was not connected to an interface. |

Remarks

None.

Locate Instrument (Ref. Tie-In)

Guides a user through measurement of a set of points, and locates the instrument in accordance with a best fit transformation between the measured and nominal points. Equivalent to the *Instrument*►*Locate (Transform to Part)*►*Measure Nominal Points* command.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument to locate. |
| 1 | Collection Object Name | Reference Group Name | The name of the group containing the reference points. |
| 2 | Collection Object Name | Actuals Group Name (to be measured) | The group into which the measured points will be placed. |
| 3 | Double | Tolerance | A tolerance for the measured points. |
| 4 | Boolean | Auto Survey | Indicate whether the instrument should automatically point and measure each nominal point (if possible). |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurement completed successfully. |
| FAILURE | The instrument or reference group was not found, or measurement failed. |

Remarks

None.

Locate Instrument (Group to Surface Quick Fit)

Locates an instrument by initially performing a points to points fit between nominal and measured points (preferably at least 3), then performs a points to surfaces fit for final alignment.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Collection Instrument ID | Instrument to Locate | The instrument ID of the instrument to locate. |
| 1 | Collection Object Name | Name of Measured Group | The group containing the measured point. |
| 2 | Collection Object Name | Name of Group containing Surface Pts | The group containing the "quick fit" points for the initial fit. |
| 3 | Collection Object Name | Surface to fit | The surface to fit to. |
| 4 | Collection Object Name Ref List | Other Objects to Transform | Additional objects to which to apply the calculated transform. |
| 5 | Double | RMS Tolerance (0.0 for none) | The RMS tolerance for the surface fit. |
| 6 | Double | Maximum Absolute Tolerance (0.0 for none) | The maximum absolute tolerance for the surface fit. |

Return Arguments

| | | | |
|---|--------|------------------------|--|
| 7 | Double | RMS Error | The calculated RMS error for the fit. |
| 8 | Double | Maximum Absolute Error | The actual maximum absolute RMS error for the fit. |

Returned Status

| | |
|-----------------|---|
| SUCCESS | The instrument was located successfully and the tolerances were not exceeded. |
| PARTIAL SUCCESS | The instrument was located successfully, but one or both tolerances were exceeded. |
| FAILURE | The instrument, measured group, surface points, surface, or other objects were not found. |

Remarks

None.

Locate Instruments (USMN)

Locates a network of instruments using Unified Spatial Metrology Network (USMN).

Input Arguments

| | | | |
|----|-----------------------------------|--|--|
| 0 | Collection Instrument ID Ref List | Instruments to Locate | A list of instruments to locate. |
| 1 | Collection Object Name | Nominals Group Name (blank for none) | An optional group containing nominal points for driving the USMN fit. |
| 2 | Collection Object Name | Output Group Name (to be established) | The name for the resulting USMN composite point group. |
| 3 | Boolean | Move in Working Frame (TRUE) or Instrument Frame (FALSE) | Controls the way in which an instrument is rotated, either using its base frame or the world frame. |
| 4 | Boolean | AutoReject Outliers and Resolve | Indicates whether outliers should automatically be rejected (once), with the network then being re-solved. |
| 5 | Show USMN Dialog | Show USMN Dialog | Indicate whether the USMN dialog should always be shown, never be shown, or only shown on tolerance violation. |
| 6 | Double | Max Acceptable RMS Error Value (0.0 for none) | A tolerance for the maximum acceptable RMS error from the USMN fit. |
| 7 | Double | Max Acceptable Error Value (0.0 for none) | A tolerance for the maximum acceptable error from the USMN fit. |
| 8 | Collection Object Name Ref List | Groups to be Excluded | The groups to exclude from the USMN calculation. |
| 9 | Boolean | Exclude Points Measured By Only One Instrument | If TRUE, points that have measurements from only one instrument are excluded from the USMN calculation. |
| 10 | Boolean | Run Uncertainty Field Analysis? | True, runs the uncertainty analysis |
| 11 | Integer | Analysis Samples | Number of samples to include for each point |
| 12 | Double | Analysis Time Limit (Minutes - 0 for none) | How long to let the analysis continue. |

Return Arguments

| | | | |
|----|--------|-----------------|---|
| 13 | Double | RMS Error Value | The calculated RMS error for the fit. |
| 14 | Double | Max Error Value | The calculated maximum error for the fit. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The USMN was completed successfully and tolerances were not exceeded. |
| PARTIAL SUCCESS | The instruments were located successfully, but one or both tolerances were exceeded. |
| FAILURE | The instruments or nominal group were not found. |

Remarks

For more information on USMN refer to Chapter 30 of the Users Manual.

Locate Templated Instruments (USMN)

Locates a network of templated instruments using Unified Spatial Metrology Network (USMN). Use Create Templated Instrument USMN to define parameters in advance for this command.

Input Arguments

| | | | |
|---|---------------------------------|--|--|
| 0 | USMN Instrument Template List | Templated Instruments to Locate | A list of instruments to locate. |
| 1 | Collection Object Name | Nominals Group Name (blank for none) | An optional group containing nominal points for driving the USMN fit. |
| 2 | Collection Object Name | Output Group Name (to be established) | The name for the resulting USMN composite point group. |
| 3 | Boolean | Move in Working Frame (TRUE) or Instrument Frame (FALSE) | Controls the way in which an instrument is rotated, either using its base frame or the world frame. |
| 4 | Boolean | AutoReject Outliers and Resolve | Indicates whether outliers should automatically be rejected (once), with the network then being re-solved. |
| 5 | Show USMN Dialog | Show USMN Dialog | Indicate whether the USMN dialog should always be shown, never be shown, or only shown on tolerance violation. |
| 6 | Double | Max Acceptable RMS Error Value (0.0 for none) | A tolerance for the maximum acceptable RMS error from the USMN fit. |
| 7 | Double | Max Acceptable Error Value (0.0 for none) | A tolerance for the maximum acceptable error from the USMN fit. |
| 8 | Collection Object Name Ref List | Groups to be Excluded | The groups to exclude from the USMN calculation. |
| 9 | Boolean | Exclude Points Measured By Only One Instrument | If TRUE, points that have measurements from only one instrument are excluded from the USMN calculation. |

Return Arguments

| | | | |
|----|--------|-----------------|---|
| 10 | Double | RMS Error Value | The calculated RMS error for the fit. |
| 11 | Double | Max Error Value | The calculated maximum error for the fit. |

Returned Status

| | |
|-----------------|--|
| SUCCESS | The USMN was completed successfully and tolerances were not exceeded. |
| PARTIAL SUCCESS | The templated instruments were located successfully, but one or both tolerances were exceeded. |
| FAILURE | The templated instruments or nominal group were not found. |

Remarks

None.

Create Templated Instrument (USMN)

Creates templated instrument for Unified Spatial Metrology Network (USMN).

Input Arguments

| | | | |
|----|--------------------------|--------------------------------|--|
| 0 | Collection Object Name | Instrument template Name | The name to give the instrument template. |
| 1 | Collection Instrument ID | Instrument ID | Name of the instrument to add the template to. |
| 2 | Double | Overall Instrument Weight | The instruments weight in the USMN |
| 3 | Boolean | Moving | Designate whether the instrument is moving. |
| 4 | Boolean | Enable X | Enable X as a degree of freedom. |
| 5 | Boolean | Enable Y | Enable Y as a degree of freedom. |
| 6 | Boolean | Enable Z | Enable Z as a degree of freedom. |
| 7 | Boolean | Enable Rx | Enable Rx as a degree of freedom. |
| 8 | Boolean | Enable Ry | Enable Ry as a degree of freedom. |
| 9 | Boolean | Enable Rz | Enable Rz as a degree of freedom. |
| 10 | Boolean | Enable Scale | Enable scale in the USMN. |
| 11 | Boolean | Enable Component Weights | Enable the weights of components in the USMN. |
| 12 | Double | Component 1 (Azimuth) Weight | Set Azimuth weight. |
| 13 | Double | Component 2 (Elevation) Weight | Set Elevation weight. |
| 14 | Double | Component 3 (Distance) Weight | Set Distance weight. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The template was successfully created for the instrument. |
| FAILURE | The instrument was not found. |

Remarks

None.

Make a USMN Templated Instrument List

Creates a list of templated instruments for Unified Spatial Metrology Network (USMN).

Input Arguments

None.

Return Arguments

| | | | |
|---|-------------------------------|-------------------------------|---|
| 0 | USMN Instrument Template List | USMN Instrument template List | The name to give the instrument template. |
|---|-------------------------------|-------------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Add a USMN Templated Instrument to a USMN Templated Instrument List

Adds a templated instrument for Unified Spatial Metrology Network (USMN) templated instrument list.

Input Arguments

| | | | |
|---|--|---------------------------------|--|
| 0 | USMN Instrument Template List | USMN Instrument template List | The name to give the instrument template. |
| 1 | Collection USMN Instrument Template Name | USMN Instrument template To Add | The name of the templated instrument to add. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | USMN template list and templated instrument to add were both found and instrument added successfully. |
| FAILURE | USMN template list and/or templated instrument to add was not found. |

Remarks

Neither argument can be left blank.

Locate Instrument (Best Fit - Group to Group)

Locates an instrument by fitting measured points to a set of reference points.

Input Arguments

| | | | |
|----|----------------------------|--|---|
| 0 | Collection Object Name | Reference Group | The group to fit to. |
| 1 | Collection Object Name | Corresponding Group | The measured points to fit. |
| 2 | Boolean | Show Interface | Indicate whether the best fit interface should be displayed. |
| 3 | Double | RMS Tolerance (0.0 for none) | A tolerance for the RMS error of the fit. |
| 4 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum absolute error allowed for the fit. |
| 5 | Boolean | Allow Scale | Indicates whether the instrument is allowed to scale in the fit. |
| 6 | Boolean | Allow X | Indicates whether the X position degree of freedom is allowed to float in the fit. |
| 7 | Boolean | Allow Y | Indicates whether the Y position degree of freedom is allowed to float in the fit. |
| 8 | Boolean | Allow Z | Indicates whether the Z position degree of freedom is allowed to float in the fit. |
| 9 | Boolean | Allow Rx | Indicates whether the Rx rotational degree of freedom is allowed to float in the fit. |
| 10 | Boolean | Allow Ry | Indicates whether the Ry rotational degree of freedom is allowed to float in the fit. |
| 11 | Boolean | Allow Rz | Indicates whether the Rz rotational degree of freedom is allowed to float in the fit. |
| 12 | Boolean | Lock Degrees of Freedom | If True, the DoF controls in the interface will be locked out such that a user cannot access them when the dialog is displayed. |
| 13 | Boolean | Generate Event | When True, an event will be generated. |
| 14 | File Path or Embedded File | File Path for CSV Text Report (requires Show Interface = TRUE) | A path for a CSV text report to create as a result of the fit (only created if the Show Interface argument is set to TRUE). |

Return Arguments

| | | | |
|----|--------------------------|----------------------------|--|
| 13 | Transform | Transform in Working | The calculated transform (in working coordinates) for the fit. |
| 14 | World Transform Operator | Optimum Transform | The calculated transform (in world coordinates) for the fit. |
| 15 | Double | RMS Deviation | The calculated RMS error for the fit. |
| 16 | Double | Maximum Absolute Deviation | The actual maximum absolute error for the fit. |
| 19 | Integer | Number of Unknowns | The number of unknowns |
| 20 | Integer | Number of Equations | The number of equations used |
| 21 | Double | Robustness | The solution robustness |

Returned Status

| | |
|---------|---|
| SUCCESS | The fit was successful and no tolerances were exceeded. |
| FAILURE | The reference or corresponding group could not be found, the fit could not be performed, or one or more tolerances were exceeded. |

Remarks

The transform will not be applied to the instrument if any tolerance is exceeded. Also, monitoring the solution robustness factor is a great way to identify the mathematical stability of the solution.

Locate Instrument (Best Fit - Nominal Geometry)

Locates an instrument by fitting measured feature center points to a set of reference point reducible features.

Input Arguments

| | | | |
|----|----------------------------|--|---|
| 0 | Collection Instrument ID | Instrument ID | Instrument to Align |
| 1 | Relationship Ref List | Geometry Relationships | The point reducible compare to nominal relationships used for the best fit. |
| 2 | Boolean | Show Interface | Indicate whether the best fit interface should be displayed. |
| 3 | Double | RMS Tolerance (0.0 for none) | A tolerance for the RMS error of the fit. |
| 4 | Double | Maximum Absolute Tolerance (0.0 for none) | A maximum absolute error allowed for the fit. |
| 5 | Boolean | Allow Scale | Indicates whether the instrument is allowed to scale in the fit. |
| 6 | Boolean | Allow X | Indicates whether the X position degree of freedom is allowed to float in the fit. |
| 7 | Boolean | Allow Y | Indicates whether the Y position degree of freedom is allowed to float in the fit. |
| 8 | Boolean | Allow Z | Indicates whether the Z position degree of freedom is allowed to float in the fit. |
| 9 | Boolean | Allow Rx | Indicates whether the Rx rotational degree of freedom is allowed to float in the fit. |
| 10 | Boolean | Allow Ry | Indicates whether the Ry rotational degree of freedom is allowed to float in the fit. |
| 11 | Boolean | Allow Rz | Indicates whether the Rz rotational degree of freedom is allowed to float in the fit. |
| 12 | Boolean | Lock Degrees of Freedom | If True, the DoF controls in the interface will be locked out such that a user cannot access them when the dialog is displayed. |
| 13 | Boolean | Generate Event | When True, an event will be generated. |
| 14 | File Path or Embedded File | File Path for CSV Text Report (requires Show Interface = TRUE) | A path for a CSV text report to create as a result of the fit (only created if the Show Interface argument is set to TRUE). |

Return Arguments

| | | | |
|----|--------------------------|----------------------------|--|
| 13 | Transform | Transform in Working | The calculated transform (in working coordinates) for the fit. |
| 14 | World Transform Operator | Optimum Transform | The calculated transform (in world coordinates) for the fit. |
| 15 | Double | RMS Deviation | The calculated RMS error for the fit. |
| 16 | Double | Maximum Absolute Deviation | The actual maximum absolute error for the fit. |
| 19 | Integer | Number of Unknowns | The number of unknowns |
| 20 | Integer | Number of Equations | The number of equations used |
| 21 | Double | Robustness | The solution robustness |

Returned Status

| | |
|---------|---|
| SUCCESS | The fit was successful and no tolerances were exceeded. |
| FAILURE | The reference or corresponding group could not be found, the fit could not be performed, or one or more tolerances were exceeded. |

Remarks

The transform will not be applied to the instrument if any tolerance is exceeded. Also, monitoring the solution robustness factor is a great way to identify the mathematical stability of the solution.

Get Instrument Transform

Retrieves the transform of an instrument.

Input Arguments

| | | | |
|---|--------------------------|-----------------|---|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument of interest. |
| 1 | Collection Object Name | Reference Frame | The reference frame in which to express the instrument's transform. |

Return Arguments

| | | | |
|---|-----------|-----------|---|
| 2 | Transform | Transform | The transform of the instrument expressed in the reference frame. |
|---|-----------|-----------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The transform was successfully obtained. |
| FAILURE | The instrument or reference frame could not be found. |

Remarks

None.

Set Instrument Transform

Sets an instrument's transform.

Input Arguments

| | | | |
|---|--------------------------|-----------------------|---|
| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
| 1 | Transform | Destination Transform | The destination transform for the instrument. |
| 2 | Collection Object Name | Reference Frame | The frame in which to the destination transform should be expressed. |
| 3 | Integer | Number of Steps | The number of animation steps to use when animating the instrument to its new position in the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument's transform was successfully set. |
| FAILURE | The instrument or reference frame could not be found. |

Remarks

None.

Set Tracker/EDM Theodolite Uncertainties

Sets the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button(used with Laser Trackers and Total Stations).

Input Arguments

| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|-------------------------------|---|
| 1 | Double | Theta Resolution (arcseconds) | Theta or Horizontal Angle |
| 2 | Double | Phi Resolution (arcseconds) | Phi or Vertical Angle |
| 3 | Double | Distance Error | Distance Measurement Error (in job units) |
| 4 | Double | PPM | Distance Measurement parts per million |
| 5 | Double | Aperture Resolution | Aperture error threshold (in job units) |
| 6 | Double | Aperture PPM | Aperture parts per million |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Get Tracker/EDM Theodolite Uncertainties

Returns the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button(used with Laser Trackers and Total Stations).

Input Arguments

| | | | |
|---|--------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|--------------------|---|

Return Arguments

| | | | |
|---|--------|-------------------------------|---|
| 1 | Double | Theta Resolution (arcseconds) | Theta or Horizontal Angle |
| 2 | Double | Phi Resolution (arcseconds) | Phi or Vertical Angle |
| 3 | Double | Distance Error | Distance Measurement Error (in job units) |
| 4 | Double | PPM | Distance Measurement parts per million |
| 5 | Double | Aperture Resolution | Aperture error threshold (in job units) |
| 6 | Double | Aperture PPM | Aperture parts per million |

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Set PCMM Instrument XYZ Uncertainties

Sets the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button(used with PCMM Arms). The XYZ uncertainty values are with respect to the instruments base frame.

Input Arguments

| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|--------------------|---|
| 1 | Double | X Uncertainty | Uncertainty value for X |
| 2 | Double | Y Uncertainty | Uncertainty value for Y |
| 3 | Double | Z Uncertainty | Uncertainty value for Z |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Get PCMM Instrument XYZ Uncertainties

Returns the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button(used with PCMM Arms). The XYZ uncertainty values are with respect to the instruments base frame.

Input Arguments

| | | | |
|---|--------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|--------------------|---|

Return Arguments

| | | | |
|---|--------|---------------|-------------------------|
| 1 | Double | X Uncertainty | Uncertainty value for X |
| 2 | Double | Y Uncertainty | Uncertainty value for Y |
| 3 | Double | Z Uncertainty | Uncertainty value for Z |

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Set Instrument XYZ Uncertainties

Sets the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button. The XYZ uncertainty values are with respect to the instruments base frame.

Input Arguments

| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|--------------------|---|
| 1 | Double | X Uncertainty | Uncertainty value for X |
| 2 | Double | Y Uncertainty | Uncertainty value for Y |
| 3 | Double | Z Uncertainty | Uncertainty value for Z |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Get Instrument XYZ Uncertainties

Returns the uncertainty parameters saved within the selected instruments properties under the [Edit Uncertainty Variables](#) button. The XYZ uncertainty values are with respect to the instruments base frame.

Input Arguments

| | | | |
|---|--------------------------|--------------------|---|
| 0 | Collection Instrument ID | Instrument to Move | The collection instrument ID of the instrument of interest. |
|---|--------------------------|--------------------|---|

Return Arguments

| | | | |
|---|--------|---------------|-------------------------|
| 1 | Double | X Uncertainty | Uncertainty value for X |
| 2 | Double | Y Uncertainty | Uncertainty value for Y |
| 3 | Double | Z Uncertainty | Uncertainty value for Z |

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument's uncertainty parameters was successfully returned. |
| FAILURE | The instrument could not be found or is of the wrong type. |

Remarks

Refer to the Uncertainty chapter of the users manual for addition information.

Get Instrument Weather Setting

Retrieves the weather settings for an instrument.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
|---|--------------------------|-----------------|--|

Return Arguments

| | | | |
|---|---------|--|---|
| 1 | Double | Temperature (F) | The temperature setting for the instrument (in degrees F). |
| 2 | Double | Pressure (mmHg) | The pressure setting for the instrument (in mmHg). |
| 3 | Double | Humidity (%Rel) | The relative humidity setting for the instrument (in %). |
| 4 | Boolean | Was Set Automatically? (using Inst or external sensor) | Indicates whether the weather was set automatically (using the instrument's sensors) or an external sensor/manually). |

Returned Status

| | |
|---------|--|
| SUCCESS | The weather settings were retrieved. |
| FAILURE | The instrument could not be found, or an error occurred retrieving the weather data. |

Remarks

None.

Set Instrument Weather Setting

Sets the weather settings for an instrument.

Input Arguments

| | | | |
|---|--------------------------|--|---|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
| 1 | Double | Temperature (F) | The temperature setting for the instrument (in degrees F). |
| 2 | Double | Pressure (mmHg) | The pressure setting for the instrument (in mmHg). |
| 3 | Double | Humidity (%Rel) | The relative humidity setting for the instrument (in %). |
| 4 | Boolean | Set Automatically? (Ignore above values) | Indicates whether the weather settings should be set to their sensed values (therefore ignoring the above values) or whether the provided values in Arguments 1-3 should be used. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The weather settings were set successfully. |
| FAILURE | The instrument could not be found, or there was a general error setting the weather values. |

Remarks

None.

Get Instrument Part Temperature

Retrieves the part temperature from an instrument.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
|---|--------------------------|-----------------|--|

Return Arguments

| | | | |
|---|--------|------------------|--|
| 1 | Double | Part Temperature | The instrument's reading for part temperature. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The part temperature was retrieved successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection.

Compute CTE Scale Factor

Computes a scale factor based on the Coefficient of Thermal Expansion.

Input Arguments

| | | | |
|---|--------|-------------------------|---|
| 0 | Double | Material CTE (1/Deg F) | The coefficient of thermal expansion for the material of interest (per degree F). |
| 1 | Double | Initial Temperature (F) | The initial temperature of interest (in degrees F). |
| 2 | Double | Final Temperature (F) | The final temperature of interest (in degrees F). |

Return Arguments

| | | | |
|---|--------|--------------|------------------------------|
| 3 | Double | Scale Factor | The calculated scale factor. |
|---|--------|--------------|------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set (multiply) Instrument Scale Factor (CAUTION!)

Multiplies an instrument's current scale by a new scale factor.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
| 1 | Double | Scale Factor | The scale factor to multiply by. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scale factor was modified successfully. |
| FAILURE | The instrument was not found. |

Remarks

Use extreme caution with this command. This command multiplies the instrument's current scale factor by the provided scale factor--it does not SET the scale factor to the provided value.

Set (absolute) Instrument Scale Factor (CAUTION!)

Sets an instrument's current scale to a new scale factor.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
| 1 | Double | Scale Factor | The scale factor to set the instrument to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The scale factor was set successfully. |
| FAILURE | The instrument was not found. |

Remarks

Use extreme caution with this command. This command sets the instrument's current scale factor to a new value--it does NOT multiply the existing scale factor by the provided value.

Get Instrument Scale Factor

Retrieves the scale factor for an instrument.

Input Arguments

| | | | |
|---|--------------------------|-----------------|--|
| 0 | Collection Instrument ID | Instrument's ID | The instrument ID of the instrument of interest. |
|---|--------------------------|-----------------|--|

Return Arguments

| | | | |
|---|--------|--------------|--|
| 1 | Double | Scale Factor | The instrument's current scale factor. |
|---|--------|--------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The scale factor was retrieved successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection.

Transform Instrument - Frame To Frame

Transforms an instrument using the 6-DOF delta between a source frame and a destination frame.

Input Arguments

| | | | |
|---|--------------------------|------------------------|--|
| 0 | Collection Instrument ID | Instrument to move | The instrument ID of the instrument to move. |
| 1 | Frame Name | Initial Frame Name | The name of the source (starting) frame. |
| 2 | Frame Name | Destination Frame Name | The name of the destination (ending) frame. |
| 3 | Integer | Number of Steps | The number of animation steps to use when animating the instrument's movement in the graphical view. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The instrument was transformed successfully. |
| FAILURE | The instrument, source frame, or destination frame was not found. |

Remarks

The instrument and frames must be in the active collection.

Transform Instrument by Delta

Transforms an instrument by a provided 6-DOF delta transform.

Input Arguments

| | | | |
|---|--------------------------|--|---|
| 0 | Collection Instrument ID | Instrument to Transform | The instrument ID of the instrument to transform. |
| 1 | World Transform Operator | Delta Transform | The delta transform to apply to the instrument (in world transform coordinates). |
| 2 | Boolean | Apply Scale from Transform to Instrument | Indicates whether the scale specified in the transform in Argument 1 should be applied to the instrument. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The instrument was transformed successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection.

Transform Multiple Instruments by Delta

Transforms one or more instruments by a provided 6-DOF delta transform.

Input Arguments

| | | | |
|---|-----------------------------------|--|--|
| 0 | Collection Instrument ID Ref List | Instruments to Move | A list of collection instrument IDs specifying the instruments to transform. |
| 1 | World Transform Operator | Delta Transform | The delta transform to apply to the instruments (in world transform coordinates). |
| 2 | Boolean | Apply Scale from Transform to Instrument | Indicates whether the scale specified in the transform in Argument 1 should be applied to the instruments. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The instruments were transformed successfully. |
| PARTIAL SUCCESS | At least one instrument (but not all) was not found. |
| FAILURE | The instruments were not found. |

Remarks

None.

Instrument Operational Check

Executes a behavior on an instrument that is typically unique to that instrument or class of instruments.

Input Arguments

| | | | |
|---|--------------------------|---------------------|---|
| 0 | Collection Instrument ID | Instrument to Check | The instrument ID of the instrument in question. |
| 1 | String | Check Type | A string command indicating the type of check to perform. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------------|
| SUCCESS | The check was performed successful. |
| FAILURE | The check was not successful. |

Remarks

Command strings are listed below:

Laser Tracker Commands

| Laser Trackers | |
|------------------|--|
| Time Out [] | [] is the time in seconds to wait for an MP command that supports a time out period to be assigned as the maximum allowed time for the command to complete successfully before automatically failing at the expiration of the period. Initially, only the "Home" command supports Time Out. The Time Out can be disabled by issuing a command of "Time Out 0.0". |
| Retry On | Enable retries for MP commands. If a command is specified correctly but the instrument has not responded, it will retry after the "Retry Wait" period. |
| Retry Wait [] | When "Retry On" is enabled, [] specifies the number of seconds after the command has failed before a retry attempt ([] not part of string). |
| Retry Off | Turns retries off for MP commands (Default behavior). |
| Set Targ [] | Set active target to that designated by [] ([] not part of string). |
| Set Auto Meas [] | Set SA requested discrete point acquisition to that specified by []. Any meas profile with a discrete acquisition can be referenced and it's settings used to set the SA Interaction >Measurement Requests profile. |
| Motors On | Turn motors on. |
| Motors Off | Turn motors off. |
| AutoLock On | Turns on PowerLock, SmartFind, or i-Vision. |

| Laser Trackers | |
|--------------------------------|--|
| AutoLock Off | Turns off PowerLock, SmartFind, or i-Vision. |
| HoldPositionNoBeamLock On | Hold position on, but tracking off (tracker won't lock if it sees a target). For Leica emScon trackers, this turns PowerLock off. For Faro trackers, when the tracker is pointed at a target, the tracker will not attempt a lock or search. |
| HoldPositionNoBeamLock Off | Hold position on, and tracking returns to default behavior (will lock on if it sees a target). For Leica emScon trackers, this turns PowerLock on. For Faro trackers, the tracker will attempt to lock on if pointed at a target. |
| Start | Re-start tracker (performs initialization as well). |
| Initialize | Initialize the tracker (NOTE: does not apply to API trackers). |
| Home | Home tracker. |
| Two Face Here | Perform a two-face ops check at the current location. |
| ADM/IFM Here | Perform ADM ops check (with respect to IFM) at current location. |
| IFM | Start an IFM ops check/cal. (two station or scale bar) at current location. |
| Closure | Perform closure check at current location with respect to current home. |
| Drift | Start drift ops check. |
| ADM | Start ADM ops check/cal ui for multiple locations (with respect to IFM). |
| ADM Drive | Equivalent to pressing the "ADM Drive" button. |
| ADM Reset [<i>d</i>] | Equivalent to pressing the "ADM Reset" button. If [<i>d</i>] is included ([] not part of string), the seed distance will be set to the distance <i>d</i> . |
| ADM Search Radius [<i>r</i>] | Sets the ADM search radius to the value in inches designated by <i>r</i> ([] not part of string) |
| ADM Timeout [<i>t</i>] | Sets the ADM reset timeout to the amount in seconds designated by <i>t</i> ([] not part of string). |
| Jog Up [<i>d</i>] | Jogs the tracker up by the amount in degrees designated by <i>d</i> ([] not part of string). |
| Jog Down [<i>d</i>] | Jogs the tracker down by the amount in degrees designated by <i>d</i> ([] not part of string). |
| Jog Left [<i>d</i>] | Jogs the tracker left by the amount in degrees designated by <i>d</i> ([] not part of string). |
| Jog Right [<i>d</i>] | Jogs the tracker right by the amount in degrees designated by <i>d</i> ([] not part of string). |
| Camera View | Pops the tracker's overview video camera view. |
| Ball Bar | Start Ball Bar ops check. |
| Reflector Center | Start Reflector Center ops check. |
| Is Laser Warmed Up | Return value indicates whether the laser is warmed up |
| Valid Distance | Return value indicates whether tracker is locked on a reflector. |
| Is Measuring | Return value indicates whether the tracker is measuring. |
| Change Face | Change face. |
| Measure Level | Initiate level measurement--sends frame to SA with Z-axis 'up'. |
| Monitor Level | Initiate level monitor. |
| Level Compensator ON | Turn ON Level Compensation if this tacker is equipped. if not, this returns true so as not to interrupt the script. NOTE: This command will Re-Initialize the tracker if needed. |
| Level Compensator OFF | Turn OFF Level Compensation if this tacker is equipped. if not, this returns true so as not to interrupt the script. NOTE: This command will Re-Initialize the tracker if needed. |

| Laser Trackers | |
|---|---|
| Outdoor Mode On | For Leica 40x trackers only: Turns ON Outdoor measurement mode. If not a 40x tracker, this returns true so as not to interrupt the script. |
| Outdoor Mode Off | For Leica 40x trackers only: Turns OFF Outdoor measurement mode, and sets tracker to Fast Point mode. If not a 40x tracker, this returns true so as not to interrupt the script. |
| Beam Break Auto-IFM | Set beam break behavior to automatically home to the last used home position (tracker mounted nest, or remote home). |
| Beam Break Auto-ADM | Set beam break behavior to automatically use ADM to reset the distance. |
| Shut Down Tracker | For LMF (AT9x0, ATS) only – turns off the tracker and controller, and closes the interface |
| Go To Sleep, Wake Up From Now d, h::m::s | For LMF (930/960, ATS) only – turns off the laser, and turns it back on after the amount of time designated by d, h::m::s has passed, where d is the integer number of days, h is hours, m is minutes, and s is seconds. The comma and colons are required parts of the string, and the letters are to be replaced by the appropriate integer numbers. This time is FROM THE CURRENT LOCAL TIME of your PC. So the time to turn the laser back on is RELATIVE to the current time, therefore, the MP will always be valid. This command does not close the interface, but you can close it if you wish. The laser will still come back on after the designated time. You can in fact check the tracker controller, it will tell you when it is scheduled to turn the laser back on. |
| Measure All You Can See | For LMF (930/960, ATS) only – tells tracker to measure all targets visible in the camera's field of view |
| Spiral Search On | For Leica 901 – favors spiral search over power lock – no retry |
| Spiral Search Off | For Leica 901 – favors power lock over spiral search – no retry |
| Allow No Tip On | Allow tipless measurement (for Leica T-Products). |
| Allow No Tip Off | Disallow tipless measurement (for Leica T-Products). |
| Set External Trigger For TMAC-Touch Probe | Sets EmScon triggering to External, and makes related settings to ready the system for measuring with the TMAC-I touch probe. This MUST be called before doing touch-triggered measurement. |
| Set Internal Trigger | Sets EmScon triggering to Internal Application, the “normal” use case where measurements are initiated from the interface or from MP commands. |
| Select Compensation [] | Initially for Leica 930/960. Selects the tracker compensation by name, designated by [] ([] not part of string). Results logged to Inst. History. |
| Set Faro Internal Trigger | Sets tracker to the default “Internal” triggering, where the interface controls measurement. |
| Set Faro External Trigger | Sets tracker to accept external trigger signal connected to tkr to control measurement. In main interface, go to General Settings >> Faro Settings for detailed information. |
| Faro Beam Mode IFM Only | Set Faro tracker to IFM-only beam mode--disables ADM. |
| Faro Beam Mode ADM Only | Set Faro tracker to ADM-only beam mode--disables IFM. |
| Faro Beam Mode IFM Set By ADM | Set Faro tracker to set distances with ADM only, but count IFM fringes when measuring distance. |
| Enable Faro Camera Search | For Faro trackers that are video-capable: Enables the camera search such that an ADM target search will try the camera search before trying the spiral search (provided the target is within the acceptable camera search range). |
| Disable Faro Camera Search | For Faro trackers that are video-capable: Disables the camera search such that an ADM target search will only try a spiral search. |

| Laser Trackers | |
|--|---|
| Run Faro CompIT | Run Faro's CompIT utility. |
| Run Faro Self Comp | Run Faro self-compensation directly (no Java Applet). Note: This command calls "Quick Comp" on Vantage trackers. |
| Run Faro Angular Accuracy Check | Run Faro angular accuracy check directly (no Java Applet). |
| Run Faro ADM/IFM Check | Run Faro ADM/IFM check directly (no Java Applet). |
| Run Faro Quick Comp | Runs Faro's no-UI quick compensation on the currently-locked target. |
| Run Faro AAC | Runs Faro's no-UI angular accuracy check on the currently-locked target. |
| Run Faro Angular Accuracy Check | Run Faro Angular Accuracy Check directly (no Java Applet). |
| Run Faro ADM/IFM Check | Run Faro ADM/IFM Check directly (no Java Applet). |
| Run Faro Quick Comp | Run Faro No UI Quick Compensation (on the currently locked target) |
| Run Faro AAC | Run Faro No UI AAC (on the currently locked target, newer trackers) |
| Run Faro Angular Accuracy Check | Run Faro Direct Angular Accuracy Check (on the currently locked target, older trackers) |
| Add Remote Home [] | Add a new remote home position with name designated by [] ([] not part of string). |
| Go To Remote Home [] | Go to and lock onto the remote home designated by [] ([] not part of string). |
| Delete Remote Home [] | Delete the remote home designated by [] ([] not part of string). |
| API DI Virtual Level | Perform API Device Interface virtual level routine. |
| API DI iProbe Offset | Pop API Device Interface iProbe Offset Calibration window. |
| API DI iProbe Global | Pop API Device Interface iProbe Global Calibration window. |
| API DI iVision Dlg | Pop the API Device Interface iVision control/settings dialog. |
| API DI iVision Multi-SMR Timeout Seconds [t] | The automated Innovo multi-smr measurement has no error handling. This provides a maximum time to wait for the measurement to return success, so that the MP command can fail when appropriate. The time is designated by t ([] not part of string). 300 seconds is the default. |
| API DI iVision Measure Time Seconds [t] | This sets the acquisition time for each Innovo measured point. The time is designated by t ([] not part of string). 0.5 to 5.0 seconds is recommended. 0.5 seconds is the default. |
| API DI Perform iVision Multi SMR Measurement | Enable the Innovo camera, and put it in multi-SMR measurement mode. Return success only if the measurement succeeds. |
| API DI Enable iVision | Enable the Innovo camera for catching the beam. |
| API DI Disable iVision | Disable the Innovo camera. |
| API DI Run iVision Teach | This causes the API iVision dialog to display, waiting for you to click on the SMRs in the camera's field of view that you would like the Teach Measurement (see below) to measure. |
| API DI iVision Teach Meas Iterations [] | Sets the number of iterations designated by [] which the iVision Teach measurement will be performed. |
| API DI iVision Teach Meas XML Path [C:\Temp\TeachMeas.xml] | Sets the path for the iVision Teach measurement XML file designated by []. (Path shown as an example). |
| API DI Run iVision Teach Meas | This will run the iVision-taught points (see above) and will use the iterations and path set by the above commands. |
| Set Motor Mode Tracking | For API — put motors in "Tracking" mode. For the Radian, this can be used to tell the camera to lock onto the SMR. |
| API TTL Trigger ON | Set the external TTL trigger mode on for the next temporal scan measurement to use it. |
| API TTL Trigger OFF | Set the external TTL trigger mode off, so that the next temporal scan measurement will not use it. |

| Laser Trackers | |
|----------------------------------|---|
| Show Big Group/Target Window | Pops the resizable group/target window. The window will persist its size and placement. Returns success if the dialog is already showing. |
| Close Big Group/Target Window | Closes the group/target window if it is open. Returns success if the dialog is already closed. |
| Hide UI | Hide the main tracker interface window and show the SA tracker ToolBar (no retry). |
| Show UI | Hide the SAToolBar and show the main tracker interface window (no retry). |
| Show RMS Monitor | Display the RMS Monitor window. Do nothing if it is already up. |
| Hide RMS Monitor | Close the RMS Monitor window. Do nothing if it is already closed. |
| Close Camera View | Attempts to close an open overview video camera window if one is open. |
| Set Scan Profile [] | For Leica AS1 scanner. Sets the RDS scan profile by name, designated by [] ([] not part of string) |
| Start Scan | For Leica AS1 scanner. Begins scanning. |
| Stop Scan | For Leica AS1 scanner. Ends scanning. |
| Get Reflector Position | For Leica AS1 scanner. Gets the reflector center position as a measurement, for a future auto-reset onto the scanner. |
| Set Profile '[1]''[2]'' to [3]'' | <p>Sets a measure profile's parameter to stipulated value. (Profile name designated by [1], string in single quotes. Parameter name designated by [2], string in single quotes. Value [3], include the word "to" then the number without quotes.</p> <p>Currently works for a spatial scan's increment parameter, e.g. [Set Profile 'Spatial Scan to SA' 'Increment' to 0.125]</p> <p>This was added for use with the arm controlled simultaneous arm and tracker measurement feature, to automate setting the tracker's spatial increment (in inches).</p> |

| AT960 Automation with the AS1 Scanner | |
|---------------------------------------|--|
| Connect LMF IO [] | Establishes a connection to the CB21 (Control Box) at the IP address specified by [] ([] not part of string). This command, like all LMF_IO commands, will work whether connected to the AS1 via RDS, or to the AP21 via LMF. But the normal operating mode is with an RDS connection to the AS1 scanner. "AT960Simulator" can be used as an IP address to open the LMFIO simulator. |
| Disconnect LMF IO | Removes the connection to the CB21 Digital IO Interface.. |
| Is Robot Program Active | Checks if a robot program is running. Returns "Success" if the RobotProgramActive I/O value is TRUE (1). Returns Partial success if RobotProgramActive I/O value is FALSE (0). Returns failure if SA cannot read the RobotProgramActive I/O value. |
| Send Robot Go [timeout] | Sets the RobotGo bit to 1. Waits for the robot program to set RobotPositionReached to 0 (the timeout is expressed in seconds) Set RobotGo back to 0. |
| Wait For Robot Position Reached [] | Waits for the robot program to set RobotPositionReached to 1 (the timeout is expressed in seconds, 0 = no timeout!) |
| Switch to LMF/RDS | This command forces a connection either to RDS or LMF (enter LMF or RDS). In addition to switching to the desired interface, this command will also deactivate the Auto Switching. If the connection to the specified interface is already established, this command will deactivate AutoSwitching and immediately return as success. |

| AT960 Automation with the AS1 Scanner | |
|--|--|
| Toggle LMF<>RDS AutoSwitching [] | This command toggles the Auto switching between LMF and RDS On and Off. |
| Get Reflector Position. | This command records the position of the prism of the currently locked on face of the AP21. This command will create a point on the tree view with the name defined on the interface. This is normally used to define the lock-on positions. |
| AutoLock On/Off | Toggle Powerlock On/Off. It works while being connected with either interface RDS/LMF. |
| Valid Distance | When locked on the AS1, this command will return success if the Scanner is Ready To Measure. Can be used before calling «Scan Start» to ensure that the scanner is Ready. |
| Start Scan | Starts the measurement, this is equivalent to pressing the trigger on the scanner. |
| Stop Scan | Stops the measurement, this is equivalent to releasing the trigger on the scanner. |
| Set Scan Profile [] | Sets the RDS scan Profile by name. The available profiles can be found on General Tracker Settings>Leica AT9x0>Available MP Commands. |
| Set Line Scan Thinning Factor [] | Sets the thinning factor for line scanners. |
| Toggle Line Scan Min Angle Filter On/Off | Turns the curvature based filter on and off. |
| Min Angle Filter [] Degrees [] Max Distance mm | Sets the parameters for the curvature based filter. |

| Leica Automation Interface | |
|-----------------------------|--|
| Set Automation Mode On | This puts the AIC Interface in Automation Mode. In this mode, the next 'Measure' command from MPs will cause the Interface to expect Digital I/O signals from the robot to drive scanning. |
| Set Automation Mode Off | This puts the AIC Interface in Manual Mode. In this mode, the next 'Measure' command from MP will simply turn the scanner on. The scanner will stay on until the next "Stop Active Measurement Mode" command from MP. NOTE: This is the default startup mode of the AIC Interface. |
| Select Device Trigger Probe | Set AIC Device Selection to Trigger Probe (emScon). |
| Select Device T-Scan | Set AIC Device Selection to T-Scan. |
| Select Tracker 1 | Set AIC to Tracker 1, Jump SA Instrument to designated Collection::Instrument Index (set in AIC Driver), Connect and Start Interface to T-Scan or emScon based on current MUX and AIC Device Selection |
| Select Tracker 2 | Set AIC to Tracker 2, Jump SA Instrument to designated Collection::Instrument Index (set in AIC Driver), Connect and Start Interface to T-Scan or emScon based on current MUX and AIC Device Selection |
| Select Tracker 3 | Set AIC to Tracker 3, Jump SA Instrument to designated Collection::Instrument Index (set in AIC Driver), Connect and Start Interface to T-Scan or emScon based on current MUX and AIC Device Selection |
| Select Tracker 4 | Set AIC to Tracker 4, Jump SA Instrument to designated Collection::Instrument Index (set in AIC Driver), Connect and Start Interface to T-Scan or emScon based on current MUX and AIC Device Selection |
| Increment Group/Cloud Name | Increment the Current Group/Cloud Name by 1. This name is used for Point Group when measuring points, and Cloud Name when scanning with T-Scan. |
| Release Motors | Release motors if there is a current emScon or T-Scan connection. |

| Leica Automation Interface | |
|--|---|
| Is Laser Locked | For the T-Mac or T-Scan: Returns success if the beam is locked, fail if not locked, and the status window shows what kind of target the laser is locked onto. |
| Is 6D Status Valid | Returns success if the 6D Status is valid, fail otherwise. NOTE: This command only works properly for the TMac interface. The COM connection to TScan Collect does not have a working version of this function—it will return TRUE even if the laser is locked on a 3D target (e.g. SMR). |
| Is Laser Locked on TScan | For use with TScan Collect. Added because “Is 6D Status Valid” does not work properly for TScan Collect. |
| Is Laser Locked on TMac | For use with TScan Collect. Added because “Is 6D Status Valid” does not work properly for TScan Collect. |
| Is Laser Locked on TMac MultiSide | When called with T-Mac active, this will report whether the tracker is locked on the T-Mac MS, and if so, will also report the number of sides, and the locked side. When called with TScan Collect active, added because “Is 6D Status Valid” does not work properly for TScan Collect. |
| Set emScon Measure Time [] | Set the Measure Time for Discrete measurements when Trigger Probe is the Current Device (emScon connection) to that designated by [] Seconds ([] not part of string) |
| Set Scan Point To Point Distance [] | Set T-Scan Point to Point Distance to that designated by [] mm ([] not part of string) |
| Set Scan Line To Line Distance [] | Set T-Scan Line to Line Distance to that designated by [] mm ([] not part of string) |
| Set Scan Maximum Angle of Incidence [] | Set T-Scan Maximum Angle of Incidence to that designated by [] degrees ([] not part of string) |
| Close Scan Gaps Up To [] | Tell T-Scan Collect to close gaps in scan data up to that designated by [] mm ([] not part of string). A setting of 0.0 means do not close gaps. |
| AIC Move Robot | Perform the Go-Position Reached sequence without scanning. |
| Send Confirmed Go | Same as “AIC Move Robot”, but does not wait for position reached, only confirms that Go gets sent. |
| Wait For Position Reached | Only waits for the position reached signal from I/O. |
| Scanner Power On | Turn the T-Scan laser on. |
| Scanner Power Off | Turn the T-Scan laser off. |
| Set Alignment Sphere Radius [] | Set the calibration sphere radius in mm for the Scanner Alignment. |
| Set Alignment Iterations [] | Set the number of iterations for the Scanner Alignment. |
| Set Alignment Use Auto Clipping On | Set the scanner alignment to use auto-clipping. |
| Set Alignment Use Auto Clipping Off | Set the scanner alignment to NOT use auto-clipping. |
| Start Scanner Alignment | Begin the Scanner Alignment data acquisition. Ensure that the scanner is on before calling this. Normally, you will insert an ‘Ask for User Decision’ step after this, so the user can tell the MP that the data collection is complete. NOTE: Use in ‘Manual’ mode with the trigger button, or ‘Automation’ mode with subsequent ‘Measure’ command(s). |
| Calculate Scanner Alignment | Once the data collection is complete from the ‘Start Scanner Alignment’ step, this will Calculate the alignment transform, and if successful, present the alignment error, and give the user a chance to accept the results and apply them, or reject the results. If the alignment error is less than 1500, this step automatically succeeds, and applies the alignment. |
| Reduce Scanner Waviness | Call the scanner ReduceWaviness function. |
| TPWizard Measurement Check 3D | Run automated TrackerPilot Wizard check. |
| TPWizard Measurement Check 6D | Run automated TrackerPilot Wizard check. |

| Leica Automation Interface | |
|---------------------------------|--|
| TPWizard Stylus Check | Run automated TrackerPilot Wizard check. |
| TPWizard Stylus Compensation | Run automated TrackerPilot Wizard compensation. |
| TPWizard Shank Compensation | Run automated TrackerPilot Wizard compensation. |
| TPWizard Create Virtual Stylus | Run automated TrackerPilot Wizard. |
| TPWizard Edit Virtual Stylus | Run automated TrackerPilot Wizard. |
| Start Instrument Interface | Start the Interface for the 'Instrument's ID' in SA. 'Interface Type' 0 runs the SA Laser Tracker interface. 'Interface Type' 1 runs the AIC interface. |
| Point At Target | Point the Instrument Designated by 'Instrument ID' at a 'Point Name' in the Designated "Collection::Group::Target" |
| Set Instrument Group and Target | For the Instrument Designated by 'Instrument ID', this sets the current Collection::Group::Target designated by 'Point Name'. NOTE: The Group name is used as the Cloud name when using the T-Scan. |
| Measure | This causes the T-Scan connected to 'Instrument ID' to begin scanning if the AIC Interface is connected and NOT in Automation Mode. In Automation Mode, this causes the AIC Interface to enter automated measurement with digital I/O handshaking. If the SA Laser Tracker interface is connected, this will cause the current active Measure Profile to be run. |
| Stop Active Measurement Mode | This causes the T-Scan connected to 'Instrument ID' to end scanning if the AIC Interface is connected. If the SA Laser Tracker interface is connected, this will cause the current active Measure Profile to be stopped if measuring. NOTE: For the T-Scan, this command will be called by the robot using the digital I/O handshaking when in Automation Mode. |
| Stop Instrument Interface | Stop the Interface for the 'Instrument's ID' in SA. |

| Leica T-Scan | |
|--|--|
| Start Scan | Begins a scan pass |
| Stop Scan | Ends a scan pass |
| Increment Group/Cloud Name | Increment the Current Group/Cloud Name by 1. This name is used for clouds when scanning. |
| Is Laser Locked | Succeeds if the laser is locked. Fails if not. |
| Release Motors | Releases torque on the tracking motors in order to lock the beam by hand. |
| Set Scan Point To Point Distance [] | Set Point to Point Distance to that designated by [] mm ([] not part of string) |
| Set Scan Line To Line Distance [] | Set Line to Line Distance to that designated by [] mm ([] not part of string) |
| Set Scan Maximum Angle of Incidence [] | Set Maximum Angle of Incidence to that designated by [] degrees ([] not part of string) |
| Set Scan Exposure Time [] | Sets the scanner's exposure time , where [] is in milliseconds ([] not part of string). Use 0.25-20.0ms for T-Scan Collect version 10 and higher, or 0.01-9.98 ms for PROBEScan. |
| Set Scan Width Iteration [] | Sets the width of the scan line where [] is a value from 1-12 ([] not part of string). 0=100%, 12=40% (decrements by 5%). |
| Set Scan Reflection Filter [] | Sets the reflection filter type, where [] is a value between 1 and 4 ([] not part of string). 1 = Standard, 2 = Low, 3 = Medium, 4 = High. |
| Close Scan Gaps Up To [] | Maximum allowable closed gap in mm ([] not part of string). Use 0.0 to disable this option. |
| Scanner Power On | Turns the scanner on. |
| Scanner Power Off | Turns the scanner off. |

Portable CMM Arm Commands

| Portable CMM Arms | |
|---|--|
| Send Measured Points | Send measured points with all measure modes |
| Don't Send Measured Points | Don't send measured points with any measure modes |
| Calibrate | Initiate arm calibration. |
| Auto-Prox with Scanner | For SA Auto-Correspond with Proximity Trigger |
| Auto-Prox with Probe | For SA Auto-Correspond with Proximity Trigger |
| Set Geom Name [] | Set the Geometry Name to text designated by [] ([] not part of string) |
| Mouse Mode On | Put the arm in mouse mode |
| Mouse Mode Off | Take the arm out of mouse mode. |
| Set Stream Points Spatial Increment Inches [] | Set the Spatial Probe Stream Increment to the amount designated by [] in inches. |
| Tracker Meas Control On | Set Stream Points measure mode to control simultaneous tracker measurement. (Same as right-clicking on the Stream Points measurement button and checking the "Control Tracker Meas with Stream On/Off" check box. |
| Tracker Meas Control Off | Set Stream Points measure mode to NOT control simultaneous tracker measurement. (Same as right-clicking on the Stream Points measurement button and unchecking the "Control Tracker Meas with Stream On/Off" check box. |
| "Stream Points Frequency [1] Increment [2] [3]" | Sets the Stream Points profile's sampling frequency ([1] in Hz) and spatial increment ([2]). Units [3] can be "in" or "mm". Example: [Stream Points Frequency 120 Increment 0.1 in] The command also ensures Stream is set active and set to Spatial (not Temporal). This was added for use with the control for simultaneous measurement with a laser tracker, but can be used any time. |

Theodolite & Total Station Commands

| Leica Total Station Interface | |
|-------------------------------|--|
| Set Laser On | Turns on the red laser pointer |
| Set Laser Off | Turns off the red laser pointer |
| PowerSearch | PowerSearch performs an automatic and rapid search for a prism. |
| Set Tracking On | Enables Tracking, an already targeted prism is followed automatically. |
| Set Tracking Off | Disables Tracking |
| Set ATR On | Enables ATR (Automatic Target Aiming) for precision target centering |
| Set ATR Off | Disables ATR centering |

| Theodolite Manager | |
|-----------------------|--|
| Set TwoFace ON | Enable two face measurement |
| Set TwoFace OFF | Disable two face measurement |
| Set SEPOBS ON | Enable Send Front/Back as Separate Observations |
| Set SEPOBS OFF | Disable Send Front/Back as Separate Observations |
| Set MeasMode Standard | Sets the measurement mode to "Standard" (Leica instruments). |
| Set MeasMode Precise | Sets the measurement mode to "Precise" (Leica instruments). |
| Set MeasMode Fast | Sets the measurement mode to "Fast" (Leica instruments). |

| Theodolite Manager | |
|--------------------------|---|
| Set Laser On | Turns on the laser. |
| Set Laser Off | Turns off the laser. |
| Acquire | Attempts to acquire the target. |
| PowerSearch | Performs a PowerSearch for the target (Leica instruments). |
| Query Angles | Perform a query – angles only |
| Query Distance | Perform a query – angles + distance |
| Set Tracking Off | Turns off tracking mode. |
| Set Tracking TrackOnly | Puts the instrument in “track only” mode. |
| Set Tracking Updates | Tells the instrument to begin sending updates. |
| Set Tracking SpatialScan | Puts the instrument into a spatial scan mode (if applicable). |
| Set Tracking StablePoint | Puts the instrument into a stable point mode. |
| Camera Telescope | Sets the camera to telescope on supported scopes |
| Camera Overview | Sets the sets the camera to overview on supported scopes |

* To change Targets use the “Set Instrument Targeting” command and specify the target by name.

Laser Radar Commands

| Nikon Laser Radar | |
|--|---|
| Linearization | Perform linearization test. |
| FlipTest | Perform a flip test. |
| SelfTest | Perform a self test. |
| StareTest | Perform a stare test. |
| LOSeparation <region> | MV200/300 Only - Perform an LO separation test. Region should be Region12, Region23, or Region34. |
| FullComp | Performs a full compensation. |
| QuickComp | Performs a quick compensation (currently functions the same as a full compensation). |
| MCMCalibration <ptgroup> | Performs an MCM calibration using a point group. (Point group syntax is Collection::PtGroupName). |
| MCMCalibration <ptgroup1>, <ptgroup2>, ..., <ptgroupN> | Performs a relative MCM calibration using several point groups. |
| MCMCalibration <ptgroup1>, ..., <ptgroupN> TRUE | Validate a relative MCM calibration using several point groups. |
| AutoFocus | Initiates an autofocus operation. |
| SetFocusLimits <min> <max> | Sets the focus limits using the specified min and max distances. |
| mirror on <mirror> | Set an active mirror, where <mirror> is the name of the mirror to use. |
| mirror off | Disable use of a mirror |
| Start Video | Starts the video display. |
| Stop Video | Stops the video display. |
| HWCapability <capability> | Offers the ability to check if the LR supports hardware capabilities (SVIGrouping, ScanEngineLicense, HighSpeedScan, GaugingEngine, GaugingEngineLicensed, MCMDatabase, NetworkVideo, InternalAccelerometer, VibrationAnalysis, InstDopplerCorrection, GEPartialScanProgress) |

Laser Projector Commands

| Aligned Vision LaserGuide Projector | |
|-------------------------------------|--|
| Pause | Pause the current projection. |
| Resume | Resume the current projection. |
| Current | Project the current pattern (part) file. |
| Next | Project the next pattern file in the current file's folder. |
| Previous | Project the previous pattern file in the current file's folder. |
| Cross | Project the field of view and center crosshair for location. |
| Field Of View | Project the field of view and center crosshair for location. |
| Set Part Name [] | Set active part PATH to that designated by [] ([] not part of string). |
| Add Patterns to Existing Part | Tell the interface to add projections from SA to the current selected part file. |
| Make New Part | Tell the interface to make a new part file, located in the current part file's directory, but carrying the name of the first projection object in the projection from SA. |
| Set Auto Align File [] | Set the path to an alignment file, designated by [] ([] not part of string). This path will be updated in the UI, and will be used for the next Auto Alignment, whether commanded from MP, or from the Auto Align control in the UI. NOTE: The file must have the usual Aligned Vision Target File format: name, x, y, z, phi, theta, and the usual padding of 0's at the end. This is not the same as an ascii file exported from SA, as described in the "Manual Align..." command. |
| Auto Align | Perform and Auto Alignment using the file shown in the "Align" section of the UI. |
| Manual Align from Exported File [] | Start the Manual (mouse drive) Alignment with an ASCII file exported from SA. The existing MP command, "Export ASCII Points", can be used with all default settings, but populate ASCII File Path with the desired file path and name, the Group Name to Export, Data Delimiter of Space, Target Name Format of Target, and Desired Coordinate System of Cartesian. Then, for this command, the [] is the ASCII File Path argument from above ([] not part of string). NOTE: This command expects an ascii file of format "name x y z", NOT a projector Target File as described in the "Set Auto Align File..." command. When this command is processed, the interface will create the AV Target File corresponding to the coordinates in the exported ascii file. The Target File will go to the same directory as the exported file, and will have the same name, but prepended with "AV_". |

| LAP Laser Projector | |
|-------------------------------|--|
| Pause | Pauses the current projection. |
| Resume | Resumes the current projection. |
| Current | Projects the current pattern (part) file. |
| Next | Projects the next pattern file in the current file's folder. |
| Previous | Projects the previous pattern file in the current file's folder. |
| Cross | Toggles the center cross (+) for aiming the projector. |
| Set Part Name [] | Sets the active part PATH to that designated by []. |
| Add Patterns to Existing Part | Adds projections from SA to the current selected part file. |
| Make New Part | Makes a new part file, located in the current part file's directory, but carrying the name of the first projection object in the projection from SA. |

| LAP Laser Projector | |
|------------------------|--|
| Set Auto Align File [] | Set the path to an alignment file, designated by [] ([] not part of string). This path will be updated in the UI, and will be used for the next Auto Alignment, whether commanded from MP, or from the Auto Align control in the UI. The file must have the usual alignment file format: name, x, y, z, phi, theta, search radius. |
| Auto Align | Perform and Auto Alignment using the file shown in the "Align" section of the UI. |
| Set Color Red | Sets the current projection color to red. |
| Set Color Green | Sets the current projection color to green. |
| Set Color Yellow | Sets the current projection color to Yellow. |

| LPT Projector | |
|-----------------------------|--|
| Pause | Pause the current projection. |
| Resume | Resume the current projection. |
| Current | Project the selected ply in the current part. |
| Next | Project the next ply in the current part. |
| Previous | Project the previous ply in the current part. |
| Cross | Project the center cross hair for location. |
| Field Of View | Project the projector's field of view for location. |
| Set Part Name [] | Set active part to that designated by [] ([] not part of string). |
| Add Plies to Projector Part | Tell the interface to add plies from SA to the current part in the database. |
| Add Plies to Offline Part | Tell the interface to add plies from SA to the current offline part. |

| Virtek Iris Projector | |
|--------------------------------|---|
| Pause | Pause the current projection. |
| Resume | Resume the current projection. |
| Current | Project the selected ply in the current part. |
| Next | Project the next ply in the current part. |
| Previous | Project the previous ply in the current part. |
| Cross | Project the center cross hair for location. |
| Field Of View | Project the projector's field of view for location. |
| Select or Create Work Order [] | Select Work Order if a folder is found which matches the name designated by [] ([] not part of string). Otherwise, create a new Work Order folder in the active Work Orders. The new Work Order will be highlighted in the interface's Work Orders list, ready for a subsequent "Add Cal File from Exported File []" command. Also, that new Work Order will not be valid (and cannot be opened) until it contains a valid .cal file and a valid .ply file. These are requirements of Virtek's Server Manager. NOTE: This command will fail if you have not set up a valid Work Cell in Iris. See your Virtek representative for more info regarding initial setup. |

| Virtek Iris Projector | |
|------------------------------------|--|
| Add Cal File from Exported File [] | Add the .cal file to the existing selected Work Order with the points from an ASCII file exported from SA. The existing MP command, "Export ASCII Points", can be used with all default settings, but populate ASCII File Path with the desired file path and name, the Group Name to Export, Data Delimiter of Space, Target Name Format of Target, and Desired Coordinate System of Cartesian. Then, for this command, the [] is the ASCII File Path argument from above ([] not part of string). NOTE: This command expects an ascii file of format "name x y z", NOT a projector .cal file defined in a Work Order. When this command is processed, the interface will create a .cal file corresponding to the coordinates in the exported ascii file. The .cal file will go to the current selected Work Order, and will have the same name, but will have the extension ".cal". This command will fail if there is no current active Work Order. See your Virtek representative for more info regarding initial setup. Also note that until there is also a valid .ply file in the Work Order, Virtek will not allow the Work Order to be Opened. So you will need to add a .ply BEFORE you will be able to run the initial Manual Align on the .cal file in a new Work Order. |
| Auto Align | Perform and Auto Alignment. NOTE: This requires that you have defined and selected a valid Work Order, and that the work order has a valid .cal file in it. This alignment will be performed using the targets defined in that file. |

Photogrammetry System Commands

| AICON MoveInspect | |
|---------------------|---|
| Set MeasureMode [] | Set the mode to "Single", "Continuous", "Targeting", or "Probing" in place of []. Note that there are two modes for each measurement type so you may need to call this command twice to switch between Single/Continuous, then Targeting/Probing. |
| Set Measure Mode [] | Set the mode to "Probing" or "Tracking" in place of []. |
| Set Filter [] | Sets the measurement action to "Coded", "Noncoded", "Adapter-Frames", "AdapterPoints" followed by "True" or "False" in place of []. For example: "Set Filter AdapterFrames True". |
| Connect | Connects to the server and connects SA to MoveInspect. |
| Stop Server | Disconnects the communication between the interface and the MoveInspect software |
| Measure | Initiates a Single Measurement. |
| Measure Continuous | Initiates a Continuous Measurement. |
| Stop | Stops the current measurement. |
| Snapshot | Toggles the current Adapter frame setting from updating an existing frames transform to recording separate frames with each measurement. |
| Set Reference [] | Where [] is the Name of the desired existing dynamic reference system. No name specified deactivates reference. |

| GSI V-STARS | |
|----------------------|---|
| Select Data: Probe | Set interface radio button data type to: Probe |
| Select Data: Targets | Set interface radio button data type to: Targets |
| Select Data: Cameras | Set interface radio button data type to: Cameras |
| Select Data: Cloud | Set interface radio button data type to: Cloud (Pro Spot) |

| GSIV-STARS | |
|--------------------------|---|
| Select Data: Dream | Set interface radio button data type to: Dream Probe |
| Use VStars Target Label | TRUE or False (use the interface target name string) |
| Target Labels Use All [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels CODE [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels NUGGET [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels TARGET [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels _S [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels _T [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Target Labels SB [] | Set Target Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels Use All [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels CODE [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels NUGGET [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels TARGET [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels _S [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels _T [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Cloud Labels SB [] | Set Cloud Labels to use: replace [] with TRUE or FALSE |
| Trigger | MMode Trigger function |
| Take Picture | SMode Data to SA function |
| Open Template | SMode operation used to open a Template file |
| Save Picture | SMode operation used to Save a picture |
| Process Data | SMode operation used to Process Data |
| Send Data | SMode operation used to Send Data |
| Close Project | SMode operation used to Close a Project |

| Metronor Portable Measurement System | |
|--------------------------------------|--|
| Set MeasureMode Single | Set the measurement mode to discrete point measurement. |
| Set MeasureMode Continuous | Set the measurement mode to continuous point measurement. |
| Set Action Points | Sets the measurement action to send points to SA. |
| Set Action Updates | Sets the measurement action to send updates to SA. |
| Set Action Frames | Sets the measurement action to send frames to SA. |
| Set Action Batch | Sets the measurement action to send an LED batch of points to SA. |
| Set Action Unit | Sets the measurement action to send an LED target unit to SA. |
| Set Tip [] | Sets the probe tip to [], where [] must match one of the names in the probe drop down ID list. |

| Nikon Surveyor | |
|------------------------|---|
| Measure [] | Initiate measurement using the provided frame name. |
| 6D Measure Mode On [] | Turn on 6D streaming mode for the named frame. |
| 6D Measure Mode Off [] | Turn off 6D streaming mode for the named frame. |
| Stream 6D On [] | Initiate 6D streaming for the named frame. |
| Stream 6D Off [] | Stop 6D streaming for the named frame. |
| Listen On [] | Starts listening to the named frame. |
| Listen Off [] | Stops listening to the named frame. |

| Vicon Tracker | |
|-----------------------------|--|
| Get Objects | Gets all available Vicon objects, and populates the dropdown list. (Same as clicking the Set Objects button in the main dialog). |
| Set Active Object [] | Set the Vicon object to measure, designated by [] ([] not part of string). |
| Set Measure Mode [] | Set the mode to "Discrete", "Spatial", or "Temporal" in place of []. |
| Set Measurement Time Out [] | Set the time (in seconds) that the interface will wait for a Vicon object (frame) to come into view. |

| Vicon Tracker | |
|---------------------------|--|
| Set Data Type [] | Set type to "3D" or "6D" in place of []. |
| Set SA Data Type [] | Set type to "Measurement" or "Update" in place of []. |
| Set Spatial Increment [] | Sets the spatial increment for the temporal measure mode to the amount in inches designated by [] ([] not part of string). |
| Set Temporal Increment [] | Sets the temporal increment for the temporal measure mode to the amount in seconds designated by [] ([] not part of string). |
| Get Unlabeled Markers | Gets unlabeled markers from the current active Vicon object (see "Set Active Object []" above). |
| Get Labeled Markers | Gets labeled markers from the current active Vicon object (see "Set Active Object []" above). |

Room Scanner System Commands

| Surphaser | |
|--------------------|--|
| Send Scan to SA [] | Sends surphaser scan to SA, where [] is the full path of the ptx file, brackets are not to be included in the command string |

Get Number of Observations on Target

Returns the number of observations of a specified target.

Input Arguments

| | | | |
|---|------------|------------|------------------------------------|
| 0 | Point Name | Point Name | The name of the target to examine. |
|---|------------|------------|------------------------------------|

Return Arguments

| | | | |
|---|---------|-----------------|---|
| 1 | Integer | Number of Shots | The number of observations of the target in question. |
|---|---------|-----------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The number of shots was returned successfully. |
| FAILURE | The point could not be found. |

Remarks

None.

Get Instruments with Observations on Target

Returns the list of instruments that have observations on (measurements of) the specified target.

Input Arguments

| | | | |
|---|------------|------------|-------------------------------------|
| 0 | Point Name | Point Name | The name of the target of interest. |
|---|------------|------------|-------------------------------------|

Return Arguments

| | | | |
|---|-----------------------------------|--|---|
| 1 | Collection Instrument ID Ref List | Resultant Collection Instrument Reference List | The list of instruments that have observations on the target of interest. |
|---|-----------------------------------|--|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of instruments was returned successfully. |
| FAILURE | The specified point could not be found. |

Remarks

None.

Get Targets Measured by Instrument

Returns the list of points measured by an instrument.

Input Arguments

| | | | |
|---|--------------------------|-------------------------|--------------------------------------|
| 0 | Collection Instrument ID | Measuring Instrument ID | The name of the measuring instrument |
|---|--------------------------|-------------------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------------------|-----------------------------------|--|
| 1 | Point Name Ref List | Points Measured by the Instrument | The list of targets that were measured by the instrument identified. |
|---|---------------------|-----------------------------------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of targets was returned successfully. |
| FAILURE | The specified instrument could not be found. |

Remarks

None.

Set Observation Status

Sets the status (active or inactive) for any observation of a target.

Input Arguments

| | | | |
|---|------------|-------------------|--|
| 0 | Point Name | Point Name | The name of the point to modify. |
| 1 | Integer | Observation Index | The index of the observation to modify. Index 0 is the first observation for the target, Index 1 is the second, etc. |
| 2 | Boolean | Active? | Indicates whether the observation should be set as active or not. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The observation was modified successfully. |
| FAILURE | The point or observation index could not be found. |

Remarks

None.

Get Observation Info

Obtains details for a given observation, including the source instrument, azimuth/elevation/distance (for spherical measurement devices), and the status for the observation (active vs. inactive).

Input Arguments

| 0 | Point Name | Point Name | The name of the point to examine |
|---|------------|-------------------|---|
| 1 | Integer | Observation Index | The index of the observation to examine. Index 0 is the first observation for the target, Index 1 is the second, etc. |

Return Arguments

| | | | |
|----|--------------------------|----------------------|--|
| 2 | Collection Instrument ID | Resulting Instrument | The instrument that obtained the observation. |
| 3 | Vector | Resultant Vector | The distance, azimuth, and elevation of the observation (in that order).** |
| 4 | Boolean | Active? | Indicates whether the observation is active. |
| 5 | String | Timestamp | Indicates measurement time accurate to seconds. |
| 6 | Double | RMS Error | RMS error of observation. |
| 7 | Double | Temperature (degF) | Temperature from observation info data |
| 8 | Double | Pressure (in. Hg) | Pressure from observation info data |
| 9 | Double | Humidity (% RH) | Humidity from observation info data |
| 10 | String | Info Data | Full info data string from selected observation |

Returned Status

| | |
|---------|--|
| SUCCESS | The observation information was obtained successfully. |
| FAILURE | The point or observation index could not be found. |

Remarks

A zero will be returned for the RMS error for those observations that do not have them, while -1 will be returned for temperature/pressure and humidity.

**The theta values returned from this check are raw instrument specific values. For example, a point 15 degrees toward Y from X is reported as either 345 or -15 depending on the instrument (theta or 360-theta). Confirm which value your instrument returns, particularly with Total Stations.

Fabricate Observations

Converts a group of constructed points into measured targets by adding fabricated observations to the points.

Input Arguments

| | | | |
|---|--------------------------|-----------------------------|---|
| 0 | Collection Instrument ID | Instrument to shoot | The instrument to measure the specified points. |
| 1 | Collection Object Name | Group name to shoot | The group of constructed points for which to fabricate observations. |
| 2 | Boolean | Introduce instrument error? | True - induces a small random error based on the instruments current uncertainty settings, simulating real life measurements. |
| 3 | Boolean | Limit Distance? | True - limits the fabrication to only points within the Min and Max distances |
| 4 | Double | Min Distance | Min distance used to limit measurement fabrication when Limit Distance is true. |
| 5 | Double | Max Distance | Max distance used to limit measurement fabrication when Limit Distance is true |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The observations were fabricated successfully. |
| FAILURE | The instrument or group was not found. |

Remarks

Random Error generation is only available for polar instruments (Trackers and Total Stations for example). PCMM arms and photogrammetry systems for example are not included in this operation.

Get Obscured Points from Instrument

Returns a list of points that are not visible from a specified instrument plant.

Input Arguments

| | | | |
|---|--------------------------|---------------------|--|
| 0 | Collection Instrument ID | Instrument to shoot | The instrument ID of the instrument in question. |
| 1 | Point Name Ref List | Candidate Points | The list of points to consider |
| 2 | Boolean | Show Obscured Shots | True- displays shot lines to the obscured points |

Return Arguments

| | | | |
|---|---------------------|-----------------|-------------------------|
| 3 | Point Name Ref List | Obscured Points | List of obscured points |
|---|---------------------|-----------------|-------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The obscured points were returned successfully |
| FAILURE | The instrument or points were not found. |

Remarks

None.

Set Instrument Measurement Mode/Profile

Sets a named measurement profile or mode active for an instrument.

Input Arguments

| | | | |
|---|--------------------------|-------------------|---|
| 0 | Collection Instrument ID | Instrument to set | The instrument ID of the instrument in question. |
| 1 | String | Mode/Profile | The string name of the measurement mode or profile. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The mode/profile was set successfully. |
| FAILURE | The instrument, mode, or profile was not found. |

Remarks

For some instruments (listed below), you can pass special strings to obtain additional behavior:

| Portable CMM Arms | |
|-------------------|--|
| Discrete | Single Discrete Point measurement mode. |
| Stream | Scan points per user option setting (spatial or temporal measurement). |
| Patch | Measure patch, or projected point. |
| Pin | Measure pin, or outside circle with projection plane. |
| Hole | Measure hole, or inside circle with projection plane. |
| Slot | Measure slot, or two inside circles with projection plane. |
| Line | Measure line per user option setting (two point, averaged, or edge). |
| Circle | Measure circle, inside, outside, or on face, with no planar offset. |
| Plane | Measure Plane |
| Sphere | Measure Sphere. |
| Section | Measure Cross Sections (multiple if cross value not equal to zero). |
| Frame | Measure frames (origin at probe center, using probe orientation). |
| Batch | Perform Guided Measurement (invoked from SA with Batch of pts). |
| Scanner | If available, use installed line scanner to measure cloud points. |
| Average | Single averaged point. |
| Geom Trigger | Measure across array of planar geometry triggers. |

| Nikon Laser Radar | |
|-------------------|---|
| mirroron <mirror> | Sets an active mirror, where <mirror> is the name of the mirror to use. |
| mirroroff | Disables use of a mirror. |
| StartVideo | Starts the video display. |
| StopVideo | Stops the video display. |

Set Instrument Group and Target

Sets the collection, group and target names in the instrument interface for future measurements.

Input Arguments

| | | | |
|---|--------------------------|---------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Point Name | Point Name | The name of the point to use in the instrument interface. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The point name was set successfully. |
| FAILURE | The instrument was not found. |

Remarks

For instruments that produce a point cloud, i.e. Surphaser, the cloud will take the name of the Group specified and the Collection specified. The target name will be ignored.

Set Instrument Targeting

Sets the targeting (tooling offsets, etc.) for an instrument.

Input Arguments

| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
|---|--------------------------|----------------|---|
| 1 | String | Targeting Name | The name of the target in the instrument interface. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The target was set successfully. |
| FAILURE | The instrument or target name was not found. |

Remarks

For example, in the laser tracker interface, you might use the string SMR: 1.5" Tkr Nest.

Set Target Computation Options

Sets the User Option control available on the Analysis Tab for how points with multiple observations are used in point computation.

Input Arguments

| | | | |
|---|---------------------------|------------------------------|--|
| 0 | Target Computation Method | Target Computation Method | Pick the desired method from the list. |
| 1 | Boolean | Ignore Distance Measurements | True sets the flag to ignore |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------|
| SUCCESS | This command always succeeds |
|---------|------------------------------|

Remarks

None.

Set Observation Mirror Cube Shot Face

Designates a particular observation on a point as a mirror cube shot.

Input Arguments

| | | | |
|---|------------|-------------------------------|--|
| 0 | Point Name | Point Name | Name of Point to Edit |
| 1 | Integer | Observation index | Observation to mark |
| 2 | Boolean | Is Mirror Cube Shot? | True marks the observation as the designated face of a mirror cube |
| 3 | Integer | Mirror Cube shot Face (1...6) | The mirror cube face measured |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The observation was marked successfully |
| FAILURE | The point or index could not be found. |

Remarks

SA uses the convention for cube faces as follows:

- Face 1 - is the top (x)
- Face 2 - is the side 1 (y)
- Face 3 - is the side 2 (z)
- Face 4 - is the side 3 (-y)
- Face 5 - is the side 4 (-z)
- Face 6 - is the attachment or bottom (-x)

Set Observation Collimation Shot Options

Designates a particular observation on a point as a collimation shot.

Input Arguments

| | | | |
|---|--------------------------|----------------------|--|
| 0 | Point Name | Point Name | Name of Point to Edit |
| 1 | Integer | Observation index | Observation to mark |
| 2 | Boolean | Is Collimation Shot? | True marks the observation as collimation shot |
| 3 | Collection Instrument ID | Targeted instrument | Name of the targeted Instrument |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The observation was marked successfully |
| FAILURE | The point or index could not be found. |

Remarks

None.

Collimation

Performs an *Instrument > Collimation* operation

Input Arguments

| | | | |
|----|-----------------------------|-----------------------------------|--|
| 0 | Collection Instrument ID | Stationary Instrument ID | ID of the stationary instrument |
| 1 | Collection Instrument ID | Moving Instrument ID | ID of the instrument to move |
| 2 | Point Name | Collimation Point | Name of the collimation point |
| 3 | Boolean | Zero Moving Instrument | Used to zero the azimuth of the move instrument in the active instrument control |
| 4 | Collimation Type | Collimation Baseline Mode | Choose from Full Collimation or No-Tilt Collimation |
| 5 | Collimation Baseline Method | Collimation Baseline Method | Choose from Determined By Value, From Scale, or from Know Point |
| 6 | Double | Baseline Distance | Used in "Determined by Value" baseline method and to define the length of the scale-bar. |
| 7 | Point Name | Scale Point 1 | First Scale-Bar Point |
| 8 | Point Name | Scale Point 2 | Second Scale-Bar Point |
| 9 | Point Name | Not Measured by Moving Instrument | First measured point |
| 10 | Point Name | As Measured by Moving Instrument | Second measured point |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The observation was marked successfully |
| FAILURE | The point or index could not be found. |

Remarks

There are 3 baseline methods for this command selected in Argument 5. For more details on operation refer to the Instrument Manual under Measuring with Total Stations.

Get Instrument Target Status

Retrieves information on the instrument's target.

Input Arguments

| | | | |
|---|--------------------------|---------------|---------------------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the instrument in question. |
|---|--------------------------|---------------|---------------------------------------|

Return Arguments

| | | | |
|---|---------|-----------------|--|
| 1 | Boolean | Is Locked? | Returns TRUE if the instrument is locked on a target, FALSE if not. |
| 2 | String | Name | The name of the currently active target. |
| 3 | Integer | Number of Faces | The number of faces on the selected target (for Leica's multi-face T-MAC). |
| 4 | Integer | Locked Face | The face currently locked on (for Leica's multi-face T-MAC). |

Returned Status

| | |
|---------|---|
| SUCCESS | The status was retrieved successfully. |
| FAILURE | The specified instrument was not found. |

Remarks

This command only applies to laser trackers.

Make Surface Face List from Point Proximity

Returns a list of surface face ID's (numbers) based on querying a list of points to the available CAD in the file.

Input Arguments

| | | | |
|---|---------------------|-----------------|--|
| 0 | Point Name Ref List | Measured Points | List of points to use for face detection |
|---|---------------------|-----------------|--|

Return Arguments

| | | | |
|---|---------------|-------------------------|-------------------------------------|
| 1 | Surface Faces | Selected Surfaces Faces | Returns a list of surface face ID's |
|---|---------------|-------------------------|-------------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The status was retrieved successfully. |
| FAILURE | The specified points could not be found or no surface were available. |

Remarks

The list of surface faces can be referenced by *Make GD&T Feature Check Annotations* or *Scan CAD Faces* for example.

Scan within Perimeter

Initiates an instrument scan inside of a specified perimeter.

Input Arguments

| | | | |
|---|--------------------------|---------------------|---|
| 0 | Collection Instrument ID | Instrument to scan | The instrument ID of the instrument in question. |
| 1 | Collection Object Name | Scan perimeter name | The name of the perimeter to use. |
| 2 | String | Parameter set name | The named parameter set to use for the scan (from the instrument's interface). |
| 3 | Collection Object Name | Group name | The name of the Point Cloud or group to scan into. |
| 4 | Boolean | Wait for Completion | Indicates whether the MP should pause until this step is complete, or whether it should continue executing while scanning is occurring. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scan was completed successfully. |
| FAILURE | The instrument, scan perimeter, or supplied parameter set name was not found. |

Remarks

There are a number of instrument specific variations on how the *Parameter Set Name* [Argument 2] can be used.

Laser Radar:

When using the laser radar the *Parameter Set Name* should be used to set the desired target definition. Select from the available targets defined for use with a perimeter such as Open or Closed Metrology or Vision Scans.

Laser Trackers:

Leica ATS600

When using the Leica ATS600 the scan resolution can be set using a string in the *Parameter Set Name* field. Use the Dialog entry method in [A2] for more clarity or specify a string directly as follows:

Area Scan Specifications:

- “inPxx” or “mmPxx” to set point to point spacing only
- “inPxxGSWxx” or “mmPxxGSWxx” to set point to point distance (P) and Grow/Shrink Width (GSW)

For example “mmP10” would deliver a scan with 10mm spacing within the specified perimeter.

Use “C” or “P” at the end of any string to specify Point Cloud (C) or Point Group (P) return.

Line Scan Specification (used with both open and closed perimeters):

- “uuPxxLWxx” - line scan using the specified Point to Point distance (Pxx) and Line Width (LWxx)
- “uuPxxLWxxLtoLxx” - cross line scan using the specified Point to Point distance (Pxx), Line Width (LWxx), and Line to Line distance (LtoLxx)

Room Scanners:

Surphaser

Region scans can be accomplished with both *Configure and Measure* and *Scan within Perimeter*. The saved *Parameter Set* specifies naming and resolution for a scan as well as target extraction. But in *Configure and Measure* the saved region definition is used to define the scan extents, where as *Scan within Perimeter* ignores the saved region in preference for the specified perimeter defined in the command. *See notes on zeroing azimuth in the Quickstart Guide.

Scan CAD Faces

Initiates an instrument scan inside of a specified perimeter defined by a CAD face or selection of faces. It also provides the ability to designate exclusion zones within a face perimeter such that holes within a face are excluded.

Input Arguments

| | | | |
|---|--------------------------|---------------------|---|
| 0 | Collection Instrument ID | Instrument to scan | The instrument ID of the instrument in question. |
| 1 | Surface Faces | Surface Faces | List of surface faces to scan |
| 2 | String | Parameter set name | The named parameter set to use for the scan (from the instrument's interface). |
| 3 | Boolean | Enable exclusions? | When true, holes within a selected face will be included as exclusion zones and will not be scanned. |
| 4 | Boolean | Wait for Completion | Indicates whether the MP should pause until this step is complete, or whether it should continue executing while scanning is occurring. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scan was completed successfully. |
| FAILURE | The instrument, scan perimeter, or supplied parameter set name was not found. |

Remarks

Leica ATS600

When using the Leica ATS600 the scan resolution can be set using a string in the *Parameter Set Name* [A2] field. Use the Dialog Entry method for more clarity. The following strings can be built and used directly:

Area Scan Specifications:

- “inPxx” or “mmPxx” to set point to point spacing only where “xx” defines the resolution
- “inPxxGSWxx” or “mmPxxGSWxx” to set point to point distance (P) and Grow/Shrink Width (GSW)

For example “mmP10” would deliver a scan with 10mm spacing within the face perimeter.

You can also add “C” or “P” to the end of the string to specify Point Cloud (C) or Point Group (P) as a return.

Edge Scan Measurement

Commands a laser radar to detect the edge of a part. Two points (a seed or starting point and a point specifying the scan direction) are supplied, and the scan proceeds from the starting point in the direction of the second point and detects an edge. When detected, an interpolated point is created on the edge.

Input Arguments

| | | | |
|---|--------------------------|--------------------------------|--|
| 0 | Collection Instrument ID | Instrument to scan | The instrument ID of the instrument in question. |
| 1 | Point Name | Point near edge | A point near the edge to scan to act as a starting point for the edge detection. |
| 2 | Point Name | Point in edge search direction | A point that defines a vector (from the "Point near edge") for the direction of the edge search. |
| 3 | String | Parameter set name | The name of the instrument's parameter set to use for the scan. |
| 4 | Collection Object Name | Group Name | The group name into which to put the edge point. |
| 5 | String | Target Name | The target name for the edge point. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scan was completed successfully. |
| FAILURE | The instrument, scan perimeter, or supplied parameter set name was not found. |

Remarks

The instrument must be in the active collection.

Track Tape Measurement

Commands an instrument to scan along a set of linearly-arranged retroreflective targets. These may be targets on reflective tape, or individually placed.

Input Arguments

| | | | |
|---|--------------------------|-----------------------|---|
| 0 | Collection Instrument ID | Instrument to scan | The instrument ID of the instrument in question. |
| 1 | Point Name | Point on Tape | A point defining the where search will be initiated for the first retroreflective target. |
| 2 | Point Name | Point on Part | A point used to orient the scan for the target. |
| 3 | Point Name | Point for Direction | A point that, coupled with the "Point on Tape", defines the scan direction. |
| 4 | Point Name | Point for Termination | A point defining the end of the scan. When the scan extends past this point, the search terminates. |
| 5 | String | Parameter set name | The named parameter set from the instrument interface indicating the parameters for the scan. |
| 6 | Collection Object Name | Group Name | The name of the group into which to place the detected target centers. |
| 7 | String | Initial Target Name | The name for the first detected target. Influences the names of subsequent detected targets. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The scan was completed successfully. |
| FAILURE | The instrument, point on tape, point on part, point for direction, point for termination, or parameter set name were not found. |

Remarks

The instrument must be in the active collection.

Auto Measure Points

Automatically measures a set of points. This command points an instrument at a set of reference points (in the “reference group”), performs a search for a nearby target, then initiates a measurement after locking onto the target.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Reference Group Name | The group of reference (nominal) points for the measurement. |
| 2 | Collection Object Name | Actuals Group Name (to be measured) | A group name in which to place the measured points. |
| 3 | Boolean | Show complete dialog? | Indicates whether the full auto-measure dialog is displayed, or whether an abbreviated dialog is displayed. |
| 4 | Boolean | Wait for Completion? | Indicates whether the MP should pause on this command while the measurement is being performed, or whether the MP should continue executing subsequent commands while the measurement is occurring. |
| 5 | Boolean | Auto Start? | Indicates whether measurement should automatically start, or whether it should be triggered by the user. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurements were completed successfully. |
| FAILURE | The instrument or reference group were not found, or measurement failed. |

Remarks

The measured points, while placed into the “Actuals” group, inherit the same target names as the reference points.

Auto-Measure Vectors

Automatically measures a set of nominal vectors. The instrument is first pointed at the base of the first nominal vector (the “seed” point), and a search is initiated to lock onto a reflector (for non-targetless instruments). A measurement is taken and a vector is created between the measured point and the seed point.

If the Project Point to Vector argument is set to TRUE, the measured point will first be projected to a vector, and the vector will instead be created between the projected point and the seed point. In the case of targetless instruments such as a laser radar, the projected point will be used as the new “seed” point for another measurement, and the process will be repeated until the measured point is within a specified tolerance of the nominal vector (defined by the measurement profile).

Input Arguments

| | | | |
|---|--------------------------|-------------------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Vector Group Name | The name of the nominal vector group to measure. |
| 2 | Collection Object Name | Actuals Group Name (to be measured) | A group name for the measured points. |
| 3 | Boolean | Project Point to Vector | Indicates whether the measured point should first be projected to the nominal vector before creating the resulting vector. |
| 4 | Double | Angle Tolerance | Indicates the maximum acceptable acute angle between the line of sight and the nominal vector. Angles above this threshold are skipped. |
| 5 | Double | High Tolerance | A high tolerance to apply to the resulting vector group (for colorization purposes). |
| 6 | Double | Low Tolerance | A low tolerance to apply to the resulting vector group (for colorization purposes). |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurements were completed successfully. |
| FAILURE | The instrument or reference vectors were not found, or measurement failed. |

Remarks

The angle tolerance is primarily for targetless devices such as a laser radar, in which measurements are either not possible or not accurate above some threshold. In those cases, this tolerance saves a significant amount of time on measurement.

Auto-Measure Surface Vector Intersections

This command initiates a surface vector intersection measurement process currently supported by the Laser Radar and ATS that searches for the surface vector intersection point and measures that location for each vector in the group.

Input Arguments

| | | | |
|---|--------------------------|------------------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Vector Group Name (to be measured) | Vector group to be measured. |
| 2 | Collection Object Name | Resulting Group Name | Name of the resulting Point Group |
| 3 | Boolean | Wait for Complete | Indicates whether the script should block (hold) until the auto-measurement is complete. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurement was completed successfully. |
| FAILURE | The instrument or object was not found, or measurement failed. |

Remarks

None.

Auto-Measure Specified Geometry

Automatically measures specified geometry with the Laser Radar.

This function currently supports perimeters and circles (holes).

Input Arguments

| | | | |
|---|--------------------------|-------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Object | The object to measure. |
| 2 | String | Mode/Profile | The measurement profile/mode to use for the measurement. |
| 3 | Boolean | Wait for Complete | Indicates whether the script should block (hold) until the auto-measurement is complete. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurement was completed successfully. |
| FAILURE | The instrument or object was not found, or measurement failed. |

Remarks

None.

Auto-Correspond Closest Point

Initiates the Auto-Correspond Closest Point measurement routine. This method automatically names measured points to match the names of the reference points.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Reference Group Name | The reference (nominal) points to measure. |
| 2 | Collection Object Name | Actuals Group Name (to be measured) | A group name for the measured points. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The measurements were completed successfully. |
| FAILURE | The instrument or reference group was not found, or measurement failed. |

Remarks

The instrument must be in the active collection, and its interface must be running.

Auto-Correspond with Proximity Trigger

Initiates the Auto-Correspond with Proximity Trigger measurement routine. This method automatically triggers measurement when the probe has moved within a definable threshold of a nominal point group or vector group.

Input Arguments

| | | | |
|----|--------------------------|---|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Nominal Point Group or Vector Group | The nominal points or vectors to measure. |
| 2 | Collection Object Name | Results Point Group for measurements | The collection/group name for the resulting measured points. |
| 3 | Double | Point distance threshold | The probe must be within this radius from the base of the vectors to trigger measurement. |
| 4 | Double | Vector axis threshold | The probe must be within this radius from the nominal vector axis to trigger measurement. |
| 5 | Boolean | Project results to nominal vector | Indicates whether the measured points should be projected to the nominal vectors. |
| 6 | Double | Warbler ramp start zone distance | The warbler audio tone will begin ramping up when the probe is at this radius from the nominal vector. |
| 7 | Boolean | Show Watch window on startup | Indicates whether a watch window depicting deviation between the current probe position and the closest vector should be displayed. |
| 8 | Vector Group Name | Vector Group to make while Measuring (blank means ignore) | If specified, a vector group will be created depicting deviation between the nominal vector base and the measured point (or projected measured point, if argument 5 is TRUE). |
| 9 | Boolean | Make unmeasured group when done | Indicates whether a point group of the points that weren't measured should be created. |
| 10 | Boolean | Measure each point only once | False allows multiple observations per point to be collected as the probe approaches the target |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The measurements were completed successfully. |
| FAILURE | The instrument or reference group/vector was not found, or measurement failed. |

Remarks

The instrument must be in the active collection, and its interface must be running. The measure each point only once check box control was added to facilitate shank measurement and other applications where the closest sample to the target points should be obtained. This option should be set to True when using a scanner.

Drifts the Mirror From Plane

Creates a Drifts the Mirror interface to match a specified plane.

Input Arguments

| | | | |
|---|--------------------------|--------------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument for defining the mirror. |
| 1 | Collection Object Name | Reference Group Name | The name of the reference point group. |
| 2 | Collection Object Name | Plane | The plane to use to define the mirror. |
| 2 | Collection Object Name | Actuals Group (to be measured) | placed. |
| 3 | Double | Tolerance | An allowable tolerance for each individual point being checked. |

Return Arguments

None.

Return Arguments

Returned Status

| | |
|-----------------|---|
| SUCCESS | The mirror was constructed successfully. |
| FAILURE | The instrument or plane could not be found. |
| SUCCESS | The measurements were completed successfully and tolerances were not exceeded. |
| PARTIAL SUCCESS | Measurement completed, but at least one point exceeded the allowable tolerance. |
| FAILURE | The instrument or reference group was not found, measurement failed, or all measurements exceeded specified tolerances. |

None.

Remarks

The instrument must be in the active collection, and its interface must be running.

Create New Dynamic Reference

Used specifically with AICON (Hexagon) MoveInspect systems.

Input Arguments

| | | | |
|---|--------------------------|-----------------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The name of the master instrument. |
| 1 | Point Name Ref List | Points defining Dynamic Reference | The points selected in this list are used to define the reference system |
| 2 | String | Dynamic Reference Name | The name used for this reference system |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The Dynamic Reference Frame was passed to MoveInspect |
| FAILURE | The reference frame could not be defined |

Remarks

None.

Calculate TCP Fixture Uncertainties

This will process measured points using closest point associations to TCP Fixture nominal points and a best-fit point-to-point transform performed with respect to (wrt) the TCP frame to determine the uncertainty covariance matrix wrt the TCP frame.

Input Arguments

| | | | |
|---|------------------------|----------------------|------------------------------|
| 0 | Collection Object Name | TCP Fixture | The name of the fixture |
| 1 | Transform | TCP in Working Frame | Working transform of the TCP |
| 2 | Point Name Ref List | TCP Measurements | List of TCP Measurements |

Return Arguments

| | | | |
|----|-------------|------------------------------------|--------------------------------------|
| 3 | Boolean | Solution Valid | True indicates the solution is valid |
| 4 | Transform | Refined TCP in Working Frame | |
| 5 | Double List | Uncertainties in TCP Fixture Frame | |
| 6 | Double List | Uncertainties in Working Frame | |
| 7 | Double | RMS Error | |
| 8 | Double | Max Abs Error | |
| 9 | Double | Goodness of fit | |
| 10 | Double | Robustness | |
| 11 | Edit Text | Result Notes | |

Returned Status

| | |
|---------|--|
| SUCCESS | The TCP Fixture Uncertainties were computed successfully |
| FAILURE | The inputs were not valid |

Remarks

None.

Construct TCP Fixture

Creates an entity that will compare newly measured points to reference nominal points to determine the uncertainty covariance matrix with respect to the TCP frame.

Input Arguments

| | | | |
|---|------------------------|------------------------------|---|
| 0 | Collection Object Name | TCP Fixture | The name of the fixture |
| 1 | Double | Point Match Threshold | |
| 2 | Boolean | Replace Existing TCP Fixture | True indicates replace the existing Fixture if found. |
| 3 | Collection Object Name | Resulting TCP Fixture | |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The TCP Fixture Uncertainties were computed successfully |
| FAILURE | The inputs were not valid |

Remarks

None.

Add Nominal Point to TCP Fixture

Provides an entry method for adding nominal points and their associated uncertainty covariance matrices to the TCP Fixture.

Input Arguments

| | | | |
|---|------------------------|------------------------|-------------------------|
| 0 | Collection Object Name | TCP Fixture | The name of the fixture |
| 1 | String | Nominal Point Name | |
| 2 | Vector | Nominal Point Location | |
| 3 | Double | Var XX | |
| 4 | Double | Var YY | |
| 5 | Double | Var ZZ | |
| 6 | Double | CoVar XY | |
| 7 | Double | Covar XZ | |
| 8 | Double | Covar YZ | |

Return Arguments

None

Returned Status

| | |
|---------|--|
| SUCCESS | The nominal point was added successfully |
| FAILURE | The TCP Fixture could not be found. |

Remarks

None.

Get Last Solved TCP Fixture Uncertainty Covariance Matrix

Provides access to the last performed TCP Fixture uncertainty determination which can be used to set the instrument base uncertainties with respect to (wrt) instrument base using the “Set Instrument Base Uncertainty Covariance Matrix WRT Base” MP command.

Input Arguments

| | | | |
|---|------------------------|-------------|-------------------------|
| 0 | Collection Object Name | TCP Fixture | The name of the fixture |
|---|------------------------|-------------|-------------------------|

Return Arguments

| | | | |
|---|-------------|-------------|--|
| 1 | Double List | Covar Row 1 | |
| 2 | Double List | Covar Row 2 | |
| 3 | Double List | Covar Row 3 | |
| 4 | Double List | Covar Row 4 | |
| 5 | Double List | Covar Row 5 | |
| 6 | Double List | Covar Row 6 | |

Returned Status

| | |
|---------|--|
| SUCCESS | Covariance Matrix was returned successfully. |
| FAILURE | The TCP Fixture could not be found. |

Remarks

None.

Set Instrument Base Uncertainty Covariance Matrix WRT Base

Provides a method for setting the uncertainty covariance matrix for an instrument with respect to (wrt) the instrument base frame.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | |
| 1 | Double List | Covar Row 1 | |
| 2 | Double List | Covar Row 2 | |
| 3 | Double List | Covar Row 3 | |
| 4 | Double List | Covar Row 4 | |
| 5 | Double List | Covar Row 5 | |
| 6 | Double List | Covar Row 6 | |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Instrument Uncertainty was set successfully. |
| FAILURE | The Instrument could not be found. |

Remarks

None.

Set Instrument Base Uncertainty Covariance Matrix WRT World

Provides a method for setting the uncertainty covariance matrix for an instrument with respect to (wrt) the WORLD frame.

Input Arguments

| 0 | Collection Instrument ID | Instrument ID | |
|---|--------------------------|---------------|--|
| 1 | Double List | Covar Row 1 | |
| 2 | Double List | Covar Row 2 | |
| 3 | Double List | Covar Row 3 | |
| 4 | Double List | Covar Row 4 | |
| 5 | Double List | Covar Row 5 | |
| 6 | Double List | Covar Row 6 | |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | Instrument Uncertainty was set successfully. |
| FAILURE | The Instrument could not be found. |

Remarks

None.

Get Instrument Base Uncertainty Covariance Matrix WRT World

Provides a method for retrieving the uncertainty covariance matrix for an instrument with respect to (wrt) the WORLD frame.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | |
|---|--------------------------|---------------|--|

Return Arguments

| | | | |
|---|-------------|-------------|--|
| 1 | Double List | Covar Row 1 | |
| 2 | Double List | Covar Row 2 | |
| 3 | Double List | Covar Row 3 | |
| 4 | Double List | Covar Row 4 | |
| 5 | Double List | Covar Row 5 | |
| 6 | Double List | Covar Row 6 | |

Returned Status

| | |
|---------|--|
| SUCCESS | Instrument Uncertainty was set successfully. |
| FAILURE | The Instrument could not be found. |

Remarks

None.

Construct Measured Point Uncertainty Ellipsoids

Provides the capability for displaying the uncertainty covariance matrix for a measured point that is the combined uncertainty of the measurement with respect to the instrument base frame and the uncertainty of the instrument base with respect to the WORLD frame. Using this function allows the user to explore the results when uncertainty inputs are varied.

Input Arguments

| | | | |
|---|---------------------|--------------|--|
| 0 | Point Name Ref List | Measurements | Points from which the uncertainty values are recorded. |
|---|---------------------|--------------|--|

Return Arguments

None

Returned Status

| | |
|---------|---|
| SUCCESS | The ellipsoids were computed successfully. |
| FAILURE | The points were not found or did not contain uncertainties. |

Remarks

None.

Measure Nominal Feature

Points a laser radar at a nominal feature (e.g. hole/circle) and performs a measurement.

Input Arguments

| | | | |
|---|--------------------------|----------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Feature Name | The name of the feature to measure. |
| 2 | Point Name | Resulting Point Name | The resulting measured point name. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The feature was measured successfully. |
| FAILURE | The instrument or feature was not found, or measurement failed. |

Remarks

The instrument must be in the active collection, and its interface must be running.

The measured feature will match the nominal feature in type, and its name will reflect the point name provided in argument 2.

Guide Objects in 6D based on Point Measurements

Initiate the “Guide Objects in 6D using point measurements” build routine. This routine moves objects in 6-DOF based on discrete 3-DOF point measurements.

Input Arguments

| | | | |
|---|---------------------------------|---|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to perform the measurement. |
| 1 | Collection Object Name | Destination Group (goal) | The group representing the preferred destination for a moving reference group. |
| 2 | Collection Object Name | Moving Reference Group (attached to objects) | A moving point group that, when in the goal position, lines up with the destination group. |
| 3 | Collection Object Name Ref List | Objects to Move | A list of objects (other than the moving reference group) to move along with the moving group. |
| 4 | Collection Object Name | Initially surveyed Group (First Position Measurements - Optional) | An optional group of points to be measured in the initial position. |
| 5 | Vector Tolerance | Positional Tolerance - Optional | A vector describing the allowable X, Y, and Z tolerances on the goal position. |
| 6 | Vector Tolerance | Rotational Tolerance - Optional | A vector describing the allowable Rx, Ry, and Rz tolerances on the goal position. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The measurement completed successfully. |
| PARTIAL SUCCESS | One or more objects to move (but not all) could not be found. |
| FAILURE | The instrument, destination group, moving reference group, objects to move, or initially surveyed group could not be found. |

Remarks

The instrument must be in the active collection, and its interface must be running.

Zero tolerances indicate that no tolerance should be applied.

Move Objects in 6D using Instrument Updates

Uses transformation data from a 6-DOF device (such as an API STS or Leica T-Mac) to apply 6-DOF transformations to a list of objects.

Input Arguments

| | | | |
|---|---------------------------------|------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument used to measure. |
| 1 | Collection Object Name Ref List | Objects to Move | A list of objects to move. |
| 2 | String | Measurement Mode | The 6-DOF measurement mode to use when measuring. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The mode completed successfully. |
| PARTIAL SUCCESS | One or more objects to move (but not all) could not be found. |
| FAILURE | The instrument, objects to move, or measurement mode could not be found, or measurement failed. |

Remarks

The instrument's interface must be running.

Align Two Targets with Axis (WCF-X)

Guides the user through the alignment of two points along the working coordinate frame's X axis.

Input Arguments

| | | | |
|---|--------------------------|---------------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to use for measurement. |
| 1 | Point Name | First Point On Axis | The first of two points defining the axis to align. |
| 2 | Point Name | Second Point On Axis | The second of two points defining the axis to align. |
| 3 | Collection Object Name | Initial Measured Group | The first group into which to place the pair of measured points. |
| 4 | Vector Tolerance | Rotational Tolerance - Optional | A tolerance on the Y and Z components will colorize the pitch/yaw results based on their current condition. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The mode completed successfully. |
| FAILURE | The instrument or first/second point could not be found. |

Remarks

The instrument must be in the active collection and its interface must be running. The axis is defined from the first point to the second point. On each successive measurement of first/second points, the group name will be incremented.

Get Instrument Interface Response Timeout

Obtains the instrument interface response timeout value for the specified instrument.

Input Arguments

| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
|---|--------------------------|--------------------------------|--|
| 1 | Double | Resulting Timeout Value (secs) | The current timeout value of the instrument, in seconds. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The timeout was obtained successfully. |
| FAILURE | The instrument was not found. |

Remarks

None.

Set Instrument Interface Response Timeout

Sets the instrument interface response timeout value for the specified instrument.

Input Arguments

| | | | |
|---|--------------------------|----------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument in question. |
| 1 | Double | Timeout (secs) | The desired timeout value of the instrument, in seconds. |

Return Arguments

None.

Returned Status

| | |
|---------|-----------------------------------|
| SUCCESS | The timeout was set successfully. |
| FAILURE | The instrument was not found. |

Remarks

None.

Get Current Trapping Status

Returns that status of trapping including whether or not trapping is active and if so, which feature has focus for inspection.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------------------------|-----------------------------------|---|
| 0 | Boolean | Trapping Active? | True indicates that trapping is active. |
| 1 | Collection Object Name | Relationship / Feature Check Name | The name of the feature that currently has focus in the inspection bar. |
| 2 | Collection Instrument ID | Instrument ID | ID of the instrument with trapping focus. |

Returned Status

| | |
|---------|------------------------------|
| SUCCESS | This command always succeeds |
|---------|------------------------------|

Remarks

None.

Wait For Trapping To Complete

Provides a means for a script to wait for a trapping activity to complete prior to progressing to the next step.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------|
| SUCCESS | This command always succeeds |
|---------|------------------------------|

Remarks

If Trapping is not active this step will not have any influence on the MP operation, but if it is, it will wait for trapping to advance in order for the script to continue.

Jump Instrument to New Location

Stops a live instrument's interface, adds a new instrument, starts that new instrument, and optionally hides the previous instrument.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|---|
| 0 | Collection Instrument ID | Live Instrument ID | The instrument ID of a live instrument. |
| 1 | Boolean | Hide the Previous Instrument? | Indicates whether the previous instrument should be hidden. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The new interface was started successfully. |
| FAILURE | The instrument was not found or its interface was not running. |

Remarks

None.

Quick Align

Aligns an instrument to one or more CAD surfaces using QuickAlign to CAD.

Input Arguments

| | | | |
|---|-----------------------------------|--|---|
| 0 | Collection Instrument ID Ref List | Instrument IDs | A list of one or more instruments to apply the resulting transform to. |
| 1 | Collection Object Name Ref List | Objects | A list of one or more objects to transform during the alignment. |
| 2 | Point Name Ref List | Nominal Points (optional) | Nominal points to measure. |
| 3 | String Ref List | Nominal Point of View Names (optional) | A list of named views that (sequentially) match the list of nominal points. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The new instruments were aligned successfully. |
| FAILURE | The instrument, objects, nominal points (if specified), or nominal views (if specified) could not be found. |

Remarks

None.

Align Cloud to CAD

Aligns cloud to CAD surfaces and return the resultant transform, RMS, average and max errors.

Input Arguments

| | | | |
|---|---------------------------------|-------------------------------------|---|
| 0 | Collection Object Name | Cloud Name | The name of the cloud to align. |
| 1 | Collection Object Name Ref List | Surfaces | The name of the reference surface to be used in the alignment. |
| 2 | Double | Maximum Course CAD Mesh Edge Length | The mesh edge length defines its resolution. Use a larger number for a faster solution. |
| 3 | Boolean | Use Fine CAD Mesh (50% of Coarse)? | Use a fine mesh in the alignment process. |
| 4 | Boolean | Execute Alignment? | Choose whether to move instrument during the alignment. |

Return Arguments

| | | | |
|---|-----------|--------------------------------------|---|
| 5 | Double | RMS Deviation | Resulting RMS deviation of the alignment. |
| 6 | Double | Average Deviation | Average deviation of the alignment. |
| 7 | Double | Maximum Absolute Deviation | Maximum deviation of the alignment. |
| 8 | Transform | Resultant Transform in Working Frame | The data transformation after alignment. |

Returned Status

| | |
|---------|--|
| SUCCESS | The cloud was aligned successfully. |
| FAILURE | The cloud or surface could not be found. |

Remarks

Mesh Edge length (A2) used to be a fixed 10mm but that made the conversion process take a lot time with large models. Use the finest mesh for the best results but this will take longer to compute. Values <5mm will be automatically set to exactly 5mm.

With *Execute Alignment* disabled the opportunity exists to move other objects in the job (such as CAD surfaces) using the inverse of the resulting analytical transform.

Start GD&T Inspection Design

Initiates a GD&T inspection design routine.

Input Arguments

| | | | |
|---|-----------------|----------------------------|---|
| 0 | Collection Name | Collection Name | The collection containing the GD&T inspection routine to design. |
| 1 | String | Filter (ALL/CHECKS/DATUMS) | Indicates whether all checks should be designed, whether only feature checks should be designed, or only datums should be designed. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The inspection design routine was completed successfully. |
| FAILURE | The collection was not found or did not contain any GD&T feature checks/datums. |

Remarks

None.

Start GD&T Inspection Rehearse

Initiates a GD&T inspection rehearsal routine.

Input Arguments

| | | | |
|---|-----------------|----------------------------|--|
| 0 | Collection Name | Collection Name | The collection containing the GD&T inspection routine to rehearse. |
| 1 | String | Filter (ALL/CHECKS/DATUMS) | Indicates whether all checks should be rehearsed, only feature checks should be rehearsed, or only datums should be rehearsed. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The inspection rehearsal was completed successfully. |
| FAILURE | The collection was not found or did not contain any GD&T feature checks/datums. |

Remarks

None.

Start GD&T Inspection

Initiates a GD&T inspection routine.

Input Arguments

| | | | |
|---|--------------------------|----------------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument to use for inspection. |
| 1 | Collection Name | Collection Name | The name of the collection containing the GD&T inspection. |
| 2 | String | Filter (ALL/CHECKS/DATUMS) | Indicates whether all checks should be inspected, only feature checks should be inspected, or only datums should be inspected. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The inspection routine was initiated successfully. |
| FAILURE | The instrument, collection, or GD&T inspection routine could not be found. |

Remarks

None.

Set Remeasure Failed Checks Only

This function mimics the option available through the right-click Feature Check category option in the tree. It clears the points from any failed feature checks, sets lockout trapping for the passing checks and starts Trapping to the first check in the list.

Input Arguments

| | | | |
|---|-----------------|-----------------|------------------------|
| 0 | Collection Name | Collection Name | Collection to consider |
|---|-----------------|-----------------|------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The inspection routine was initiated successfully. |
| FAILURE | The collection could not be found. |

Remarks

None.

Associate Objects with Instrument

Associates one or more objects with an instrument.

Input Arguments

| | | | |
|---|---------------------------------|---------------|---|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument to associate with. |
| 1 | Collection Object Name Ref List | Objects | A list of one or more objects to associate with the instrument. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The association was successful. |
| PARTIAL SUCCESS | One or more objects (but not all) could not be found. |
| FAILURE | The instrument or objects could not be found. |

Remarks

None.

Disassociate Objects from Instrument

Disassociates one or more objects from an instrument.

Input Arguments

| | | | |
|---|---------------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | The collection instrument ID of the instrument to disassociate from. |
| 1 | Collection Object Name Ref List | Objects | A list of one or more objects to disassociate from the instrument. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The disassociation was successful. |
| PARTIAL SUCCESS | One or more objects (but not all) could not be found. |
| FAILURE | The instrument or objects could not be found. |

Remarks

None.

Make Collection Object Name Ref List from Objects associated with Instrument

Creates a list of objects associated with one or more specified instruments.

Input Arguments

| | | | |
|---|-----------------------------------|----------------|--------------------------------------|
| 0 | Collection Instrument ID Ref List | Instrument IDs | The list of instruments in question. |
|---|-----------------------------------|----------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------------------------------|---------------------------------------|--|
| 1 | Collection Object Name Ref List | Resultant Collection Object Name List | The list of objects associated with the provided instrument. |
|---|---------------------------------|---------------------------------------|--|

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | One or more instruments was found. |
| FAILURE | No reference instrument was found. |

Remarks

None.

Combine Point Groups

This command duplicates the menu operation *Instrument>Measurement Grouping> Combine Groups (for Bundling)*. This will build a new group collecting the measurement observations from the commonly measured points.

Input Arguments

| | | | |
|---|---------------------------------|----------------------|---|
| 0 | Collection Object Name Ref List | Groups to Combine | All the point groups that are to be combined. |
| 1 | Collection Object Name | Combined Point Group | New combined point group name. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | One or more point groups were found. |
| FAILURE | No point groups were found. |

Remarks

None.

Dissect Point Groups

Dissects a point group.

Input Arguments

| 0 | Collection Object Name | Group to Dissect | The point group to dissect. |
|---|------------------------|-------------------------------|--|
| 1 | String | Base New for Dissected Groups | The base name for the groups that will be dissected. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Point group was dissected successfully. |
| FAILURE | No point group was found. |

Remarks

None.

Synchronized Measurement (Master/Slave)

Initiates a synchronized measurement between two instruments.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|--|
| 0 | Collection Instrument ID | Master Instrument | The name of the master instrument. |
| 1 | Collection Instrument ID | Slave Instrument | The name of the slave instrument. |
| 2 | String | Slave Group Suffix | The suffix to be used on points in the slave group. |
| 3 | Boolean | Locate One of the Instrument? | Designate whether to locate an instrument. |
| 4 | Boolean | Locate Master (False = Slave) | Choose which instrument you want to locate. |
| 5 | Boolean | Wait for Completion? | Whether to wait to finish measurement before MP moves to next command. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | Both instruments were found and measurement window opens. |
| FAILURE | The master and/or slave instrument were not found. |

Remarks

None.

Crib Sheet Operations

Run Crib Sheet

Starts execution of a crib sheet.

Input Arguments

| | | | |
|---|--------------------------|-----------------|---|
| 0 | Collection Name | Collection Name | The name of the collection containing the crib sheet. |
| 1 | String | Crib Sheet Name | The name of the crib sheet to execute. |
| 2 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument to use for measurement. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The crib sheet was executed successfully. |
| FAILURE | The collection, crib sheet, or instrument were not found. |

Remarks

The instrument must be in the active collection and its interface must be running.

Laser Projection

Project Objects

Initiates projection of a list of objects on a projector.

Input Arguments

| | | | |
|---|---------------------------------|--------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the projector to use. |
| 1 | Collection Object Name Ref List | Objects to Project | The list of objects to project with the projector. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The projection was started successfully. |
| PARTIAL SUCCESS | At least one (but not all) objects was not found. |
| FAILURE | The instrument or objects were not found. |

Remarks

None.

Stop Projection

Stops projection on a specified projector.

Input Arguments

| | | | |
|---|--------------------------|---------------|---|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the projector to stop. |
|---|--------------------------|---------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The projection was stopped successfully. |
| FAILURE | The instrument could not be found, or the projection could not be stopped. |

Remarks

The projector must be in the active collection, and its interface must be running.

Advanced Instrument Operations

Issue Instrument Actuator Command

Sends a command string to an instrument interface. This function is used as a general method to send strings to an interface port. The instrument interface manages the output interface protocol. The command strings are specific to the instrument interface.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the instrument of interest. |
| 1 | String | Command | The string command to send to the instrument. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The command was sent successfully. |
| FAILURE | The instrument could not be found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Set Instrument Axes

Sets instrument axis positions (joint values), typically used with robot interfaces.

Input Arguments

| | | | |
|---|--------------------------|----------------------|--|
| 0 | Collection Instrument ID | Instrument to Adjust | The instrument ID of the instrument to adjust (typically a robot). |
| 1 | Double List | Axis Values | A list of joint values for the instrument. The sequences of values in this list correspond to the sequence of joints in the interface model. |
| 2 | Integer | Number of Steps | The number of animation steps used in the graphical view to animate the instrument model to the new position. |

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------|
| SUCCESS | The axes were set successfully. |
| FAILURE | The instrument was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Set Alignment Projector

Initiates a projector's alignment routine.

Input Arguments

| | | | |
|---|--------------------------|-------------------|--|
| 0 | Collection Instrument ID | Instrument ID | The instrument ID of the projector of interest. |
| 1 | String | Projector Profile | The name of the profile to trigger on the projector. |
| 2 | String | User Prompt | A prompt to display to the user prior to initiating the routine. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The routine was initiated successfully. |
| FAILURE | The instrument or profile was not found. |

Remarks

The instrument must be in the active collection, and its interface must be active.

Nikon Metrology - Laser Radar (LR)

LR Hardware Disconnect

Disconnects the LR interface from the LR hardware.

Input Arguments

| | | | |
|---|--------------------------|---------------|------------------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR of to disconnect. |
|---|--------------------------|---------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The LR was disconnected successfully. |
| FAILURE | The LR could not be disconnected. |

Remarks

None.

LR Verify Hardware Connection

Verifies that the LR hardware is connected to the interface.

Input Arguments

| | | | |
|---|--------------------------|---------------|-----------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to verify. |
|---|--------------------------|---------------|-----------------------------|

Return Arguments

| | | | |
|---|---------|------------------------|--|
| 1 | Boolean | Connected to Hardware? | Indicates whether hardware is connected. |
|---|---------|------------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR Set Red Laser Intensity

Sets the intensity of the red laser on the LR.

Input Arguments

| | | | |
|---|--------------------------|-------------------|-------------------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to verify. |
| 1 | Integer | Intensity (0-100) | The intensity for the laser to use. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The intensity was set successfully. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR Self Test

Initiates a self-test on the laser radar, and returns results.

Input Arguments

| | | | |
|---|--------------------------|---------------|--|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR on which to perform the test. |
|---|--------------------------|---------------|--|

Return Arguments

| | | | |
|----|---------|--|---|
| 1 | Double | Ref Arm Length (Inches) | The ref arm length, in inches. |
| 2 | Double | Ref Arm Quality | The ref arm quality. |
| 3 | Integer | Mirror Measurement Count | The mirror measurement count. |
| 4 | Double | Mirror Measurement Range - Mean (Inches) | The mean mirror measurement range, in inches. |
| 5 | Double | Mirror Measurement Range - StdDev (Inches) | The standard deviation of the mirror measurement range, in inches. |
| 6 | Double | Mirror Measurement Quality - Mean | The average mirror measurement quality |
| 7 | Double | Mirror Measurement Quality - StdDev | The standard deviation of the mirror measurement quality. |
| 8 | Boolean | Passed Ref Arm Quality Threshold? | Indicates whether the ref arm quality threshold was exceeded (FALSE) or not (TRUE). |
| 9 | Boolean | Passed Mirror Offset Delta Threshold? | Indicates whether the mirror offset delta passed (TRUE) or failed (FALSE) based on the threshold. |
| 10 | Boolean | Passed Mirror Offset StdDev Threshold? | Indicates whether the standard deviation of the mirror offset passed (TRUE) or failed (FALSE) based on the threshold. |
| 11 | Boolean | Passed Mirror Mean Quality Threshold? | Indicates whether the average mirror quality passed (TRUE) or failed (FALSE) based on the threshold. |
| 12 | Boolean | Passed Overall? | Indicates whether the instrument passed (TRUE) or failed (FALSE) the overall test. |

Returned Status

| | |
|---------|---|
| SUCCESS | The test successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR Self Test - Linearization

Performs a linearization self test on the hardware.

Input Arguments

| | | | |
|---|--------------------------|---------------|---------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to test. |
|---|--------------------------|---------------|---------------------------|

Return Arguments

| | | | |
|---|--------|-----------------|--------------------------|
| 1 | Double | Linearity (kHz) | The resulting linearity. |
|---|--------|-----------------|--------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR Self Test - Flip Test

Performs a flip test on the Laser Radar.

Input Arguments

| | | | |
|---|--------------------------|---------------|---------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to test. |
|---|--------------------------|---------------|---------------------------|

Return Arguments

| | | | |
|----|--------|--|---|
| 1 | Double | Front Measurement - Range (Inches) | The range of the front face measurement, in inches. |
| 2 | Double | Front Measurement - Azimuth (Degs) | The azimuth of the front face measurement, in degrees. |
| 3 | Double | Front Measurement - Elevation (Degs) | The elevation of the front face measurement, in degrees. |
| 4 | Double | Front Measurement - Quality | The quality of the front face measurement. |
| 5 | Double | Back Measurement - Range (Inches) | The range of the back face measurement, in inches. |
| 6 | Double | Back Measurement - Azimuth (Degs) | The azimuth of the back face measurement, in degrees. |
| 7 | Double | Back Measurement - Elevation (Degs) | The elevation of the back face measurement, in degrees. |
| 8 | Double | Back Measurement - Quality | The quality of the back face measurement. |
| 9 | Double | Front/Back Difference - Range (Inches) | The difference between the front and back face ranges, in inches. |
| 10 | Double | Front/Back Difference - Azimuth (Degs) | The difference between the front and back face azimuth, in degrees. |
| 11 | Double | Front/Back Difference - Elevation (Degs) | The difference between the front and back face elevation, in degrees. |

Returned Status

| | |
|---------|---|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR Self Test - LO Sep

Performs an LO Sep self test on a Laser Radar (does not apply to the APDIS).

Input Arguments

| | | | |
|---|--------------------------|---|---|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to test. |
| 1 | Integer | Region (1=Region12, 2=Region23, 3=Region34) | The region to test. |
| 2 | Integer | Num Range Measurements | The number of range measurements to take. |

Return Arguments

| | | | |
|----|---------|--|--|
| 3 | Integer | Primary LO (indexed from 1) | The primary LO. |
| 4 | Integer | Secondary LO (indexed from 1) | The secondary LO. |
| 5 | Integer | Primary LO Measurement Count | The number of measurements for the primary LO. |
| 6 | Double | Primary LO Measurement Range - Mean (Inches) | The average range of primary LO measurements, in inches. |
| 7 | Double | Primary LO Measurement Range - StdDev (Inches) | The standard deviation of primary LO measurements, in inches. |
| 8 | Double | Primary LO Measurement Quality - Mean | The average measurement quality of the primary LO. |
| 9 | Double | Primary LO Measurement Quality - StdDev | The standard deviation of the measurement quality of the primary LO. |
| 10 | Integer | Secondary LO Measurement Count | The number of measurements for the secondary LO. |
| 11 | Double | Secondary LO Measurement Range - Mean (Inches) | The average range of secondary LO measurements, in inches. |
| 12 | Double | Secondary LO Measurement Range - StdDev (Inches) | The standard deviation of secondary LO measurements, in inches. |
| 13 | Double | Secondary LO Measurement Quality - Mean | The average measurement quality of the secondary LO. |
| 14 | Double | Secondary LO Measurement Quality - StdDev | The standard deviation of the measurement quality of the secondary LO. |

Returned Status

| | |
|---------|---|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

LR APDIS Perform MCM Calibration

Provides the ability to perform an MCM calibration.

Input Arguments

| | | | |
|---|--------------------------|-------------------------|---|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to Calibrate |
| 1 | Collection Object Name | Nominal Group Name | Reference group to use for the calibration process. |
| 2 | Boolean | Use Matte Tooling Ball? | True - uses matte tooling balls as apposed to shinny. |
| 3 | Sting | New Calibration Name | Name to apply with the new calibration |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found or a problem occurred with the Calibration. |

Remarks

To proceed with manual MCM calibration, enter a point group containing 8-12 nominal points in SA and a tooling ball type. LR will measure points from both faces and calculate MCM delta values.

Once finished, the results will be sent back to SA as _FS (front-side) and _BS (back-side) point groups. MCM record with predefined <New Calibration Name> will be stored in the MCM Database on the LR.

LR APDIS Activate MCM Calibration

Sets the specified MCM calibration as the active calibration on the instrument.

Input Arguments

| | | | |
|---|--------------------------|-----------------------------|---------------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to adjust. |
| 1 | String | Calibration Name (Optional) | Name of the desired calibration |
| 2 | Integer | Calibration ID (Optional) | ID of the desired calibration |

Return Arguments

| | | | |
|---|--------|-----------------|---------------------------------------|
| 1 | String | Active MCM Name | Name of the newly set MCM calibration |
|---|--------|-----------------|---------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument or calibration was not found. |

Remarks

On APDIS models, and internal accelerometer is present, capable for reading the laser radar's orientation with respect to gravity. If both optional input arguments (name is empty and id = -1) are not specified, then the function triggers AOC-based MCM lookup by reading the accelerometer of the Laser Radar. If MCM Database record that matches the accelerometer reading within a fixed tolerance is found, then this record will be activated; otherwise, the factory calibration will be activated.

If both optional input arguments (name and id) are specified, then the MCM calibration is set by id. The function checks if the record active MCM name and requested MCM name match. If name doesn't match the id's record, the function returns partial success;

If MCM record name (case-sensitive) is specified and record's id is not, then this record will be activated if found in the database. If multiple MCMs exist in the database with the specified name, the most recently inserted (highest id) will be selected for activation. If record is not found, then the function will return error and no changes will be made.

If MCM record id is specified (and no name), then if found the MCM with the specified id will be activated.

If input record id set to 0, it indicates no MCM (use factory calibration defaults).

LR APDIS Get Active MCM Calibration

Provides a means to ask the specified instrument what calibration is active by name.

Input Arguments

| | | | |
|---|--------------------------|---------------|---------------------------|
| 0 | Collection Instrument ID | Instrument ID | The ID of the LR to query |
|---|--------------------------|---------------|---------------------------|

Return Arguments

| | | | |
|---|--------|-----------------|------------------------------------|
| 1 | String | Active MCM Name | Name of the active MCM Calibration |
|---|--------|-----------------|------------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The command successfully executed. |
| FAILURE | The specified instrument was not found. |

Remarks

None.

13 ROBOT OPERATIONS

Add Robot/Machine (.SAMachine)

Adds an .SAMachine to the job file

Input Arguments

| | | | |
|---|----------------------------|-----------------|------------------------------------|
| 0 | File Path or Embedded File | .SAMachine File | Path to the .SAMachine File to add |
|---|----------------------------|-----------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine was added successfully. |
| FAILURE | The robot/machine could not be found. |

Remarks

None.

Add Robot/Machine (.ManipKin)

Adds a .ManipKin to the job file

Input Arguments

| | | | |
|---|----------------------------|----------------|-----------------------------------|
| 0 | File Path or Embedded File | .ManipKin File | Path to the .Manipkin File to add |
|---|----------------------------|----------------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine was added successfully. |
| FAILURE | The robot/machine could not be found. |

Remarks

None.

Delete Robot/Machine

Deletes the specified Robot/Machine from the job file

Input Arguments

| | | | |
|---|-----------------------|------------|-------------------------|
| 0 | Collection Machine ID | Machine ID | Robot/Machine to delete |
|---|-----------------------|------------|-------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine was deleted successfully. |
| FAILURE | The robot/machine could not be found. |

Remarks

None.

Move Robot/Machine to Frame

Moves the end effector of a robot or machine to a specified coordinate frame.

Input Arguments

| | | | |
|---|------------|---------------------|--|
| 0 | Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
| 1 | Frame Name | Destination Frame | The frame to which the end effector should be moved. |
| 2 | Boolean | Use SA Kinematics | Indicates whether SA's kinematic model should be used for the move, or whether the machine's own kinematic model should be used. |
| 3 | Boolean | Acknowledge Arrival | Pauses the MP until the robot/machine has arrived at its final destination. |

Return Arguments

| | | | |
|---|-----------|--------------------------------------|---|
| 4 | Transform | Actual Transform In Working (result) | The actual transform used to get from the current position to the goal position (expressed in working coordinates). |
|---|-----------|--------------------------------------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The robot/machine moved to the specified frame successfully. |
| FAILURE | The robot/machine or destination frame was not found. |

Remarks

The robot/machine must be in the active collection, and its interface must be active.

Move Robot/Machine through Path

Moves the end effector of a robot or machine in a path through a series of intermediate frame positions.

Input Arguments

| | | | |
|---|---------------------------------|---------------------|---|
| 0 | Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
| 1 | Collection Object Name Ref List | Path Frames | The frames through which the end effector should travel. |
| 2 | Boolean | Use SA Kinematics | Indicates whether SA's kinematic model should be used for the move, or whether the machine's own kinematic model should be used. |
| 3 | Boolean | Linear Segments | Indicates whether the frames should be interpolated linearly or using the controller's own B-spline interpolation between the frames. |
| 4 | Boolean | Acknowledge Arrival | Pauses the MP until the robot/machine has arrived at its final destination. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The robot/machine travelled the path successfully. |
| FAILURE | The robot/machine or provided frames were not found or were invalid. |

Remarks

The robot/machine must be in the active collection, and its interface must be active.

Move Robot/Machine to Named Destination

Moves a robot or machine to a named position in joint space (as defined in the instrument interface).

Input Arguments

| | | | |
|---|------------|---------------------|--|
| 0 | Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
| 1 | String | Destination Name | The named destination in joint space (as defined in the instrument interface). |
| 2 | Boolean | Acknowledge Arrival | Pauses the MP until the robot/machine has arrived at its final destination. |

Return Arguments

| | | | |
|---|-----------|--------------------------------------|--|
| 3 | Transform | Actual Transform In Working (result) | The actual transform used to get from the current position to the named position (expressed in working coordinates). |
|---|-----------|--------------------------------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine moved to the specified position successfully. |
| FAILURE | The robot/machine or named position was not found. |

Remarks

The robot/machine must be in the active collection, and its interface must be active.

Set Robot/Machine Parameter

Sets a parameter on the specified robot/machine. The parameter name is machine-specific.

Input Arguments

| | | | |
|---|-----------------------|-----------------|--|
| 0 | Collection Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
| 1 | String | Parameter Name | The machine-specific parameter name to set. |
| 2 | Double | Parameter Value | The value to associate with the specified parameter. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The parameter was set successfully. |
| FAILURE | The robot/machine or parameter was not found. |

Remarks

The robot/machine's interface must be active.

Get Robot/Machine Parameter

Gets a parameter on the specified robot/machine. The parameter name is machine-specific.

Input Arguments

| | | | |
|---|--------------------------|----------------|---|
| 0 | Collection Instrument ID | Machine ID | The instrument ID of the robot/machine in question. |
| 1 | String | Parameter Name | The machine-specific parameter name to set. |

Return Arguments

| | | | |
|---|--------|-----------------|--|
| 2 | Double | Parameter Value | The value associated with the specified parameter. |
|---|--------|-----------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The parameter was retrieved successfully. |
| FAILURE | The robot/machine or parameter was not found. |

Remarks

The robot/machine's interface must be active.

Set Robot/Machine Model Link Parameters

Provides a means to set the link parameters for a robot/machine model

Input Arguments

| | | | |
|----|--------------------------|-----------------------------------|---|
| 0 | Collection Instrument ID | Machine ID | The instrument ID of the robot/machine in question. |
| 1 | String | Link Name | |
| 2 | Link Type | Link Type | |
| 3 | Double | DH ALPHA Component | |
| 4 | Double | DH A Component | |
| 5 | Double | DH D Component | |
| 6 | Double | DH Theta Component | |
| 7 | Double | DH X-Axis Deflection Factor | |
| 8 | Double | DH Y-Axis Deflection Factor | |
| 9 | Double | DH Z-Axis Deflection Factor | |
| 10 | Double | 6D Of X Component | |
| 11 | Double | 6D Of Y Component | |
| 12 | Double | 6D Of Z Component | |
| 13 | Double | 6D Of RX Component | |
| 14 | Double | 6D Of RY Component | |
| 15 | Double | 6D Of RZ Component | |
| 16 | Joint Component | Active Joint Component | Select from the list of components |
| 17 | Double | Encoder Offset Value | |
| 18 | Double | Minimum Encoder Limit | |
| 19 | Double | Maximum Encoder Limit | |
| 20 | Boolean | Encoder Sense Negative | |
| 21 | Boolean | Include Additional Encoder | |
| 22 | Integer | Additional Encoder Index Offset | |
| 23 | Boolean | Additional Encoder Sense Negative | |
| 24 | Double | Segment Origin Mass in Kg | |
| 25 | Double | Segment CG Mass in Kg | |
| 26 | Vector | Segment CG in Segment | |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The Model Link Parameters were set successfully. |
| FAILURE | The Machine could not be found or components were improperly configured. |

Remarks

None.

Get Robot/Machine Model Link Parameters

Provides a means to access the link parameters for a robot/machine model

Input Arguments

| | | | |
|---|--------------------------|------------|---|
| 0 | Collection Instrument ID | Machine ID | The instrument ID of the robot/machine in question. |
| 1 | String | Link Name | |

Return Arguments

| | | | |
|----|-----------------|-----------------------------------|------------------------------------|
| 2 | Link Type | Link Type | |
| 3 | Double | DH ALPHA Component | |
| 4 | Double | DH A Component | |
| 5 | Double | DH D Component | |
| 6 | Double | DH Theta Component | |
| 7 | Double | DH X-Axis Deflection Factor | |
| 8 | Double | DH Y-Axis Deflection Factor | |
| 9 | Double | DH Z-Axis Deflection Factor | |
| 10 | Double | 6D Of X Component | |
| 11 | Double | 6D Of Y Component | |
| 12 | Double | 6D Of Z Component | |
| 13 | Double | 6D Of RX Component | |
| 14 | Double | 6D Of RY Component | |
| 15 | Double | 6D Of RZ Component | |
| 16 | Joint Component | Active Joint Component | Select from the list of components |
| 17 | Double | Encoder Offset Value | |
| 18 | Double | Minimum Encoder Limit | |
| 19 | Double | Maximum Encoder Limit | |
| 20 | Boolean | Encoder Sense Negative | |
| 21 | Boolean | Include Additional Encoder | |
| 22 | Integer | Additional Encoder Index Offset | |
| 23 | Boolean | Additional Encoder Sense Negative | |
| 24 | Double | Segment Origin Mass in Kg | |
| 25 | Double | Segment CG Mass in Kg | |
| 26 | Vector | Segment CG in Segment | |

Returned Status

| | |
|---------|---|
| SUCCESS | The Model Link Parameters were returned successfully. |
| FAILURE | The Machine or link could not be found. |

Remarks

None.

Start Robot/Machine Interface

Starts the interface for a robot/machine.

Input Arguments

| | | | |
|---|--------------------------|-------------------|---|
| 0 | Collection Instrument ID | Machine ID | The instrument ID of the robot/machine in question. |
| 1 | Integer | Interface Type | A machine-specific interface type value (the default 0 runs the SARobotDriver). |
| 2 | Boolean | Run in Simulation | Indicates if the interface should be started in simulation mode. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The machine interface was started successfully. |
| FAILURE | The interface could not be started. |

Remarks

None.

Stop Robot/Machine Interface

Stops the interface for a robot/machine.

Input Arguments

| | | | |
|---|--------------------------|------------|---|
| 0 | Collection Instrument ID | Machine ID | The instrument ID of the robot/machine in question. |
|---|--------------------------|------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The machine interface was stopped successfully. |
| FAILURE | The specified machine was not found. |

Remarks

None.

Compute Robot/Machine Adjusted Goal Frame

Computes an adjusted goal frame to compensate for discrepancies between a machine's kinematic model and its real-world behavior. In this case, the goal is to move a robot/machine to a goal frame from its current position. In reality, the robot/machine moves to some slightly different position than what was commanded, due to real-world imperfections in the machine's kinematic model. This end position is measured, and based on how the machine moved, a new adjusted goal frame is computed that will bring the machine closer to the desired goal frame.

Input Arguments

| | | | |
|---|------------------------|--------------------------|--|
| 0 | Collection Object Name | Original Goal Frame | The goal frame to achieve. |
| 1 | Collection Object Name | Last Adjusted Goal Frame | The computed modified goal frame from a previous iteration of the command. |
| 2 | Collection Object Name | Actual Measured Frame | The measured frame for the current machine position. |
| 3 | Collection Object Name | Modified Goal Frame | A name for an adjusted goal frame that will be created. |

Return Arguments

| | | | |
|---|-----------|-----------------|---|
| 4 | Transform | Transform Value | The transform from the current position (measured frame) to the modified goal frame (in working coordinates). |
|---|-----------|-----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The modified goal frame was calculated successfully. |
| FAILURE | The original goal frame, last adjusted goal frame, or actual measured frame could not be found. |

Remarks

The robot/machine must be in the active collection, and its interface must be active. The general use pattern for this command is to:

Command the machine to the goal frame.

Run this command, feeding the goal frame to the *Last Adjusted Goal Frame* argument.

Command the machine to the computed Modified Goal Frame.

Repeat this command as many times as desired, feeding the previously calculated *Modified Goal Frame* to the *Last Adjusted Goal Frame* parameter.

Move Robot/Machine to Joint Pose (6DOF)

Poses a 6 degree of freedom robot or machine based on explicit joint settings that you provide.

Input Arguments

| 0 | Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
|---|------------|------------|--|
| 1 | Double | Joint 1 | The value to set for joint 1. |
| 2 | Double | Joint 2 | The value to set for joint 2. |
| 3 | Double | Joint 3 | The value to set for joint 3. |
| 4 | Double | Joint 4 | The value to set for joint 4. |
| 5 | Double | Joint 5 | The value to set for joint 5. |
| 6 | Double | Joint 6 | The value to set for joint 6. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine moved to the specified pose successfully. |
| FAILURE | The robot/machine was not found. |

Remarks

The robot/machine must be in the active collection, and its interface must be active.

Simulate Robot/Machine Path, Output CSV File

Simulates a robot passing through a set of provided path frames, and exports the associated joint settings to a .CSV file.

Input Arguments

| | | | |
|---|---------------------------------|-----------------|--|
| 0 | Machine ID | Machine ID | The instrument ID of the robot/machine to command. |
| 1 | Collection Object Name Ref List | Path Frames | The set of frames defining the path for the robot. |
| 2 | File Path or Embedded File | Output CSV File | The path to a CSV file to create. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The robot/machine moved to the specified pose successfully. |
| FAILURE | The robot/machine was not found, the path had no inverse kinematic solution, or one or more frames was not found. |

Remarks

The robot/machine must be in the active collection.

Create Robot Calibration

Creates a named calibration for a machine.

Input Arguments

| | | | |
|---|-----------------------|------------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine to create the calibration for. |
| 1 | String | Calibration Name | The name of the calibration to create. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The calibration was created successfully. |
| FAILURE | The specified robot/machine was not found. |

Remarks

None.

Delete Robot Calibration

Deletes a named calibration for a machine.

Input Arguments

| | | | |
|---|-----------------------|------------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to delete. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration was deleted successfully. |
| FAILURE | The specified robot/machine or calibration was not found. |

Remarks

None.

Import Poses Match to Measurements

Imports a joint set file and associates those poses with existing measurements.

Input Arguments

| | | | |
|---|----------------------------|---------------------------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to modify. |
| 2 | Point Name Ref List | Point Names | The list of measurements to associated with the poses. |
| 3 | File Path or Embedded File | FilePath for CSV Joint Set File | The path to the CSV joint set file. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The poses were associated successfully. |
| FAILURE | The specified robot/machine, calibration, points, or joint set file was not found. |

Remarks

None.

Import Poses Match to Frames

Imports a joint set file and associates those poses with existing frames.

Input Arguments

| | | | |
|---|--------------------------------|---------------------------------|---|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to modify. |
| 2 | Collection Frame Name Ref List | Frame Names | The list of frames to associate with the poses. |
| 3 | File Path or Embedded File | FilePath for CSV Joint Set File | The path to the CSV joint set file. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The poses were associated successfully. |
| FAILURE | The specified robot/machine, calibration, frames or joint set file was not found. |

Remarks

None.

Perform Robot Calibration

Calculate a robot calibration based on existing measurements and poses.

Input Arguments

| | | | |
|---|--|--------------------|---|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to calculate. |
| 2 | Machine Cal Degrees-Of-Freedom Options | Degrees of Freedom | The allowable degrees of freedom for the calculation. |
| 3 | Boolean | Show Interface | Indicates if the calibration interface should be displayed. |

Return Arguments

| | | | |
|----|--------|----------------|---|
| 4 | Double | XYZ Max | The maximum XYZ error. |
| 5 | Double | XYZ Average | The average XYZ error. |
| 6 | Double | XYZ RMS | The RMS XYZ error. |
| 7 | Double | Orient Max | The max orientation error. |
| 8 | Double | Orient Average | The average orientation error. |
| 9 | Double | Orient RMS | The RMS orientation error. |
| 10 | Double | Robustness | The robustness factor (numeric stability) of the calculation. |

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration was calculated successfully. |
| FAILURE | The specified robot/machine or calibration was not found, or a solution could not be found. |

Remarks

None.

Perform Robot Calibration (Alternate)

Alternate way to calculate a robot calibration based on existing measurements and poses.

Input Arguments

| | | | |
|---|-----------------------|--------------------------|---|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to calculate. |
| 2 | BASE DOF Settings | BASE Degrees of Freedom | The degrees of freedom of the BASE. |
| 3 | ROBOT DOF Settings | ROBOT Degrees of Freedom | The degrees of freedom of the ROBOT. |
| 4 | TOOL DOF Settings | TOOL Degrees of Freedom | The degrees of freedom of the TOOL. |
| 5 | Boolean | Show Interface | Display calibration interface. |

Return Arguments

| | | | |
|----|--------|----------------|---|
| 6 | Double | XYZ Max | The maximum XYZ error. |
| 7 | Double | XYZ Average | The average XYZ error. |
| 8 | Double | XYZ RMS | The RMS XYZ error. |
| 9 | Double | Orient Max | The max orientation error. |
| 10 | Double | Orient Average | The average orientation error. |
| 11 | Double | Orient RMS | The RMS orientation error. |
| 12 | Double | Robustness | The robustness factor (numeric stability) of the calculation. |

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration was calculated successfully. |
| FAILURE | The specified robot/machine or calibration was not found, or a solution could not be found. |

Remarks

None.

Start/Stop Robot Calibration Trapping

Starts or stop measurement trapping for a robot calibration.

Input Arguments

| | | | |
|---|--------------------------|-------------------------------|---|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to start or stop trapping. |
| 2 | Collection Instrument ID | Instrument ID | The ID of the instrument performing the calibration measurements. |
| 3 | Boolean | Start Trapping (FALSE = Stop) | Indicates whether or not trapping should be started. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The trapping was started or stopped successfully. |
| FAILURE | The specified robot/machine or calibration was not found, or the instrument was not found or active. |

Remarks

None.

Set Active Robot Calibration

Sets a calibration active for a robot.

Input Arguments

| | | | |
|---|-----------------------|------------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to activate. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration was activated. |
| FAILURE | The specified robot/machine or calibration was not found. |

Remarks

None.

Set Robot Calibration Tool Frame

Sets the frame used for calibrating the robot.

Input Arguments

| | | | |
|---|-----------------------|---------------------------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to activate. |
| 2 | Transform | Tool Frame (relative to flange) | The coordinates of the tool frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration tool frame was set. |
| FAILURE | The specified calibration tool frame was not found. |

Remarks

None.

Set Robot Calibration Measurement Offset in Tool Frame

Sets the measurement offset used for robot calibration.

Input Arguments

| | | | |
|---|-----------------------|--------------------------------------|---|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | String | Calibration Name | The name of the calibration to activate. |
| 2 | Transform | Measurement Frame (relative to tool) | The coordinates of the measurement frame. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The calibration measurement offset was activated. |
| FAILURE | The specified calibration measurement offset was not found. |

Remarks

None.

Get Robot Pose for a Frame

Gets the robot pose for a specified frame.

Input Arguments

| | | | |
|---|------------------------|----------------|--|
| 0 | Collection Machine ID | Machine ID | The ID of the robot/machine in question. |
| 1 | Collection Object Name | Goal Frame | The specified frame to use. |
| 2 | Double List | Reference Pose | The robot pose to reference |

Return Arguments

| | | | |
|---|-------------|-----------|--|
| 3 | Double List | Goal Pose | The robot pose looking to ge achieved. |
|---|-------------|-----------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The robot pose was successfully achieved.. |
| FAILURE | The specified machine ID, goal frame, and/or reference pose were not found. |

Remarks

If no reference pose is provided, the current robot pose is used as reference pose.

Get Calibration Appliance Integer Value

Retrieves the value of the calibration appliances integer.

Input Arguments

| | | | |
|---|---------|--------------|--------------------------------------|
| 0 | Integer | Index Offset | The amount the calibration appliance |
|---|---------|--------------|--------------------------------------|

Return Arguments

| | | | |
|---|---------|---------------|---|
| 1 | Integer | Integer Value | The integer value received in the packet. |
|---|---------|---------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The packet was sent successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Set Calibration Appliance Integer Value

Creates the value of the calibration appliances integer.

Input Arguments

| | | | |
|---|---------|---------------|--|
| 0 | Integer | Index Offset | The amount the calibration appliance |
| 1 | Integer | Integer value | The value to set calibration appliance to. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The appliance integer was set successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Get Calibration Appliance Real Value

Retrieves the real value of the calibration appliance.

Input Arguments

| | | | |
|---|---------|--------------|---------------------------------------|
| 0 | Integer | Index Offset | The amount the calibration appliance. |
|---|---------|--------------|---------------------------------------|

Return Arguments

| | | | |
|---|--------|------------|--|
| 1 | Double | Real Value | The real value received in the packet. |
|---|--------|------------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The packet was sent successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Set Calibration Appliance Real Value

Creates the real value of the calibration appliances integer.

Input Arguments

| | | | |
|---|---------|--------------|---|
| 0 | Integer | Index Offset | The amount the calibration appliance |
| 1 | Double | Real Value | The real value to set calibration appliance to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The real value for the calibration appliance was set successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Get Calibration Appliance Data

Retrieves the data of the calibration appliance.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------------|----------------|--|
| 0 | Integer List | Integer Values | The amount of the calibration appliance. |
| 1 | Double List | Real Values | The real values of the appliance. |

Returned Status

| | |
|---------|---|
| SUCCESS | The packet was sent successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Set Calibration Appliance Data

Creates the real value of the calibration appliances integer.

Input Arguments

| | | | |
|---|--------------|----------------|---|
| 0 | Integer List | Integer Values | The amountthe calibration appliance |
| 1 | Double List | Real Values | The real value to set calibration appliance to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The data for the calibration appliance was set successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Robot Calibration Appliance Node Operations...

The following additional MP commands are also available. For more details on their use please refer to the [SA Robot Calibration Appliance Users Manual](#) or reach out to NRK for assistance.

- **Add Calibration Appliance Node**
- **Delete Calibration Appliance Node**
- **Connect/Disconnect Calibration Appliance Node**
- **Set Calibration Appliance Node Instrument**
- **Set Calibration Appliance Node Measurement Profile**
- **Set Calibration Appliance Node Measurement Target**
- **Enable/Disable Calibration Appliance Node Instrument Auto Point**
- **Set Calibration Appliance Node Instrument Dwell Time**
- **Skip Calibration Appliance Node Measurement**
- **Set Calibration Appliance Node Measurement Frame**
- **Set Calibration Appliance Node Measurement Offset Transform**
- **Set Calibration Appliance Node Measurement Point Group**
- **Set Calibration Appliance Node Calibration Appliance IP Address**
- **Set Calibration Appliance Node Trapping Node ID**
- **Enable/Disable Calibration Appliance Node Trap Manager**
- **Clear Calibration Appliance Node Trap Manager Requests**
- **Set Calibration Appliance Node Integer Value**
- **Get Calibration Appliance Node Integer Value**
- **Set Calibration Appliance Node Real Value**
- **Get Calibration Appliance Node Real Value**
- **Set Calibration Appliance Node Data**
- **Get Calibration Appliance Node Data**
- **Set Calibration Appliance Node Display Robot**
- **Update Calibration Appliance Node Display Robot Joints**
- **Get Calibration Appliance Node Status**

This Page Intentionally Left Blank.

14 UTILITY OPERATIONS

OPC DA Server

Set OPC DA Tag Value Double

Creates a double value with the specified tag in the SA OPC server's address space. If the address space already contains a tag with the specified name, the previous value will be overwritten.

Input Arguments

| | | | |
|---|--------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name to associate with the data in the address space. |
| 1 | Double | Value | The value to associate with the specified tag name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

It is assumed that the OPC server is running when this command is executed.

Get OPC DA Tag Value Double

Retrieves the double value associated with the specified tag from the SA OPC server's address space.

Input Arguments

| | | | |
|---|--------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name associated with the data in the address space. |
|---|--------|------------------------|---|

Return Arguments

| | | | |
|---|--------|-------|---|
| 1 | Double | Value | The value associated with the specified tag name. |
|---|--------|-------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The value was retrieved successfully. |
| FAILURE | The OPC server is not running, or the specified tag does not exist in the server's address space. |

Remarks

If the address space does not contain a value with the specified tag, the command will fail.

Set OPC DA Tag Value Integer

Creates an integer value with the specified tag in the SA OPC server's address space. If the address space already contains a tag with the specified name, the previous value will be overwritten.

Input Arguments

| | | | |
|---|---------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name to associate with the data in the address space. |
| 1 | Integer | Value | The value to associate with the specified tag name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

It is assumed that the OPC server is running when this command is executed.

Get OPC DA Tag Value Integer

Retrieves the integer value associated with the specified tag from the SA OPC server's address space.

Input Arguments

| | | | |
|---|--------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name associated with the data in the address space. |
|---|--------|------------------------|---|

Return Arguments

| | | | |
|---|---------|-------|---|
| 1 | Integer | Value | The value associated with the specified tag name. |
|---|---------|-------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The value was retrieved successfully. |
| FAILURE | The OPC server is not running, or the specified tag does not exist in the server's address space. |

Remarks

If the address space does not contain a value with the specified tag, the command will fail.

Set OPC DA Tag Value String

Creates a string with the specified tag in the SA OPC server's address space. If the address space already contains a tag with the specified name, the previous value will be overwritten.

Input Arguments

| | | | |
|---|--------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name to associate with the data in the address space. |
| 1 | String | Value | The string to associate with the specified tag name. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

It is assumed that the OPC server is running when this command is executed.

Get OPC DA Tag Value String

Retrieves the string associated with the specified tag from the SA OPC server's address space.

Input Arguments

| | | | |
|---|--------|------------------------|---|
| 0 | String | OPC Server DA Tag Name | The tag name associated with the data in the address space. |
|---|--------|------------------------|---|

Return Arguments

| | | | |
|---|--------|-------|--|
| 1 | String | Value | The string associated with the specified tag name. |
|---|--------|-------|--|

Returned Status

| | |
|---------|---|
| SUCCESS | The value was retrieved successfully. |
| FAILURE | The OPC server is not running, or the specified tag does not exist in the server's address space. |

Remarks

If the address space does not contain a string with the specified tag, the command will fail.

Language

Set Active Integrated Language

This command allows a user to set the default language translation for SA.

Input Arguments

| | | | |
|---|-----------------|---------------|---|
| 0 | Active Language | Language Name | Choose the desired language from the list of available built in languages |
|---|-----------------|---------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The supported languages are also available through the Menu command *File>Language Translation*.

Set Active Custom Language

This command allows a user to set a custom language translation for SA.

Input Arguments

| | | | |
|---|----------------------------|--------------------|--|
| 0 | File Path or Embedded File | Language File Name | Select the desired language translation (*.lan) file |
| 1 | Font Type | Font | Sets the base font for the language file |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Custom languages can also be set through the Users Options > Display Tab using the [Custom Language File](#) button.

Get Active Language

This command allows a user to get the current translation for SA.

Input Arguments

None.

Return Arguments

| | | | |
|---|----------------------------|--------------------|--|
| 0 | Boolean | Custom Language? | True indicates custom, false indicates integrated language translation file. |
| 0 | File Path or Embedded File | Language File Name | The active language translation (*.lan) file |
| 1 | Font Type | Font | Returns the base font for the language file |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The results will be blank if no language file is in use and SA is using its native English settings.

Folders

Delete Folder

Deletes the specified folder from the tree.

Input Arguments

| | | | |
|---|--------|-------------|-----------------------------------|
| 0 | String | Folder Path | The path to the folder to delete. |
|---|--------|-------------|-----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The folder was deleted successfully. |
| FAILURE | The specified folder was not found. |

Remarks

A single folder is deleted by specifying a name only. A folder in a hierarchy is specified by separating folder names by one or two colons. For example, **A::B::C** deletes the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Move Collection to Folder

Moves a collection to a specific folder in the tree.

Input Arguments

| 0 | Collection Name | Collection | The name of the collection to move. |
|---|-----------------|-------------|--|
| 1 | String | Folder Path | The path to the folder into which to place the collection. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The collection was moved successfully. |
| FAILURE | The specified collection or folder was not found. |

Remarks

A folder path is specified by separating folder names by one or two colons. For example, **A::B::C** places the collection into the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Move Folder to Folder

Moves a source folder into a destination folder in the tree.

Input Arguments

| | | | |
|---|--------|-------------------------|---|
| 0 | String | Source Folder Path | The path of the folder to move. |
| 1 | String | Destination Folder Path | The path of the folder into which to place the source folder. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The folder was moved successfully. |
| FAILURE | The source or destination folders were not found. |

Remarks

A folder path is specified by separating folder names by one or two colons. For example, **A::B::C** places the source folder into the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Get Folders by Wildcard

Obtains a set of folders that match specified wildcard criteria, returning them as a list of strings.

Input Arguments

| | | | |
|---|---------|-----------------------|---|
| 0 | String | Search String | The wildcard search string |
| 1 | Boolean | Case Sensitive Search | Indicates whether the search for folder names should be case-sensitive. |

Return Arguments

| | | | |
|---|-----------------|-------------|-------------------------------------|
| 2 | String Ref List | Folder List | The resulting list of folder names. |
|---|-----------------|-------------|-------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Standard SA wildcard characters (* and ?) apply for the wildcard search string.

Get Folder Notes

Retrieves the notes associated with a specified folder.

Input Arguments

| | | | |
|---|--------|-------------|-------------------------------------|
| 0 | String | Folder Path | The path to the folder in question. |
|---|--------|-------------|-------------------------------------|

Return Arguments

| | | | |
|---|-----------|-------|---|
| 1 | Edit Text | Notes | The notes associated with the specified folder. |
|---|-----------|-------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The folder notes were retrieved successfully. |
| FAILURE | The folder was not found. |

Remarks

A folder path is specified by separating folder names by one or two colons. For example, **A::B::C** retrieves the notes from the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Set Folder Notes

Sets the note associated with a specified folder.

Input Arguments

| | | | |
|---|-----------|----------------------------|---|
| 0 | String | Folder Path | The path to the folder in question. |
| 1 | Edit Text | Notes | The text to place in the note field. |
| 2 | Boolean | Append?(FALSE = Overwrite) | Append text following existing text notes |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The folder notes were set successfully. |
| FAILURE | The folder was not found. |

Remarks

A folder path is specified by separating folder names by one or two colons. For example, **A::B::C** refers to the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Get Folder Collections

Retrieves the collections inside a specified folder.

Input Arguments

| | | | |
|---|--------|-------------|-------------------------------------|
| 0 | String | Folder Path | The path to the folder in question. |
|---|--------|-------------|-------------------------------------|

Return Arguments

| | | | |
|---|-----------------|-----------------|---|
| 1 | String Ref List | Collection List | A list of all collections inside the specified folder (as strings). |
|---|-----------------|-----------------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The folder's collections were retrieved successfully. |
| FAILURE | The folder was not found. |

Remarks

A folder path is specified by separating folder names by one or two colons. For example, **A::B::C** refers to the folder “C” within the parent “B”, which itself is in a parent folder “A”. **A:B:C** would have the same effect.

Notes

Get Collection Notes

Retrieves a collection's notes as a string.

Input Arguments

| | | | |
|---|-----------------|------------|----------------------------|
| 0 | Collection Name | Collection | The collection to examine. |
|---|-----------------|------------|----------------------------|

Return Arguments

| | | | |
|---|-----------|-------|-------------------------|
| 1 | Edit Text | Notes | The collection's notes. |
|---|-----------|-------|-------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The notes were retrieved successfully. |
| FAILURE | The collection was not found. |

Remarks

The returned string includes a hidden line return. This means that direct name comparisons may require that the last character be removed.

Set Collection Notes

Sets a collection's notes.

Input Arguments

| | | | |
|---|-----------------|----------------------------|---|
| 0 | Collection Name | Collection | The collection to modify. |
| 1 | Edit Text | Notes | The notes to apply to the collection. |
| 2 | Boolean | Append?(FALSE = Overwrite) | Append text following existing text notes |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The notes were set successfully. |
| FAILURE | The collection was not found. |

Remarks

None.

Get Object Notes

Retrieves an object's notes as a string.

Input Arguments

| | | | |
|---|------------------------|--------|------------------------|
| 0 | Collection Object Name | Object | The object to examine. |
|---|------------------------|--------|------------------------|

Return Arguments

| | | | |
|---|-----------|-------|---------------------|
| 1 | Edit Text | Notes | The object's notes. |
|---|-----------|-------|---------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The notes were retrieved successfully. |
| FAILURE | The object was not found. |

Remarks

The returned string includes a hidden line return. This means that direct name comparisons may require that the last character be removed.

Set Object Notes

Sets an object's notes.

Input Arguments

| | | | |
|---|------------------------|----------------------------|---|
| 0 | Collection Object Name | Object | The object to modify. |
| 1 | Edit Text | Notes | The notes to apply to the object. |
| 2 | Boolean | Append?(FALSE = Overwrite) | Append text following existing text notes |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The notes were set successfully. |
| FAILURE | The object was not found. |

Remarks

None.

Get Point Notes

Retrieves a point's notes as a string.

Input Arguments

| | | | |
|---|------------|-------|-----------------------|
| 0 | Point Name | Point | The point to examine. |
|---|------------|-------|-----------------------|

Return Arguments

| | | | |
|---|-----------|-------|--------------------|
| 1 | Edit Text | Notes | The point's notes. |
|---|-----------|-------|--------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The notes were retrieved successfully. |
| FAILURE | The point was not found. |

Remarks

The returned string includes a hidden line return. This means that direct name comparisons may require that the last character be removed.

Set Point Notes

Sets a point's notes.

Input Arguments

| | | | |
|---|------------|----------------------------|---|
| 0 | Point Name | Point | The point to modify. |
| 1 | Edit Text | Notes | The notes to apply to the point. |
| 2 | Boolean | Append?(FALSE = Overwrite) | Append text following existing text notes |

Return Arguments

None.

Returned Status

| | |
|---------|----------------------------------|
| SUCCESS | The notes were set successfully. |
| FAILURE | The point was not found. |

Remarks

None.

Units

Get Active Units

Determines the current length, angle, and temperature units of the current SA file.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|-------------|------------------------------------|
| 0 | String | Length | The length units of the file. |
| 1 | String | Angular | The angular units of the file. |
| 2 | String | Temperature | The temperature units of the file. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Active Units

Sets the length and temperature units for the current SA file.

Input Arguments

| | | | |
|---|-------------------|------------------------------|--|
| 0 | Units | Length | The length units for the file. |
| 1 | Boolean | Display Inch Fractions? | True will set display to fractional units |
| 2 | Double | Inch Fractional Denominator? | Minimum or base denominator used. |
| 3 | Boolean | Simplify Inch Fraction? | Setting this to True will cause fractions to be reduced automatically. |
| 4 | Temperature Units | Temperature | The temperature units of the file. |
| 5 | Angular Units | Angular | The angular units of the file. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The units are saved with the current SA file. Fractional unit display can be set for either inches or feet length designations. When set to feet the display will include feet, inches and fractional inches. This is currently used for Watch Windows, Dimensions and Vector labels and Vector Callouts.

Set Decimal Digits for Display

Sets the number of digits used to display length values for the current SA file. This directly changes the settings on the Display tab of the User Options.

Input Arguments

| | | | |
|---|---------|-------------|---|
| 0 | integer | Length | The number of digit used for length |
| 1 | integer | Angle | The number of digit used for angle |
| 2 | integer | Scale | The number of digit used for scale |
| 3 | integer | Unit Vector | The number of digit used for unit vectors |
| 4 | integer | Weight | The number of digit used for weight |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The units are saved with the current SA file.

Set Angular Representation

Sets the way angles will be represented within the current SA file.

Input Arguments

| | | | |
|---|---------|-------------------------|--|
| 0 | Boolean | 0-360, (FALSE = +/-180) | Distinguish which representation to use. |
|---|---------|-------------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Input zero or type "false" for +/-180 angular representation. Input any number other than 0 or type true for 0-360 angular representation.

Get Angular Representation

Determines the angular representation of the current SA file.

Input Arguments

None.

Return Arguments

| | | | |
|---|---------|------------------------|---|
| 0 | Boolean | 0-360 (FALSE = +/-180) | The angular representation that is set in the file. |
|---|---------|------------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Step Comment

Adds a comment to a script. This command does not perform any operation.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set User Interface Profile

Sets the current user interface profile.

Input Arguments

| | | | |
|---|----------------------------|------------------------------|--|
| 0 | String | Profile Name | The name of the user interface profile. |
| 1 | File Path or Embedded File | Profile File Name (Optional) | The path to a saved user interface profile in any directory. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The profile was set successfully. |
| FAILURE | The specified profile was not found. |

Remarks

When using the optional path designation **both** the path to the file and the profile name must be set. This name will also be used to create a template file that will be associated with this name.

For example, if you have a profile anywhere on an accessible drive, say “C:\Analyzer Data\abcxyz.saprofile” (or any other name) and import it using the MP function with a profile name of “My New Profile”, a new file will be created (if it does not already exist named “C:\Analyzer Data\Templates\My New Profile.saprofile” and “My New Profile” will be retained in the registry as the current user profile. This way, the next time SA comes up, it will come up with “My New Profile” which must access the “C:\Analyzer Data\Templates\My New Profile.saprofile” file in order to be instantiated – the original file location is not referenced any longer.

Set MP Step Mode

Sets the playback method for the current MP.

Input Arguments

| | | | |
|---|--------------|--------------|---|
| 0 | MP Step Mode | MP Step Mode | Specifies whether the MP pauses and requires the user to step through each command individually (Single Step), or whether the MP runs automatically (Auto Run). |
|---|--------------|--------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command can essentially be used to pause an MP at a specified step.

Delay for Specified Time

Adds a time delay to the script. Essentially pauses the script for a specified amount of time.

Input Arguments

| | | | |
|---|------------|---------------|--|
| 0 | Double | Time to delay | The desired delay value. |
| 1 | Time Units | Time units | The units for the value in argument 0. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Get Tick Count

Returns an approximately millisecond-accurate tick count (timer value) of the computer clock, in seconds.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|-------------------|---|
| 0 | Double | Tick Count (secs) | The current tick count of the computer's CPU. |
|---|--------|-------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This number is a single value and reports the number of seconds since a specific point in time. To calculate the duration of time for a series of steps, obtain the tick count before and after the operation, then subtract the former from the latter to obtain the elapsed time.

This value is approximately millisecond accurate.

Speak To User

Uses the SA speech server (text to speech synthesizer) to speak a specified string to the user through the computer's audio out port.

Input Arguments

| | | | |
|---|--------|-----------------|----------------------------------|
| 0 | String | String to speak | The string to speak to the user. |
|---|--------|-----------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The string was spoken successfully. |
| FAILURE | The SA speech server was not configured correctly. |

Remarks

None.

Set Working Frame

Sets a specified frame as the working frame.

Input Arguments

| | | | |
|---|------------------------|------------------------|--|
| 0 | Collection Object Name | New Working Frame Name | The name of the frame to make working. |
|---|------------------------|------------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---------------------------------------|
| SUCCESS | The frame was successfully activated. |
| FAILURE | The specified frame was not found. |

Remarks

None.

Get Working Frame Properties

Obtains information about the current working frame.

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|-----------------|--------------------------------------|
| 0 | String | Frame Name | The name of the working frame. |
| 1 | String | Collection Name | The collection of the working frame. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Working Color

Sets the current working color.

Input Arguments

| | | | |
|---|-------|------------------------|------------------------------------|
| 0 | Color | New Working Color Name | The name of the new working color. |
|---|-------|------------------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Working Color Auto Increment

Indicates whether the working color should automatically increment after an object is created.

Input Arguments

| | | | |
|---|---------|----------------|---|
| 0 | Boolean | Auto Increment | Indicates whether or not working color auto increment should be active. |
|---|---------|----------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Object(s) Color

Sets one or more objects to a specified color.

Input Arguments

| | | | |
|---|---------------------------------|------------------------|--|
| 0 | Collection Object Name Ref List | Objects to change | The list of objects to color. |
| 1 | Color | New Working Color Name | The color to change the specified objects to. |
| 2 | Boolean | Auto Increment | Indicates whether the color should just be automatically incremented from the current color. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The objects were colored successfully. |
| PARTIAL SUCCESS | At least one (but not all) objects were not found. |
| FAILURE | No objects were found. |

Remarks

You may supply the color as a string variable in the format R,G,B (ex. 128, 64, 128).

If the Auto Increment argument is set to TRUE, the supplied color will be ignored in favor of the next color in the color list.

Get Object Color

Retrieves the objects color.

Input Arguments

| | | | |
|---|------------------------|-------------|---|
| 0 | Collection Object Name | Object Name | The name of the object whose color to retrieve. |
|---|------------------------|-------------|---|

Return Arguments

| | | | |
|---|-------|--------------|--------------------------|
| 1 | Color | Object Color | The color of the object. |
|---|-------|--------------|--------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Object(s) Translucency

Sets the rendering type and opacity of a list of objects.

Input Arguments

| | | | |
|---|---------------------------------|-------------------|---|
| 0 | Collection Object Name Ref List | Objects to change | The list of objects to modify. |
| 1 | Translucency Type | Rendering Type | The rendering type (solid, translucent, or wire-frame). |
| 2 | Double | Opacity Value | The opacity of the specified objects. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The objects were changed successfully. |
| FAILURE | One or more objects in the list were not found. |

Remarks

The opacity value only applies if the rendering type is set to Translucent. Opacity values should be between 0 (fully transparent) and 1 (fully opaque). Values outside of this range will be set to the closest valid value.

Send MP Result to External Device

Sends an “Undone”, “Success”, “Partial Success”, or “Failure” message to another device via the TCP/IP network protocol.

Input Arguments

| | | | |
|---|-------------------------|-----------------------------|---|
| 0 | String | IP Address or computer name | The IP address/computer name of the device to which the message should be sent. |
| 1 | Integer | Socket Port | The port over which to send the message. |
| 2 | Measurement Plan Result | Result | One of the possible result messages to send. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The message was sent successfully. |
| FAILURE | The address or port was not valid. |

Remarks

None.

Send MP Step's Status to External Device

Sends the result of a specific step to another device via the TCP/IP network protocol.

Input Arguments

| | | | |
|---|---------|-----------------------------|---|
| 0 | String | IP Address or computer name | The IP address/computer name of the device to which the message should be sent. |
| 1 | Integer | Socket Port | The port over which to send the message. |
| 2 | Step ID | Step ID | The step whose result should be transmitted. |

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------------|
| SUCCESS | The message was sent successfully. |
| FAILURE | The address or port was not valid. |

Remarks

None.

Delete Objects

Deletes one or more objects from the current SA file.

Input Arguments

| | | | |
|---|---------------------------------|--------------|--------------------------------|
| 0 | Collection Object Name Ref List | Object Names | The list of objects to delete. |
|---|---------------------------------|--------------|--------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The objects were deleted successfully. |
| PARTIAL SUCCESS | One or more objects (but not all) could not be found. |
| FAILURE | No objects could be found. |

Remarks

None.

Delete Items

Deletes one or more items from the current SA file. An “Item” is a generic term that can apply to anything in the tree other than specific points. This includes things that are not objects such as reports or charts or GD&T annotations for example.

Input Arguments

| | | | |
|---|-------------------------------|-----------|------------------------------|
| 0 | Collection Item Name Ref List | Item List | The list of items to delete. |
|---|-------------------------------|-----------|------------------------------|

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The items were deleted successfully. |
| PARTIAL SUCCESS | One or more items (but not all) could not be found. |
| FAILURE | No items could be found. |

Remarks

Built to reference Make a Collection Item Name Reference List - Wildcard Selection.

Lock Imported Items

This command controls the Direct CAD Import setting “Lock Imported Items”. It controls the default behavior of imported CAD surfaces and associated geometry. When locked the transform of these imported objects cannot be modified by any operation within SA until it is unlocked.

Input Arguments

| | | | |
|---|---------|-------------|--|
| 0 | Boolean | Lock Items? | True sets the status to Lock for imported items. |
|---|---------|-------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|------------------------------|
| SUCCESS | This command always succeeds |
|---------|------------------------------|

Remarks

Imported items that can be locked include:

- Imported Surfaces
- SA Objects (excluding frames, point groups, and clouds)

Lock/Unlock Selected Items

This command controls the locked status of items in the tree. When locked the transform of these items cannot be modified by any operation within SA until it is unlocked.

Input Arguments

| | | | |
|---|-----------------------------------|-------------|--|
| 0 | Collection Item Name Ref List | Item List | Items to which the locked status will be applied |
| 1 | Collection Instrument ID Ref List | Instruments | Instruments to which the locked status will be applied |
| 2 | Boolean | Lock Items? | True sets the status to status to Locked for the selected items. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The locked status was applied successfully. |
| PARTIAL SUCCESS | One or more objects (but not all) could not be found. |
| FAILURE | No objects could be found. |

Remarks

Items that can be locked include:

- Imported Surfaces
- Instruments
- SA Objects (excluding frames, point groups, and clouds) not yet associated with instruments.

Highlight Objects

Sets the highlight state for one or more objects.

Input Arguments

| | | | |
|---|---------------------------------|-----------------------------------|---|
| 0 | Collection Object Name Ref List | Object Names (Empty to clear all) | The list of objects to highlight/unhighlight. |
| 1 | Boolean | HighLight Objects? | Indicates whether or not the specified objects should be highlighted. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The objects were highlighted/unhighlighted successfully. |
| PARTIAL SUCCESS | One or more objects (but not all) could not be found. |
| FAILURE | No objects could be found. |

Remarks

None.

Highlight Objects

Sets the highlight state for one or more relationships.

Input Arguments

| | | | |
|---|-----------------------|------------------------------------|---|
| 0 | Relationship Ref List | Relationships (Empty to clear all) | The list of relationships to highlight/unhighlight. |
| 1 | Boolean | HighLight Relationships? | Indicates whether or not the specified relationships should be highlighted. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | The relationships were highlighted/unhighlighted successfully. |
| PARTIAL SUCCESS | One or more relationships (but not all) could not be found. |
| FAILURE | No relationships could be found. |

Remarks

None.

Highlight Point

Sets the highlight state for a point.

Input Arguments

| | | | |
|---|------------|---------------------------------|---|
| 0 | Point Name | Point Name (Empty to clear all) | The point to highlight or unhighlight. |
| 1 | Boolean | Show Point? | Indicates whether or not the specified point should be highlighted. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The point was highlighted/unhighlighted successfully. |
| FAILURE | The point could not be found. |

Remarks

None.

Move Objects Drag Graphically

Puts the user in a mode such that a set of objects can be dragged graphically in the view.

Input Arguments

| | | | |
|---|---------------------------------|---------|------------------------------|
| 0 | Collection Object Name Ref List | Objects | The list of objects to move. |
|---|---------------------------------|---------|------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | One or more objects were successfully moved. |
| FAILURE | The user cancelled the move operation by pressing ESC. |

Remarks

None.

Scale Objects

Scales a list of objects about the working coordinate frame.

Input Arguments

| | | | |
|---|---------------------------------|--------------|---|
| 0 | Collection Object Name Ref List | Objects | The list of objects to scale. |
| 1 | Double | Scale Factor | The scale factor to apply to the objects. |

Return Arguments

None.

Returned Status

| | |
|-----------------|--|
| SUCCESS | All objects were successfully scaled. |
| PARTIAL SUCCESS | At least one (but not all) object was successfully scaled. |
| FAILURE | No objects could be found. |

Remarks

None.

Move Instruments Drag Graphically

Puts the user in a mode such that a set of instruments can be dragged graphically in the view.

Input Arguments

| | | | |
|---|-----------------------------------|-------------|----------------------------------|
| 0 | Collection Instrument ID Ref List | Instruments | The list of instruments to move. |
|---|-----------------------------------|-------------|----------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | One or more instruments were successfully moved. |
| FAILURE | The user cancelled the move operation by pressing ESC. |

Remarks

None.

Set Special MP Mode

Puts MP scripting into a special mode. Reserved for specific use.

Input Arguments

| | | | |
|---|---------|----------------------|---|
| 0 | String | Keyword | The keyword identifying the mode in which to place MPs. |
| 1 | Boolean | Enable Special Mode? | Indicates whether the identified mode should be turned on or off. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------|
| SUCCESS | The mode was set successfully. |
| FAILURE | The mode was invalid. |

Remarks

None.

Increment Point Name

Increments a point name by an integer value.

Input Arguments

| | | | |
|---|------------|----------------------|--|
| 0 | Point Name | 'Base' Point Name | The "base" prefix to use for the point name. |
| 1 | Integer | Increment | The amount by which to increment the base name. |
| 2 | Point Name | Resultant Point Name | The point name after the increment has been applied. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Refresh Views

Refreshes and updates the graphical view.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Logging State

Controls whether or not file logging is active.

Input Arguments

| | | | |
|---|---------|---------|--|
| 0 | Boolean | Active? | Indicates whether or not file logging is active. |
|---|---------|---------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Trim Log File

Trims the log, deleting all entries beyond the specified entry count.

Input Arguments

| | | | |
|---|---------|---------------------------|--|
| 0 | Integer | Number of Entries to Keep | Enter the desired number of entries to keep... all previous entries will be deleted. |
|---|---------|---------------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command replicates the Trim Log button in the log.

Set Auto Event Creation

Turns automatic event generation on or off. This is the MP equivalent to the *Allow Automatic Event Generation* checkbox in the *User Options* ▶ *Reporting* tab.

Input Arguments

| | | | |
|---|---------|---------|--|
| 0 | Boolean | Active? | Indicates whether or not automatic event generation is active. |
|---|---------|---------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Write to Log

Writes a string to the current job's log file.

Input Arguments

| | | | |
|---|--------|-----------|---|
| 0 | String | Log Entry | The string to write to the active file's log. |
|---|--------|-----------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Remove Specified Characters From String

Removes the appearance of one or more characters from a string.

Input Arguments

| | | | |
|---|--------|----------------------|--|
| 0 | String | Characters to remove | A string of individual characters to remove. |
| 1 | String | String to process | The string to search for the characters to remove. |
| 2 | String | Resultant String | The "string to process" after the "characters to remove" have been stripped out. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Close All Watch Windows

Automatically close all open watch windows.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Status Dialog

Displays a dialog window with a progress bar and an optional “time remaining” counter.

Input Arguments

| | | | |
|---|---------|--------------------------|--|
| 0 | String | Dialog Title | The title to display in the dialog. |
| 1 | String | Text Message | The text to display in the dialog. |
| 2 | Integer | Current Position | The current integer value for the progress bar. |
| 3 | Integer | Upper Limit | The integer value at which the progress bar is at 100%. |
| 4 | Boolean | Suppress Time Remaining? | Indicates whether a timer should be displayed that estimates the amount of time remaining for the operation. |
| 5 | Boolean | Close Dialog? | Indicates whether the dialog (as named in argument 0) should be displayed or closed. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Set Working Directory

Sets a directory as the working directory.

Input Arguments

| | | | |
|---|----------------|-------------------|--|
| 0 | Directory Path | Working Directory | The path to the desired working directory. |
|---|----------------|-------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The specified directory was made working. |
| FAILURE | The specified directory was not found. |

Remarks

None.

Make Directory

Creates a directory (or nested directories) on the file system.

Input Arguments

| | | | |
|---|----------------|-----------|--------------------------|
| 0 | Directory Path | Directory | The directory to create. |
|---|----------------|-----------|--------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The directory was created successfully. |
| FAILURE | The directory could not be created. |

Remarks

Nested directories can be created by listing multiple folders as part of the path. For example, `C:\first\second\third`.

Directories can be created relative to the running MP's directory by using the relative path symbol. For example, to create `mySubdirectory` in the current MP's folder, use `.\mySubdirectory`.

Directory Existence

Indicates whether or not the specified directory exists.

Input Arguments

| | | | |
|---|----------------|-----------|---------------------------|
| 0 | Directory Path | Directory | The directory to examine. |
|---|----------------|-----------|---------------------------|

Return Arguments

| | | | |
|---|---------|---------|---|
| 1 | Boolean | Exists? | True if the specified directory exists. |
|---|---------|---------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Delete Directory

Deletes a directory (or nested directories) from the file system.

Input Arguments

| | | | |
|---|----------------|-----------|--------------------------|
| 0 | Directory Path | Directory | The directory to delete. |
|---|----------------|-----------|--------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The directory was deleted successfully. |
| FAILURE | The directory could not be deleted. |

Remarks

A nested directory can be deleted by listing multiple folders as part of the path. For example, `C:\first\second\third` would delete the third directory.

Directories can be deleted relative to the running MP's directory by using the relative path symbol. For example, to delete `mySubdirectory` in the current MP's folder, use `.\mySubdirectory`.

Copy Directory

Copies a directory (or nested directories) to a specified file system location.

Input Arguments

| | | | |
|---|----------------|-----------------------|--|
| 0 | Directory Path | Source Directory | The directory to copy. |
| 1 | Directory Path | Destination Directory | The directory inside which the source directory should be copied. |
| 2 | Boolean | Replace Existing? | Indicates whether existing files should be overwritten if they conflict with files being copied. |
| 3 | Boolean | Show Progress? | Indicates whether a progress dialog should be displayed during the copy operation. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The directory was copied successfully. |
| FAILURE | The source directory could not be found or copied. |

Remarks

Directories can be copied relative to the running MP's directory by using the relative path symbol. For example, to copy **mySubdirectory** in the current MP's folder, use **. \mySubdirectory** as the source directory path.

Generate Random Number

Generates a random number of type double. The number can be distributed normally about 0 (Gaussian distribution) or can have a uniform distribution.

Input Arguments

| | | | |
|---|---------|-----------------------|--|
| 0 | Boolean | Gaussian distributed? | Determines whether the random numbers should have a Gaussian distribution (like a bell curve) or be distributed uniformly. |
|---|---------|-----------------------|--|

Return Arguments

| | | | |
|---|--------|---------------|------------------------------|
| 1 | Double | Random Number | The resulting random number. |
|---|--------|---------------|------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

If the random number is generated uniformly (not Gaussian distributed), it will be between 0 and 1. If Gaussian distributed, it will have no bounds (as a Gaussian distribution is unbounded), but will be centered about 0.

Set View Idle Update Frequency

Sets the frequency at which the graphical view is updated (for script optimization purposes).

Input Arguments

| | | | |
|---|---------|------------|---|
| 0 | Integer | Idle Count | The number of "idle counts" between graphical view updates. |
|---|---------|------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The idle update counts were set successfully. |
| FAILURE | An invalid (negative) idle count was specified. |

Remarks

Idle counts have no specific length--so you cannot set a specific frequency (as in one update per second).

A higher idle count will update the graphical view less frequently, and as a result scripts will run faster (at the expense of a less-frequently updated view).

Set Automatic Backup State

Enables or disables automatic file and/or measurement backups in SA.

Input Arguments

| | | | |
|---|---------|--------------------------------------|---|
| 0 | Boolean | Auto Job File Restore Points Active? | Indicates whether automatic full file backups should be enabled. |
| 1 | Boolean | Auto Measurements Backup Active? | Indicates whether automatic measurement-only backups should be enabled. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Since this command changes a Machine Configuration setting, the change will apply until the setting is changed again. If set by an MP, the setting will persist, even between different executions of SA.

Set Notification Cancel Override

Shows or hides the **X** and **Cancel** buttons displayed in later “Ask For...” MP commands.

Input Arguments

| | | | |
|---|---------|------------------|---|
| 0 | Boolean | Prohibit Cancel? | Indicates whether the “X” and “Cancel” buttons should be visible. |
|---|---------|------------------|---|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This is a state command—it sets the visibility state of the **X** and **Cancel** buttons for all future “Ask for ...” commands until the command is executed again to set the state to a different value.

This can be helpful in programming user interaction commands that require an entry from the user in order for the script to continue. It is not available for “Notify User” commands because these commands perform no action and always continue to the next step regardless of the button pressed.

Set Interaction Mode

Specifies how Measurement Plan or SDK scripts will interact with the SA application.

Input Arguments

| | | | |
|---|----------------------------|--|--|
| 0 | SA Interaction Mode | SA Interaction Mode | Specifies the level of interaction SA will have with a user while an MP is running. "Silent" implies very little to no dialogues requiring user interaction and "Manual" implies the most interaction with the user, whereas "Automatic" is a balance in the middle. |
| 1 | MP Interaction Mode | Measurement Plan Interaction Mode | Specifies how MPs react to a failure or partial success in a command. They can be set to continue executing, halt on a failure only, or halt on a failure or partial success. |
| 2 | MP Dialog Interaction Mode | Measurement Plan Dialog Interaction Mode | Specifies whether the user is allowed to interact with the user interface during MP execution (Allow Application Interaction) or whether the user is locked out from user interface interaction (Block Application Interaction). |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Set the MP Interaction mode to "Never Halt" to enable the ability for an MP to perform full error checking.

This is a *state* setting. The interaction mode can be changed at any time. The new settings take effect immediately, until the next Set Interaction Mode command or the end of the MP is reached.

- **SA Interaction Mode.** This setting persists for the life of the application. If SA is exited and then restarted, the setting will continue to persist until changed.
- **MP Interaction Mode.** This setting persists for a run of an entire script, including any subroutines or jumped scripts, until a script completely exits. At that time, the MP Interaction mode is set back to "Halt on Failure or Partial Success." Put another way, the *Set Interaction Mode* command need not be set in each subroutine—it will persist for the life of the script until changed.
- **MP Dialog Interaction Mode.** This setting persists for the life of the application. If SA is exited and then restarted, the setting will continue to persist until changed.

UDP Send String

Transmits a UTF-8 encoded string to a host and port using the UDP transmission protocol.

Input Arguments

| | | | |
|---|---------|------------------|---|
| 0 | String | Destination Host | The destination host for the UDP packet (IP address). |
| 1 | Integer | Destination Port | The destination port for the UDP packet. |
| 2 | String | String to Send | The string to send across the network. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The packet was sent successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

UDP Receive String

Receives a UTF-8 encoded string transmitted by the UDP transmission protocol over the network.

Input Arguments

| | | | |
|---|---------|-----------------------------|--|
| 0 | Integer | Local Port | The port to use to listen for the packet. |
| 1 | Integer | Timeout (secs), 0 for none. | The timeout to use to wait for the packet. If none, the command will wait infinitely until a packet is received on the specified port. |

Return Arguments

| | | | |
|---|---------|-----------------|---|
| 2 | String | Received String | The string received in the packet. |
| 3 | String | Sender Host | The host (IP address) that sent the packet. |
| 4 | Integer | Sender Port | The port from which the packet was transmitted. |

Returned Status

| | |
|---------|---|
| SUCCESS | The packet was sent successfully. |
| FAILURE | The destination host or port was invalid. |

Remarks

None.

Get Screen Resolution

Retrieves screen resolution.

Input Arguments

| | | | |
|---|---------|------------------------|----------------------------------|
| 0 | Integer | Display (-1 = Primary) | Choose screen resolution display |
|---|---------|------------------------|----------------------------------|

Return Arguments

| | | | |
|---|---------|------------------------------------|--|
| 1 | Integer | Integer Window Top Left X Position | Resolution will be in the top left x position. |
| 2 | Integer | Integer Window Top Left Y Position | Resolution will be in the top left Y position. |
| 3 | Integer | Integer Width | The width of the screen resolution. |
| 4 | Integer | Integer Height | The height of the screen resolution. |

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Using -1 will default to primary screen resolution.

Set Wild Card Asterisk Mode

Set the Auto Wrap behavior desired for all Wildcard search commands

Input Arguments

| | | | |
|---|---------|--------------------------|------------------------------------|
| 0 | Boolean | Auto Wrap Search String? | TRUE = Auto Wrap [default bahvior] |
|---|---------|--------------------------|------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Traditionally MP wildcard search commands were set up to find as many matching names as possible. Each search string was automatically wrapped as *search string*. This allowed you to search for "P1" and find "AP123". However, more granular control is required for some searches. Rather than add a boolean control to every MP command a global status was added that can be set using this command.

Lock/Unlock Trapping Control

Locks and unlocks trapping for Relationships, Datum, and Feature Checks

Input Arguments

| | | | |
|---|------------------------|------------------------|--|
| 0 | Relationship Ref List | Relationship Ref List | The relationships to lock or unlock trapping. |
| 1 | Feature Check Ref List | Feature Check Ref List | The feature checks to lock or unlock trapping. |
| 2 | Datum Ref List | Datum Ref List | The datums to lock or unlock trapping. |
| 3 | Boolean | Lock Out Trapping? | Whether to lock or unlock trapping. |

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | This command is always successfully. |
|---------|--------------------------------------|

Remarks

None.

This Page Intentionally Left Blank.

15

ACCUMULATOR MATH OPERATIONS

Accumulator Clear

Clears the accumulator for use, resetting it to Zero.

Input Arguments

None.

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Add

Adds a double value to the existing accumulator value

Input Arguments

| | | | |
|---|--------|-----------------|---------------------|
| 0 | Double | Double Argument | Double to be added. |
|---|--------|-----------------|---------------------|

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Subtract

Subtracts a double value to the existing accumulator value

Input Arguments

| | | | |
|---|--------|-----------------|--------------------------|
| 0 | Double | Double Argument | Double to be subtracted. |
|---|--------|-----------------|--------------------------|

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Multiply

Multiplies the accumulator value by a double value.

Input Arguments

| | | | |
|---|--------|-----------------|---------------------|
| 0 | Double | Double Argument | Double to multiply. |
|---|--------|-----------------|---------------------|

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Divide

Divides the accumulator value by a double value.

Input Arguments

| | | | |
|---|--------|-----------------|--|
| 0 | Double | Double Argument | Current accumulator value is divided by this double value. |
|---|--------|-----------------|--|

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Power

Raises the accumulator value by a the specified power.

Input Arguments

| | | | |
|---|--------|-----------------|--|
| 0 | Double | Double Argument | Current accumulator value is raised to the specified double power. |
|---|--------|-----------------|--|

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Invert

Inverts the accumulator value

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Accumulator Change Sign

Changes the sign of the accumulator value

Input Arguments

None.

Return Arguments

| | | | |
|---|--------|-------------|---------------------------------|
| 1 | Double | Accumulator | The resulting accumulator value |
|---|--------|-------------|---------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

This Page Intentionally Left Blank.

16 SCALAR MATH OPERATIONS

Integer Math Operation

Performs a math operation on two integers.

Input Arguments

| | | | |
|---|---------------|--------------|--|
| 0 | Integer | First Value | The first value. |
| 1 | Math Operator | Operation | The math operation to perform. Choose from addition(+), subtraction(-), multiplication(*), division(/), and modulo(%). |
| 2 | Integer | Second Value | The second value. |

Return Arguments

| | | | |
|---|---------|-----------------|-----------------------------------|
| 3 | Integer | Resultant Value | The result of the math operation. |
|---|---------|-----------------|-----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Be careful not to perform an invalid math operation, such as dividing by zero. Doing so will cause termination of the application.

Double Math Operation

Performs a math operation on two doubles.

Input Arguments

| | | | |
|---|---------------|--------------|--|
| 0 | Double | First Value | The first value. |
| 1 | Math Operator | Operation | The math operation to perform. Choose from addition(+), subtraction(-), multiplication(*), division(/), and modulo(%). |
| 2 | Double | Second Value | The second value. |

Return Arguments

| | | | |
|---|--------|-----------------|-----------------------------------|
| 3 | Double | Resultant Value | The result of the math operation. |
|---|--------|-----------------|-----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Be careful not to perform an invalid math operation, such as dividing by zero. Doing so will cause termination of the application.

Double Comparison

Compares two double values.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Double | Double A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | Double | Double B | The second value. |
| 3 | Step ID | Step if TRUE | The step to jump to if the comparison is true. |
| 4 | Step ID | Step if FALSE | The step to jump to if the comparison is false. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Double Comparison (result)

Compares two double values and gives result.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Double | Double A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | Double | Double B | The second value. |

Return Arguments

| | | | |
|---|---------|-----------------|--------------------------------------|
| 3 | Boolean | Resultant Value | The result of the double comparison. |
|---|---------|-----------------|--------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Integer Comparison

Compares two integer values.

Input Arguments

| | | | |
|---|-----------------|--------------------|--|
| 0 | Integer | Integer A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | Integer | Integer B | The second value. |
| 3 | Step ID | Step if TRUE | The step to jump to if the comparison is true. |
| 4 | Step ID | Step if FALSE | The step to jump to if the comparison is false. |
| 5 | Boolean | Use for % complete | True reports percent complete as a function of the reference value |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The Use for % Complete (A5) was added to help keep track of the progress of an MP working through a long looping process. This provides the ability to get a display of the % complete as function of the two reference values. The loop iteration percentage is displayed in the MP bar header.

Integer Comparison (result)

Compares two double values and gives result.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | Integer | Integer A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | Integer | Integer B | The second value. |

Return Arguments

| | | | |
|---|---------|-----------------|---------------------------------------|
| 3 | Boolean | Resultant Value | The result of the integer comparison. |
|---|---------|-----------------|---------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None

String Comparison

Compares two string values, providing the ability to jump to the appropriate step depending on whether or not the strings are the same.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | String | String A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | String | String B | The second value. |
| 3 | Boolean | Case sensitive? | True requires the strings to be of the same case as well as the same characters |
| 4 | Step ID | Step if TRUE | The step to jump to if the comparison is true. |
| 5 | Step ID | Step if FALSE | The step to jump to if the comparison is false. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command performs an alphabetic comparison between the two strings. A word appearing earlier in the dictionary is considered “smaller”.

String Comparison (result)

Compares two string values, but unlike a *String Comparison* step that provides the ability to jump to the appropriate step, this command simply returns a result which can be handled as needed within the script.

Input Arguments

| | | | |
|---|-----------------|-----------------|--|
| 0 | String | String A | The first value. |
| 1 | Comparison Type | Comparison Type | The comparison to perform. Choose from equal to(=), less than(<), greater than(>), less than or equal to(<=), greater than or equal to(>=), or not equal to(!=). |
| 2 | String | String B | The second value. |
| 3 | Boolean | Case sensitive? | True requires the strings to be of the same case as well as the same characters |

Return Arguments

| | | | |
|---|---------|-----------------|--------------------------------------|
| 4 | Boolean | Resultant Value | The result of the string comparison. |
|---|---------|-----------------|--------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This command performs an alphabetic comparison between the two strings. A word appearing earlier in the dictionary is considered “smaller”.

Change String Case

Changes a string to all uppercase or all lowercase characters.

Input Arguments

| | | | |
|---|---------|-------------|---|
| 0 | String | String | The string to modify. |
| 1 | Boolean | Upper Case? | TRUE sets the string to all capital letters. FALSE set the string to all lowercase letters. |

Return Arguments

| | | | |
|---|--------|------------------|----------------------|
| 2 | String | Resultant String | The modified string. |
|---|--------|------------------|----------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Does String Contain Sub-String

Searches for a string within another string.

Input Arguments

| | | | |
|---|---------|-----------------------|--|
| 0 | String | String to Check | The string to search through. |
| 1 | String | Sub-string | The string to find within the search string. |
| 2 | Boolean | Case sensitive? | Indicates whether the search is case sensitive. If no, case is ignored when determining matches. |
| 3 | Step ID | Step if TRUE | The step to jump to if the sub-string is found. |
| 4 | Step ID | Step if FALSE | The step to jump to if the sub-string is not found. |
| 5 | Integer | First Character index | The index of the first character of the sub-string in the search string. |

Return Arguments

| | | | |
|---|---------|-----------------------|--|
| 5 | Integer | First Character index | The index of the first character of the sub-string in the search string. |
|---|---------|-----------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

The returned “first character index” is zero based. So, if the sub-string appears at the beginning of the search string, the returned index is zero. For example, searching for ology in metrology would return an index of 4.

Boolean Comparison

Compares two boolean values and allows the a jump step to be specified based upon the results.

Input Arguments

| | | | |
|---|---------|-------------------|--|
| 0 | Boolean | Boolean A | The first value. |
| 1 | Boolean | Boolean B | The second value. |
| 2 | Step ID | Step if Same | The step to jump to if the two values are the same. |
| 3 | Step ID | Step if Different | The step to jump to if the two values are different. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Boolean Comparison (result)

Compares two boolean values and returns the result only. A returned value of True indicates that the values match, both true or both false.

Input Arguments

| | | | |
|---|---------|--------------|---|
| 0 | Boolean | Boolean A | The first value. |
| 1 | Boolean | Boolean B | The second value. |
| 2 | Step ID | Step if Same | The step to jump to if the two values are the same. |

Return Arguments

| | | | |
|---|---------|-----------------|--|
| 3 | Boolean | Resultant Value | True indicates that the boolean values compared match. |
|---|---------|-----------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Color Comparison

Compares two RGB colors and jumps to the specified Step ID based upon the results. “Same” if all three colors components (RGB) of a specified object’s color match within their respective deviations ranges, “Different” if any one or more colors exceeds the allowable range.

Input Arguments

| | | | |
|---|---------|-----------------------------|---|
| 0 | Color | Color A | The source color. |
| 1 | Color | Color B | The comparison color. |
| 2 | Integer | Allowable Deviation (Red) | The allowable integer deviation as a +/- value |
| 3 | Integer | Allowable Deviation (Green) | The allowable integer deviation as a +/- value |
| 4 | Integer | Allowable Deviation (Blue) | The allowable integer deviation as a +/- value |
| 5 | Step ID | Step if Same | The step to jump to if all 3 colors match with the specified range. |
| 6 | Step ID | Step if Different | The step to jump to if the colors do not match. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Color Comparison (result)

Compares two RGB colors. “Same” or True if all three colors components (RGB) of a specified object’s color match within their respective deviations ranges, “Different” or False if any one or more colors exceeds the allowable range.

Input Arguments

| | | | |
|---|---------|-----------------------------|--|
| 0 | Color | Color A | The source color. |
| 1 | Color | Color B | The comparison color. |
| 2 | Integer | Allowable Deviation (Red) | The allowable integer deviation as a +/- value |
| 3 | Integer | Allowable Deviation (Green) | The allowable integer deviation as a +/- value |
| 4 | Integer | Allowable Deviation (Blue) | The allowable integer deviation as a +/- value |

Return Arguments

| | | | |
|---|---------|-----------------|--|
| 5 | Boolean | Resultant Value | True indicates that the colors match within the allowable range. |
|---|---------|-----------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Integer Absolute Value

Returns the absolute value of an integer.

Input Arguments

| | | | |
|---|---------|------------|---------------------|
| 0 | Integer | Integer In | The source integer. |
|---|---------|------------|---------------------|

Return Arguments

| | | | |
|---|---------|----------------|---|
| 1 | Integer | Integer Result | The absolute value of the source integer. |
|---|---------|----------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Double Absolute Value

Returns the absolute value of a double.

Input Arguments

| | | | |
|---|--------|-----------|--------------------|
| 0 | Double | Double In | The source double. |
|---|--------|-----------|--------------------|

Return Arguments

| | | | |
|---|--------|---------------|--|
| 1 | Double | Double Result | The absolute value of the source double. |
|---|--------|---------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Double Square Root

Returns the square root of a double.

Input Arguments

| | | | |
|---|--------|-----------|--------------------|
| 0 | Double | Double In | The source double. |
|---|--------|-----------|--------------------|

Return Arguments

| | | | |
|---|--------|---------------|---------------------------------------|
| 1 | Double | Double Result | The square root of the source double. |
|---|--------|---------------|---------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Round Double

Rounds a double value.

Input Arguments

| | | | |
|---|---------|-------------------|---|
| 0 | Double | Double In | The source double. |
| 1 | Integer | Decimal Precision | The number of decimals to which the double should be rounded. |

Return Arguments

| | | | |
|---|--------|---------------|--------------------|
| 2 | Double | Double Result | The rounded value. |
|---|--------|---------------|--------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Trig Function

Performs a trigonometric function on one (or two) double values.

Input Arguments

| 0 | Trig Function | Function | The trigonometric function to perform. Choose from $\sin(x)$, $\cos(x)$, $\tan(x)$, $\operatorname{acos}(x)$, $\operatorname{asin}(x)$, $\operatorname{atan}(x)$, $\operatorname{atan2}(y, x)$. |
|---|---------------|-------------|--|
| 1 | Double | X Double In | The X value for the function. |
| 2 | Double | Y Double In | The Y value for the function (if applicable). |

Return Arguments

| | | | |
|---|--------|---------------|---|
| 3 | Double | Double Result | The result of the trigonometric function. |
|---|--------|---------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Logarithmic Function

Performs a logarithmic function on a double value.

Input Arguments

| | | | |
|---|----------------------|-------------|--|
| 0 | Logarithmic Function | Function | The logarithmic function to perform. Choose from $\ln(x)$, $\log(x)$, or e . |
| 1 | Double | X Double In | The X value for the function. |

Return Arguments

| | | | |
|---|--------|---------------|---|
| 2 | Double | Double Result | The result of the trigonometric function. |
|---|--------|---------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Double Angle Conversion

Converts angle units.

Input Arguments

| | | | |
|---|---------------|--------------------|-----------------------------------|
| 0 | Angular Units | Input An Units | The initial units of the angle. |
| 1 | Double | Input Angle Double | The numerical value of the angle. |
| 2 | Double | Output Ang Units | The desired units of angle. |

Return Arguments

| | | | |
|---|--------|---------------------|--|
| 3 | Double | Output Angle Double | The desired value of the angle in the new units. |
|---|--------|---------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

17 VECTOR MATH OPERATIONS

Vector Addition

Adds two vectors together.

Input Arguments

| | | | |
|---|--------|---------------|---------------------------|
| 0 | Vector | First Vector | The first vector. |
| 1 | Vector | Second Vector | The second vector to add. |

Return Arguments

| | | | |
|---|--------|------------------|--|
| 2 | Vector | Resultant Vector | The resulting vector from the vector addition. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Subtraction

Subtracts one vector from another.

Input Arguments

| | | | |
|---|--------|---------------|---|
| 0 | Vector | First Vector | The first vector. |
| 1 | Vector | Second Vector | The vector to subtract from the first vector. |

Return Arguments

| | | | |
|---|--------|------------------|---|
| 2 | Vector | Resultant Vector | The resulting vector from the vector subtraction. |
|---|--------|------------------|---|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Dot Product

Computes the dot product of two vectors.

Input Arguments

| | | | |
|---|--------|---------------|--|
| 0 | Vector | First Vector | The first vector. |
| 1 | Vector | Second Vector | The vector to dot with the first vector. |

Return Arguments

| | | | |
|---|--------|------------------|--|
| 2 | Vector | Resultant Vector | The resulting vector from the dot product. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Cross Product

Computes the cross product of two vectors.

Input Arguments

| | | | |
|---|--------|---------------|--|
| 0 | Vector | First Vector | The first vector. |
| 1 | Vector | Second Vector | The vector to cross with the first vector. |

Return Arguments

| | | | |
|---|--------|------------------|--|
| 2 | Vector | Resultant Vector | The resulting vector from the cross product. |
|---|--------|------------------|--|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Scaling

Scales the magnitude of a vector.

Input Arguments

| | | | |
|---|--------|--------------|----------------------------|
| 0 | Vector | Vector | The vector to scale. |
| 1 | Double | Scale Factor | The scale factor to apply. |

Return Arguments

| | | | |
|---|--------|------------------|------------------------------|
| 2 | Vector | Resultant Vector | The resulting scaled vector. |
|---|--------|------------------|------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Normalize

Normalizes a vector so that its magnitude is 1.

Input Arguments

| | | | |
|---|--------|--------|--------------------------|
| 0 | Vector | Vector | The vector to normalize. |
|---|--------|--------|--------------------------|

Return Arguments

| | | | |
|---|--------|------------------|----------------------------------|
| 2 | Vector | Resultant Vector | The resulting normalized vector. |
|---|--------|------------------|----------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

Vector Magnitude (Length)

Calculates the magnitude (length) of a vector.

Input Arguments

| | | | |
|---|--------|--------|-------------------------|
| 0 | Vector | Vector | The vector in question. |
|---|--------|--------|-------------------------|

Return Arguments

| | | | |
|---|--------|-----------------|---------------------------------------|
| 1 | Double | Resultant Value | The magnitude of the supplied vector. |
|---|--------|-----------------|---------------------------------------|

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

None.

18 MP SUBROUTINES

Run Subroutine

Calls an MP subroutine.

Input Arguments

| 0 | File Path or Embedded File | MP Subroutine File Path | The path to the MP subroutine to call. |
|------|----------------------------|-------------------------|---|
| 1 | Boolean | Share Parent Variables? | Specifies whether variables in-scope in the calling MP are accessible from the called subroutine. |
| 2->n | USER | USER | Additional arguments which are input arguments for the selected subroutine. |

Return Arguments

None.

Returned Status

| | |
|-----------------|---|
| SUCCESS | The subroutine was found and returned a SUCCESS return value. |
| PARTIAL SUCCESS | The subroutine was found and returned a PARTIAL SUCCESS return value. |
| FAILURE | The subroutine could not be found or returned a FAILURE return value. |

Remarks

Argument 0 supports both absolute paths (ex. `C:\test.mp`) and relative paths (ex. `.\test.mp`).

Once the subroutine has been specified in Argument 0 (via the “Browse” entry method), a list of additional arguments (2 -> n) will appear if the entered subroutine has input arguments defined. Note that this list of arguments is only refreshed when the subroutine is selected in Argument 0. To refresh the list, re-select the subroutine in Argument 0.

Define Subroutine Input Values

Defines the input arguments for a subroutine. This is equivalent to defining subroutine or function input arguments in a traditional programming language.

Input Arguments

| | | | |
|------|------|------|-------------------------|
| 0->n | USER | USER | User-defined Arguments. |
|------|------|------|-------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This must be the first command of any subroutine that is created.

Any number of input arguments may be added, edited, or removed using the buttons below the comment area of the MP editor. Notice that when adding arguments, you specify the data type and description for each of the arguments. Leaving the list of arguments empty indicates that the subroutine takes no arguments.

Return from Subroutine Now

Returns immediately from a subroutine, passing back a specified return status. This returned status is the resulting status of the Run Subroutine command in the parent MP.

Input Arguments

| 0 | Measurement Plan Result | MP Subroutine Return Step Result | The step status to return back to the calling MP. |
|------|-------------------------|----------------------------------|---|
| 1->n | USER | USER | User-defined return arguments. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Once a Define Subroutine Return Values command has been specified on the last line of the subroutine, you can click the Refresh Arguments button to refresh the list of arguments that will be returned to the calling MP. Use this list to specify what values to return from the subroutine.

The arguments listed in each Return from Subroutine Now command must match those in the Define Subroutine Return Values command.

Define Subroutine Return Values

Defines the return arguments for a subroutine, passing back a specified return status. This returned status is the resulting status of the Run Subroutine command in the parent MP. This is equivalent to defining one or more function return values in a traditional programming language.

Input Arguments

| | | | |
|------|-------------------------|----------------------------------|---|
| 0 | Measurement Plan Result | MP Subroutine Return Step Result | The returned status that will be passed back to the calling MP. |
| 1->n | USER | USER | Additional User-defined arguments. |

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

This must be the last command of any subroutine that is created.

Any number of return arguments may be added, edited, or removed using the buttons below the comment area of the MP editor. Notice that when adding arguments, you specify the data type and description for each of the arguments. Leaving the list of arguments empty indicates that the subroutine returns no arguments.

This Page Intentionally Left Blank.

19 VARIABLES

Set Integer Variable

Sets the value of an integer variable.

Input Arguments

| | | | |
|---|---------|-------|--|
| 0 | String | Name | The name of the integer variable to set. |
| 1 | Integer | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Integer Variable

Retrieves the value of an integer variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the integer variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|---------|-------|---|
| 1 | Integer | Value | The value stored in the specified variable. |
|---|---------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Double Variable

Sets the value of a double variable.

Input Arguments

| | | | |
|---|--------|-------|-----------------------------------|
| 0 | String | Name | The name of the variable to set. |
| 1 | Double | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Double Variable

Retrieves the value of a double variable.

Input Arguments

| | | | |
|---|--------|------|--|
| 0 | String | Name | The name of the double variable to retrieve. |
|---|--------|------|--|

Return Arguments

| | | | |
|---|--------|-------|---|
| 1 | Double | Value | The value stored in the specified variable. |
|---|--------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Named Double List Variable

Sets the value of a double list variable.

Input Arguments

| | | | |
|---|-------------|----------------------|---------------------------------|
| 0 | String | Name | The name of the list variable. |
| 1 | Double List | Double List Variable | The value of the list variable. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid list variable name was provided. |

Remarks

None.

Get Named Double List Variable

Retrieves the value of a double list variable.

Input Arguments

| | | | |
|---|--------|------|---------------------------------------|
| 0 | String | Name | The name of the double list variable. |
|---|--------|------|---------------------------------------|

Return Arguments

| | | | |
|---|-------------|----------------------|---------------------------------|
| 1 | Double List | Double List Variable | The value of the list variable. |
|---|-------------|----------------------|---------------------------------|

Returned Status

| | |
|---------|---|
| SUCCESS | The value was retrieved successfully. |
| FAILURE | An invalid list variable name was provided. |

Remarks

None.

Add Double to Named Double List Variable

Adds a double to a double list variable.

Input Arguments

| | | | |
|---|--------|--------------|--|
| 0 | String | Name | The name of the double list variable. |
| 1 | Double | Double Value | The double to add to the double list variable. |

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The double value was added successfully. |
| FAILURE | An invalid list variable name was provided. |

Remarks

None.

Get Named Double List Variable Min/Max

Retrieves the minimum and maximum values of a double list variable.

Input Arguments

| | | | |
|---|--------|------|---------------------------------------|
| 0 | String | Name | The name of the double list variable. |
|---|--------|------|---------------------------------------|

Return Arguments

| | | | |
|---|--------|---------------|---------------------------------|
| 1 | Double | Minimum Value | Min value of the list variable. |
| 2 | Double | Maximum Value | Max value of the list variable. |

Returned Status

| | |
|---------|---|
| SUCCESS | The min and max values were retrieved successfully. |
| FAILURE | An invalid list variable name was provided. |

Remarks

None.

Clear Named Double List Variable

Clears the double list variable.

Input Arguments

| | | | |
|---|--------|------|---------------------------------------|
| 0 | String | Name | The name of the double list variable. |
|---|--------|------|---------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|---|
| SUCCESS | The double list variable was cleared uccessfully. |
| FAILURE | An invalid list variable name was provided. |

Remarks

None.

Get i-th Double From List

Gets the value of a double from a list.

Input Arguments

| | | | |
|---|-------------|-------------------|---------------------------------------|
| 0 | Double List | Double List | The name of the list. |
| 1 | Integer | Double List Index | The number of the double in the list. |

Return Arguments

| | | | |
|---|--------|-------|--|
| 2 | Double | Value | The value of the double in the specified list. |
|---|--------|-------|--|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid double list was provided. |

Remarks

None.

Get number of Doubles in List

Gets the number of doubles in a list.

Input Arguments

| | | | |
|---|-------------|-------------|-----------------------|
| 0 | Double List | Double List | The name of the list. |
|---|-------------|-------------|-----------------------|

Return Arguments

| | | | |
|---|---------|-------|---|
| 1 | Integer | Value | The number of the double in the specified list. |
|---|---------|-------|---|

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid double list was provided. |

Remarks

None.

Clear Double List

Clears the double list.

Input Arguments

| | | | |
|---|-------------|-------------|-----------------------|
| 0 | Double List | Double List | The name of the list. |
|---|-------------|-------------|-----------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--------------------------------------|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid double list was provided. |

Remarks

None.

Set String Variable

Sets the value of a string variable.

Input Arguments

| | | | |
|---|--------|-------|-----------------------------------|
| 0 | String | Name | The name of the variable to set. |
| 1 | String | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get String Variable

Retrieves the value of a string variable.

Input Arguments

| | | | |
|---|--------|------|--|
| 0 | String | Name | The name of the string variable to retrieve. |
|---|--------|------|--|

Return Arguments

| | | | |
|---|--------|-------|---|
| 1 | String | Value | The value stored in the specified variable. |
|---|--------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Boolean Variable

Sets the value of an boolean variable.

Input Arguments

| | | | |
|---|---------|-------|--|
| 0 | String | Name | The name of the boolean variable to set. |
| 1 | Boolean | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Boolean Variable

Retrieves the value of a boolean variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the boolean variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|---------|-------|---|
| 1 | Boolean | Value | The value stored in the specified variable. |
|---|---------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Vector Variable

Sets the value of a vector variable.

Input Arguments

| | | | |
|---|--------|-------|--|
| 0 | String | Name | The name of the vector variable to set. |
| 1 | Vector | Value | The vector value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Vector Variable

Retrieves the value of a vector variable.

Input Arguments

| | | | |
|---|--------|------|--|
| 0 | String | Name | The name of the vector variable to retrieve. |
|---|--------|------|--|

Return Arguments

| | | | |
|---|--------|-------|---|
| 1 | Vector | Value | The value stored in the specified variable. |
|---|--------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Font Variable

Sets the value of a font variable.

Input Arguments

| | | | |
|---|-----------|-------|-----------------------------------|
| 0 | String | Name | The name of the variable to set. |
| 1 | Font Type | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Font Variable

Retrieves the value of a font variable.

Input Arguments

| | | | |
|---|--------|------|--|
| 0 | String | Name | The name of the font variable to retrieve. |
|---|--------|------|--|

Return Arguments

| | | | |
|---|-----------|-------|---|
| 1 | Font Type | Value | The value stored in the specified variable. |
|---|-----------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Transform Variable

Sets the value of a transform variable.

Input Arguments

| | | | |
|---|-----------|-------|---|
| 0 | String | Name | The name of the variable to set. |
| 1 | Transform | Value | The transform value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Transform Variable

Retrieves the value of a transform variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the transform variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|-----------|-------|---|
| 1 | Transform | Value | The value stored in the specified variable. |
|---|-----------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Point Name Variable

Sets the value of a point name variable.

Input Arguments

| | | | |
|---|------------|-------|-----------------------------------|
| 0 | String | Name | The name of the variable to set. |
| 1 | Point Name | Value | The value to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The value was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Point Name Variable

Retrieves the value of a point name variable.

Input Arguments

| | | | |
|---|--------|------|--|
| 0 | String | Name | The name of the point name variable to retrieve. |
|---|--------|------|--|

Return Arguments

| | | | |
|---|------------|-------|---|
| 1 | Point Name | Value | The value stored in the specified variable. |
|---|------------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The value was retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Collection Object Ref List Variable

Sets the list of values of a collection object name reference list variable.

Input Arguments

| | | | |
|---|---------------------------------|-------|--|
| 0 | String | Name | The name of the variable to set. |
| 1 | Collection Object Name Ref List | Value | The list of values to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The values were set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Collection Object Ref List Variable

Retrieves the values of a collection object name reference list variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the collection object name reference list variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|---------------------------------|-------|--|
| 1 | Collection Object Name Ref List | Value | The values stored in the specified variable. |
|---|---------------------------------|-------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The values were retrieved. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Relationship Ref List Variable

Sets the list of relationship names in a name reference list variable.

Input Arguments

| | | | |
|---|-----------------------|-------|---|
| 0 | String | Name | The name of the variable to set. |
| 1 | Relationship Ref List | Value | The list of relationships to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The values were set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Relationship Ref List Variable

Returns the list of relationship names in a name reference list variable.

Input Arguments

| | | | |
|---|--------|------|---------------------------|
| 0 | String | Name | The name of the variable. |
|---|--------|------|---------------------------|

Return Arguments

| | | | |
|---|-----------------------|-------|--|
| 1 | Relationship Ref List | Value | The list of relationships in the variable. |
|---|-----------------------|-------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The values were returned successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Set Report Items Reference List Variable

Sets the list of report item names in a name reference list variable.

Input Arguments

| | | | |
|---|-----------------------|-------|---|
| 0 | String | Name | The name of the variable to set. |
| 1 | Report Items Ref List | Value | The list of items to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The values were set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

Items are the most generic list of things contained within SA. An item list allows free selection of anything within SA, such as relationships, dimensions, annotations, reports, pictures, etc.

Get Report Items Reference List Variable

Returns the list of report item names in a name reference list variable.

Input Arguments

| | | | |
|---|--------|------|---------------------------|
| 0 | String | Name | The name of the variable. |
|---|--------|------|---------------------------|

Return Arguments

| | | | |
|---|-----------------------|-------|------------------------------------|
| 1 | Report Items Ref List | Value | The list of items in the variable. |
|---|-----------------------|-------|------------------------------------|

Returned Status

| | |
|---------|--|
| SUCCESS | The values were returned successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Set Point Name Ref List Variable

Sets the list of point names in a point name reference list variable.

Input Arguments

| | | | |
|---|---------------------|-------|--|
| 0 | String | Name | The name of the variable to set. |
| 1 | Point Name Ref List | Value | The list of values to set the variable to. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The values were set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Point Name Ref List Variable

Retrieves the point names in a point name ref list variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the point name reference list variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|---------------------|-------|---|
| 1 | Point Name Ref List | Value | The list of point names stored in the specified variable. |
|---|---------------------|-------|---|

Returned Status

| | |
|---------|--|
| SUCCESS | The list of point names were retrieved successfully. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set String Ref List Variable

Stores a list of strings as a variable.

Input Arguments

| | | | |
|---|-----------------|-------|---|
| 0 | String | Name | The name for the string reference list variable. |
| 1 | String Ref List | Value | The list of strings to store under the variable name. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The list of strings were stored successfully under the point name. |
| FAILURE | The list of strings could not be found. |

Remarks

None.

Get String Ref List Variable

Retrieves the string list associated with a string ref list variable.

Input Arguments

| | | | |
|---|--------|------|---|
| 0 | String | Name | The name of the point name reference list variable to retrieve. |
|---|--------|------|---|

Return Arguments

| | | | |
|---|-----------------|-------|---|
| 1 | String Ref List | Value | The list of strings stored in the specified variable. |
|---|-----------------|-------|---|

Returned Status

| | |
|---------|---|
| SUCCESS | The list of strings was retrieved successfully. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Set Collection Object Name Variable

Sets a variable for a collection object name.

Input Arguments

| | | | |
|---|------------------------|-------|---|
| 0 | String | Name | The name for the variable. |
| 1 | Collection Object Name | Value | The collection object name to assign to the variable. |

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The variable was set successfully. |
| FAILURE | An invalid variable name was provided. |

Remarks

None.

Get Collection Object Name Variable

Retrieves a collection object name associated with a variable.

Input Arguments

| | | | |
|---|--------|------|----------------------------|
| 0 | String | Name | The name for the variable. |
|---|--------|------|----------------------------|

Return Arguments

| | | | |
|---|------------------------|-------|--|
| 1 | Collection Object Name | Value | The resulting collection object name stored with the variable. |
|---|------------------------|-------|--|

Returned Status

| | |
|---------|--|
| SUCCESS | The variable was retrieved successfully. |
| FAILURE | The variable was not found. |

Remarks

None.

Delete Variable

Deletes a variable from memory.

Input Arguments

| | | | |
|---|--------|------|-------------------------------------|
| 0 | String | Name | The name of the variable to remove. |
|---|--------|------|-------------------------------------|

Return Arguments

None.

Returned Status

| | |
|---------|--|
| SUCCESS | The variable was deleted successfully. |
| FAILURE | The specified variable does not exist. |

Remarks

None.

Delete Variables - Wildcard Match

Deletes several variables from memory that match a specified wildcard selection criteria.

Input Arguments

| | | | |
|---|--------|----------------------------|--|
| 0 | String | Variable Wildcard Criteria | A wildcard string specifying the names of variables to remove. |
|---|--------|----------------------------|--|

Return Arguments

None.

Returned Status

| | |
|---------|-------------------------------|
| SUCCESS | This command always succeeds. |
|---------|-------------------------------|

Remarks

Enter wildcard values for the variable name using the same conventions as elsewhere in SA. Asterisks (*) are wildcards for one or more characters, while question marks (?) are placeholders for single characters. To remove all variables that start with s1, use a wildcard selection criteria of s1*.

This Page Intentionally Left Blank.